

IBM SPSS Modeler 15 スクリプト  
とオートメーション ガイド



注：この情報とサポートされている製品をご使用になる前に、「注意事項」( p. ) の一般情報をお読みください。

本版は IBM SPSS Modeler 15 , および新版で指示されるまで後続するすべてのリリースおよび変更に対して適用されます。

Adobe 製品のスクリーンショットは Adobe Systems Incorporated の許可を得て転載しています。

Microsoft 製品のスクリーンショットは Microsoft 社の許可を得て転載しています。

Licensed Materials - Property of IBM

© Copyright IBM Corporation 1994, 2012.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# はじめに

IBM® SPSS® Modeler は、IBM Corp. が開発した企業強化用のデータ マイニング ワークベンチです。SPSS Modeler を使用すると、企業はデータを詳しく調べることで顧客および一般市民とのリレーションシップを強化することができます。企業は、SPSS Modeler を使って得られた情報に基づいて利益をもたらす顧客を獲得し、抱き合わせ販売の機会を見つけ、新規顧客を引き付け、不正を発見し、リスクを減少させ、政府機関へのサービスの提供を改善することができます。

SPSS Modeler の視覚的インターフェイスを使用すると、特定ビジネスの専門知識を適用し、より強力な予測モデルを実現し、解決までの時間を短縮します。SPSS Modeler では、予測、分類、セグメント化、および関連性検出アルゴリズムなど、さまざまなモデル作成手法を提供しています。モデルを作成した後は、IBM® SPSS® Modeler Solution Publisher により、企業全体の意思決定者やデータベースにモデルを配布することが可能になります。

## IBM Business Analytics について

IBM Business Analytics ソフトウェアは、意思決定者がビジネス パフォーマンスを向上させるために信頼する完全で、一貫した正確な情報を提供します。ビジネス インテリジェンス、予測分析、財務実績および戦略管理、および分析アプリケーションの包括的なポートフォリオを利用することによって、現在の実績を明確、迅速に理解し、将来の結果を予測することができます。豊富な業界のソリューション、実績ある実例、専門サービスと組み合わせ、さまざまな規模の組織が、高い生産性を実現、意思決定を自信を持って自動化し、より良い決定をもたらします。

このポートフォリオの一部として、IBM SPSS Predictive Analytics ソフトウェアを使用する組織は、将来のイベントを予測し、その洞察に基づいて積極的に行動し、より優れた業績を実現することができます。全世界の企業、政府、学術分野のお客様が IBM SPSS の技術を活用し、不正行為を減少させ、リスクを軽減させながら、顧客の獲得、保持、成長において、競争優位を高めることができます。IBM SPSS ソフトウェアを日々の業務に取り入れることによって、組織は業務目標を達成し、大きな競争的優位を獲得することができるよう、意思決定を方向付け、自動化することができるようになります。お問い合わせは、<http://www.ibm.com/spss> を参照してください。

## テクニカル サポート

お客様はテクニカル サポートをご利用いただけます。IBM Corp. 製品の使用方法、または対応するハードウェア環境へのインストールについてサポートが必要な場合は、テクニカル サポートにご連絡ください。テクニカ

ル サポートの詳細は、IBM Corp. Web ページ <http://www.ibm.com/support> を参照してください。ご本人、組織、サポートの同意を確認できるものをご用意ください。

---

# 内容

|  |          |
|--|----------|
| <b>1 IBM SPSS Modeler について</b>   | <b>1</b> |
| IBM SPSS Modeler 製品  | 1        |
| IBM SPSS Modeler   | 1        |
| IBM SPSS Modeler Server  | 2        |
| IBM SPSS Modeler Administration Console                                  | 2        |
| IBM SPSS Modeler Batch   | 3        |
| IBM SPSS Modeler Solution Publisher                                      | 3        |
| IBM SPSS Modeler Server の IBM SPSS Collaboration and Deployment Services | 3        |
| IBM SPSS Modeler エディション  | 3        |
| IBM SPSS Modeler ドキュメント  | 5        |
| SPSS Modeler Professional ドキュメント   | 5        |
| SPSS Modeler Premium ドキュメント  | 6        |
| アプリケーションの例   | 7        |
| Demos フォルダ   | 7        |

## パート I: スクリプトとスクリプト言語

|                                |           |
|--------------------------------|-----------|
| <b>2 スクリプトの概要</b>              | <b>10</b> |
| スクリプトの種類                       | 10        |
| ストリーム スクリプト                    | 11        |
| ストリーム スクリプトの例 :ニューラル ネットワークの学習 | 13        |
| スタンドアロン スクリプト                  | 14        |
| スタンドアロン スクリプトの例 :モデルの保存とロード    | 15        |
| スタンドアロン スクリプトの例 :フィールド選択モデルの生成 | 16        |
| スーパーノード スクリプト                  | 17        |
| スーパーノード スクリプトの例                | 18        |
| スクリプトの実行と中断                    | 19        |
| 検索と置換                          | 19        |

### 3 スクリプト言語 23

|                              |    |
|------------------------------|----|
| スクリプト言語の概要                   | 23 |
| スクリプトのシンタックス                 | 23 |
| ノードの参照                       | 24 |
| オブジェクトの取得                    | 26 |
| 現在のオブジェクトの設定                 | 27 |
| ストリームとその他のオブジェクトを開く          | 28 |
| 複数ストリームの作業                   | 28 |
| ローカル スクリプト変数                 | 29 |
| ストリーム、セッション、およびスーパーノード パラメータ | 30 |
| スクリプトの実行の制御                  | 32 |
| スクリプト内の演算子                   | 32 |
| スクリプト内の CLEM 式               | 33 |
| コメントと継続の挿入                   | 33 |
| リテラル テキストのブロック               | 34 |

### 4 スクリプト コマンド 36

|                     |    |
|---------------------|----|
| 一般のスクリプト コマンド       | 36 |
| execute_all         | 36 |
| execute_script      | 36 |
| exit                | 36 |
| for...endfor        | 37 |
| if..then...else...  | 38 |
| set コマンド            | 39 |
| var コマンド            | 42 |
| ノード オブジェクト          | 42 |
| create NODE         | 43 |
| connect NODE        | 44 |
| delete NODE         | 44 |
| disable NODE        | 44 |
| disconnect NODE     | 45 |
| duplicate NODE      | 45 |
| enable NODE         | 45 |
| execute NODE        | 45 |
| export NODE as FILE | 46 |
| flush NODE          | 47 |

|  |    |
|--|----|
| get node NODE . . . . .                        | 47 |
| load node FILENAME . . . . .                   | 47 |
| position NODE. . . . .                         | 47 |
| rename NODE as NEWNAME . . . . .               | 48 |
| ノードの REPOSITORY_PATH の取得 . . . . .             | 48 |
| save node NODE as FILENAME . . . . .           | 49 |
| store node NODE as REPOSITORY_PATH . . . . .   | 49 |
| モデル オブジェクト . . . . .                           | 49 |
| モデル ナゲット名 . . . . .                            | 49 |
| 重複するモデル名の回避 . . . . .                          | 52 |
| delete model MODEL . . . . .                   | 52 |
| export model MODEL as FILE . . . . .           | 53 |
| insert model MODEL . . . . .                   | 54 |
| load model FILENAME . . . . .                  | 54 |
| retrieve model REPOSITORY_PATH . . . . .       | 55 |
| save model MODEL as FILENAME . . . . .         | 55 |
| store model MODEL as REPOSITORY_PATH . . . . . | 55 |
| ストリーム オブジェクト . . . . .                         | 55 |
| create stream DEFAULT_FILENAME . . . . .       | 56 |
| close STREAM . . . . .                         | 56 |
| clear stream . . . . .                         | 56 |
| get stream STREAM. . . . .                     | 57 |
| load stream FILENAME . . . . .                 | 57 |
| open stream FILENAME . . . . .                 | 57 |
| retrieve stream REPOSITORY_PATH . . . . .      | 57 |
| save STREAM as FILENAME . . . . .              | 58 |
| store stream as REPOSITORY_PATH . . . . .      | 59 |
| with stream STREAM . . . . .                   | 59 |
| プロジェクト オブジェクト . . . . .                        | 60 |
| execute_project . . . . .                      | 60 |
| load project FILENAME . . . . .                | 60 |
| retrieve project REPOSITORY_PATH . . . . .     | 60 |
| save project as FILENAME . . . . .             | 60 |
| store project as REPOSITORY_PATH . . . . .     | 60 |
| ステート型オブジェクト . . . . .                          | 61 |
| load state FILENAME . . . . .                  | 61 |
| 結果オブジェクト . . . . .                             | 61 |
| value RESULT. . . . .                          | 61 |
| ファイル オブジェクト . . . . .                          | 62 |
| close FILE . . . . .                           | 62 |
| open FILE. . . . .                             | 62 |
| write FILE. . . . .                            | 62 |

|  |           |
|--|-----------|
| 出力オブジェクト . . . . .   | 63        |
| 出力形式名 . . . . .  | 63        |
| delete output OUTPUT . . . . .   | 64        |
| export output OUTPUT . . . . .   | 64        |
| get output OUTPUT . . . . .  | 64        |
| load output FILENAME . . . . .   | 65        |
| retrieve output REPOSITORY_PATH . . . . .  | 65        |
| save output OUTPUT as FILENAME . . . . .   | 65        |
| store output OUTPUT as REPOSITORY_PATH . . . . .                                       | 65        |
| <br>   |           |
| <b>5 スクリプトのヒント . . . . .</b>   | <b>66</b> |
| ストリーム実行の変更 . . . . .   | 66        |
| ノードのループ . . . . .  | 66        |
| IBM SPSS Collaboration and Deployment Services Repository 内のオブジェクトへの<br>アクセス . . . . . | 67        |
| 暗号化パスワードの生成 . . . . .  | 69        |
| スクリプトの検査 . . . . .   | 70        |
| コマンドラインからのスクリプト . . . . .  | 71        |
| 旧リリースとの互換性 . . . . .   | 71        |
| <br>   |           |
| <b>6 スクリプトの例 . . . . .</b>   | <b>73</b> |
| データ型ノード レポート . . . . .   | 73        |
| ストリーム レポート . . . . .   | 76        |
| <br>   |           |
| <b>7 コマンドラインの引数 . . . . .</b>  | <b>79</b> |
| ソフトウェアの起動 . . . . .  | 79        |
| コマンドライン引数の使用 . . . . .   | 79        |
| 複数の引数の組み合わせ . . . . .  | 80        |
| サーバー接続の引数 . . . . .  | 81        |
| IBM SPSS Collaboration and Deployment Services Repository 接続の引数 . . . . .              | 82        |
| システムの引数 . . . . .  | 83        |
| パラメータの引数 . . . . .   | 85        |



## 8 CLEM 言語に関するリファレンス

86

|               |     |
|---------------|-----|
| CLEM リファレンス概要 | 86  |
| CLEMデータ型      | 86  |
| 整数            | 87  |
| 実数            | 87  |
| 文字            | 88  |
| 文字列           | 88  |
| リスト           | 88  |
| Fields.       | 89  |
| 日付(D)         | 89  |
| Time          | 90  |
| CLEM演算子       | 91  |
| 関数のリファレンス     | 93  |
| 関数の表記方法について   | 94  |
| 情報関数          | 95  |
| 変換関数          | 96  |
| 比較関数          | 97  |
| 論理関数          | 99  |
| 数値関数          | 100 |
| 三角関数          | 101 |
| 確率関数          | 102 |
| ビット単位の整数演算    | 102 |
| 乱数関数          | 104 |
| 文字列関数         | 104 |
| SoundEx 関数    | 110 |
| 日付および時刻の関数    | 110 |
| シーケンス関数       | 115 |
| グローバル関数       | 120 |
| 空白値とヌル値処理関数   | 121 |
| 特殊フィールド       | 122 |

## パート II: プロパティ参照

### 9 プロパティ参照

126

|                     |     |
|---------------------|-----|
| プロパティ参照の概要          | 126 |
| プロパティのシンタックス        | 126 |
| ノードおよびストリームのプロパティの例 | 128 |

|                                 |            |
|---------------------------------|------------|
| ノードのプロパティの概要                    | 129        |
| 共通のノード プロパティ                    | 130        |
| <b>10 ストリームのプロパティ</b>           | <b>131</b> |
| <b>11 プロジェクトのプロパティ</b>          | <b>134</b> |
| <b>12 入力ノードのプロパティ</b>           | <b>135</b> |
| 入力ノードの共通プロパティ                   | 135        |
| cognosimportnode のプロパティ         | 137        |
| databasenode のプロパティ             | 139        |
| datacollectionimportnode のプロパティ | 141        |
| excelimportnod のプロパティ           | 144        |
| evimportnode のプロパティ             | 145        |
| fixedfilenode のプロパティ            | 146        |
| sasimportnode のプロパティ            | 148        |
| statisticsimportnode のプロパティ     | 149        |
| userinputnode のプロパティ            | 149        |
| variablefilenode のプロパティ         | 151        |
| xmlimportnode のプロパティ            | 154        |
| <b>13 レコード設定ノードのプロパティ</b>       | <b>155</b> |
| appendnode のプロパティ               | 155        |
| aggregatenode のプロパティ            | 155        |
| balancenode のプロパティ              | 156        |
| distinctnode のプロパティ             | 157        |
| mergenode のプロパティ                | 158        |
| rfmaggregatenode のプロパティ         | 160        |
| samplenode のプロパティ               | 161        |
| selectnode のプロパティ               | 164        |
| sortnode のプロパティ                 | 164        |

## 14 フィールド設定ノードのプロパティ

165

|                               |     |
|-------------------------------|-----|
| anonymizenode のプロパティ          | 165 |
| autodatapreinode のプロパティ       | 166 |
| binningnode のプロパティ            | 169 |
| derivinode のプロパティ             | 172 |
| ensemblenode のプロパティ           | 174 |
| fillernode のプロパティ             | 175 |
| filternode のプロパティ             | 176 |
| historynode のプロパティ            | 177 |
| partitionnode のプロパティ          | 178 |
| reclassifynode のプロパティ         | 179 |
| reordernode のプロパティ            | 181 |
| restructurenode のプロパティ        | 181 |
| rfanalysisnode のプロパティ         | 182 |
| settoflagnode のプロパティ          | 184 |
| statisticstransformnode プロパティ | 185 |
| timeintervalsnode のプロパティ      | 185 |
| transposenode のプロパティ          | 190 |
| typenode のプロパティ               | 191 |

## 15 グラフ作成ノードのプロパティ

197

|                         |     |
|-------------------------|-----|
| グラフ作成ノードの共通のプロパティ       | 197 |
| collectionnode のプロパティ   | 198 |
| distributionnode のプロパティ | 199 |
| evaluationnode のプロパティ   | 200 |
| graphboardnode のプロパティ   | 202 |
| histogramnode のプロパティ    | 204 |
| multiplotnode のプロパティ    | 206 |
| plotnode のプロパティ         | 207 |
| timeplotnode のプロパティ     | 209 |
| webnode のプロパティ          | 211 |

|                             |     |
|-----------------------------|-----|
| 一般的なモデル作成ノードのプロパティ          | 213 |
| anomalydetectionnode のプロパティ | 214 |
| apriorinode のプロパティ          | 215 |
| autoclassifiernode のプロパティ   | 217 |
| アルゴリズム プロパティの設定             | 219 |
| autoclusternode のプロパティ      | 220 |
| autonumericnode のプロパティ      | 221 |
| bayesnetnode プロパティ          | 223 |
| c50node のプロパティ              | 224 |
| carmanode のプロパティ            | 226 |
| cartnode のプロパティ             | 227 |
| chaidnode のプロパティ            | 230 |
| coxregnode のプロパティ           | 232 |
| decisionlistnode のプロパティ     | 234 |
| discriminantnode のプロパティ     | 236 |
| factornode のプロパティ           | 237 |
| featureselectionnode のプロパティ | 239 |
| genlinnode のプロパティ           | 241 |
| glimmnode のプロパティ            | 245 |
| kmeansnode のプロパティ           | 248 |
| knnnode のプロパティ              | 249 |
| kohonenode のプロパティ           | 251 |
| linearnode プロパティ            | 252 |
| logregnode のプロパティ           | 254 |
| neuralnetnode のプロパティ        | 258 |
| neuralnetworknode プロパティ     | 261 |
| questnode のプロパティ            | 263 |
| regressionnode のプロパティ       | 265 |
| sequencenode のプロパティ         | 267 |
| slrmnode のプロパティ             | 269 |
| statisticsmodelnode のプロパティ  | 270 |
| svmnnode プロパティ              | 270 |
| timeseriesnode のプロパティ       | 271 |
| twostepnode のプロパティ          | 273 |

## 17 モデル ナゲット ノードのプロパティ

275

|                                   |     |
|-----------------------------------|-----|
| applyanomalydetectionnode のプロパティ  | 275 |
| applypriorinode のプロパティ            | 276 |
| applyautoclassifiernode のプロパティ    | 276 |
| applyautoclusternode のプロパティ       | 277 |
| applyautonumericnode プロパティ        | 277 |
| applybayesnetnode のプロパティ          | 277 |
| applyc50node のプロパティ               | 278 |
| applycarmanode のプロパティ             | 278 |
| applycartnode のプロパティ              | 279 |
| applychaidnode のプロパティ             | 279 |
| applycoxregnode のプロパティ            | 280 |
| applydecisionlistnode のプロパティ      | 280 |
| applydiscriminantnode のプロパティ      | 280 |
| applyfactornode のプロパティ            | 281 |
| applyfeatureselectionnode のプロパティ  | 281 |
| applygeneralizedlinearnode のプロパティ | 281 |
| applykmeansnode のプロパティ            | 282 |
| applyknnnode プロパティ                | 282 |
| applykohonenode のプロパティ            | 282 |
| applylinearnode プロパティ             | 283 |
| applylogregnode のプロパティ            | 283 |
| applyneuralnetnode のプロパティ         | 283 |
| applyneuralnetworknode プロパティ      | 284 |
| applyquestnode のプロパティ             | 284 |
| applyregressionnode のプロパティ        | 285 |
| applyselflearningnode のプロパティ      | 285 |
| applysequencenode のプロパティ          | 285 |
| applysvmnode のプロパティ               | 286 |
| applytimeseriesnode のプロパティ        | 286 |
| applytwostepnode のプロパティ           | 286 |

## 18 データベース モデル作成ノードのプロパティ 287

|                                      |     |
|--------------------------------------|-----|
| Microsoft モデル作成ノードのプロパティ             | 288 |
| Microsoft モデル作成ノードのプロパティ             | 288 |
| Microsoft モデル ナゲットのプロパティ             | 291 |
| Oracle モデル作成ノードのプロパティ                | 293 |
| Oracle モデル作成ノードのプロパティ                | 293 |
| Oracle モデル ナゲットのプロパティ                | 299 |
| IBM DB2 モデル作成ノードのプロパティ               | 300 |
| IBM DB2 モデル作成ノードのプロパティ               | 300 |
| IBM DB2 モデル ナゲットのプロパティ               | 306 |
| IBM Netezza Analytics モデル作成ノードのプロパティ | 307 |
| Netezza モデル作成ノードのプロパティ               | 307 |
| Netezza モデル ナゲットのプロパティ               | 315 |

## 19 出カノードのプロパティ 316

|                             |     |
|-----------------------------|-----|
| analysisnode のプロパティ         | 316 |
| dataauditnode のプロパティ        | 317 |
| matrixnode のプロパティ           | 319 |
| meansnode のプロパティ            | 321 |
| reportnode のプロパティ           | 323 |
| setglobalsnode のプロパティ       | 324 |
| statisticsnode のプロパティ       | 325 |
| statisticsoutputnode のプロパティ | 326 |
| tablenode のプロパティ            | 327 |
| transformnode のプロパティ        | 329 |

## 20 エクスポート ノードのプロパティ 331

|                                 |     |
|---------------------------------|-----|
| 共通のエクスポート ノード プロパティ             | 331 |
| cognosexportnode のプロパティ         | 331 |
| databaseexportnode のプロパティ       | 332 |
| datacollectionexportnode のプロパティ | 338 |
| excelexportnode のプロパティ          | 339 |
| outputfilenode プロパティ            | 340 |

|                                  |     |
|----------------------------------|-----|
| sasexportnode のプロパティ.....        | 341 |
| statisticsexportnode のプロパティ..... | 341 |
| xmlexportnode のプロパティ.....        | 342 |

## 21 IBM SPSS Statistics ノードのプロパティ 343

|                                    |     |
|------------------------------------|-----|
| statisticsimportnode のプロパティ.....   | 343 |
| statisticstransformnode プロパティ..... | 343 |
| statisticsmodelnode のプロパティ.....    | 344 |
| statisticsoutputnode のプロパティ.....   | 345 |
| statisticsexportnode のプロパティ.....   | 346 |

## 22 スーパーノードのプロパティ 347

### 付録

## A 注意事項 350

## 索引 353





# IBM SPSS Modeler について

IBM® SPSS® Modeler は、ビジネスの専門知識を活用して予測モデルを迅速に作成したり、また作成したモデルをビジネス オペレーションに展開して意志決定を改善できるようにする、一連のデータ マイニング ツールです。SPSS Modeler は業界標準の CRISP-DM モデルをベースに設計されたものであり、データ マイニング プロセス全体をサポートして、データに基づいてより良いビジネスの成果を達成できるようにします。

SPSS Modeler ではさまざまなモデル作成方法を提供しています。[モデル作成] パレットを利用して、データから新しい情報を引き出したり、予測モデルを作成することができます。各手法によって、利点や適した問題の種類が異なります。

SPSS Modeler は、スタンドアロン製品として購入または SPSS Modeler Server と組み合わせてクライアントとして使用することができます。後のセクションで説明されているとおり、多くの追加オプションも使用することができます。詳細は、<http://www.ibm.com/software/analytics/spss/products/modeler/> を参照してください。

## IBM SPSS Modeler 製品

製品と関連するソフトウェアの IBM® SPSS® Modeler ファミリの構成は次のとおりです。

- IBM SPSS Modeler
- IBM SPSS Modeler Server
- IBM SPSS Modeler Administration Console
- IBM SPSS Modeler Batch
- IBM SPSS Modeler Solution Publisher
- IBM SPSS Modeler Server の IBM SPSS Collaboration and Deployment Services

## IBM SPSS Modeler

SPSS Modeler はこの製品のすべての機能を搭載したバージョンであり、コンピュータにインストールし、そのコンピュータで実行します。スタンドアロン製品としてローカル モードで SPSS Modeler を実行するか、大規

模なデータ セットを使用する場合にパフォーマンスを向上させるために IBM® SPSS® Modeler Server と組み合わせて実行することができます。

SPSS Modeler を使用して、プログラミングの必要なく、正確な予測モデルを迅速かつ直感的に構築することができます。独自のビジュアル インターフェイスを使用すると、データ マイニング プロセスを簡単に視覚化することができます。製品に組み込まれている高度な分析の支援を受けて、データ内に隠れたパターンやトレンドを発見することができます。結果をモデル化し、ビジネスチャンスを活用してリスクを軽減できるようになり、それらに影響を与える要因を理解することができます。

SPSS Modeler は SPSS Modeler Professional および SPSS Modeler Premium の 2 つのエディションで使用できます。 [詳細は、IBM SPSS Modeler エディション in IBM SPSS Modeler 15 ユーザー ガイド](#) を参照してください。

## IBM SPSS Modeler Server

SPSS Modeler は、クライアント/サーバー アーキテクチャを使用し、リソース主体の操作が必要な要求を、強力なサーバー ソフトウェアへ分散されるようになりました。その結果、規模が比較的大きいデータ セットを処理するパフォーマンスを実現しました。

SPSS Modeler Server は、1 つまたは複数の IBM® SPSS® Modeler のインストールと組み合わせてサーバー ホストで分散分析モードで継続的に実行する、別途ライセンスが必要な製品です。このように、メモリー集中型の操作は、クライアントコンピュータにデータをダウンロードせずにサーバー上で実行することができるため、SPSS Modeler Server は大きなデータセットに対し優れたパフォーマンスを示すことができます。IBM® SPSS® Modeler Server は、パフォーマンスと自動化のさらなる利点を提供し、SQLの最適化とデータベース内のモデリング機能をサポートしています。

## IBM SPSS Modeler Administration Console

Modeler Administration Console は多くの SPSS Modeler Server 設定オプションを管理し、オプション ファイルによって設定可能なグラフィカルアプリケーションです。アプリケーションには、SPSS Modeler Server のインストールを監視、構成するコンソール ユーザー インターフェイスが用意されており、しかも、現在の SPSS Modeler Server のお客様には無料で提供されます。アプリケーションは Windows コンピュータにのみインストールできますが、サポートされる任意のプラットフォームにインストールされたサーバーを管理できます。

## IBM SPSS Modeler Batch

データマイニングは、通常、対話型のプロセスですが、グラフィカル ユーザー インターフェースを必要とせずに、コマンドラインから SPSS Modeler を実行することも可能です。たとえば、ユーザーの介入なしで実行する長期実行または反復的なタスクがあります。SPSS Modeler Batch は、通常のユーザーインターフェイスにアクセスせずに SPSS Modeler の完全な分析機能のサポートを提供する製品の特別バージョンです。SPSS Modeler Batch を使用するには、SPSS Modeler Server ライセンスが必要です。

## IBM SPSS Modeler Solution Publisher

SPSS Modeler Solution Publisher は、外部ランタイムで実行することができ、外部アプリケーションに埋め込まれる SPSS Modeler ストリームのパッケージ版を作成することができるツールです。このように、SPSS Modeler がインストールされていない環境で使用するための完全な SPSS Modeler ストリームを公開して展開することができます。SPSS Modeler Solution Publisher は、個別のライセンスが必要とされている IBM SPSS Collaboration and Deployment Services - Scoring サービスの一部として配布されています。このライセンスを使用すると、SPSS Modeler Solution Publisher Runtime を受信し、公開されたストリームを実行することができます。

## IBM SPSS Modeler Server の IBM SPSS Collaboration and Deployment Services

さまざまな IBM® SPSS® Collaboration and Deployment Services アダプタを使用すると、SPSS Modeler および SPSS Modeler Server が IBM SPSS Collaboration and Deployment Services リポジトリとインタラクティブに機能させることができます。このように、リポジトリにデプロイされた SPSS Modeler ストリームは、複数のユーザーで共有したり、またはシンククライアントアプリケーション IBM SPSS Modeler Advantage からアクセスできます。リポジトリをホストするシステム上のアダプタをインストールします。

## IBM SPSS Modeler エディション

SPSS Modeler は次のエディションで使用できます。

## SPSS Modeler Professional

SPSS Modeler Professional は、CRM システムで追跡する行動や対話、人口統計データ、購入行動や販売データなど、多くの構造化データを処理するために必要なすべてのツールを提供しています。

## SPSS Modeler Premium

SPSS Modeler Premium は、エンティティの分析やソーシャル ネットワーキングなどの特化したデータ、又は構造化されていないテキスト データを処理するために SPSS Modeler Professional を拡張する、別途ライセンスが必要な製品です。SPSS Modeler Premium は次のコンポーネントで構成されています。

**IBM® SPSS® Modeler Entity Analytics** が新しい次元を IBM® SPSS® Modeler の予測分析に追加します。予測分析は過去のデータから将来の行動を予測しようとするのに対し、エンティティ分析ではレコードの中でアイデンティティの競合を解決することで現在のデータの干渉性と一貫性を改善することに焦点を当てます。アイデンティティは、個人、組織、オブジェクトまたは曖昧さの存在する他のエンティティとなります。アイデンティティの解決は、顧客関係の管理、不正行為の検出、マネーロンダリング防止、国内および国際的なセキュリティなどのさまざまなフィールドにおいて重要になります。

**IBM SPSS Modeler Social Network Analysis** は、関係に関する情報を、個人およびグループの社会的行動を特徴づけるフィールドに変換します。ソーシャル ネットワークの基底となる関係を説明するデータを使用して、IBM® SPSS® Modeler Social Network Analysis はネットワークの他の人の行動に影響を与えるソーシャル リーダーを識別します。また、他のネットワーク参加者に最も影響を受ける人を確認できます。これらの結果を他の指標と組み合わせることによって、予測モデルの基準となる個人の包括的なプロフィールを作成できます。この社会的情報を含むモデルは、含まないモデルに比べてパフォーマンスが高くなります。

**Text Analytics for IBM® SPSS® Modeler** は、高度な言語技術と Natural Language Processing (NLP) を使用して、多様な未構築のテキスト データを急速に処理し、重要なコンセプトを抽出および組織化、そしてそのコンセプトをカテゴリ別に分類します。抽出されたコンセプトとカテゴリを、人口統計のような既存の構造化データと組み合わせ、SPSS Modeler の豊富なデータ マイニング ツールを適用する方法で、焦点を絞ったより良い決定を下すことができます。

## IBM SPSS Modeler ドキュメント

オンライン ヘルプ形式のドキュメントは、SPSS Modeler の [ヘルプ] メニューから使用できます。SPSS Modeler、SPSS Modeler Server、および SPSS Modeler Solution Publisher のアプリケーション ガイドやその他 サポート資料が含まれています。

各製品の PDF 形式の完全なドキュメント（インストール手順を含む）は、各製品 DVD の ¥Documentation フォルダにもあります。インストール マニュアルは、Web サイト <http://www-01.ibm.com/support/docview.wss?uid=swg27023172> からダウンロードできます。

これらの形式のドキュメントは、SPSS Modeler インフォメーション センター <http://publib.boulder.ibm.com/infocenter/spssmodl/v15r0m0/> から入手できます。

## SPSS Modeler Professional ドキュメント

SPSS Modeler Professional のドキュメント スイート（インストール手順を除く）は次のとおりです。

- **IBM SPSS Modeler ユーザー ガイド:**SPSS Modeler の使用方法への全体的な入門で、データ ストリームの構築方法、欠損地の処理方法、CLEM 式の処理方法、プロジェクトおよびレポートの処理方法、IBM SPSS Collaboration and Deployment Services、予測アプリケーション製品、または IBM SPSS Modeler Advantage へ展開するストリームのパッケージ化方法が含まれています。
- **IBM SPSS Modeler 入力ノード、プロセス ノード、出力ノード:** さまざまな形式のデータを読み込み、処理し、出力するために使用するすべてのノードの説明があります。これは、モデル作成ノード以外のすべてのノードについての説明です。
- **IBM SPSS Modeler モデル作成ノード:** データ マイニング モデルの作成に使用するすべてのノードの説明。IBM® SPSS® Modeler には、マシン学習、人工知能、および統計に基づいたさまざまなモデル作成手法が用意されています。詳細は、[3 章 モデル作成ノードの概要 in IBM SPSS Modeler 15 モデル作成ノード](#) を参照してください。
- **IBM SPSS Modeler アルゴリズム ガイド:**SPSS Modeler で使用されている手法の数学的な基礎の説明があります。このガイドは、PDF 形式のみです。
- **IBM SPSS Modeler アプリケーション ガイド:** 本ガイドの例では、特定のモデル作成手法および技術に関する簡単で、目的に沿った説明を行います。本ガイドのオンライン バージョンは、[ヘルプ] メニューから利用できます。詳細は、[アプリケーションの例 in IBM SPSS Modeler 15 ユーザー ガイド](#) を参照してください。

- **IBM SPSS Modeler スクリプトとオートメーション:** スクリプトの実行によるシステムのオートメーションの情報で、ノードおよびストリームを操作するために使用することができるプロパティが含まれています。
- **IBM SPSS Modeler 展開ガイド:**SPSS Modeler のストリームやシナリオを IBM® SPSS® Collaboration and Deployment Services Deployment Manager のジョブを処理するステップとしての実行についての情報。
- **IBM SPSS Modeler CLEF 開発者ガイド:**CLEF では、 SPSS Modeler のノードとしてデータ処理ルーチンやモデル作成アルゴリズムなどのサードパーティ製のプログラムを統合します。
- **IBM SPSS Modeler データベース内 マイニング ガイド:** ユーザーのデータベースを最大限に活用して、パフォーマンスを改善する方法と、サードパーティー製のアルゴリズムを使用して分析可能な範囲を拡大する方法についての情報があります。
- **IBM SPSS Modeler Server 管理およびパフォーマンス ガイド**IBM® SPSS® Modeler Server の設定と管理の方法について説明します。
- **IBM SPSS Modeler 管理コンソール ユーザー ガイド:**SPSS Modeler Server を監視して設定するためのコンソール ユーザー インターフェイスのインストールおよび使用に関する情報。コンソールは、Deployment Manager アプリケーションへのプラグインとして実装されます。
- **IBM SPSS Modeler Solution Publisherガイド:** SPSS Modeler Solution Publisher はアドオン コンポーネントです。組織はこれを使用すると、標準的な SPSS Modeler 環境の外部へストリームを公開できます。
- **IBM SPSS Modeler CRISP-DM Guide:** CRISP-DM 手法を使用して SPSS Modeler によるデータ マイニングを行う段階的なガイドです。
- **IBM SPSS Modeler Batch ユーザー ガイド:** バッチ モードの実行およびコマンドラインの引数の詳細を含む、IBM SPSS Modeler をバッチ モードで使用するための完全ガイド。このガイドは、PDF 形式のみです。

## SPSS Modeler Premium ドキュメント

SPSS Modeler Premium のドキュメント スイート（インストール手順を除く）は次のとおりです。

- **IBM SPSS Modeler Entity Analytics ユーザー ガイド:** リポジトリのインストールと設定、エンティティ分析ノード、管理タスクについて説明した、SPSS Modeler でのエンティティ分析の使用に関する情報。
- **IBM SPSS Modeler Social Network Analysis ユーザー ガイド:** グループ分析および拡散分析を含む SPSS Modeler によるソーシャル ネットワーク分析を実行するためのガイド。

- **Text Analytics for SPSS Modeler ユーザー ガイド:** テキスト マイニング ノード、インタラクティブ ワークベンチ、テンプレート、その他のリソースについて説明した、SPSS Modeler でのテキスト分析の使用に関する情報。
- **Text Analytics for IBM SPSS Modeler 管理コンソール ユーザー ガイド:** Text Analytics for SPSS Modeler と使用するために IBM® SPSS® Modeler Server を監視して設定するためのコンソール ユーザー インターフェイスのインストールおよび使用に関する情報。コンソールは、Deployment Manager アプリケーションへのプラグインとして実装されます。

## アプリケーションの例

SPSS Modeler のデータ マイニング ツールは、多様なビジネスおよび組織の問題解決を支援しますが、アプリケーションの例では、特定のモデル作成手法および技術に関する簡単な、目的に沿った説明を行います。ここで使用されるデータセットは、データ マイニング 作業によって管理された巨大なデータ ストアよりも非常に小さいですが、関係するコンセプトや方法は実際のアプリケーションに対して大規模です。

SPSS Modeler の [ヘルプ] メニューから [アプリケーションの例] を選択すると、例にアクセスすることができます。データ ファイルとサンプル ストリームは、製品のインストール ディレクトリの Demos フォルダにインストールされています。詳細は、[Demos フォルダ in IBM SPSS Modeler 15 ユーザー ガイド](#) を参照してください。

**データベース モデル作成の例:** 例は、『IBM SPSS Modeler データベース内マイニング ガイド』を参照してください。

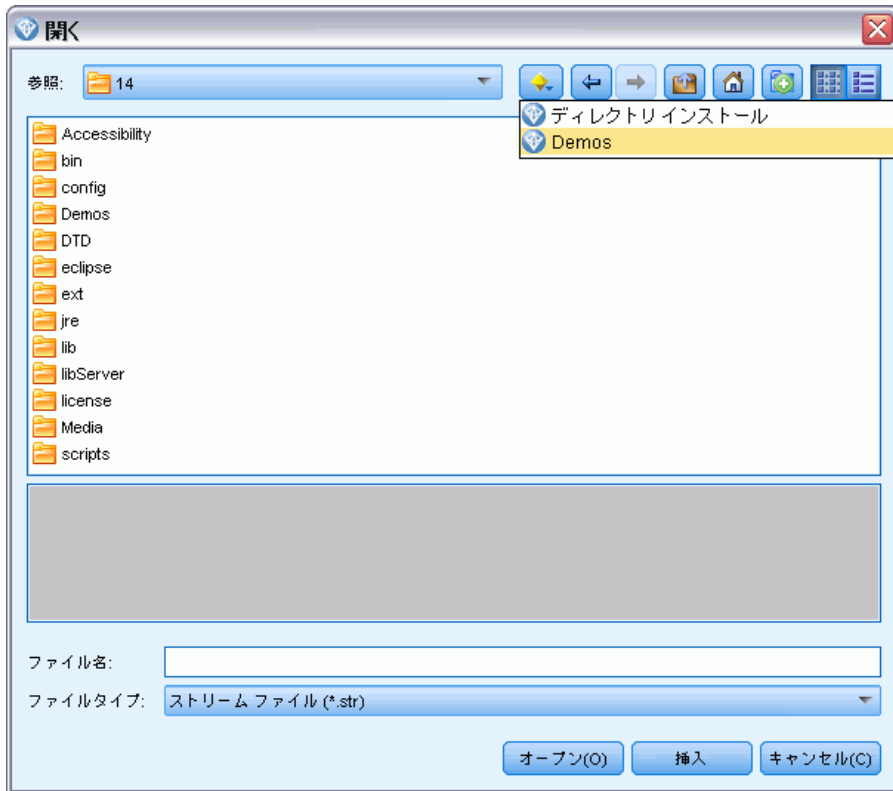
**スクリプトの例:** 例は、『IBM SPSS Modeler スクリプトとオートメーション ガイド』を参照してください。

## Demos フォルダ

アプリケーションの例で使用されるデータ ファイルとサンプル ストリームは、製品のインストール ディレクトリの Demos フォルダにインストールされています。このフォルダには、Windows [スタート] メニューの **IBM SPSS Modeler 15** プログラム グループから、または [ファイルを開く] ダイ

アログ ボックスの最近使ったディレクトリの一覧から [Demos] を選択してアクセスすることもできます。

図 1-1  
最近使用されたディレクトリの一覧から Demos フォルダを選択





# パート I: スクリプトとスクリプト言語

# スクリプトの概要

IBM® SPSS® Modeler のスクリプトは、ユーザー インターフェースのプロセスを自動化する強力なツールです。スクリプトで、マウスやキーボードを使用した場合と同じ種類のアクションを実行できます。また、頻繁に繰り返したり手動で実行するのに時間がかかるタスクを自動化するために使用できます。

次の処理にスクリプトを使用できます。

- ストリームでノードを実行する特定の順序を指定する。
- CLEM (Control Language for Expression Manipulation) のサブセットを使用して、ノードにオプロパティを設定したり、フィールドを作成したりする。
- 通常はユーザーとの対話によって実行される一連の操作（たとえば、モデルを作成してテストするなど）を自動化する。
- 十分なユーザーとの対話が必要な複雑な処理（たとえば、モデルの生成とテストを繰り返す交差検証手順など）を設定する。
- ストリームを操作するプロセスを設定する。たとえば、モデル学習ストリームの取得および実行と、対応するモデル テスト ストリームの生成を自動的に実行できます。

この章では、ストリームレベルのスクリプト、スタンドアロン スクリプト、および SPSS Modeler インターフェースのスーパーノード内のスクリプトに関する高度な説明と例を記述しています。スクリプト言語、シンタックス、およびコマンドは、以後の章で説明します。

注： SPSS Modeler 内の IBM® SPSS® Statistics で作成されたスクリプトはインポートおよび実行できません。

## スクリプトの種類

IBM® SPSS® Modeler では、次の 3 種類のスクリプトが使用されます。

- **ストリーム スクリプト** は、ストリーム プロパティとして格納されるため、特定のストリームと一緒に保存およびロードされます。たとえば、モデル ナゲットの学習と適用のプロセスを自動化するストリーム スクリプトを書くことができます。また、特定のストリームが実行されたときは常に、そのストリームのキャンパスの内容ではなく、スクリプトが実行されるように指定することもできます。

- **スタンドアロン スクリプト**は、どのストリームとも関連付けがなく、外部のテキスト ファイルに保存されます。スタンドアロン スクリプトは、たとえば、複数のストリームを一緒に操作する場合に使用できます。
- **スーパーノード スクリプト**は、スーパーノード ストリーム プロパティとして格納されます。スーパーノード スクリプトは、ターミナル スーパーノードでのみ使用可能です。スーパーノード スクリプトは、スーパーノードの内容のシーケンスの実行を制御するのに使用できます。ターミナル以外の（ソースまたはプロセス）スーパーノードの場合、ストリーム スクリプト内で直接、スーパーノードまたはスーパーノード内のノードにプロパティを定義できます。

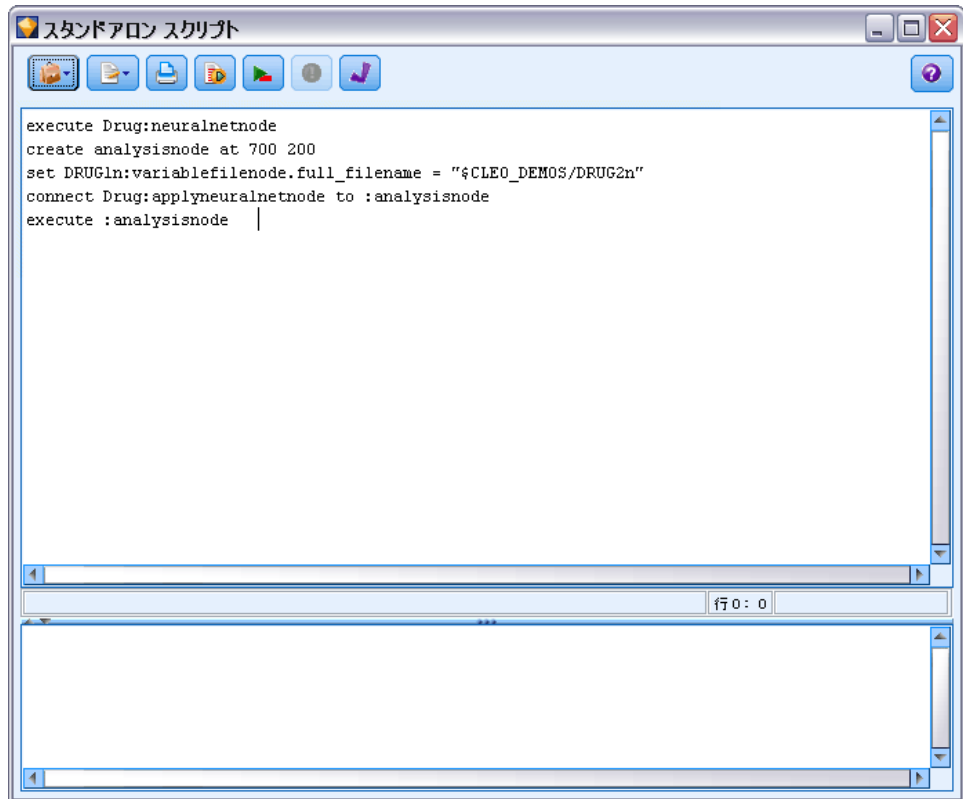
## ストリーム スクリプト

スクリプトを使って特定のストリーム内の操作をカスタマイズできます。また、スクリプトをそのストリームとともに保存することができます。ストリーム スクリプトは、ストリーム内のターミナル ノードの、特定の実行順序を指示するために使用されます。ストリーム スクリプト ダイアログ ボックスを使用して、現在のストリームとともに保存されているスクリプトを編集します。

**[ストリームのプロパティ] ダイアログ ボックスの [ストリーム スクリプト] タブにアクセスするには**

- ▶ [ツール] メニューから次の各項目を選択します。  
ストリームのプロパティ > スクリプト...
- ▶ [スクリプト] タブをクリックして、現在のストリームに関連するスクリプトの作業を行います。

図 2-1  
[ストリーム スクリプト] ダイアログ ボックス



このダイアログ ボックスの一番上にあるツールバー アイコンを使用すると、次のような作業を実行できます。

- ウィンドウに既存のスタンドアロン スクリプトの内容をインポートする。
- スクリプトをテキスト ファイルとして保存する。
- スクリプトを印刷する。
- デフォルト スクリプトを追加する。
- 現在のスクリプト全体を実行する。
- スクリプトから選択した行を実行する。
- スクリプトのシンタックスをチェックして、エラーが見つければ、ダイアログ ボックスの下部パネルにそれを表示する。

さらに、ストリームが実行されたときにこのスクリプトが実行されるべきか、それとも実行されないべきかを指定できます。[このスクリプトを実行]を選択すると、ストリームの実行時に常に、スクリプトに指定された実行順序でこのスクリプトが実行されます。この設定により、ストリームレ

ベルでの自動化を実現でき、素早いモデル構築が可能になります。ただし、デフォルトでは、ストリーム実行時にこのスクリプトは無視されます。[このスクリプトを無視] オプションを選択した場合でも、常にこのダイアログ ボックスで直接スクリプトを実行できます。

## ストリーム スクリプトの例 :ニューラル ネットワークの学習

ストリームは実行時に、ニューラル ネットワーク モデルの学習に使用できます。通常、モデルをテストするには、モデル作成ノードを実行してモデルをストリームに追加し、適切な接続を確立して、精度分析ノードを実行します。

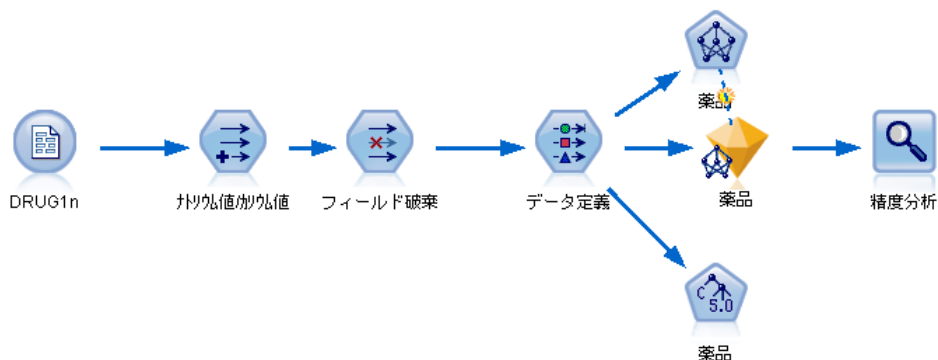
IBM® SPSS® Modeler スクリプトを使用すると、モデル ナゲット作成後のテスト プロセスを自動化できます。たとえば、デモ ストリーム `druglearn.str` (SPSS Modeler インストールの下の `/Demos/streams/` フォルダ内) をテストする次のストリーム スクリプトは、[ストリーム プロパティ] ダイアログ ([ツール] → [ストリームのプロパティ] → [スクリプト]) で実行できます。

```
execute Drug:neuralnetworknode
create analysisnode at 700 200
set DRUG1n.variablefilename.full_filename = "$CLEO_DEMOS/DRUG2n"
connect :applyneuralnetworknode to :analysisnode
execute :analysisnode
```

このスクリプト例の各行について、次に説明します。

- 1 行目によって、デモ ストリーム内にすでに見つかった **Drug** と呼ばれるニューラル ノードが実行され、モデル ナゲットが作成され、ストリームですでにデータ型ノードに接続しているストリーム領域に配置されます。
- 2 行目で、分析ノードが作成され、それがキャンバス位置 700 x 200 に配置されます。
- 3 行目で、ストリーム内で使用された元のデータ ソースが、**DRUG2n** と呼ばれるテスト データセットに切り替えられます。
- 4 行目で、ニューラル ネットワーク モデル ナゲットが精度分析ノードに接続します。ストリームには他の同様のノードが存在していないので、ニューラル ネットワーク モデル ナゲットまたは精度分析ノードを表すのに名前は使用されないことに注意してください。
- 最後に、分析ノードが実行されて、分析レポートが生成されます。

図 2-2  
結果のストリーム



このスクリプトは、既存のストリームとともに機能するように設計されています。Drug ニューラル ノードはすでに存在すると想定されています。ただし、空のストリーム領域から、ストリームを作成して実行するスクリプトを使用することも可能です。スクリプト言語一般については、「スクリプト言語の概要」( p. 23 ) を参照してください。スクリプト コマンドの詳細は、「スクリプト コマンド」( p. 36 ) を参照してください。

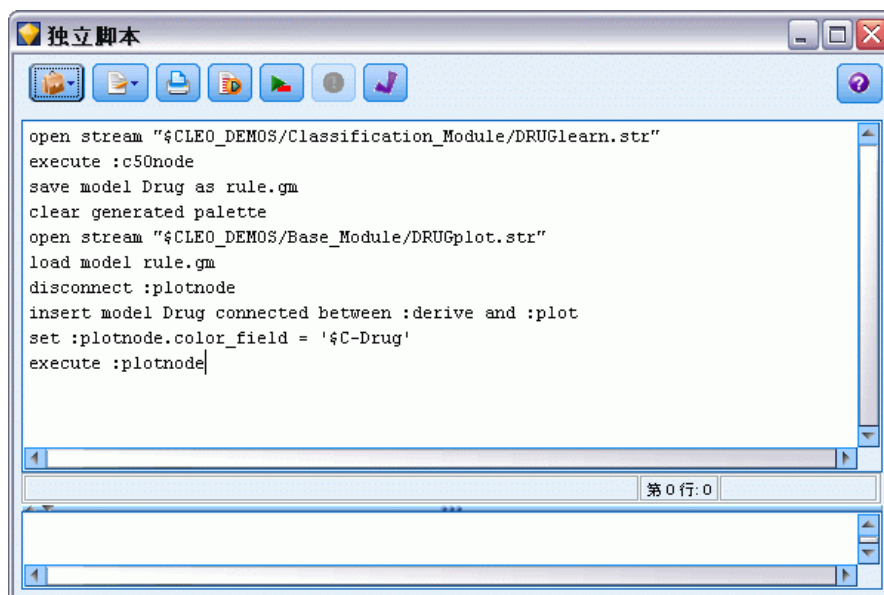
## スタンドアロン スクリプト

[スタンドアロン スクリプト] ダイアログ ボックスでは、テキスト ファイルとして保存されるスクリプトを作成したり編集したりします。このダイアログ ボックスには、ファイル名が表示されます。スクリプトのロード、保存、インポート、および実行の機能が備わっています。

### スタンドアロン スクリプトのダイアログ ボックスにアクセスするには

- ▶ メイン メニューから次の各項目を選択します。  
ツール > [スタンドアロン スクリプト]

図 2-3  
[スタンドアロン スクリプト] ダイアログ ボックス



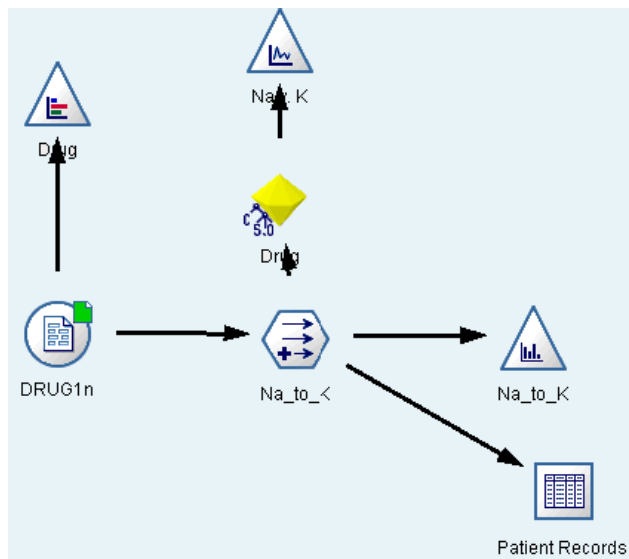
スタンドアロン スクリプトでは、ストリーム スクリプトと同じツールバーやスクリプト シンタックス検査オプションを使用することができます。詳細は、[p. 11 ストリーム スクリプト](#) を参照してください。

## スタンドアロン スクリプトの例 :モデルの保存とロード

スタンドアロン スクリプトは、ストリームを操作するときに役立ちます。2 種類のストリームがある場合を想定します。1 つはモデルを作成するストリームであり、もう 1 つはグラフを使用して最初のストリームと既存のデータ フィールドから生成されたルール セットを探索するストリームです。この場合のスタンドアロン スクリプトは次のようになります。

```
open stream "$CLEO_DEMOS/streams/druglearn.str"
execute :c50node
save model Drug as rule.gm
clear generated palette
open stream "$CLEO_DEMOS/streams/drugplot.str"
load model rule.gm
disconnect :plotnode
insert model Drug connected between :derive and :plot
set :plotnode.color_field = '%C-Drug'
execute :plotnode
```

図 2-4  
結果のストリーム



注： スクリプト言語一般については、「スクリプト言語の概要」（ p. 23 ）を参照してください。スクリプト コマンドの詳細は、「スクリプト コマンド」（ p. 36 ）を参照してください。

## スタンドアロン スクリプトの例 :フィールド選択モデルの生成

この例では、空の領域からフィールド選択モデルを生成するストリームを構築し、そのモデルに適用して、指定された対象に関連するもっとも重要な上位 15 のフィールドを表示するテーブルを作成します。

```
create stream 'featureselection'
create statisticsimportnode
position :statisticsimportnode at 50 50
set :statisticsimportnode.full_filename = "$CLEO_DEMOS/customer_dbase.sav"
```

```
create typenode
position :typenode at 150 50
set :typenode.direction.'response_01' = Target
connect :statisticsimportnode to :typenode
```

```
create featureselectionnode
position :featureselectionnode at 250 50
set :featureselectionnode.screen_missing_values=true
set :featureselectionnode.max_missing_values=80
set :featureselectionnode.criteria = Likelihood
set :featureselectionnode.important_label = "Check Me Out!"
```



```

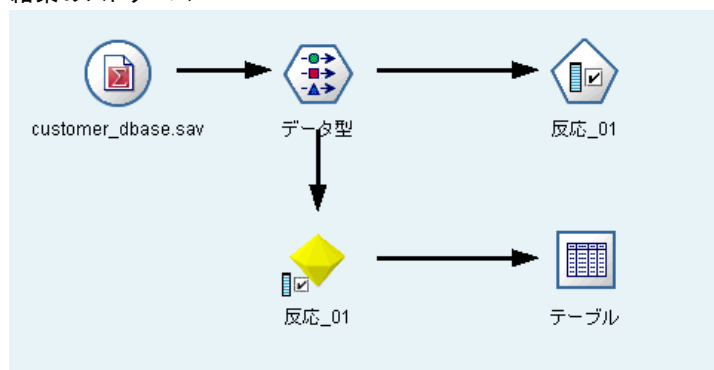
set :featureselectionnode.selection_mode = TopN
set :featureselectionnode.top_n = 15
connect :typenode to :featureselectionnode
execute :featureselectionnode

create tablenode
position :tablenode at 250 250
connect response_01:applyfeatureselectionnode to :tablenode
execute :tablenode

```

このスクリプトで、データを読み込む入力ノードを作成し、response\_01 フィールドの役割を **Target** に設定するデータ型ノードを使用し、その後フィールド選択ノードを作成して実行します。また、読みやすいレイアウトになるように、ストリーム領域で各ノードを接続し、配置します。その後、作成されるモデル ナゲットがテーブル ノードへ接続されます。テーブル ノードでは、**selection\_mode** プロパティと **top\_n** プロパティに設定されたとおりに、もっとも重要な上位 15 フィールドが一覧表示されます。詳細は、16 章 p.239 **featureselectionnode** のプロパティを参照してください。

図 2-5  
結果のストリーム



## スーパーノード スクリプト

IBM® SPSS® Modeler のスクリプト言語を使用して、スクリプトを作成し、任意のターミナル スーパーノード内に保存できます。これらのスクリプトはターミナル スーパーノードにのみ使用でき、テンプレート ストリームの作成時、およびスーパーノードの内容に特定の実行順序を指定する際に使用できます。スーパーノード スクリプトを使用すると、ストリーム内で複数のスクリプトを実行することもできます。

たとえば、複雑なストリームで実行の順序を指定する必要があり、スーパーノードには、散布図ノードで使用される新しいフィールドを作成する前に実行される必要のあるグローバル ノードを含む、いくつかのノードが

あるとします。この場合、まずグローバル ノードを実行するスーパーノード スクリプトを作成できます。このノードが計算する平均や標準偏差などの値は、散布図ノードを実行するときに使用します。

スーパーノード スクリプト内では、ほかのスクリプトの場合と同様の方法で、ノード プロパティを指定できます。また、ストリーム スクリプトから直接に、任意のスーパーノードまたはカプセル化されたノードのプロパティを変更または定義することもできます。[詳細は、22 章 p.347 スーパーノードのプロパティ を参照してください。](#) この手法は、ソース スーパーノード、プロセス スーパーノード、およびターミナル スーパーノードに適用できます。

注：独自のスクリプトを実行することができるのはターミナル スーパーノードの場合だけなので、[スーパーノード]ダイアログ ボックスの [スクリプト] タブは、ターミナル スーパーノードの場合にだけ利用可能です。

### メイン キャンバスから [スーパーノード スクリプト] ダイアログ ボックスを開くには

- ▶ ストリーム キャンバスでターミナル スーパーノードを選択して、[スーパーノード] メニューから次の項目を選択します。  
[スーパーノード スクリプト(S)...]

### ズーム インしたスーパーノード キャンバスから [スーパーノード スクリプト] ダイアログ ボックスを開くには

- ▶ スーパーノード キャンバス上を右クリックして表示されるコンテキストメニューから、次の項目を選択します。  
[スーパーノード スクリプト(S)...]

[詳細は、9 章 スーパーノードとスクリプト in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード を参照してください。](#)

## スーパーノード スクリプトの例

次のスーパーノード スクリプトでは、スーパーノード内のターミナルノードが実行されるべき順序が宣言されます。この順序によって、まずグローバル ノードが実行されて、別のノードを実行したときに、このノードによって算出される値が使用されるようになります。

```
execute 'Set Globals'  
execute 'gains'  
execute 'profit'  
execute 'age v. $CC-pep'  
execute 'Table'
```

## スクリプトの実行と中断

その他多くの方法でスクリプトを実行できます。たとえば、ストリーム スクリプトまたはスタンドアロンのスクリプトのダイアログで、[このスクリプトを実行] ボタンをクリックすると、完全なスクリプトを実行します。

図 2-6  
[このスクリプトを実行] ボタン



[選択した行] ボタンをクリックすると、スクリプト内で選択した 1 行または隣接する行のブロックを実行します。

図 2-7  
[選択した行を実行] ボタン



スクリプトの実行は、次のいずれかの方法で行います。

- ストリーム スクリプトまたはスタンドアロン スクリプトのダイアログ ボックスの [このスクリプトを実行] または [選択した行を実行] をクリックします。
- デフォルトの実行方法として [このスクリプトを実行] が設定されているストリームを実行する。
- 起動時にインタラクティブ モードで `-execute` フラグを使用します。 [詳細は、7 章 p.79 コマンド ライン引数の使用 を参照してください。](#)

注： [スーパーノード] ダイアログ ボックスで [このスクリプトを実行] を選択しているかぎり、スーパーノード スクリプトは、スーパーノードの実行時に実行されます。

### スクリプト実行の中断

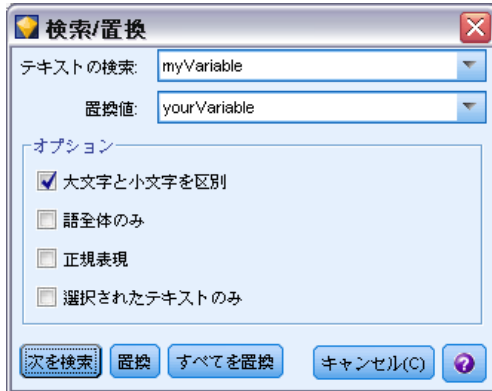
[ストリーム スクリプト] ダイアログ ボックスのツールバーにある赤い中止ボタンは、スクリプト実行時に有効になります。このボタンを使用すると、スクリプトおよび現在のストリームの実行を中止することができます。

## 検索と置換

[検索/置換] ダイアログ ボックスは、スクリプト エディタ、CLEM 式ビルダーなど、スクリプトまたは式のテキストを編集する場合、またはレポート ノードでテンプレートを定義する場合に使用できます。これらの領域のいずれかでテキストを編集する場合、**Ctrl + F** キー を押してダイアログ ボックスにアクセスし、カーソルがテキスト領域にフォーカスしているこ

とを確認します。[フィルタ] ノードを使用している場合、たとえば、[設定] タブのテキスト領域から、または CLEM 式ビルダーのテキスト フィールドからダイアログ ボックスにアクセスできます。

図 2-8  
[検索/置換] ダイアログ ボックス



- ▶ テキスト領域内にカーソルを置いて、Ctrl + F キーを押して [検索/置換] ダイアログ ボックスにアクセスします。
- ▶ 検索するテキストを入力するか、最近検索した項目のドロップダウン リストから選択します。
- ▶ 置換テキストがある場合は、入力します。
- ▶ [次を検索] をクリックして、検索を開始します。
- ▶ [置換] をクリックして現在の選択内容を置換するか、[すべて置換] をクリックしてすべてまたは選択したインスタンスを更新します。
- ▶ 各操作が終了すると、ダイアログ ボックスが閉じます。テキスト領域で F3 を押すと最後の検索操作が繰り返され、または Ctrl + F キーを押すとダイアログに再度アクセスします。

### 検索オプション

**大文字と小文字を区別：** 検索操作で、たとえば myvar が myVar と位置するかどうかなど、大文字と小文字を区別するかどうかを指定します。この設定に関係なく、置換テキストは常に入力したとおりに挿入されます。

**語全体のみ：** 検索操作が語内に埋め込まれたテキストに一致するかどうかを指定します。このオプションを選択すると、spider に関する検索は、spiderman または spider-man に一致しません。

**正規表現：** 正規表現のシンタックスを使用するかどうかを指定します（次項参照）。このオプションを選択すると、[語全体のみ] オプションは無効化され、その値は無視されます。

**選択されたテキストのみ** :[すべて置換] オプションを使用する場合、検索の範囲を制御します。

## 正規表現シンタックス

正規表現を使用すると、タブまたは改行文字などの特殊文字、a から d までなど文字のクラスまたは範囲、行の開始または終了などの境界について検索することができます。次の種類の表現がサポートされています。

### 文字の一致

| 文字     | 一致                                      |
|--------|---|
| x      | 文字 x                                    |
| ¥¥     | 円記号                                     |
| ¥0n    | 8 進法の値を持つ文字 0n (0 ≤ n ≤ 7)              |
| ¥0n    | 8 進法の値を持つ文字 0nn (0 ≤ n ≤ 7)             |
| ¥0mnn  | 8 進法の値を持つ文字 0mnn (0 ≤ m ≤ 3, 0 ≤ n ≤ 7) |
| ¥xhh   | 16 進法の値を持つ文字 0xhh                       |
| ¥uhhhh | 16 進法の値を持つ文字 0xhhhh                     |
| ¥t     | タブ文字 ( '¥u0009' )                       |
| ¥n     | 改行文字 ( '¥u000A' )                       |
| ¥r     | 復帰文字 ( '¥u000D' )                       |
| ¥f     | 改ページ文字 ( '¥u000C' )                     |
| ¥a     | アラート (ベル) 文字 ( '¥u0007' )               |
| ¥e     | エスケープ文字 ( '¥u001B' )                    |
| ¥cx    | xに対応する制御文字                              |

### 文字クラス的一致

| 文字クラス         | 一致   |
|---------------|--|
| [abc]         | a、b、または c (単純クラス)                                |
| [^abc]        | a、b、または c 以外の文字 (減法)                             |
| [a-zA-Z]      | a から z または A から Z の各文字 (範囲)                      |
| [a-d[m-p]]    | a から d、または m から p (統合) また、[a-dm-p] と指定することもできます。 |
| [a-z&&[def]]  | a から z、および d、e、または f (交差)                        |
| [a-z&&[^bc]]  | b と c 以外の a から z (減法) また、[a-d-z] と指定することもできます。   |
| [a-z&&[^m-p]] | a から z、m から p を除く (減法) また、[a-lq-z] と指定することもできます。 |

### 事前設定された文字クラス

| 事前設定された文字クラス | 一致                              |
|--------------|---------------------------------|
| .            | 任意の文字（行末に一致する場合または一致しない場合があります） |
| ¥d           | 任意の数字： [0-9]                    |
| ¥D           | 数字以外： [^0-9]                    |
| ¥s           | 空白文字： [ ¥t¥n¥x0B¥f¥r]           |
| ¥S           | 空白文字以外： [^¥s]                   |
| ¥w           | 語文字： [a-zA-Z_0-9]               |
| ¥W           | 語文字以外： [^¥w]                    |

### 境界の一致

| 境界の一致 | 一致            |
|-------|---------------|
| ^     | 行頭            |
| \$    | 行末            |
| ¥b    | 語の境界          |
| ¥B    | 語以外の境界        |
| ¥A    | 入力の開始         |
| ¥Z    | 最後の行末以外の入力の終了 |
| ¥z    | 入力の終了         |

# スクリプト言語

## スクリプト言語の概要

IBM® SPSS® Modeler スクリプト言語の構成要素を次に示します。

- ノード、ストリーム、プロジェクト、出力、およびその他の SPSS Modeler オブジェクトを参照するフォーマット
- 上記オブジェクトを操作するのに使用されるスクリプト ステートメントまたはコマンドのセット
- 変数、パラメータ、およびその他のオブジェクトに値を設定するためのスクリプト式の言語
- コメント、行の継続、およびリテラル テキストのブロックのサポート

このセクションでは、スクリプト言語を使用するための基本的なシンタックス（構文規則）を説明します。特定のプロパティとコマンドについての情報は、以後のセクションにあります。

## スクリプトのシンタックス

解析時の明確性を向上させるために、IBM® SPSS® Modeler でスクリプトに関する作業を行う際には、次の規則に従ってください。

- `income` や `referrerID` などの変数名には、引用符を付けないでください。
- すでに値が設定されている既存の変数を参照するときは、`^mystream` のように、先頭にcaret (^) 記号を付けます。ただし、宣言時や変数の値の設定時には、caret (^) 記号を使用しません。 [詳細は、p. 24 ノードの参照](#) を参照してください。
- `'$P-Maxvalue'` のように、セッション、ストリーム、およびスーパーノード パラメータへの参照は、単一引用符で囲む必要があります。
- 二重引用符が使用されている場合、`"Web graph of BP and Drug"` のように式は文字列リテラルとして処理されます。単一引用符と二重引用符を不注意に使用した場合、予期せぬ結果を生じる場合があります（例：`"$P-Maxvalue"` は、パラメータに格納されている値への参照ではなく、文字列になります）。
- `"druglearn.str"` のようなファイル名は、二重引用符で囲む必要があります。
- `datanode` または `Na_to_K` のようなノード名は、引用符を付けないか、単一引用符で囲むことができます。注：名前にスペースや特殊文字が含まれている場合は、引用符で囲む必要があります。ただし、

スクリプト中で '2a\_referrerID' のような、数字から始まるノード名を使用することはできません。

- フラグ型プロパティは、**true** および **false** の値を使用して読み込まれるか、設定される必要があります（ここに示したとおりに小文字を使用）。**Off**、**OFF**、**off**、**No**、**NO**、**no**、**n**、**N**、**f**、**F**、**False**、**FALSE**、または **0** なども値の設定時に認識されますが、プロパティ値の読み込み時にエラーが発生する場合があります。その他の値はすべて **true** と見なされます。**true** と **false** を使用すると、こうした混乱が避けられます。
- 改行、スペース、または単一または二重引用符をブロック内に含むリテラル文字列またはブロックは、三重引用符で囲むことができます。詳細は、[p. 34 リテラル テキストのブロック](#) を参照してください。
- "Age >= 55" のような CLEM 式は、次のように、二重引用符で囲む必要があります。

```
set :derivnode.flag_expr = "Age >= 55"
```

- CLEM 式の中で引用符を使用する場合は、次のように各引用符の前に円記号 (\) を挿入します。

```
set :node.parameter = "BP = \"HIGH\""
```

すべての場合に必ず必要ではありませんが、これらの内容は明確性を向上させるためにお勧めします。すべてのスクリプト ダイアログ ボックスで利用できるスクリプト 検査機能をしようすると、不明確（不正）なシンタックスにはフラグが立てられ、メッセージが表示されます。

## ノードの参照

スクリプト内でノードを参照するための多くの方法があります。

- ノードは、**DRUG1n** などの名前指定できます。この名前は、ノードの種類で修飾することもできます。たとえば、**Drug:neuralnetworknode** は、**Drug** というニューラル ノードであることを示します。
- ノードの種類のみで名前を指定できます。たとえば、**:neuralnetworknode** はすべてのニューラル ノードを参照します。また、**samplenode**、**neuralnetworknode**、および **kmeansnode** などの他の有効なノードの種類も使用できます。接尾辞の **node** は省略できますが、スクリプト内のエラーを識別しやすくするので、省略しないことをお勧めします。
- ノードごとに [注釈] タブに表示される一意の ID でそれぞれのノードを参照できます。"@" 記号の後 ID を使用します（例：**@id5E5GJK23L.custom\_name = "My Node"**）。詳細は、[5 章 \[注釈\] in IBM SPSS Modeler 15 ユーザー ガイド](#) を参照してください。

**生成されたモデル：** 生成されたモデル ノードにも、同じ規則が適用されます。マネージャ ウィンドウ内の生成されたモデルのパレットに表示されたとおりのノード名を使用できます。または、生成されたモデル ノードを



ノードの種類で参照できます。生成されたモデルをマネージャ内で参照するのに使用する名前は、スコアリングの目的でストリームに追加されたモデルに使用する名前と異なります（后者は接頭辞の“apply”を使用します）。詳細は、4章 p.49 [モデル ナゲット名](#) を参照してください。

### 変数を使用したノードの参照

カレット (^) シンタックスを使用すると、ノードの名前と種類をローカルスクリプト変数の値として提供できます。たとえば、ノード名が必要な場合、`^n` は変数 `n` にノードの名前が保存されているノードであり、`Drug:^t` は変数 `t` にノードの種類が保存されている `Drug` という名前のノードです。

ノード参照は、(var ステートメントを使用して宣言された) ローカルスクリプト変数に保存できますが、ストリーム、セッション、またはスーパーノードの各パラメータには保存できません。ノードの参照があいまいにならないように、ノードの作成時に一意なノード ID を変数に割り当てます。

```
var x
set x = create typenode
set ^x.custom_name = "My Node"
```

- 1 行目では `x` という名前の変数を作成します。
- 2 行目では、新しいデータ型ノードを作成し、そのノードへの参照を `x` に保存します。`x` は、ノード名ではなくノード自体への参照を保存していることに注意してください。
- これをオフにするには、プロパティ `custom_name` に値 `"My Node"` を次のように設定します。`x` がノードではなく変数名であることを示すために、カレットが使用されています。(カレットがない場合、システムは `x` という名前のノードを探そうとします。たとえば、`var` コマンドの右辺が変数でしかないため、変数の宣言時と設定時にカレットは必要ありません。ただし、3行目で `x` は、論理的には変数ではなくノード名である可能性もあるため、両者を区別するためにカレットが必要になります。)

最初に変数を宣言しないで変数にノードへの参照を保存しようとするのは、よくある誤りです。

```
set x = create typenode
set ^x.custom_name = "My Node"
```

この場合、`SET` コマンドは、`x` を変数ではなくストリーム、セッション、またはスーパーノードパラメータとして作成しようとしていますが、パラメータにはノードへの参照を保存できないため、エラーが返されます。

### ID によるノードの参照

以下のようにして、一意のノード ID を変数に格納することもできます。

```
var n
set n = "id5E5GJK23L"
set @^n.custom_name = "My Node"
```

**ストリーム内でのノードのループ** `:stream.nodes` プロパティを使用してストリーム内の全ノードのリストを返し、個々のノードにアクセスするためにそのリストを繰り返し参照することもできます。詳細は、6 章 p.76 [ストリーム レポート](#) を参照してください。

## 例

NAME:TYPE

NAME はノードの名前で、TYPE はノードの種類です。NAME または TYPE のどちらか 1 つを省略することはできますが、どちらか一方を省略することはできません。たとえば、次のコマンドは新しいフィールド作成ノードを、`drug1n` という名前の既存の変長ノードと既存の散布図ノードの間に作成します（新規ノードはコロンを使用しません）。

```
create derivenode connected between drug1n and :plotnode
```

また、次の例のように、NAME または TYPE の先頭に ^ 記号を付けて、パラメータの存在を示すこともできます。

Drug:^t

この参照は、Drug というノードを意味し、t は、ノードの種類を指定するパラメータです。たとえば ^t の値が `c50node` の場合、上記の参照は次のように変換できます。

Drug:c50node

同じように、ノード名にパラメータを使用することもできます。たとえば、次の例はノード名が必要なコンテキストで両方とも使用できます。

^n:derivenode

^n

## オブジェクトの取得

`get` コマンドは、ストリーム、ノード、または出力オブジェクトへの参照を返し、スクリプトを使用してこれらのオブジェクトを操作できるようにします。次に例を示します。

```
var mynode
set mynode = get node flag1:derivenode
position ^mynode at 400 400
```

```
var mytable = get output :tableoutput
export output ^mytable as c:/mytable.htm format html
```

```
set stream = get stream 'Stream1'
set ^stream.execute_method = "Script"
```

## 現在のオブジェクトの設定

次の特殊変数は、現在のオブジェクトを参照するために使用できます。

- node
- stream
- output
- project

`project` の例外はありますが、上記の変数は、現在のコンテキストを変更するためにリセットできます。ほかのスクリプト変数とは異なり、これらの変数は事前に定義されているので、最初に `var` コマンドで宣言する必要はありません。

```
set node = create typenode
rename ^node as "mytypenode"
```

```
set output = get output :statisticsoutput
export output ^output as c:/myoutput.htm format html
```

これらの特殊変数は、変数が参照するオブジェクトの名前と一致するので、変数とオブジェクトの区別がある種の場合にあいまいとなる可能性があります。その結果、使用法に微妙な区別が必要です。 [詳細は、4章 p.39 set コマンド を参照してください。](#)

### コメント

誤った種類の値を特殊変数に割り当てると（ノード オブジェクトを `stream` 変数に設定するなど）、ランタイム エラーになります。

特殊変数を使用できる場合は、任意の変数も利用できます。たとえば、現在のストリームを保存する場合は、次のように指定できます。

```
save stream as 'C:/My Streams/Churn.str'
```

これはまた、次のように指定することもできます。

```
save my_stream as 'C:/My Streams/Churn.str'
```

ここで `my_stream` には、事前にストリーム値が割り当てられています。

## ストリームとその他のオブジェクトを開く

スタンドアロンのスクリプト内で、たとえば次のようにファイル名と場所を指定して、ストリームを開くことができます。

```
open stream "c:/demos/druglearn.str"
```

その他の種類のオブジェクトは、次のように **load** コマンドを使用して開くことができます。

```
load node c:/mynode.nod
```

```
load model c:/mymodel.gm
```

**ストリームを開くこととストリームを読み込むことの対比** `:load stream` コマンドは、現在のストリームのノードをクリアしないで、指定されたストリームをストリーム領域へ追加します。このコマンドは、初期のリリースでは現在より機能が多かったのですが、複数のストリーム間でのノードの展開、管理、コピーの機能によって、機能が大きく縮小されました。

## 複数ストリームの作業

ファイル システム、または IBM® SPSS® Collaboration and Deployment Services Repository からストリームにアクセスするために使用されるコマンド (**open**、**load**、および **retrieve**) に加えて、ほとんどのスクリプト コマンドは現在のストリームに自動的に適用されます。しかし、スタンドアロン スクリプト内で、同じスクリプトから複数のストリームを開いて操作したい場合があります。これは、任意の開いているストリームへの参照の設定、または **with... endwith** コマンドを使用して現在ストリームを一時的に転換することで、実行できます。

たとえば、現在のストリーム以外のストリームを閉じるには、希望のストリームを参照するために **get stream** コマンドを使用できます。

```
set stream = get stream "druglearn"  
close stream
```

このスクリプトでは、特殊変数の `stream` を **druglearn** ストリームへ転換し(本質的に現在ストリームにする)、その後そのストリームを閉じます。

または、次のように、**with stream** を使用して現在ストリームが一時的に転換されます。

```
with stream 'druglearn'  
  create typenode  
  execute_script  
endwith
```

このステートメントで、**create** アクションを実行し、指定したストリームを現在ストリームに設定して、そのストリームのスクリプトを実行します。各ステートメントが実行されると、元のストリームが現在のストリームに戻ります。条件文やループなどを指定することもできます。次に例を示します。

```
with stream 'druglearn'
  create tablenode at 500 400
  create selectnode connected between :typenode and :tablenode
  for l from 1 to 5
    set :selectnode.condition = 'Age >' << (l * 10)
    execute :selectnode
  endfor
endwith
```

このステートメントは、ループ内のすべての式の現在ストリームを **STREAM** に設定し、ループの処理が完了したら元の値が復元されます。

## ローカル スクリプト変数

ローカル スクリプト変数は、**var** コマンドで宣言され、また、現在のスクリプト専用設定されます。変数は、パラメータとは異なります。パラメータは、セッション、ストリーム、またはスーパーノードに設定でき、文字列または数値だけを含むことができます。

```
var my_node
set my_node = create distributionnode
rename ^my_node as "Distribution of Flag"
```

既存の変数を参照する場合、カレット (^) 記号をパラメータ名の前に付けます。たとえば、上記のスクリプトを考えてみます。

- 最初の行で変数を宣言します。
- 2行目で、その値を設定します。
- 3 行目で、(変数自体ではなく) この変数に参照されるノードの名前を変更します。カレットで、**^my\_node** がノードのリテラル名ではなく、変数名であることを示しています(カレットがない場合、**rename** コマンドは **my\_node** という名前のノードを探そうとします。**var** コマンドの右辺は変数のみであるため、1 行目と 2 行目にカレットは必要ありません。カレットは、すでに値が設定されている変数を参照するときのみ必要です。この場合、カレットを削除すると、参照先があいまいになります)。
- 変数の参照を解決するときは、セッション、ストリーム、またはスーパーノード パラメータのリストを検索する前に、ローカル変数のリストが検索されます。たとえば、ローカル変数とセッション パラメータの両方に変数 **x** が存在している場合、スクリプトのステートメントで

シンタックス '**\$P-X**' を使用すると、ローカル変数の代わりセッション パラメータが使用されます。

注： 実際には、**var** コマンドを使用して最初に宣言しないで変数を設定した場合、現在スクリプトのコンテキストに応じて、ストリーム、セッション、またはスーパー ノードのパラメータが作成されます。たとえば、次のコーディングで、**z** と命名されたローカル スクリプト変数が作成され、その値が **[1 2 3]** に設定されます。

```
var z  
set z = [1 2 3]
```

**var** コマンドが省略されて、さらにまだ存在しない名前の変数またはノードが想定されている場合は、変数ではなくパラメータとして、**z** が作成されます。

## ストリーム、セッション、およびスーパーノード パラメータ

パラメータは、CLEM 式とスクリプトで使用するために定義できます。実際のところ、パラメータはユーザー定義の変数であり、保存されて、現在のストリーム、セッション、またはスーパーノードで持続します。さらに、スクリプトを使用する場合と同様に、ユーザー インターフェイスからもアクセスできます。たとえば、ストリームを保存すると、そのストリームに設定されているパラメータも保存されます。(これは、ローカル スクリプト変数と異なる点です。ローカル スクリプト変数は、宣言されたスクリプト内でのみ使用できます。)通常パラメータは、スクリプト中でパラメータ値を指定する CLEM 式の一部として使用されます。

パラメータの有効範囲は、それがどこで設定されたかによって異なります。

- ストリーム パラメータは、ストリーム スクリプト内またはストリーム プロパティのダイアログ ボックス内で設定でき、ストリーム内のすべてのノードで使用できます。Clem 式ビルダーの [パラメータ] リストに表示されます。
- セッション パラメータは、スタンドアロン スクリプト内または [セッション パラメータ] ダイアログ ボックス内で設定できます。セッション パラメータは、現在のセッションのすべてのストリーム ([マネージャ] ウィンドウの [ストリーム] タブに表示されているすべてのストリーム) で利用できます。

パラメータは、スーパーノード用にも設定できます。この場合、スーパーノード内にカプセル化されたノードでだけ表示できます。 [詳細は、9 章 スーパーノードのパラメータの定義 in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード を参照してください。](#)

## スクリプト内でのパラメータの設定

パラメータは、スクリプト内で `set` コマンドと次のシンタックスを使用して設定できます。

```
set foodtype = pizza
```

現在のスクリプト内に `foodtype` という名前のノードまたは変数が宣言されていない場合、このコマンドにより、デフォルト値 `pizza` を持つパラメータ `foodtype` が作成されます。

**ユーザー インターフェイス：** パラメータはユーザー インターフェイスからも設定または表示できます。[ツール] メニューから [ストリームのプロパティ] または [セッションパラメータの設定] を選択します。これらのダイアログ ボックスで、スクリプトからは利用できないストレージのタイプなど、追加のオプションを設定できるようになります。詳細は、[5 章 ストリームとセッション パラメータの設定 in IBM SPSS Modeler 15 ユーザー ガイド](#) を参照してください。

**コマンドライン：** パラメータは、コマンドラインからも設定できます。この場合、セッション パラメータとして作成されます。

## スクリプト内におけるパラメータの参照

`$P` を先頭に指定して単一引用符で囲むことにより、前に作成されたパラメータを参照することができます (例: `'$P-minvalue'`)。また、`minvalue` のように単にパラメータ名だけを参照することもできます。パラメータの値は、常に文字列または数値です。たとえば、次のシンタックスで `foodtype` パラメータを参照し、新しい値を設定することができます。

```
set foodtype = pasta
```

スクリプト中で使用されている CLEM 式のコンテキスト中のパラメータを参照することもできます。次のスクリプトに、その例を示します。この例では、`Age` の値が `cutoff` というストリーム パラメータの値よりも大きいレコードを条件抽出ノードに含めるためのプロパティを設定します。パラメータは、CLEM 用の適切なシンタックス (`'$P-cutoff'`) を使用して、CLEM 式内で使用されます。

```
set :selectnode {  
  mode = "Include"  
  condition = "Age >= '$P-cutoff'"  
}
```

このスクリプトでは、ストリーム パラメータ `cutoff` のデフォルト値が使用されます。新しいパラメータ値は、上記の条件抽出ノードの指定に次のシンタックスを追加して、指定できます。

```
set cutoff = 50
```

この行を追加すると、Age の値が 50 より大きいすべてのレコードが選択されます。

詳細は、7 章 ストリーム、セッション、およびスーパーノード パラメータ in IBM SPSS Modeler 15 ユーザー ガイド を参照してください。

## スクリプトの実行の制御

スクリプトでは通常、ステートメントを 1 つずつ順に処理します。しかし、条件ステートメントの if 文や、for ループなどを使用することによって、この処理順序に優先する指定を行えます。

```
if s.maxsize > 10000 then
s.maxsize = 10000
connect s to :derive
endif
```

for ループには、次のようにさまざまなフォーマットがあります。

```
for PARAMETER in LIST
STATEMENTS
endfor
```

このスクリプトは、リストの順序に従って、PARAMETER に割り当てられた LIST 内の各値について、STATEMENTS を 1 回ずつ実行します。リストは大かっこで囲みません。また、リストの内容は定数です。その他の多数のフォーマットも利用できます。詳細は、4 章 p.36 一般のスクリプト コマンド を参照してください。

## スクリプト内の演算子

通常の CLEM 演算子に加えて “+” および “-” 演算子を使用して、var コマンドを使用して宣言されたローカル スクリプト変数を操作できます。+ 演算子はリストに要素を追加し、- 演算子はリストから項目を除去します。次に例を示します。

```
var z #新しいローカル変数を作成
set z=[1 2 3] #1、2、および3を含むリストに設定
set z=z+4 #要素を追加;z now equals [1 2 3 4]
```

これらの演算子は、set コマンドを使用してスクリプト内で定義されたストリーム、スーパーノード、またはセッション パラメータとともに使用できません。または、フィールド作成ノードなどの、スクリプトの外側の一般的な CLEM 式で使用することもできません。



## スクリプト内の CLEM 式

IBM® SPSS® Modeler スクリプト内で CLEM 式、関数、および演算子を使用できますが、スクリプト内の式には、@ 関数、日付/時間関数、およびビット単位操作の呼び出しは指定できません。また、スクリプトで CLEM 式を使用する場合、次の規則が適用されます。

- パラメータは単一引用符で囲み、先頭に接頭辞 \$P- を指定する必要があります。
- CLEM 式は、引用符で囲む必要があります。CLEM 式自体に引用符で囲まれた文字列やフィールド名がある場合、それらの引用符の前には円記号 (¥) を付ける必要があります。 [詳細は、 p.23 スクリプトのシンタックス を参照してください。](#)

スクリプト内で GLOBAL\_MEAN(Age) のようなグローバル値を使用することができますが、スクリプト環境内で @GLOBAL 関数を使用することはできません。

スクリプト中で使用される CLEM 式の例を次に示します。

```
set :balancenode.directives = [{1.3 "Age > 60"}]

set :fillernode.condition = "(Age > 60) and (BP = \"High\")"

set :derivernode.formula_expr = "substring(5, 1, Drug)"

set Flag:derivernode.flag_expr = "Drug = X"

set :selectnode.condition = "Age >= '$P-cutoff'"

set :derivernode.formula_expr = "Age - GLOBAL_MEAN(Age)"
```

## コメントと継続の挿入

スクリプト内では、コメントや継続行を表すために、次の文字が使用されます。

| 文字 | 使用方法                                     | 例                |
|----|--|------------------|
| #  | ハッシュ (シャープ) 記号はコメントを表します。行の残りの部分は無視されます。 | #これは 1 行のコメントです。 |
| ¥  | バックスラッシュで終わる行は、文が次の行に継続することを表しています。      | 下の例を参照してください。    |

| 文字  | 使用方法  | 例             |
|-----|---|---------------|
| /*  | 2 の文字の組み合わせ /* は、コメントの先頭を表しています。ここから、コメントの終了を示す */ までのすべての文字列は無視されます。   | 下の例を参照してください。 |
| """ | 改行、スペース、または単一または二重引用符をブロック内に含むリテラル文字列またはブロックは、三重引用符で囲むことができます。詳細は、 <a href="#">p. 34 リテラルテキストのブロック</a> を参照してください。 |               |

### 例

```
/* これは
複数行の
コメントです
*/
```

#以下は複数行のステートメントです。

```
set :fixedfilenode.fields = [{"Age" 1 3}\
{"Sex" 5 7} {"BP" 9 10} {"Cholesterol" 12 22}\
{"Na" 24 25} {"K" 27 27} {"Drug" 29 32}]
```

## リテラル テキストのブロック

三重引用符で囲むと、スペース、タブ、および改行を含むリテラル テキスト ブロックをスクリプト内に含めることができます。スペース、改行、および埋め込まれた単一引用符および二重引用符を含む引用符で囲まれた任意のテキストは、リテラル テキストとしてそのまま扱われます。行継続記号や、エスケープ文字は必要ありません。

たとえば、このテクニックは次のように、ツリー成長ディレクティブのセットをスクリプトに含めるために使用できます。

```
set :cartnode.tree_directives = """
Create Root_Node
Grow Node Index 0 Children 1 2 SplitOn ("DRUG",
  Group ("drugA", "drugB", "drugC" )
  Group ("drugY", "drugX" ))
End Tree
"""
```

また、パス、注釈のような場合にも便利です。次に例を示します。

```
set :node.annotation = """このノードは、インディケータの
Dairy
```

```
Fish  
Vegetable  
Meat  
Pastries  
Confectionary
```

の通常とは異なる売上傾向を識別できるように構築されました""

IBM® SPSS® Modeler では、開始リテラル マーカー以降のすべての改行は無視されます。たとえば、次の例は、前の例と同じです。

```
set :node.annotation = ""
```

このノードは次のインディケータのいずれかを識別できるように作成されました。

```
Etc...
```

```
""
```

# スクリプト コマンド

このセクションは、IBM® SPSS® Modeler スクリプトで使用されるコマンドの要約です。コマンドは、オブジェクトの種類ごとに整理されています。スクリプト言語の詳細は、「3 章」を参照してください。ノード、ストリーム、プロジェクト、およびスーパーノードのプロパティの詳細は、9 章 から 22 章 を参照してください。

## 一般のスクリプト コマンド

特に指定しないかぎり、以下のコマンドは、スタンドアロン、ストリーム、スーパーノードのすべての種類のスクリプトで使用できます。

### execute\_all

```
execute_all
```

現在ストリーム内のすべてのターミナル ノードを実行します。

```
open stream "c:/demos/druglearn.str"  
execute_all
```

### execute\_script

```
execute_script
```

スタンドアロン スクリプト専用。現在ストリームに関連付けられているストリーム スクリプトを実行します（ストリーム スクリプト自体を呼び出す結果になるので、この使用は、スタンドアロン スクリプトに限定されます）。

```
open stream "c:/demos/mysample.str"  
execute_script
```

### exit

```
exit CODE
```

現在のスクリプトを終了します。exit CODE は、スクリプトやストリームまたはノードの条件を評価するのに使用できます。-以下にその例を示します。

```

create tablenode
create variablefilenode
connect :variablefilenode to :tablenode

set :variablefilenode.full_filename = "$CLEO_DEMOS/DRUG1n"
execute 'Table'

set param = value :tablenode.output at 1 1

if ^param = 23 then
  create derivenode
  else exit 2
endif

```

## for...endfor

**for...endfor** は、条件に基づいて一連のステートメントをループさせます。このコマンドには多くのフォーマットがありますが、すべて同じ一般構造に従っています。

```

for PARAMETER in LIST
  STATEMENTS
endfor

```

**for PARAMETER in LIST** : このスクリプトは、リストの順序に従って、**PARAMETER** に割り当てられた **LIST** 内の各値について、**STATEMENTS** を 1 回ずつ実行します。たとえば、次のように複数のフィールドの **Filter.include** プロパティを **true** に設定できます。

```

for f in Age Sex
  set Filter.include.^f=true
endfor

```

**for PARAMETER from N to M :N ~ M** の各整数に対して、**STATEMENTS** を 1 回ずつ実行します。以下にその例を示します。

```

for l from 1 to 5
  set :selectnode.condition = 'Age >' >> (l * 10)
  execute :selectnode
endfor

```

**for PARAMETER in\_fields\_to NODE :NODE** の上流側で、各フィールドに 1 回ずつ **STATEMENTS** を実行します。たとえば、次の例は **include** プロパティを、以前 **false** に設定されたフィールドも含めて、すべてのフィールドに **true** を設定します。

```

for f in_fields_to Filter
  set Filter.include.^f = "true"
endfor

```

注： ノードに同じ名前 - 「結合」または「レコード追加」など - の複数入力フィールドがある場合、この方法では、競合の発生を避けるために、上流ではなく下流のフィールドのリストが返されます。

**for PARAMETER in\_fields\_at NODE** : 指定の **NODE** から出力されるそれぞれのフィールド（または下流）に対して **STATEMENTS** を 1 回ずつ実行します。 ノードがフィルタの場合は、通過したフィールドのみが含まれ、フィールドは返されないの、そのノードはターミナル ノードである必要はありません。 たとえば、上の例とは反対に、次のスクリプトには何の効果もありません。このループは、すでに **true** に設定されたフィールドにのみ実行されるからです。

```
for f in_fields_at Filter
  set Filter.include.^f = "true"
endfor
```

**for PARAMETER in\_models** : [モデル] パレット内の各モデル ナゲットに対して 1 回ずつ、**STATEMENTS** を実行します。 たとえば次のスクリプトは、各モデルをパレットから現在ストリームへ挿入します（ストリーム領域で次々とノードを一番上にスタックしていくのを避けるために、**xpos** 変数が使用されます）。

```
var xpos
set xpos = 100
for m in_models
  set xpos = xpos + 100
  insert model ^m at ^xpos 100
endfor
```

**for PARAMETER in\_streams** : スタンドアロン スクリプト専用。[ストリーム] パレットに表示されているロード済みの各ストリームに対して、**STATEMENTS** を 1 回ずつ実行します。 **PARAMETER** が特殊変数 **stream** の場合、ループ中の **STATEMENTS** に現在のストリームが設定されます。 ループが終了すると、**stream** の元の値が復元されます。

## if...then...else...

```
if EXPR then
  STATEMENTS 1
else
  STATEMENTS 2
endif
```

指定された式が **true** の場合は **STATEMENTS 1** を、**false** の場合は **STATEMENTS 2** を実行します。 **else** 句はオプションです。

```
if :sampler.use_max_size = true then
  set x = "yes"
```

```
else
  set x = "no"
endif
```

## set コマンド

```
set VARIABLE = EXPRESSION
set PARAMETER = EXPRESSION
set PROPERTY = EXPRESSION
```

ローカル スクリプト変数、特殊変数、パラメータ、またはプロパティの値を設定します。

### 変数の設定

ローカル スクリプト変数に値を設定するには、**var** コマンドを使用して最初に変数を宣言します。以下にその例を示します。

```
var xpos
var ypos
set xpos = 100
set ypos = 100
```

変数の値は、スクリプト内で有効な CLEM 式、値を返すスクリプト コマンド (**load**、**create**、または **get** など)、またはリテラル値であってもかきません。

```
set xpos = ^xpos + 50
```

```
var x
set x = create typenode
```

```
var s
set s = get stream 'Druglearn'
```

### 特殊変数の参照オブジェクトへの設定

特殊変数の **node**、**stream**、**output**、および **project** は、それぞれの「現在」オブジェクトを参照するのに使用されます。 **project** の例外はありますが、上記の変数は、現在のコンテキストを変更するためにリセットできます。ほかのスクリプト変数とは異なり、これらの変数は事前に定義されているので、最初に **var** コマンドで宣言する必要はありません。

```
set node = create typenode
rename ^node as "mytypenode"
```

```
set output = get output :statisticsoutput
export output ^output as c:/myoutput.htm format html
```

これらの変数は役に立ちますが、その一方で、次の例に示すように、その使用法に些細な違いがあります。

```
set stream = get stream 'Stream7'
set ^stream.execute_method = "Script"
save stream as c:/sample7.str
close stream
```

- 最初の行で現在のストリームをリセットします。さらに詳しく説明すると、特殊変数 `stream` の値を設定します（つまり、`stream` はコマンドの一部ではなく、変数なのです）。
- 2 行目では、現在ストリームのプロパティの設定にこの変数を使用しています（プロパティの詳細は下を参照）。カレットは、`^stream` がノードなどのオブジェクトの名前ではなく変数名であることを示すために使用されています（カレットがない場合、`set` コマンドは `stream` という名前のノードを探そうとします）。
- 最後の 2 行で、現在のストリームを保存して閉じます。前の例と同様に `stream` は変数ですが、この場合は、この例で使用される `save` および `close` コマンドがストリームにだけ適用されるため、カレットが使用されません（カレットは通常、それがないとあいまいな参照になる場合にだけ使用されます）。

**現在のプロジェクトの参照：** 特殊変数の `project` は、現在のプロジェクトを参照するために使用されます（プロジェクトのプロパティを設定する例は、下を参照）。一度に 1 つのプロジェクトのみを開くことができるので（したがって、これが現在プロジェクト）、`project` をリセットすることはできません。

### パラメータの設定

ストリーム、セッション、およびスーパーノードのパラメータは、変数と同じようにして値が設定されますが、`var` コマンドは不要です。

```
set p = 1
set minvalue = 21
```

注： 実際的な観点から、`set` コマンドの右辺が宣言された変数、特殊変数、またはノードなどの名前と一致しない場合は、パラメータが作成されます。詳細は、3 章 p.30 [ストリーム、セッション、およびスーパーノード パラメータ](#) を参照してください。

### ノード、ストリーム、およびプロジェクトのプロパティの設定

ノード、ストリーム、およびプロジェクトのプロパティも設定できます。以下にその例を示します。

```
set :variablefilenode.full_filename = "$CLEO_DEMOS/DRUG1n"
```



```
set ^stream.execute_method = "Script"
```

```
load project "C:/myproject.cpj"
set ^project.structure = Phase
```

ノード、ストリーム、およびプロジェクトに利用できるプロパティの完全なリストは、「[プロパティ参照](#)」( p.126 ) を参照してください。

**複数のプロパティの設定**：単一の操作で、ノードやその他のオブジェクトのプロパティに複数の式を割り当てることができます。この方法は、データ モデルを決定する前に、ノードに複数の変更を行う必要がある場合に使用されます。複数のプロパティを設定するには、次のフォーマットを使用します。

```
set NODE {
  NODEPROPERTY1 = EXPRESSION1
  NODEPROPERTY2 = EXPRESSION2
}
```

次に例を示します。

```
set :samplene {
  max_size = 200
  mode = "Include"
  sample_type = "First"
}

set ^project {
  summary = "Initial modeling work on the latest data"
  ordering = NameAddedType
}
```

**フラグ値の設定 (真または偽)**：フラグ型プロパティの読み込みまたは書き込み時に、値 `true` および `false` は小文字でなければなりません。以下にその例を示します。

```
set :variablefilene.read_field_names = true
```

注： `Off`、`OFF`、`off`、`No`、`NO`、`no`、`n`、`N`、`f`、`F`、`false`、`False`、`FALSE`、または `0` なども値の設定時に認識されますが、プロパティ値の読み込み時にエラーが発生する場合があります。その他の値はすべて真と見なされます。`true` と `false` を使用すると、こうした混乱が避けられます。

### 例: ノードのプロパティの設定

各ノードのユーザー インターフェイスのダイアログ ボックスに表示されるオプションを設定するための、さまざまなノード固有のプロパティ (スロット パラメータと呼ばれることもある) が用意されています。たとえば、ストリームを作成して各ノードのオプションを指定するには、次のようなス

クリプトを使用します。 ノード、ストリーム、プロジェクト、およびスーパーノードのプロパティの詳細は、9 章 から 22 章 を参照してください。

```
create varfilenode at 100 100
set :varfilenode {
  full_filename = "demos/drug1n"
  read_field_names = true
}
create tablenode at 400 100
create samplenode connected between :varfilenode and :tablenode
set :samplenode {
  max_size = 200
  mode = "Include"
  sample_type = "First"
}
create plotnode at 300 300
create derivenode connected between drug1n and :plotnode
set :derivnode {
  new_name = "Ratio of Na to K"
  formula_expr = "'Na' / 'K'"
}
set :plotnode {
  x_field = 'Ratio of Na to K'
  y_field = 'Age'
  color_field = 'BP'
}
```

## var コマンド

```
var VARNAME
```

ローカル スクリプト変数を宣言します。

```
var my_node
set my_node = create distributionnode
rename ^my_node as "Distribution of Flag"
```

変数は、パラメータとは異なります。パラメータは、セッション、ストリーム、またはスーパーノードに設定でき、文字列または数値だけを含むことができます。実際には、**VAR** コマンドを使用して最初に宣言しないで変数を設定した場合、現在スクリプトのコンテキストに応じて、ストリーム、セッション、またはスーパーノードのパラメータが作成されます。[詳細は、3 章 p.29 ローカル スクリプト変数 を参照してください。](#)

## ノード オブジェクト

次のスクリプト コマンドは、ノード オブジェクトに対して使用できます。

## create NODE

```
create NODE
create NODE at X Y
create NODE between NODE1 and NODE2
create NODE connected between NODE1 and NODE2
```

指定された種類のノードを作成します。以下にその例を示します。

```
create statisticsimportnode
```

オプションで、位置と接続のオプションも指定できます。

```
create featureselectionnode at 400 100
```

```
create typenode between :statisticsimportnode and :featureselectionnode
```

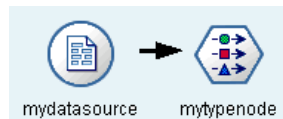
```
create selectnode connected between :typenode and :featureselectionnode
```

あいまいさを避けるために、変数を使用してノードを作成することもできます。たとえば、以下の例ではデータ型ノードが作成され、そのデータ型ノードへの参照を含む参照変数 `x` が設定されます。この変数 `x` を使用して、`x` が参照するオブジェクト（この例ではデータ型ノード）を返し、名前変更、位置設定、または新規ノードの接続などの操作を実行することができます。

```
var x
set x = create typenode
rename ^x as "mytypenode"
position ^x at 200 200
var y
set y = create varfilenode
rename ^y as "mydatasource"
position ^y at 100 200
connect ^y to ^x
```

この例では 2 つのノードが作成され、その後名前の変更と位置設定を行い、最後にそれらをストリーム領域上で接続します。

図 4-1  
変数を使用して作成されたノード



または、特殊（事前に定義された）変数の `node` は、上の例の変数 `x` と同じように使用することができます。この場合、変数は `var` コマンドで宣言する必要がなく（事前に定義されているため）、結果のスクリプトは多少読みやすくなります。

```
set node = create typenode
rename ^node as "mytypenode"
position ^node at 200 200
set node = create varfilenode
rename ^node as "mydatasource"
position ^node at 100 200
connect mydatasource to mytypenode
```

注： `node` のような特殊変数を再利用して、複数のノードを参照することができます。変数が参照するオブジェクトをリセットするには、単に `set` コマンドを使用します。詳細は、[3 章 p.27 現在のオブジェクトの設定](#) を参照してください。

**ノードの複製**：既存のノードを複製するには、`duplicate` コマンドも使用できます。詳細は、[p.45 duplicate NODE](#) を参照してください。

## connect NODE

```
connect NODE1 to NODE2
connect NODE1 between NODE2 and NODE3
```

`NODE1` を指定されたとおりにほかのノードに接続します。

```
connect :statisticsimportnode to :typenode
connect :selectnode between :typenode and :featureselectionnode
```

## delete NODE

```
delete NODE
```

指定したノードを現在のストリームから削除します。

```
delete :statisticsimportnode
delete DRUG1N:variablefilenode
```

## disable NODE

```
disable NODE
```

現在のストリームの指定されたノードを無効化します。ストリームの実行時、ノードは無視されます。これにより、ノードを削除またはバイパスする必要がなくなり、残りのノードを接続したままにできます。ノード設定を編集することもできますが、ノードを再度有効化した後で変更が有効となります。

```
disable :statisticsimportnode
```

```
disable DRUG1N:variablefilenode
```

## disconnect NODE

```
disconnect NODE
```

```
disconnect NODE1 from NODE2
```

```
disconnect NODE1 between NODE2 and NODE3
```

指定されたノードをほかのすべてのノード（デフォルト）または指定した特定のノードから切り離します。

```
disconnect :typenode
```

```
disconnect :typenode from :selectnode
```

## duplicate NODE

```
duplicate NODE as NEWNAME
```

指定されたノードの複製として、新しいノードを作成します。オプションで、位置も、絶対的な位置または相対的な位置を指定できます。

```
duplicate :derivenode as flag1 at 100 400
```

```
duplicate flag1 as flag2 connected between flag1 and flag3
```

## enable NODE

```
enable NODE
```

現在のストリームの指定されたノードを有効化します。ストリームの実行時、ノードは使用されます。無効化されたノードの設定を編集すると、変更が有効となります。

```
enable :statisticsimportnode
```

```
enable DRUG1N:variablefilenode
```

## execute NODE

```
execute NODE
```

指定されたノードを実行します。-以下にその例を示します。

```
execute :neuralnetworknode
```

ノードがターミナル ノードでない場合は、ポップアップ メニューの [ここから実行] を選択した場合と同じようにストリームが実行されます。

現在ストリーム内のすべてのターミナル ノードを実行するには

```
execute_all
```

スタンドアロン スクリプト専用。現在ストリームに関連付けられているストリーム スクリプトを実行します。

```
execute_script
```

注：別のストリームに関連付けられたスクリプトを実行するには、**with** コマンドを使用して、該当するストリームを現在のストリームとして設定します。 [詳細は、3 章 p.28 複数ストリームの作業](#) を参照してください。

## export NODE as FILE

```
export node NODE in DIRECTORY format FORMAT
```

```
export node NODE as FILE format FORMAT
```

**PMML のエクスポート**： PMML フォーマットで生成されたモデルをエクスポートするには

```
export Drug as c:/mymodel.txt format pmml
```

**SQL のエクスポート**： SQL フォーマットで生成されたモデルをエクスポートするには

```
export Drug in c:/mymodels format sql
```

```
export Drug as c:/mymodel.txt format sql
```

**ノードの詳細**： HTML またはテキスト フォーマットでノードの詳細をエクスポートするには

```
export Drug as c:\mymodel.htm format html
```

```
export Drug as c:\mymodel.txt format text
```

**ノードの要約**： HTML またはテキスト フォーマットでノードの要約をエクスポートするには

```
export Drug summary in c:/mymodels format html
```

```
export Drug summary as c:/mymodel.txt format text
```

```
export 'assocapriori' as 'C:/temp/assoc_apriori' format html
```

## flush NODE

```
flush NODE
```

ストリーム内の指定されたノードまたはすべてのノードのキャッシュをフラッシュします。指定されたノードのキャッシュが有効になっていない場合、または一杯でない場合は、この操作で何も行われません。

```
flush :mergenode
```

現在ストリーム内のすべてのノードをフラッシュするには

```
flush_all
```

## get node NODE

```
get node NODE
```

既存のノードへの参照を取得します。これは、ノードへのあいまいさがないう参照を確実なものとするために役立つ方法です。

```
var mynode  
set mynode = get node flag1:derivenode  
position ^mynode at 400 400
```

## load node FILENAME

```
load node FILENAME
```

保存されたノードを現在のストリームへ読み込みます。

```
load node c:/mynode.nod
```

## position NODE

```
position NODE at X Y  
position NODE between NODE1 and NODE2  
position NODE connected between NODE1 and NODE2
```

ストリーム領域上でノードの位置を絶対位置または相対位置の意味で決めます。オプションで、接続オプションも指定できます。

```
position DRUG1n:variablefilenode at 100 100
```

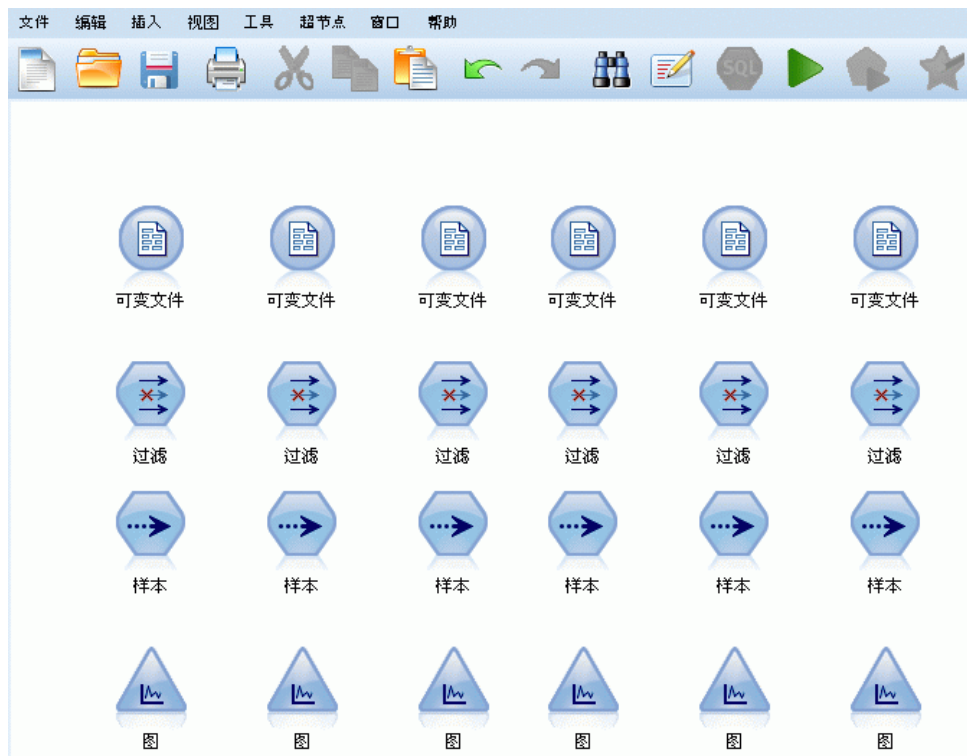
```
position Drug:net between DRUG2n and analysis
```

```
position :typenode connected between :variablefilenode and :tablenode
```

## 座標の位置設定

ストリーム領域上のノードの配置には、表示されない x-y グリッドが使用されます。 x-y グリッド座標の参照として、次の図を参照してください。

図 4-2  
x-y 座標で示す位置に作成、配置されたノード



## rename NODE as NEWNAME

```
rename NODE as NEWNAME
```

指定されたノードの名前を変更します。

```
rename :derivenode as 'Flag1'
```

```
rename :varfilenode as 'testdata'
```

## ノードの REPOSITORY\_PATH の取得

```
retrieve node REPOSITORY_PATH {label LABEL | version VERSION}
```



指定されたノードを IBM® SPSS® Collaboration and Deployment Services Repository から取得します。詳細は、5 章 p.67 IBM SPSS Collaboration and Deployment Services Repository 内のオブジェクトへのアクセスを参照してください。

```
retrieve node "/samples/drugtypenode"
```

## save node NODE as FILENAME

```
save node NODE as FILENAME
```

指定されたノードを保存します。

```
save node :statisticsimportnode as c:/mynode.nod
```

## store node NODE as REPOSITORY\_PATH

```
store node NODE as REPOSITORY_PATH {label LABEL}
```

IBM® SPSS® Collaboration and Deployment Services Repository にノードを格納します。詳細は、5 章 p.67 IBM SPSS Collaboration and Deployment Services Repository 内のオブジェクトへのアクセスを参照してください。

```
store node DRUG1n as "/samples/drug1ntypenode"
```

```
store node :typenode as "/samples/drugtypenode"
```

## モデル オブジェクト

次のスクリプト コマンドは、モデル オブジェクトに対して利用できます。

### モデル ナゲット名

モデル ナゲット（生成されたモデル）は、ノード オブジェクトと出力オブジェクトと同様に、その種類で参照できます。次の表に、モデル オブジェクトの参照名を一覧表示します。

これらの名前は、IBM® SPSS® Modeler ウィンドウの右上隅にある [モデル] パレット内のモデル ナゲットを参照するために、特に使用されます。スコアリングの目的でストリームに追加されたモデル ノードを参照するには、`apply...` の接頭辞が付いた別の名前セットが使用されます。詳細は、17 章 p.275 [モデル ナゲット ノードのプロパティ](#) を参照してください。

たとえば、次のスクリプトは、モデル ナゲットを現在のストリームへ追加し、それをデータ型ノードに接続して、テーブル ノードを作成して実行します。ストリームに追加された “apply” 付きのモデル ノードを参

照するのに使用される名前と区別するために、パレットからモデルを挿入するのに使用される別の名前に留意してください (:featureselection と :applyfeatureselectionnode)。

```
insert model :featureselection at 150 250
connect Type to :applyfeatureselectionnode
create tablenode at 250 250
connect :applyfeatureselectionnode to :tablenode
execute :tablenode
```

注：これは、例専用のコードです。通常の状態では、名前および種類の両方でモデルを参照することが、混乱を避けるために推奨されます（たとえば、response\_01:featureselection のように）。

### モデル ナゲット名 ([モデル作成] パレット)

| モデル名             | Model        |
|------------------|--------------|
| anomalydetection | 異常値          |
| apriori          | Apriori      |
| autoclassifier   | 自動分類         |
| autocluster      | 自動クラスタリング    |
| autonumeric      | 自動数値         |
| bayesnet         | ベイズ ネットワーク   |
| c50              | C5.0         |
| carma            | Carma        |
| cart             | C&R Tree     |
| chaid            | CHAID        |
| coxreg           | Cox 回帰       |
| decisionlist     | ディシジョン リスト   |
| discriminant     | 判別分析         |
| factor           | 因子分析         |
| featureselection | フィールド選択      |
| genlin           | 一般化線型回帰      |
| kmeans           | K-Means      |
| knn              | k 最近隣        |
| kohonen          | Kohonen      |
| linear           | Linear       |
| logreg           | ロジスティック回帰    |
| neuralnetwork    | ニューラル ネットワーク |
| quest            | QUEST        |
| regression       | 線型回帰         |
| sequence         | シーケンス        |

| モデル名            | Model                     |
|-----------------|---------------------------|
| slrm            | 自己学習応答モデル                 |
| statisticsmodel | IBM® SPSS® Statistics モデル |
| svm             | Support Vector Machine    |
| timeseries      | 時系列                       |
| twostep         | TwoStep                   |

### モデル ナゲット名 ([データベース モデリング] パレット)

| モデル名              | Model                  |
|-------------------|------------------------|
| db2imassoc        | IBM ISW アソシエーション       |
| db2imcluster      | IBM ISW クラスタリング        |
| db2imreg          | IBM ISW 回帰             |
| db2imsequence     | IBM ISW シーケンス          |
| db2imtree         | IBM ISW ディシジョン ツリー     |
| msassoc           | MS アソシエーション ルール        |
| msbayes           | MS Naive Bayes         |
| mscluster         | MS クラスタリング             |
| mslogistic        | MS Logistic Regression |
| msneuralnetwork   | MS Neural Network      |
| msregression      | MS Linear Regression   |
| mssequencecluster | MS シーケンス クラスタリング       |
| mstimeseries      | MS タイム シリーズ            |
| mstree            | MS ディシジョン ツリー          |
| oraabn            | Oracle Adaptive Bayes  |
| oraai             | Oracle AI              |
| oraapriori        | Oracle Apriori         |
| oradecisiontree   | Oracle Decision Tree   |
| oraglm            | Oracle GLM             |
| orakmeans         | Oracle k-Means         |
| oramdl            | Oracle MDL             |

| モデル名        | Model              |
|-------------|--------------------|
| oranb       | Oracle Naive Bayes |
| oranmf      | Oracle NMF         |
| oraocluster | Oracle O-Cluster   |
| orasvm      | Oracle SVM         |

## 重複するモデル名の回避

生成されたモデルを操作するのにスクリプトを使用する場合、重複するモデル名を使用していると、スクリプトがあいまいになることに注意する必要があります。これを避けるために、スクリプト作成時に、生成されたモデルには一意の名前を使用することをお勧めします。

重複するモデル名に関するオプションを設定するには

- ▶ メニューから次の項目を選択します。  
ツール > [ユーザー オプション]
- ▶ [通知] タブをクリックします。
- ▶ 生成されたモデルに対して重複する名前を禁止するには、[前のモデルを置換] を選択します。

あいまいなモデルの参照がある場合、スクリプト実行の動作は SPSS Modeler と IBM SPSS Collaboration and Deployment Services との間で異なります。SPSS Modeler クライアントには自動的に同じ名前を持つモデルを置き換えるオプション「以前のモデルを置き換える」があります（たとえば、スクリプトをループで反復して随時異なる名前を作成）。しかし、このオプションは、同じスクリプトが IBM SPSS Collaboration and Deployment Services で実行される場合は使用できません。ループの終了前に、モデルへの曖昧な参照を回避するために各反復で生成されるモデルの名前を変更したり、現在のモデルをクリアする（例：`clear generated palette` 文を追加する）などして、この状況を回避することができます。

## delete model MODEL

delete model MODEL

モデル ナゲット パレットから指定されたモデルを削除（または、すべてのモデルをクリア）します。

delete model Drug

delete model Drug:c50

現在のスクリプトによって挿入された最後のモデルを削除するには

```
delete last model
```

この最後のステートメントが機能するには、現在のスクリプトの実行で少なくとも 1 回、**insert model** ステートメントが実行されている必要があります。

[モデル] パレットからすべてのモデル ナゲットをクリアするには

```
clear generated palette
```

## export model MODEL as FILE

```
export model MODEL in DIRECTORY format FORMAT
```

```
export model MODEL as FILE format FORMAT
```

**PMML のエクスポート:** PMML フォーマットで生成されたモデルをエクスポートするには

```
export model Drug in c:/mymodels format pmml
```

```
export model Drug as c:/mymodel.xml format pmml
```

詳細は、[10 章 PMML としてのモデルのインポートおよびエクスポート in IBM SPSS Modeler 15 ユーザー ガイド](#) を参照してください。

**SQL のエクスポート:** SQL フォーマットで生成されたモデルをエクスポートするには

```
export Drug in c:/mymodels format sql
```

```
export Drug as c:/mymodel.txt format sql
```

注: SQL のエクスポートは、特定のモデル タイプにのみ利用可能です。詳細は、[3 章 モデル ナゲットの参照 in IBM SPSS Modeler 15 モデル作成ノード](#) を参照してください。

**モデル詳細:** モデル ナゲットの参照時に [モデル] タブ内に表示されるとおりのモデルの詳細を HTML またはテキスト フォーマットでエクスポートするには

```
export model Drug as c:\mymodel.htm format html
```

```
export model Drug as c:\mymodel.txt format text
```

注: [モデル] タブのないモデルには、これらのフォーマットを利用できません。

**モデルの要約.** モデル ナゲットの参照時に [要約] タブ内に表示されるとおりにモデルの要約を HTML またはテキスト フォーマットでエクスポートするには以下のようにします。

export model Drug summary in c:/mymodels format html

export model Drug summary as c:/mymodel.txt format text

export model 'assocapriori' as 'C:/temp/assoc\_apriori' format html

図 4-3  
HTML フォーマットでエクスポートされたアソシエーション モデルのタブ

|    | 結果   | 前提条件         | サポート % | 確信度 %  |
|----|------|--------------|--------|--------|
| 1  | 冷凍肉  | ビール and 缶詰野菜 | 16.7   | 87.425 |
| 2  | 缶詰野菜 | ビール and 冷凍肉  | 17.0   | 85.882 |
| 3  | ビール  | 冷凍肉 and 缶詰野菜 | 17.3   | 84.393 |
| 4  | 冷凍肉  | ビール          | 29.3   | 58.02  |
| 5  | 缶詰野菜 | 冷凍肉          | 30.2   | 57.285 |
| 6  | 冷凍肉  | 缶詰野菜         | 30.3   | 57.096 |
| 7  | 缶詰野菜 | ビール          | 29.3   | 56.997 |
| 8  | ビール  | 冷凍肉          | 30.2   | 56.291 |
| 9  | ビール  | 缶詰野菜         | 30.3   | 55.116 |
| 10 | ワイン  | 菓子           | 27.6   | 52.174 |
| 11 | 菓子   | ワイン          | 28.7   | 50.174 |

## insert model MODEL

insert model MODEL

insert model MODEL at X Y

insert model MODEL between NODE1 and NODE2

insert model MODEL connected between NODE1 and NODE2

現在のストリームにモデルを追加します。 オプションで、位置と接続のオプションも指定できます。

insert model Kohonen between :typenode and :analysisnode

insert model Drug:neuralnetwork connected between 'Define Types' and 'Analysis'

## load model FILENAME

load model FILENAME

保存されたモデルを [モデル] パレットへ読み込みます。

```
load model c:/mymodel.gm
```

## retrieve model REPOSITORY\_PATH

```
retrieve model REPOSITORY_PATH {label LABEL | version VERSION}
```

IBM® SPSS® Collaboration and Deployment Services Repository から、保存されたモデルを取得します。 [詳細は、5 章 p.67 IBM SPSS Collaboration and Deployment Services Repository 内のオブジェクトへのアクセス を参照してください。](#)

```
retrieve model "/my folder/Kohonen.gm"
```

## save model MODEL as FILENAME

```
save model MODEL as FILENAME
```

指定されたモデルを、生成済みモデル ファイルとして保存します。

```
save model Drug as c:/mymodel.gm
```

## store model MODEL as REPOSITORY\_PATH

```
store model MODEL as REPOSITORY_PATH {label LABEL}
```

指定されたモデルを IBM® SPSS® Collaboration and Deployment Services Repository に格納します。 [詳細は、5 章 p.67 IBM SPSS Collaboration and Deployment Services Repository 内のオブジェクトへのアクセス を参照してください。](#)

```
store model Kohonen as "/my folder/Kohonen.gm"
```

拡張子 (\*. gm) は省略できますが、モデルを格納したり取得したりする場合は必ず使用する必要があります。たとえば、ただ単に “Kohonen” として格納されている場合、そのモデルは同じ名前を取得する必要があります (別の方法では、拡張子が使用されている場合、それはモデル名の単なる一部です)。

## ストリーム オブジェクト

次のスクリプト コマンドは、ストリーム オブジェクトに対して利用できます。

## create stream DEFAULT\_FILENAME

```
create stream DEFAULT_FILENAME
```

スタンドアロン スクリプト専用。指定された名前で、メモリー内に新しいストリームを作成します。このストリームは、自動的に保存されません。

```
create stream 'Druglearn'
```

## close STREAM

```
close STREAM
```

スタンドアロン スクリプト専用。指定されたストリームを閉じます。

現在のストリームを閉じるには、次のように、すべて小文字を使用して、コマンドを入力します。

```
close stream
```

### スタンドアロン スクリプト

複数のストリームで作業している場合、**stream**（このように、小文字）は、実際は現在のストリームを参照するために使用される特殊変数です。別のストリームを閉じるために、この変数の値を再設定できます。

```
set stream = get stream 'Stream5'  
close stream
```

それとは別に、ストリームを参照する任意の宣言済み変数も指定できます。以下にその例を示します。

```
var s  
set s = get stream 'Stream2'  
save s as c:/stream2.str  
close s
```

最終的には、**with stream** コマンドを使用して、現在のストリームが一時的に転換されます。

```
with stream 'Stream1'  
close stream  
endwith
```

## clear stream

```
clear stream
```

すべてのノードを現在のストリームから削除します。



## get stream STREAM

```
get stream STREAM
```

スタンドアロン スクリプト専用。指定されたストリームへの参照を入手するために使用されます。これは、ローカル変数（または特殊変数の **stream**）へ割り当てることができます。指定されたストリームは、すでに開いてある必要があります。

```
var s
set s = get stream 'Druglearn'
close s
```

## load stream FILENAME

```
load stream FILENAME
```

スタンドアロン スクリプト専用。現在のストリームのノードをクリアしないで、指定されたストリームをストリーム領域へ追加します。

```
load stream "c:/demos/druglearn.str"
```

**ストリームを開くこととストリームを読み込むことの対比** :load stream コマンドは現在のストリームのノードをクリアしないで、指定されたストリームをストリーム領域へ追加します。このコマンドは、IBM® SPSS® Modeler の初期のリリースでは現在より機能が多かったのですが、複数のストリーム間でのノードの展開、管理、コピーの機能によって、以後のリリースでは、機能が大きく縮小されました。

## open stream FILENAME

```
open stream FILENAME
```

スタンドアロン スクリプト専用。指定されたストリームを開きます。

```
open stream "c:/demos/druglearn.str"
```

## retrieve stream REPOSITORY\_PATH

```
retrieve stream REPOSITORY_PATH {label LABEL | version VERSION}
retrieve stream URI [{#m.marker | #l.label}]
```

IBM® SPSS® Collaboration and Deployment Services Repository から指定のストリームを取得します。詳細は、[5 章 p.67 IBM SPSS Collaboration and Deployment Services Repository 内のオブジェクトへのアクセスを参照してください。](#)

```
retrieve stream "myfolder/druglearn.str"
```

```
retrieve stream "spsscr:///models/drug%20model.gm#m.0:2005-10-12%2014:15:41.281"
```

## save STREAM as FILENAME

```
save STREAM  
save STREAM as FILENAME
```

現在のストリームに対する変更を保存するには（ストリームは以前保存されていたと想定）、次のように、すべて小文字を使用して、コマンドを入力します。

```
save stream
```

新しいファイル名で初めてストリームを保存するには

```
create stream nifty  
create featureselectionnode  
save stream as c:/nifty.str
```

## スタンドアロン スクリプト

スタンドアロン スクリプト内の複数のストリームで作業している場合は、**stream**（上例のように、小文字）は、実際は現在のストリームを参照するために使用される特殊変数であることに注意してください。別のストリームを保存するために、この変数の値を再設定できます。

```
set stream = get stream 'Stream5'  
save stream
```

それとは別に、ストリームを参照する任意の宣言済み変数も指定できます。以下にその例を示します。

```
var s  
set s = get stream 'Stream2'  
save s as c:/stream2.str  
close s
```

最終的には、**with stream** コマンドを使用して、現在のストリームが一時的に転換されます。

```
with stream 'Stream1'  
save stream  
endwith
```

詳細は、[3 章 p. 28 複数ストリームの作業](#) を参照してください。

## store stream as REPOSITORY\_PATH

```
store stream as REPOSITORY_PATH {label LABEL}
store stream as URI [#l.label]
```

```
store stream as "/folder_1/folder_2/mystream.str"
```

現在のストリームを IBM® SPSS® Collaboration and Deployment Services Repository に格納します。詳細は、[5 章 p.67 IBM SPSS Collaboration and Deployment Services Repository 内のオブジェクトへのアクセスを参照してください。](#)

```
store stream as "/folder_1/folder_2/druglearn.str"
store stream as "spsscr:///folder_1/folder_2/mystream.str"
```

## スタンドアロン スクリプト

スタンドアロン スクリプト内の複数のストリームで作業している場合は、**stream**（上例のように、小文字）は、実際は現在のストリームを参照するために使用される特殊変数であることに注意してください。別のストリームを格納するために、この変数の値を再設定できます。

```
set stream = get stream 'Stream5'
store stream as "/folder_1/mystream.str"
```

それとは別に、ストリームを参照する任意の宣言済み変数を指定したり、あるいは **with stream** コマンドを使用して現在のストリームを一時的に転換したりできます。

```
with stream 'Stream6'
store stream as "/folder_1/mystream.str"
endwith
```

## with stream STREAM

```
with stream STREAM
STATEMENTS
endwith
```

スタンドアロン スクリプト専用。現在のストリームとして設定された、指定された **STREAM** で、**STATEMENTS** を実行します。ステートメントが実行されると、元のストリームが現在のストリームに戻ります。

```
with stream 'druglearn'
create typenode
execute_script
endwith
```

## プロジェクト オブジェクト

次のスクリプト コマンドは、プロジェクト オブジェクトに対して利用できます。

拡張子 (\*. gm) は省略できますが、ある特定のプロジェクトを格納したり取得したりする場合は必ず使用する必要があります。

### execute\_project

```
execute_project
```

現在のプロジェクトのデフォルトのレポートを生成します。

### load project FILENAME

```
load project FILENAME
```

指定されたプロジェクトを開きます。

```
load project "C:/clemdata/DrugData.cpj"  
set ^project.summary="Initial modeling work on the latest data."  
set ^project.ordering=NameAddedType  
execute_project
```

### retrieve project REPOSITORY\_PATH

```
retrieve project REPOSITORY_PATH {label LABEL | version VERSION}
```

IBM® SPSS® Collaboration and Deployment Services Repository からプロジェクトを取得します。 [詳細は、5 章 p.67 IBM SPSS Collaboration and Deployment Services Repository 内のオブジェクトへのアクセスを参照してください。](#)

```
retrieve project "/CRISPDM/DrugExample.cpj"
```

### save project as FILENAME

```
save project  
save project as FILENAME
```

現在のプロジェクトを保存します。

### store project as REPOSITORY\_PATH

```
store project as REPOSITORY_PATH {label LABEL}
```

現在のプロジェクトを IBM® SPSS® Collaboration and Deployment Services Repository に格納します。詳細は、5 章 p.67 IBM SPSS Collaboration and Deployment Services Repository 内のオブジェクトへのアクセスを参照してください。

```
store project as "/CRISPDM/DrugExample.cpj"
```

## ステート型オブジェクト

保存されたステートは、`load state` コマンドを使用して読み込むことができます。

### load state FILENAME

```
load state FILENAME
```

指定されたステートを読み込みます。

```
load state "c:/data/myproject.cst"
```

## 結果オブジェクト

結果には、`value` コマンドを使用してアクセスできます。

### value RESULT

```
value RESULT at ROW COLUMN
```

ターミナル ノードには、前回生成されたオブジェクトにアクセスできるようにする、`output` と呼ばれる読み込み専用パラメータがあります。行と列の表形式の出力を作成するノードの場合、このパラメータで、指定されたセルの値にアクセスすることができます。以下にその例を示します。

```
execute :tablenode
set last_row = :tablenode.output.row_count
set last_column = :tablenode.output.column_count
set last_value = value :tablenode.output at ^last_row ^last_column
var myresults
set myresults = open create 'C:/myresults.txt'
write myresults 'The value in the last cell is ' << ^last_value
```

行と列は 1 からのオフセットで指定します。出力オブジェクトが存在しない場合は、エラーが返されます。

### 結果オブジェクトのプロパティ

次のプロパティは、行と列に値が含まれる結果オブジェクト（テーブル、クロス集計など）に共通しています。

| プロパティ        | 説明            |
|--------------|---------------|
| row_count    | データ中の行数を返します。 |
| column_count | データ中の列数を返します。 |

## ファイル オブジェクト

次のスクリプト コマンドは、ファイル オブジェクトに対して使用できます。

### close FILE

```
close FILE
```

上のステートメントは、指定されたファイルを閉じます。

### open FILE

```
open create FILENAME  
open append FILENAME
```

上のステートメントは、指定されたファイルを開きます。

- **作成**：ファイルが存在しない場合はファイルを作成、すでにファイルが存在している場合は上書きします。
- **レコード追加**：既存のファイルに追加します。ファイルが存在しない場合はエラーになります。

これは、開かれているファイルのファイル ハンドルを返します。

```
var file  
set file = open create 'C:/script.out'  
for l from 1 to 3  
  write file 'Stream ' >> l  
endfor  
close file
```

### write FILE

```
write FILE TEXT_EXPRESSION  
writeln FILE TEXT_EXPRESSION
```

上記の式は、テキストをファイルに書き込みます。最初のステートメントは、テキストをそのまま書き込みますが、2 番目のステートメントは、式を書き込んだ後に改行コード（復帰改行）を書き込みます。FILE が開かれているファイル オブジェクトではない場合は、エラーが発生します。

```
var file
set file = open create 'C:/hello.txt'
writeln file 'Hello'
writeln file 'World'
write file 'Would you like to play a game?'
close file
```

## 出力オブジェクト

次のスクリプト コマンドは、出力オブジェクトに対して利用できます。

### 出力形式名

次の表に、すべての出力オブジェクトの形式と、それを作成するノードを一覧表示します。出力オブジェクトの各タイプで利用できるエクスポート形式の完全なリストは、出力形式を作成するノードのプロパティの説明を参照してください。15 章「グラフ作成ノードのプロパティ」および19 章「出力ノードのプロパティ」で利用できます。

| 出力オブジェクトの種類               | ノード   |
|---------------------------|---|
| analysisoutput            | 分析  |
| collectionoutput          | 集計棒グラフ  |
| dataauditoutput           | データ検査   |
| distributionoutput        | 分布  |
| evaluationoutput          | 評価  |
| histogramoutput           | Histogram                                       |
| matrixoutput              | クロス集計   |
| meansoutput               | 平均  |
| multiplotoutput           | 線グラフ  |
| plotoutput                | プロット  |
| qualityoutput             | 欠損値検査   |
| reportdocumentoutput      | このオブジェクトの種類はノードからのものではなく、プロジェクト レポートに作成された出力です。 |
| reportoutput              | レポート  |
| statisticsprocedureoutput | Statistics 出力                                   |

| 出力オブジェクトの種類      | ノード    |
|------------------|--------|
| statisticsoutput | 統計値    |
| tableoutput      | テーブル   |
| timeplotoutput   | 時系列グラフ |
| weboutput        | Web    |

## delete output OUTPUT

```
delete output OUTPUT
```

指定した出力をマネージャ パレットから削除します。次に例を示します。

```
delete output :statisticsoutput
```

すべての出力項目をマネージャ パレットから削除するには

```
clear outputs
```

## export output OUTPUT

```
export output OUTPUT as FILE format FORMAT
```

指定されたフォーマットで、出力をエクスポートします。利用可能なフォーマットは出力の種類に応じて異なりますが、指定された出力の参照時に、[エクスポート] メニューに利用可能なフォーマットが表示されます。

```
export output :statisticsoutput as "C:/output/statistics.html" format html
```

```
export output :matrixoutput as "C:/output/matrix.csv" format delimited
```

```
export output :tableoutput as "C:/output/table.tab" format transposed formatted
```

## get output OUTPUT

```
get output OUTPUT
```

指定された出力への参照を入手します。たとえば、一連の出力オブジェクトを入手してそれぞれを順々にエクスポートするために、ループが使用されます。

```
execute_all
```

```
for item in statisticsoutput matrixoutput tableoutput
```

```
var theoutput
```

```
set theoutput = get output :^item
```

```
set filename = 'c:/><^item ><'.htm'
```

```
export output ^theoutput as ^filename format html
```

```
endfor
```



## load output FILENAME

```
load output FILENAME
```

指定された出力を読み込みます。

```
load output 'c:/matrix.cou'
```

## retrieve output REPOSITORY\_PATH

```
retrieve output REPOSITORY_PATH {label LABEL | version VERSION}
```

指定された出力を IBM® SPSS® Collaboration and Deployment Services Repository から取得します。詳細は、[5 章 p.67 IBM SPSS Collaboration and Deployment Services Repository 内のオブジェクトへのアクセス](#) を参照してください。

```
retrieve output "/results/mytable"
```

## save output OUTPUT as FILENAME

```
save output as FILENAME
```

指定された出力を保存します。

```
save output :matrixoutput as 'c:/matrix.cou'
```

## store output OUTPUT as REPOSITORY\_PATH

```
store output OUTPUT as REPOSITORY_PATH {label LABEL}
```

指定された出力を IBM® SPSS® Collaboration and Deployment Services Repository に格納します。詳細は、[5 章 p.67 IBM SPSS Collaboration and Deployment Services Repository 内のオブジェクトへのアクセス](#) を参照してください。

```
store output "Data Audit of [6 fields]" as "/my folder/My Audit"
```

```
store output :tableoutput as "/results/mytable"
```

# スクリプトのヒント

このセクションでは、スクリプトのヒントと使い方について概要を説明します。これには、ストリームの実行を修正したり、スクリプトで暗号化されたパスワードを使用したり、また、IBM® SPSS® Collaboration and Deployment Services Repository でオブジェクトにアクセスしたりする作業が含まれます。

## ストリーム実行の変更

ストリームを実行すると、ターミナル ノードがデフォルトの状況に最適化された順番で実行されます。状況に応じて、別の順序で実行させることもできます。ストリームの実行順序を変更するには、[ストリームのプロパティ] ダイアログ ボックスの [スクリプト] タブから、次の作業を行ってください。

- ▶ 空のスクリプトを用意します。
- ▶ ツールバーの [デフォルト スクリプトの追加] ボタンをクリックして、デフォルトのストリーム スクリプトを追加します。
- ▶ デフォルト ストリーム スクリプトの文の順序を、実際に実行する順序に変更します。

## ノードのループ

for ループと `^stream.nodes` プロパティを併用してストリーム内のすべてのノードをループできます。たとえば、以下のスクリプトはすべてのノードをループし、フィルタ ノードにおけるフィールド名を大文字に変更します。

たとえ除外されるフィールドが何もなくても、このスクリプトはフィルタ ノードを持つどのようなストリームにおいても使用できます。フィールド名を全面的に大文字に変更するには、すべてのフィールドを渡すフィルタ ノードをただ単に追加するだけです。

```
var my_node
var loop_me
var var_name
```

```
for my_node in ^stream.nodes
  if ^my_node.node_type = filternode then
    for loop_me in_fields_to ^my_node:filternode
      set var_name = lowertoupper(^my_node:filternode.new_name.^loop_me)
```

```
set ^my_node.filternode.new_name.^loop_me = ^var_name
endfor
else
endif
endfor
```

スクリプトは現在のストリーム内のすべてのノードをループし、`^stream.nodes` プロパティによって返されると、各ノードが Filter であるかどうかをチェックします。各ノードが Filter ならば、スクリプトはそれぞれのフィールドをループし、`lowertoupper()` 関数を使用して名前を大文字に変更します。

ヒント : フィールド名を小文字に変更するには、`uppertolower()` 関数を使用します。

## IBM SPSS Collaboration and Deployment Services Repository 内のオブジェクトへのアクセス

注 : IBM® SPSS® Collaboration and Deployment Services リポジトリを利用するには、別途ライセンスが必要です。詳細は、<http://www.ibm.com/software/analytics/spss/products/deployment/cds/> を参照してください。

ライセンス付与された IBM® SPSS® Collaboration and Deployment Services Repository がある場合は、スクリプト コマンドを使用して、リポジトリのオブジェクトを保存、取得、ロックおよびロック解除ができます。リポジトリを使用すると、エンタープライズ規模のアプリケーション、ツール、またはソリューション環境で、データ マイニング モデルと関連する予測オブジェクトのライフ サイクルを管理できます。詳細は、9 章 IBM SPSS Collaboration and Deployment Services Repository について in IBM SPSS Modeler 15 ユーザー ガイド を参照してください。

### IBM SPSS Collaboration and Deployment Services Repository への接続

リポジトリにアクセスするには、まず、IBM® SPSS® Modeler ユーザー インターフェイスの [ツール] メニューまたはコマンド ラインから、リポジトリに対して有効な接続を設定する必要があります。(詳細は、7 章 p.82 IBM SPSS Collaboration and Deployment Services Repository 接続の引数 を参照してください。)

### オブジェクトの保存と取得

スクリプト内で、`retrieve` コマンドと `store` コマンドを使って、ストリーム、モデル、出力、ノード、およびプロジェクトなど、さまざまなオブジェクトにアクセスできます。シンタックスは、次のとおりです。

```
store object as REPOSITORY_PATH {label LABEL}
store object as URI {#.label}
```

```
retrieve object REPOSITORY_PATH {label LABEL | version VERSION}
retrieve object URI [{#m.marker | #.label}]
```

**REPOSITORY\_PATH** によって、リポジトリ内のオブジェクトの場所が決められます。パスを引用符で囲み、区切り文字としてスラッシュを使用する必要があります。大文字と小文字は区別しません。

```
store stream as "/folder_1/folder_2/mystream.str"
store model Drug as "/myfolder/drugmodel"
store model Drug as "/myfolder/drugmodel.gm" label "final"
store node DRUG1n as "/samples/drug1ntypenode"
store project as "/CRISPDM/DrugExample.cpj"
store output "Data Audit of [6 fields]" as "/my folder/My Audit"
```

オプションとして、オブジェクト名に `.str` や `.gm` などの拡張子を含むことができますが、これは、オブジェクト名に一貫性がある限り必須のことではありません。たとえば、拡張子を付けずにモデルが保存される場合、同じ名前を取得する必要があります。

```
store model "/myfolder/drugmodel"
retrieve model "/myfolder/drugmodel"
```

あるいは

```
store model "/myfolder/drugmodel.gm"
retrieve model "/myfolder/drugmodel.gm" version "0:2005-10-12 14:15:41.281"
```

オブジェクトの取得時に注意すべきことは、バージョンまたはラベルを指定しない限り、オブジェクトの最新バージョンが常に返されることです。ノード オブジェクトを取得する時には、ノードは自動的に現在ストリームに挿入されます。ストリーム オブジェクトを取得する時には、スタンドアロン スクリプトを使用する必要があります。ストリーム スクリプト内からストリーム オブジェクトを取得することはできません。

### オブジェクトのロックおよびロック解除

スクリプトから、オブジェクトをロックして、ほかのユーザーが既存のバージョンを更新したり新しいバージョンを作成しないようにすることができます。ロックされたオブジェクトのロックを解除することもできます。

オブジェクトをロックおよびロック解除するシンタックスは次のとおりです。

```
lock REPOSITORY_PATH
lock URI
```

```
unlock REPOSITORY_PATH
unlock URI
```

オブジェクトの保存および取得同様、**REPOSITORY\_PATH** によって、リポジトリ内のオブジェクトの場所が決めます。パスを引用符で囲み、区切り文字としてスラッシュを使用する必要があります。大文字と小文字は区別しません。

```
lock "/myfolder/Stream1.str"
```

```
unlock "/myfolder/Stream1.str"
```

また、オブジェクトの場所を決めるには、リポジトリ パスではなく URI (Uniform Resource Identifier) を使用できます。URI は接頭辞 **spsscr:** を含み、完全に引用符で囲まれている必要があります。パス区切り文字としてはスラッシュだけを使うことができ、スペースは暗号化する必要があります。つまり、パス内ではスペースの代わりに **%20** を使用します。URI では、大文字と小文字は区別しません。いくつか例を挙げると次の通りです。

```
lock "spsscr:///myfolder/Stream1.str"
```

```
unlock "spsscr:///myfolder/Stream1.str"
```

オブジェクトのロックはすべてのバージョンのオブジェクトに適用されます。各バージョンをロックまたはロック解除することはできません。

## 暗号化パスワードの生成

場合によっては、スクリプトにパスワードを記述する必要があるかも知れません。たとえば、パスワードで保護されたデータ ソースにアクセスしたい場合などです。暗号化パスワードは、次の場所で使用することができます。

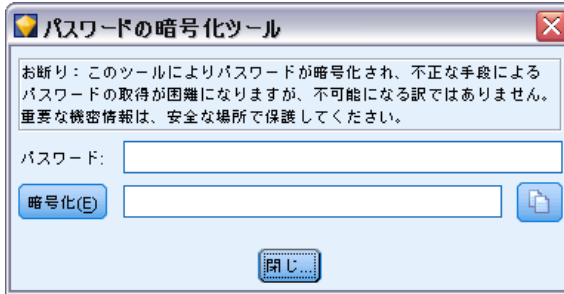
- データベース入力ノードおよび出力ノードのノード プロパティ。
- サーバーにログインするためのコマンド ライン引数。
- エクスポート ノードの [公開] タブから生成するパラメータ ファイル `.par` ファイルに保管されるデータベース接続プロパティ。

ユーザー インターフェイスから、Blowfish アルゴリズムに基づいた暗号化パスワードを生成することができます (詳細については、<http://www.schneier.com/blowfish.html> を参照してください)。パスワードを暗号化したら、そのパスワードをコピーしてスクリプト ファイルやコマンド ライン引数に指定することができます。**database** および **databaseexport** ノードに使用するノード プロパティ **epassword** は暗号化パスワードを格納します。

- ▶ 暗号化パスワードを生成するには、[ツール] メニューから次の項目を選択します。

[パスワードの暗号化(W)...]

図 5-1  
パスワードの暗号化ツール



- ▶ [パスワード] ボックスにパスワードを指定します。
- ▶ [暗号化] をクリックすると、ランダムに暗号化されたパスワードが生成されます。
- ▶ [コピー] ボタンをクリックすると、暗号化されたパスワードがクリップボードにコピーされます。
- ▶ パスワードを目的のスクリプトやパラメータに貼り付けます。

## スクリプトの検査

[スタンドアロン スクリプト] ダイアログ ボックスのツールバーにある赤い検査ボタンをクリックすれば、すべてのスクリプトのシンタックスを検査することができます。

図 5-2  
ストリーム スクリプトのツールバー アイコン



スクリプトの検査時にコードにエラーがあった場合、エラーを警告するメッセージと推奨する修正方法が表示されます。エラーのある行を表示するには、ダイアログ ボックスの下部にあるフィードバック情報をクリックしてください。エラーが赤で強調表示されます。

## コマンドラインからのスクリプト

通常はユーザー インターフェイスから行われるような操作を、スクリプトで実行することができます。IBM® SPSS® Modeler を起動する時には、コマンドライン上でスタンドアロン ストリームを指定して実行してください。次に例を示します。

```
client -script scores.txt -execute
```

`-script` フラグは指定されたスクリプトをロードすることを、`-execute` フラグはスクリプト ファイル中のすべてのコマンドを実行することを示しています。

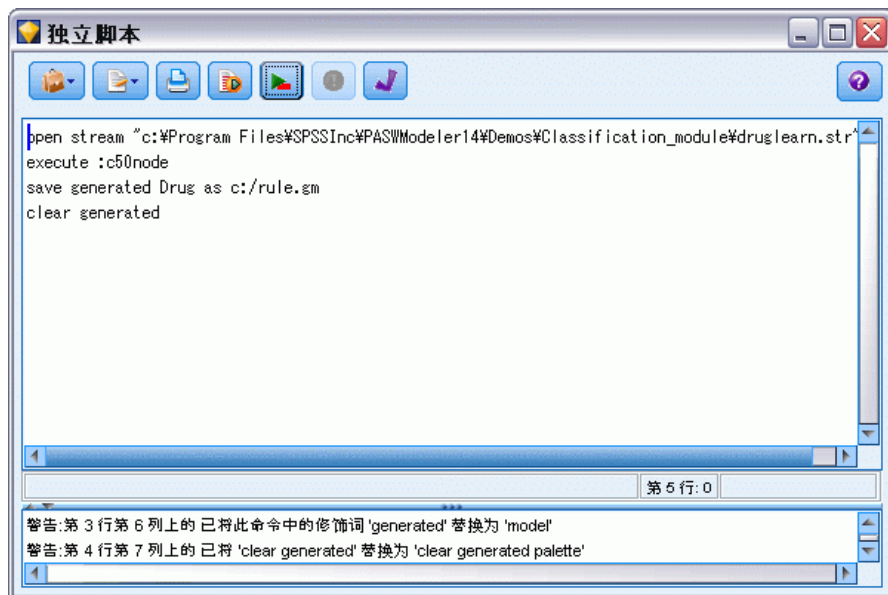
## 旧リリースとの互換性

以前の IBM® SPSS® Modeler のリリースで作成されたスクリプトは、通常現在のリリースでも変更なしで動作します。ただし、モデル ナゲットがストリームに自動的に挿入され（デフォルト設定）、ストリーム内のその種類の既存ナゲットを置き換えまたは補足する場合があります。これが実際に行われるかどうかは、[モデルをストリームに追加] オプションおよび [前のモデルを置換] オプション（[ツール] → [オプション] → [ユーザー オプション] → [通知]）の設定によって異なります。たとえば、既存のナゲットを削除して新しいナゲットを挿入し、ナゲットの置換を処理する旧リリースからのスクリプトの変更が必要な場合があります。

現在のリリースで作成したスクリプトは、以前のリリースでは動作しないことがあります。

古いリリースで作成されたスクリプトがあるコマンドを使用し、そのコマンドがリリースされてから他のコマンドに置き換えられて（または、廃止されて）いる場合は、古い形が依然としてサポートされますが、同時に警告メッセージも表示されます。たとえば、古い **generated** キーワードは **model** に、**clear generated** は **clear generated palette** に置き換えられます。古い形を使うスクリプトは依然として動作しますが、警告も表示されます。

図 5-3  
廃止されたコマンドを使うスクリプトの実行





# スクリプトの例

このセクションでは、IBM® SPSS® Modeler でスクリプトの使用方法を解説する多数の例を紹介します。

## データ型ノード レポート

このスクリプトは、現在のストリームのフィールドに関する情報を一覧表示する HTML レポートを作成します。スクリプトはインスタンス化されたデータ型ノードを持つストリームと併用でき、追加のプロパティまたはノードを簡単に取り扱えるように拡張できます。

- 標準ブラウザに表示される結果をフォーマットするため標準的な HTML タグが使用されます。
- IBM® SPSS® Modeler のデータ型ノードは、それぞれのフィールドのプロパティにアクセスするのに使用します。スクリプトを拡張して、欠損値やフィールドの役割など、データ型ノードで表示される追加プロパティを簡単に一覧表示できます。詳細は、[14 章 p.191 typenode のプロパティ](#) を参照してください。
- SPSS Modeler のスクリプト コマンドは、出力をファイルに書き込み、それぞれのプロパティにアクセスするためにフィールドをループするのに使用します。詳細は、[4 章 p.36 スクリプト コマンド](#) を参照してください。

図 6-1  
データ型ノード レポートのサンプル スクリプト

```
# This script creates an HTML file and adds data from the Type node.
var myreport
set myreport = open create "C:/typenodereport.html"

# set up the HTML page
writeln myreport "<html>"
writeln myreport "<header>Type node information from SPSS Modeler</header>"
writeln myreport "<body><br/><br/>"

#create the table and write out the headers
writeln myreport "<table border='1'"
writeln myreport "<tr bgcolor='COCOCO'"
writeln myreport "<td>Field</td><td>Type</td><td>Values</td>"
writeln myreport "</tr>"

# loop through fields and add a row for each
```

```

var current_field
for current_field in _fields_at Type
  writeln myreport "<tr>"
  write myreport "<td>" >> ^current_field >> "</td>"
  write myreport "<td>" >> Type:typenode.type.^current_field >> "</td>"

  # add values for numeric fields
  if Type:typenode.type.^current_field = Range then
    writeln myreport "<td>" >> Type:typenode.values.^current_field >> "</td>"
  endif

  # add values for flag fields
  if Type:typenode.type.^current_field = Flag then
    writeln myreport "<td>" >> Type:typenode.values.^current_field >> "</td>"
  endif

  # add values for nominal fields
  if Type:typenode.type.^current_field = Set then
    writeln myreport "<td>"
    var current_value
    for current_value in Type:typenode.values.^current_field
      writeln myreport ^current_value >> "<BR/>"
    endfor
    writeln myreport "</td>"
  endif

  writeln myreport "</tr>"
endfor
writeln myreport "</table>"
writeln myreport "</body>"
writeln myreport "</html>"
close myreport

```

## 出力ファイルの作成

新規の HTML ファイルを作成するとスクリプトが開始され、行のタイトルのフィールド、データ型、および値を一覧表示する見出し行を備えたテーブルを作成するのに必要なタグを追加します(<td></td> タグ ペアは、それぞれテーブルの行内にセルを 1 個ずつ作成します)。データ型ノードのプロパティに基づき、それぞれのフィールドについてこれらの行が読み込まれます。

```

# This script creates an HTML file and adds data from the Type node.
var myreport
set myreport = open create "C:/typenodereport.html"

# set up the HTML page
writeln myreport "<html>"

```

```
writeln myreport "<header>Type node information from SPSS Modeler</header>"
writeln myreport "<body><br/><br/>"
```

```
#create the table and write out the headers
writeln myreport "<table border='1'">"
writeln myreport "<tr bgcolor='COCOC0'">"
writeln myreport "<td>Field</td><td>Type</td><td>Values</td>"
writeln myreport "</tr>"
```

## フィールドのループ

次に、スクリプトはデータ型ノード内のすべてのフィールドをループし、フィールド名およびデータ型を示すそれぞれのフィールドについて行を追加します。

```
# loop through fields and add a row for each
var current_field
for current_field in_fields_at Type
  writeln myreport "<tr>"
  write myreport "<td>" >< ^current_field >< "</td>"
  write myreport "<td>" >< Type:typenode.type.^current_field >< "</td>"
```

## 連続型およびフラグ型フィールドの値

連続型（数値範囲型）フィールドの場合、`typenode.values` プロパティは、テーブルに表示される `[0.500517, 0.899774]` のフォーマットの最小値と最大値を返します。フラグが他フィールドの場合、同様のフォーマットで真/偽の値が表示されます。

```
# add values for numeric fields
if Type:typenode.type.^current_field = Range then
  writeln myreport "<td>" >< Type:typenode.values.^current_field >< "</td>"
endif

# add values for flag fields
if Type:typenode.type.^current_field = Flag then
  writeln myreport "<td>" >< Type:typenode.values.^current_field >< "</td>"
endif
```

## 名義型フィールドの値

名義型フィールドの場合、`typenode.values` プロパティは、定義済みの値の完全なリストを返します。スクリプトは各フィールドについてこのリストをループしてそれぞれの値を順番に挿入します。値は改行（`<br/>` タグ）で区切られます。

```
# add values for nominal fields
if Type.typhenode.type.^current_field = Set then
  writeln myreport "<td>"
  var current_value
  for current_value in Type.typhenode.values.^current_field
    writeln myreport ^current_value >< "<BR/>"
  endfor
  writeln myreport "</td>"
endif
```

### ファイルを閉じる

最後に、スクリプトは行を閉じ、`<table>`、`<body>`、および `<html>` タグを閉じて、出力ファイルを閉じます。

```
writeln myreport "</tr>"
endfor
writeln myreport "</table>"
writeln myreport "</body>"
writeln myreport "</html>"
close myreport
```

## ストリーム レポート

このスクリプトは、現在のストリーム内の各ノードについて名前、データ型、および注釈を一覧表示する HTML レポートを作成します。HTML ファイルの作成やノードおよびストリームのプロパティへのアクセスの基本要素に加えて、このスクリプトは、ストリーム内の各ノードについて特定のセットのステートメントを実行するループの作成方法を示します。これはどのストリームとも併用できます。

図 6-2  
ストリーム レポート サンプル スクリプト

```
# Create the HTML page with heading
var myfile
set myfile = open create "c:\stream_report.html"
writeln myfile "<HTML>"
writeln myfile " <BODY>"
writeln myfile " <HEAD>Report for stream " >< ^stream.name >< ".str</HEAD>"
writeln myfile " <p>" >< ^stream.annotation >< "</p>"

#Create the table with header row
writeln myfile "<TABLE border=\\"1\" width=\\"90%\>"
writeln myfile " <tr bgcolor=\\"lightgrey\" colspan=\\"3\>"
writeln myfile " <th>Node Name</th>"
```

```
writeln myfile " <th>Type</th>"
writeln myfile " <th>Annotation</th>"
writeln myfile "</tr>"

# Loop through nodes and add name, type, and annotation for each
# The ^stream.nodes property returns the list of nodes
var current_node
for current_node in ^stream.nodes
  writeln myfile "<tr>"
  writeln myfile " <td>"
  writeln myfile " ^current_node.name
  writeln myfile " </td>"
  writeln myfile " <td>"
  writeln myfile " ^current_node.node_type
  writeln myfile " </td>"
  writeln myfile " <td>"
  writeln myfile " ^current_node.annotation >> "&nbsp;"
  writeln myfile " </td>"
  writeln myfile "</tr>"
endfor

writeln myfile "</TABLE>"
writeln myfile "</BODY>"
writeln myfile "</HTML>"
close myfile
```

## レポートの作成

<BODY> および <HEAD> の要素で新規の HTML ファイルを作成するとスクリプトが開始されます。^stream.name プロパティは、見出しに挿入される現在のストリームの名前を返します。>> 演算子は、文字列をまとめて短縮するのに使用します。

```
# Create the HTML page with heading
var myfile
set myfile = open create "c:\stream_report.html"
writeln myfile "<HTML>"
writeln myfile " <BODY>"
writeln myfile " <HEAD>Report for stream " >> ^stream.name >> ".str</HEAD>"
writeln myfile " <p>" >> ^stream.annotation >> " </p>"
```

次にスクリプトは、列のタイトルのノード名、データ型、および注釈を一覧表示する見出し行を備えた HTML テーブルを作成します(<td></td> タグペアは、それぞれテーブルの行内にセルを 1 個ずつ作成します)。

```
#Create the table with header row
writeln myfile "<TABLE border=\\"1\" width=\\"90%\">"
writeln myfile " <tr bgcolor=\\"lightgrey\" colspan=\\"3\">"
writeln myfile " <th>Node Name</th>"
```

```
writeln myfile " <th>Type</th>"
writeln myfile " <th>Annotation</th>"
writeln myfile "</tr>"
```

次にスクリプトは、現在のストリーム内のすべてのノードをループします。それぞれのノードのテーブルに、名前、データ型、および注釈を一覧表示する行が追加されます。ある特定のノードについて注釈が指定されない場合に空のセルを作成するのを防止するために、注釈のあとに非表示の改行なしスペース (`&nbsp;`) が挿入されます(テーブルを表示する場合、空のセルは予期しないフォーマットになることがあります)。

```
# Loop through nodes and add name, type, and annotation for each
# The ^stream.nodes property returns the list of nodes
var current_node
for current_node in ^stream.nodes
  writeln myfile "<tr>"
  writeln myfile " <td>"
  writeln myfile   ^current_node.name
  writeln myfile " </td>"
  writeln myfile " <td>"
  writeln myfile   ^current_node.node_type
  writeln myfile " </td>"
  writeln myfile " <td>"
  writeln myfile   ^current_node.annotation >> "&nbsp;"
  writeln myfile " </td>"
  writeln myfile "</tr>"
endfor
```

最後にスクリプトは、ドキュメントおよびファイルを閉じるのに必要な HTML タグを追加します。

```
writeln myfile "</TABLE>"
writeln myfile "</BODY>"
writeln myfile "</HTML>"
close myfile
```

# コマンド ラインの引数

## ソフトウェアの起動

オペレーティング システムのコマンド ラインを使用し、次のようにして IBM® SPSS® Modeler を起動できます。

- ▶ IBM® SPSS® Modeler がインストールされているコンピュータで、DOS つまりコマンド プロンプト ウィンドウを開きます。
- ▶ SPSS Modeler インターフェイスをインタラクティブ モードで起動するには、**modelerclient** コマンドを入力し、続いてたとえば次のような適切な引数を入力します。

```
modelerclient -stream report.str -execute
```

使用可能な引数（フラグ）により、サーバーへの接続、ストリームのロード、スクリプトの実行、または必要に応じて他のパラメータの指定を行うことができます。

## コマンド ライン引数の使用

IBM® SPSS® Modeler の起動を変更するために、コマンド ラインの引数（フラグ型とも呼ばれます）を初期の **modelerclient** コマンドに追加できます。

たとえば、以下のようにして **-server**、**-stream** および **-execute** のフラグ型を使用してサーバーに接続し、ストリームをロードおよび実行できます。

```
modelerclient -server -hostname myserver -port 80 -username dminer  
-password 1234 -stream mystream.str -execute
```

ローカル クライアントのインストールと競合する場合、サーバー接続の引数は不要です。

スペースを含むパラメータ値は二重引用符で囲むことができます。たとえば、次のようになります。

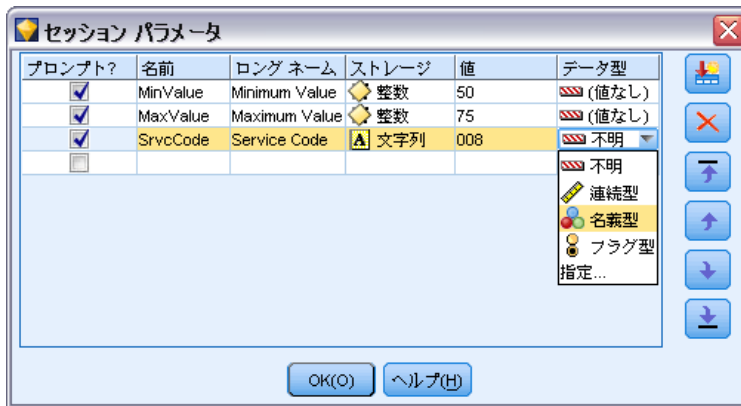
```
modelerclient -stream mystream.str -Pusername="Joe User" -execute
```

また、SPSS Modeler のステートとスクリプトも、それぞれ **-state** フラグと **-script** フラグを使用して、この方法で実行できます。

## デバッグ コマンド ラインの引数

コマンド ラインをデバッグするには `modelerclient` コマンドを使用し、適切な引数を使って SPSS Modeler を起動します。これによって、コマンドが予定通りに実行されることを検証できます。また、[セッション パラメータ] ダイアログ ボックス（[ツール] メニュー、セッション パラメータの設定）のコマンド ラインから渡されるパラメータの値を確認することもできます。

図 7-1  
セッションのパラメータの設定



## 複数の引数の組み合わせ

複数の引数を記述したコマンド ファイルを作成し、起動時に @ 記号に続けてそのファイル名を指定することができます。こうすることによって、コマンド ラインによる起動を短縮し、OS によるコマンド長の制限に関する問題を解決することができます。たとえば、以下の起動コマンドは <commandFileName> が示すファイルに指定されている引数を使用します。

```
modelerclient @<commandFileName>
```

ファイル名やコマンド ファイルへのパスにスペースがある場合は、以下のようにして引用符で囲みます。

```
modelerclient @"C:\Program Files\IBM\SPSS\Modeler\...\scripts\my_command_file.txt"
```

このコマンド ファイルには、スタートアップ時に個別に指定していたすべての引数を記述することができます。以下のようにして、1 行に 1 つの引数を記述します。

```
-stream report.str
-Porder.full_filename=APR_orders.dat
-Preport.filename=APR_report.txt
-execute
```



コマンド ファイルを記述して、コマンド ファイル名を指定する場合の制限事項を次に示します。

- 1 行につき 1 つの引数またはコマンドを記述する必要があります。
- コマンド ファイル内に、@CommandFile 引数を組み込まないでください。

## サーバー接続の引数

-server フラグは、IBM® SPSS® Modeler にパブリック サーバーに接続するよう指示し、-hostname、-use\_ssl、-port、-username、-password、および -domain のフラグを使用して、SPSS Modeler にパブリック サーバーに接続する方法を指示します。-server 引数が指定されていない場合、デフォルト サーバーまたはローカル サーバーが使用されます。

### 例

パブリック サーバーに接続するには

```
modelerclient -server -hostname myserver -port 80 -username dminer
-password 1234 -stream mystream.str -execute
```

サーバー クラスタに接続するには

```
modelerclient -server -cluster "QA Machines" \
-spsscr_hostname pes_host -spsscr_port 8080 \
-spsscr_username asmith -spsscr_epassword xyz
```

サーバー クラスタに接続するには、IBM® SPSS® Collaboration and Deployment Services を使用した Coordinator of Processes が必要です。したがって、-cluster 引数をリポジトリ接続オプション (spsscr\_\*) とともに使用する必要があります。詳細は、[p.82 IBM SPSS Collaboration and Deployment Services Repository 接続の引数](#) を参照してください。

| 引数               | 動作説明  |
|------------------|---|
| -server          | SPSS Modeler をサーバー モードで実行し、フラグ -hostname、-port、-username、-password、および -domain を使用してパブリック サーバーに接続します。 |
| -hostname <name> | サーバー マシンのホスト名を指定します。サーバー モードでしか利用できません。   |
| -use_ssl         | 接続で使用する SSL (secure socket layer) を指定します。このフラグはオプションです。SSL 使用時のデフォルト設定は not です。                       |
| -port <number>   | 指定したサーバーのポート番号。サーバー モードでしか利用できません。  |

| 引数   | 動作説明  |
|--|---|
| <b>-cluster</b> <name>                       | 名前付きサーバーではなく、サーバー クラスタへの接続を指定します。この引数は <b>hostname</b> 、 <b>port</b> 、および <b>use_ssl</b> 引数の代替です。name はクラスタ名、または IBM® SPSS® Collaboration and Deployment Services Repository 内のクラスタを識別する一意の URI です。サーバー クラスタは、IBM SPSS Collaboration and Deployment Services を使用して Coordinator of Processes で管理されます。詳細は、 <a href="#">p. 82 IBM SPSS Collaboration and Deployment Services Repository 接続の引数</a> を参照してください。 |
| <b>-username</b> <name>                      | サーバーにログオンするためのユーザー名。サーバー モードでしか利用できません。   |
| <b>-password</b> <password>                  | サーバーにログオンするためのパスワード。サーバー モードでしか利用できません。注： <b>-password</b> 引数を使用しない場合、パスワードの入力を要求するプロンプトが表示されます。  |
| <b>-epassword</b><br><encodedpasswordstring> | サーバーにログオンするための暗号化パスワード。サーバーモードでしか利用できません。注： 暗号化パスワードは、SPSS Modeler アプリケーションの [ツール] メニューから生成することができます。   |
| <b>-domain</b> <name>                        | サーバーにログオンする際に使用するドメイン名。サーバーモードでしか利用できません。   |
| <b>-P</b> <name>=<value>                     | スタートアップ パラメータの設定に使用されます。ノードのプロパティ (スロット パラメータ) の設定に使用することもできます。   |

## IBM SPSS Collaboration and Deployment Services Repository 接続の引数

注： IBM® SPSS® Collaboration and Deployment Services リポジトリを利用するには、別途ライセンスが必要です。詳細は、<http://www.ibm.com/software/analytics/spss/products/deployment/cds/> を参照してください。

コマンド ラインを経由して IBM SPSS Collaboration and Deployment Services でオブジェクトを保存したり取り出したりするには、IBM® SPSS® Collaboration and Deployment Services Repositoryに有効な接続を指定する必要があります。次に例を示します。

```
modelerclient -spsscr_hostname myhost -spsscr_port 8080
-spsscr_username myusername -spsscr_password mypassword
-stream "spsscr:///folder_1/scoring_stream.str" -execute
```

接続を設定するために使用できる引数の一覧を次の表に示します。

| 引数                                 | 動作説明   |
|------------------------------------|--|
| -spsscr_hostname <ホスト名または IP アドレス> | IBM SPSS Collaboration and Deployment Services Repository がインストールされているサーバーのホスト名または IP アドレスです。  |
| -spsscr_port <number>              | IBM SPSS Collaboration and Deployment Services Repository が接続を承認したポート番号です (通常、8080 がデフォルト値)。   |
| -spsscr_use_ssl                    | 接続で使用する SSL (secure socket layer) を指定します。このフラグはオプションです。SSL 使用時のデフォルト設定は not です。  |
| -spsscr_username <name>            | IBM SPSS Collaboration and Deployment Services Repository にログオンするためのユーザー名。   |
| -spsscr_password <password>        | IBM SPSS Collaboration and Deployment Services Repository にログオンするためのパスワード。   |
| -spsscr_epassword <暗号化パスワード>       | IBM SPSS Collaboration and Deployment Services Repository にログオンするためのエンコードされたパスワード。   |
| -spsscr_domain <name>              | IBM SPSS Collaboration and Deployment Services Repository にログオンする際に使用するドメイン名。このフラグはオプションです。LDAP または Active Directory を使ってログオンしない限り、このフラグは使用しないでください。 |

## システムの引数

ユーザー インターフェイスのコマンド ラインによる起動で利用できるシステム引数を次の表に示します。

| 引数                      | 動作説明   |
|-------------------------|--|
| @<commandFile>          | @ 文字に続けてファイル名を記述することにより、コマンド リストを指定することができます。modelerclient コマンドに @ から始まる引数を指定すると、その引数に指定されたコマンド ファイル中のコマンドが、コマンド ラインに指定されているのと同じように処理されます。<br><a href="#">詳細は、 p.80 複数の引数の組み合わせ を参照してください。</a> |
| -directory <dir>        | デフォルトの作業ディレクトリを設定します。ローカル モードでは、このディレクトリはデータと出力の両方で使用されます。   |
| -server_directory <dir> | デフォルトのデータ用サーバー ディレクトリを設定します。<br>-directory フラグで指定された作業ディレクトリは、出力に使用されます。   |
| -execute                | 起動後に、起動時にロードされたストリーム、ステート、またはスクリプトを実行します。ストリームやステートではなくスクリプトがロードされた場合は、スクリプトだけが実行されます。   |
| -stream <ストリーム>         | 起動時に、指定したストリームをロードします。複数のストリームを指定できますが、最後に指定したストリームが現在のストリームに設定されます。   |

| 引数                | 動作説明  |
|-------------------|---|
| -script <script>  | 起動時に、指定したスタンドアロン スクリプトをロードします。下で説明しているストリームやステートに加えてこれも指定できますが、起動時には 1 つのスクリプトしかロードできません。 |
| -model <model>    | 起動時に、指定の生成モデル (.gm フォーマット ファイル) をロードします。  |
| -state <ステート>     | 起動時に、指定した保存済みのステートをロードします。  |
| -project <プロジェクト> | 指定したプロジェクトをロードします。起動時には、プロジェクトを 1 つしかロードできません。  |
| -output <output>  | 起動時に、保存された出力オブジェクト (.cou フォーマット ファイル) をロードします。  |
| -help             | コマンド ライン引数のリストを表示します。このオプションを指定すると、他の引数はすべて無視されて、ヘルプ画面が表示されます。                            |
| -P <name>=<value> | スタートアップ パラメータの設定に使用されます。ノードのプロパティ (スロット パラメータ) の設定に使用することもできます。                           |

注：ユーザー インターフェイスでデフォルト ディレクトリも設定できます。このオプションにアクセスするには、[ファイル] メニューの [作業ディレクトリの設定] または [サーバー ディレクトリの設定] を選択します。

### 複数ファイルのロード

ロードされた各オブジェクトに対応する引数を繰り返し指定して、起動時にコマンド ラインから、複数のストリーム、ステート、および出力をロードすることができます。たとえば、report.str と train.str の 2 種類のストリームをロード、実行するには、コマンド ラインに次のコマンドを指定します。

```
modelerclient -stream report.str -stream train.str -execute
```

### IBM SPSS Collaboration and Deployment Services Repository からのオブジェクトのロード

ファイルまたは IBM® SPSS® Collaboration and Deployment Services Repository (ライセンスがある場合) から特定のオブジェクトを読み込むことができるため、ファイル名の接頭辞 **spsscr:** および、オプションで **file:** (ディスク上のオブジェクト) が IBM® SPSS® Modeler にオブジェクトの検索場所を示します。上記の接頭辞は、次のフラグに適用できます。

- -stream
- -script
- -output
- -model
- -project

接頭辞を使用して、オブジェクトの場所を指定する URI を作成します。たとえば、次のようになります。

`-stream "spsscr:///folder_1/scoring_stream.str"`。 `spsscr:` の接頭辞がある場合、IBM SPSS Collaboration and Deployment Services Repository への有効な接続を同じコマンドで指定する必要があります。そのため、たとえば、フルコマンドは次のようになります。

```
modelerclient -spsscr_hostname myhost -spsscr_port 8080
-spsscr_username myusername -spsscr_password mypassword
-stream "spsscr:///folder_1/scoring_stream.str" -execute
```

IBM SPSS Collaboration and Deployment Services Repository 中のオブジェクトの URI 詳細については、5 章（ p.67 ）の「[IBM SPSS Collaboration and Deployment Services Repository 内のオブジェクトへのアクセス](#)」を参照してください。コマンドラインから URI を使用する「必要がある」ことに注意してください。単純な `REPOSITORY_PATH` はサポートされていません（その場合は、スクリプト内でのみ作動します）。

## パラメータの引数

IBM® SPSS® Modeler のコマンドライン実行時に、パラメータをフラグとして使用することができます。コマンドラインの引数に `-P` フラグを使って、`-P <name>=<value>` の形式でパラメータを表すことができます。

パラメータは、次のいずれかになります。

- **単純なパラメータ**（または、CLEM 式で直接使用されるパラメータ）。
- **スロット パラメータ、ノードのプロパティ**と呼ばれることもあります。これらのパラメータは、ストリーム中のノードの設定を変更するために使用されます。 [詳細は、9 章 p.129 ノードのプロパティの概要](#) を参照してください。
- SPSS Modeler の起動を変更するために用いられる、**コマンドライン パラメータ**。

たとえば、データソースのユーザー名とパスワードを、次のようにコマンドラインのフラグとして指定することができます。

```
modelerclient -stream response.str -P:databasenode.datasourcesource={"ORA 10gR2", user1, mypsw, true}
```

形式は、`databasenode` ノード プロパティの `datasourcesource` パラメータの形式と同じです。 [詳細は、12 章 p.139 databasenode のプロパティ](#) を参照してください。

# CLEM 言語に関するリファレンス

## CLEM リファレンス概要

この項では、Control Language for Expression Manipulation (CLEM) について説明していきます。CLEM は、IBM® SPSS® Modeler ストリーム内で使われるデータの分析と操作用の非常に役に立つツールです。ノード内で CLEM を使用して、条件の評価や値の新規作成からレポートへのデータ挿入まで、作業を実行できます。詳細は、7 章 CLEM について in IBM SPSS Modeler 15 ユーザー ガイド を参照してください。

CLEM 言語のサブセットは、また、ユーザー インターフェイスでスクリプトを使用する場合に使用することができます。これによって、同じデータ操作の大部分を自動的に行うことができます。詳細は、3 章 p.33 スクリプト内の CLEM 式 を参照してください。

CLEM 式は、値、フィールド名、演算子、および関数で構成されます。正しい構文を使って、さまざまなデータ操作を作成することができます。詳細は、7 章 CLEM の例 in IBM SPSS Modeler 15 ユーザー ガイド を参照してください。

## CLEM データ型

CLEM データ型は次のいずれかで構成できます。

- 整数
- 実数
- 文字
- 文字列
- リスト
- Fields
- 日付と時刻

### 引用符の使用規則

IBM® SPSS® Modeler では、CLEM 式で使われるフィールド、値、パラメータ、および文字列などを柔軟に指定することができます。次の規則に従って式を作成することをお勧めします。

- 文字列-文字列を指定する場合は、常に二重引用符を使用してください (例: "Type 2")。単一引用符を使用することもできますが、引用符で囲まれたフィールドと誤解される危険性があります。

- フィールド-スペースや他の特殊文字を入れる必要があるような場合にだけ単一引用符を使用します (例: 'Order Number')。データ セット中に単一引用符で囲まれているのに未定義のフィールドがあると、それは文字列として読み込まれてしまいます。
- パラメーター-パラメータを使用する場合は、常に単一引用符を使用します (例: '\$P-threshold')。
- 文字-常に単一の逆引用符 ( ` ) を使用します (例: stripchar(`d`,`drugA`))。

詳細は、7 章 [値とデータ型 in IBM SPSS Modeler 15 ユーザー ガイド](#) を参照してください。これらの規則は、以降の項目で詳細に説明しています。

## 整数

整数は、10 進数のシーケンスとして表されます。必要に応じて、整数の前にマイナス符号 (-) を付けて負の数を表すことができます。たとえば、1234、999、-77 のように記述します。

CLEM 言語は任意の精度の整数を処理します。整数の最大サイズは使用するプラットフォームによって異なります。値が大きすぎて整数フィールドに表示できない場合は、通常、フィールドのデータ型を **Real** に変更すると値を正確に表示できます。

## 実数

実数は浮動小数点数を意味しています。実数は、1 つ以上の数値と、その後続く小数点、その後続く 1 つ以上の数値で表されます。CLEM 実数は倍精度で保持されます。

必要に応じて、実数の前にマイナス符号 (-) を付けて負の数を表すことができます。たとえば、1.234、0.999、-77.001 のように記述します。指数表記で実数を表すには、<数値> e<指数> の形式を使用します。たとえば、1234.0e5、1.7e-2 のように記述します。IBM® SPSS® Modeler アプリケーションがファイルから数値文字列を読み込んで自動的に数値に変換する場合、小数点の前に数字がない数値や小数点の後に数字がない数値も受け取ります。たとえば、999. や .11 などです。ただし、これらの形式は CLEM 式では不正です。

注：CLEM 式の実数を参照する場合、現在のストリームまたはロケールの設定に関わらず、小数点区切り文字としてピリオドを使用する必要があります。たとえば、次のとおりです。

Na > 0.6

次のようには、指定できません。

Na > 0.6

これは、[ストリームのプロパティ] ダイアログ ボックスでカンマが小数点として選択された場合でも適用されます。また、コード構文が特定のロケールまたは表記方法から独立する必要があるという一般的なガイドラインを検討します。

## 文字

一般的に文字（通常 **CHAR** と表記）は、CLEM 式内で文字列のテストを実行するために用いられます。たとえば、**isuppercode** 関数を使って、文字列の先頭文字が大文字かどうかを判断することができます。文字列の先頭文字に対してテストを行う必要があることを示すために、文字を使用する CLEM 式を次に示します。

```
isuppercode(subscrs(1, "MyString"))
```

CLEM 式中の特定文字のコード（場所ではなく）を表すには、単一逆引用符を `<文字>--` の形式で指定します。たとえば、``A``、``Z`` のように記述します。

注：フィールドに対する **CHAR** ストレージ タイプはありません。そのため、結果が **CHAR** となる式でフィールドが作成または置換された場合、その結果は文字列に変換されます。

## 文字列

基本的に、文字列は二重引用符で囲んでください。たとえば、文字列は `"c35product2"` や `"referrerID"` となります。文字列内で特別な文字を示すには、`"$65443"` のように円記号を使用します（円記号を示すには、`\"` のように円記号を 2 つ使用してください）。文字列を単一引用符で囲むこともできますが、その場合引用符で囲まれたフィールド（`'referrerID'`）と区別できない可能性があります。詳細は、[文字列関数 in IBM SPSS Modeler 15 ユーザー ガイド](#) を参照してください。

## リスト

リストは、順序付けられた要素のシーケンスであり、データ型が混在していることもあります。リストは、大カッコ（`[]`）で囲みます。たとえば、リストは `[1 2 4 16]` や `["abc" "def"]` となります。リストは IBM® SPSS® Modeler フィールドの値としては使用されません。リストは、**member** や **oneof** などの関数に引数を渡すために使用します。



## Fields

CLEM 式内で、関数名以外の名前はフィールド名とみなされます。これらを **Power**、**val27**、**state\_flag** のように記述できますが、または名前が数字から始まる、またはスペースなどアルファベット以外の文字（アンダースコアを除く）を含む場合、'**Power Increase**'、'**2nd answer**'、'**#101**'、'**\$P-NextField**' のように名前を単一引用符で囲みます。

注： データ セット中に単一引用符で囲まれているのに未定義のフィールドがあると、それは文字列として読み込まれてしまいます。

## 日付(D)

日付の計算は、基準日に基づいて行われます。基準日は、[ストリームのプロパティ] ダイアログ ボックスで指定します。デフォルトの基準日は、1900 年 1 月 1 日です。詳細は、[5 章 ストリームの一般的なオプションの設定 in IBM SPSS Modeler 15 ユーザー ガイド](#) を参照してください。

CLEM 言語は、次の日付のフォーマットをサポートします：

| 書式         | 例   |
|------------|---|
| DDMMYY     | 150163  |
| MMDDYY     | 011563  |
| YYMMDD     | 630115  |
| YYYYMMDD   | 19630115  |
| YYYYDDD    | 4 桁の西暦とそれに続く 3 桁の 1 月 1 日からの日数。-たとえば 2000032 は、2000 年の 32 番目の日にち、つまり 2000 年 2 月 1 日を表します。 |
| DAY        | 現在のロケールの曜日です。-たとえば、英語の場合、Monday、Tuesday などです。   |
| MONTH      | 現在のロケールの月名です。たとえば、英語の場合、January、February などです。  |
| DD/MM/YY   | 15/01/63  |
| DD/MM/YYYY | 15/01/1963  |
| MM/DD/YY   | 01/15/63  |
| MM/DD/YYYY | 01/15/1963  |
| DD-MM-YY   | 15-01-63  |
| DD-MM-YYYY | 15-01-1963  |
| MM-DD-YY   | 01-15-63  |
| MM-DD-YYYY | 01-15-1963  |
| DD.MM.YY   | 15.01.63  |

| 書式          | 例  |
|-------------|--|
| DD.MM.YYYY  | 15.01.1963   |
| MM.DD.YY    | 01.15.63   |
| MM.DD.YYYY  | 01.15.1963   |
| DD-MON-YY   | 15-JAN-63, 15-jan-63, 15-Jan-63  |
| DD/MON/YY   | 15/JAN/63, 15/jan/63, 15/Jan/63  |
| DD.MON.YY   | 15.JAN.63, 15.jan.63, 15.Jan.63  |
| DD-MON-YYYY | 15-JAN-1963, 15-jan-1963, 15-Jan-1963  |
| DD/MON/YYYY | 15/JAN/1963, 15/jan/1963, 15/Jan/1963  |
| DD.MON.YYYY | 15.JAN.1963, 15.jan.1963, 15.Jan.1963  |
| MON YYYY    | Jan 2004   |
| q Q YYYY    | 四半期を表す 1 桁の数字 (1-4) とそれに続く文字 Q、そして 4 桁の年です。たとえば 2004 年 12 月 25 日の表記は、 <b>4 Q 2004</b> になります。   |
| ww WK YYYY  | 1 年の内での週を表す 2 桁の数で、次に文字 WK と 4 桁の年が続きます。年内の週は、週の最初の日が月曜日で、また、少なくとも最初の週に 1 日以上あるという仮定の下に計算されます。 |

## Time

CLEM 言語は、次の時間のフォーマットをサポートします：

| 書式             | 例                            |
|----------------|------------------------------|
| HHMMSS         | 120112, 010101, 221212       |
| HHMM           | 1223, 0745, 2207             |
| MMSS           | 5558, 0100                   |
| HH:MM:SS       | 12:01:12, 01:01:01, 22:12:12 |
| HH:MM          | 12:23, 07:45, 22:07          |
| MM:SS          | 55:58, 01:00                 |
| (H)H:(M)M:(S)S | 12:1:12, 1:1:1, 22:12:12     |
| (H)H:(M)M      | 12:23, 7:45, 22:7            |
| (M)M:(S)S      | 55:58, 1:0                   |
| HH.MM.SS       | 12.01.12, 01.01.01, 22.12.12 |
| HH.MM          | 12.23, 07.45, 22.07          |
| MM.SS          | 55.58, 01.00                 |
| (H)H.(M)M.(S)S | 12.1.12, 1.1.1, 22.12.12     |
| (H)H.(M)M      | 12.23, 7.45, 22.7            |
| (M)M.(S)S      | 55.58, 1.0                   |

## CLEM演算子

次の演算子が利用できます。

| 操作     | コメント   | 優先順位 (次項参照) |
|--------|--|-------------|
| or     | 2 つのCLEM 式間で使用されます。どちらかが真 (true) の場合、または両方が真 (true) の場合に、値を返します。     | 10          |
| and    | 2 つのCLEM 式間で使用されます。両方が真 (true) の場合に、値を返します。                          | 9           |
| =      | 任意の比較可能な 2 つの項目間で使用されます。ITEM1 が ITEM2 と等しい場合に真が返されます。                | 7           |
| ==     | = と同等  | 7           |
| /=     | 任意の比較可能な 2 つの項目間で使用されます。ITEM1 が ITEM2 と等しくない場合に真が返されます。              | 7           |
| /==    | /= と同等   | 7           |
| >      | 任意の比較可能な 2 つの項目間で使用されます。ITEM1 が厳密に ITEM2 より大きい場合に真を返します。             | 6           |
| >=     | 任意の比較可能な 2 つの項目間で使用されます。ITEM1 が ITEM2 以上の場合に真が返されます。                 | 6           |
| <      | 任意の比較可能な 2 つの項目間で使用されます。ITEM1 が厳密に ITEM2 より小さい場合に真を返します。             | 6           |
| <=     | 任意の比較可能な 2 つの項目間で使用されます。ITEM1 が ITEM2 以下の場合に真が返されます。                 | 6           |
| &&=_0  | 2 つの整数間に用いられます。ブール式の <code>INT1 &amp;&amp; INT2 = 0</code> と同じになります。 | 6           |
| &&/=_0 | 2 つの整数間に用いられます。ブール式の <code>INT1 &amp;&amp; INT2 = 0</code> と同じになります。 | 6           |
| +      | 2 つの数値で加算します。NUM1 + NUM2。  | 5           |
| ><     | 2 つの文字列を次のように連結します。<br>STRING1 >< STRING2.                           | 5           |

| 操作        | コメント  | 優先順位 (次項参照) |
|-----------|---|-------------|
| -         | 1つの数値をもう1つの数値から減算します。NUM1 - NUM2。1つの数値の前にも使用できます。<br>- NUM。         | 5           |
| *         | 2つの数値を乗算するのに使用されます。NUM1 * NUM2。                                     | 4           |
| &&        | 2つの整数間に用いられます。結果は、INT1 と INT2 のビット単位の「論理積」になります。                    | 4           |
| &&~       | 2つの整数間に用いられます。結果は、INT1 と、INT2 のビット単位の補数との、ビット単位の「論理積」になります。         | 4           |
|           | 2つの整数間に用いられます。結果は、INT1 と INT2 のビット単位の「包括的論理和」になります。                 | 4           |
| ~~        | 整数の前に用いられます。整数 INT のビット単位の補数を生成します。                                 | 4           |
| /&        | 2つの整数間に用いられます。結果は、INT1 と INT2 のビット単位の「排他的論理和」になります。                 | 4           |
| INT1 << N | 2つの整数間に用いられます。N の数だけ位置を左にシフトした INT のビットパターンを生成します。                  | 4           |
| INT1 >> N | 2つの整数間に用いられます。N の数だけ位置を右にシフトした INT のビットパターンを生成します。                  | 4           |
| /         | 1つの数値をもう1つの数値で除算するのに使用されます: NUM1 / NUM2。NUM1 / NUM2。                | 4           |
| **        | 2つの数値間に用いられます。BASE ** POWERBASE の POWER 乗を返します。                     | 3           |
| rem       | 2つの整数間に用いられます。INT1 rem INT2 剰余 INT1 - (INT1 div INT2) * INT2 を返します。 | 2           |
| div       | 2つの整数間に用いられます。INT1 div INT2 整数の除算を実行します。                            | 2           |

## 演算子の優先順位

優先順位は、複数の 2 項演算子を使ったカッコで囲まれていない式などの、複雑な式の解析方法を決めるものです。例をあげると、次のようになります。

$3+4*5$

\* が + の前に解析されることを示すため、 $(3+4)*5$  ではなく  $3+(4*5)$  として解析します。CLEM 言語中のすべての演算子には、それに対応した優先順位があります。優先順位が低いほど、その演算子は処理リスト上で重要な意味を持ち、他の演算子よりも先に処理されます。

## 関数のリファレンス

IBM® SPSS® Modeler でデータを処理するために、次の CLEM 関数を利用できます。これらの関数は、フィールド作成ノードやフラグ設定ノードなど、さまざまなダイアログ ボックスにコードとして入力できます。または、Clem 式ビルダーを利用して、有効な CLEM 式を作成することができます。関数やフィールド名を覚えておく必要はありません。

| 関数の種類       | 説明  |
|-------------|---|
| 情報          | フィールド値を詳しく調べる場合に用いられます。たとえば、関数 <code>is_string</code> は、データ型が文字列型のすべてのレコードに対して真を返します。                   |
| 変換          | 新しいフィールドの作成や、ストレージ タイプの変換に用いられます。たとえば、関数 <code>to_timestamp</code> は選択されているフィールドをタイムスタンプに変換します。         |
| 比較          | フィールドの値を互いに比較したり、指定した文字列と比較する場合に用いられます。たとえば、 <code>&lt;=</code> は、あるフィールドの値がもう 1 つのフィールドの値以下かどうかを比較します。 |
| 論理          | <code>if</code> 、 <code>then</code> 、 <code>else</code> などの論理演算を行うために用いられます。                            |
| 数値          | フィールド値の自然ログ数の算出など、数値計算に用いられます。  |
| 三角関数        | 指定された角度の アークコサインの算出など、三角関数の計算に用いられます。   |
| Probability | 学生からの <code>t</code> 分布値が特定値に満たなくなる確率など、さまざまな分布を基準にして確率を返します。   |
| ビット単位       | 整数をビット パターンとして操作する場合に用いられます。  |
| Random      | 無作為に項目を選択したり、無作為な 数字を生成するために用いられます。   |

| 関数の種類   | 説明   |
|---------|--|
| String  | 指定した文字を削除する <code>stripchar</code> のように、文字列に関するさまざまな操作を行うために用いられます。  |
| SoundEx | 正しいスペルが分からない場合に、特定文字の発音方法についての音声的な仮定を基準にして、文字列を検索するために用いられます。  |
| 日付と時刻   | 日付、時間、タイムスタンプ フィールドに対してさまざまな操作を行うために用いられます。  |
| シーケンス   | データ セットのレコード シーケンスの詳細を調べたり、そのシーケンスに基づいた操作を行うために用いられます。   |
| グローバル   | グローバル ノードが作成したグローバル値にアクセスするために用いられます。たとえば、 <code>@MEAN</code> は、データ セット全体のフィールドのすべての値の平均を参照するために用いられます。          |
| 空白とヌル   | アクセス、フラグ設定、およびユーザーが指定した空白やシステム欠損値を埋めるために用いられます。たとえば、 <code>@BLANK(FIELD)</code> は、空白があるレコードに真のフラグを設定するために用いられます。 |
| 特殊フィールド | 調査対象の特定のフィールドを表すために用いられます。たとえば、 <code>@FIELD</code> は複数のフィールドを作成する場合に用いられます。                                     |

## 関数の表記方法について

このガイドでは、次の規約を関数中のアイテムを参照するために使用します。

| 表記方法              | 説明  |
|-------------------|---|
| BOOL              | 真 (true) または偽 (false) を示すブールまたはフラグ。                                     |
| NUM、NUM1、 NUM2    | 任意の数値。  |
| REAL、REAL1、 REAL2 | 1.234 または -77.01 のような任意の実数  |
| INT、INT1、 INT2    | 1 または -77 のような任意の実数   |
| CHAR              | 'A' のような文字コード。  |
| STRING            | "referrerID" のような文字列。   |
| リスト               | ["abc" "def"] のような、アイテムのリスト。  |
| ITEM              | Customer または <code>extract_concept</code> のような任意の実数                     |
| DATE              | <code>start_date</code> のような日付フィールド。ここで、値のフォーマットは DD-MON-YYYY のようになります。 |
| TIME              | <code>power_flux</code> のような時刻フィールド。ここで、値のフォーマットは HHMMSS のようになります。      |

このガイドにある関数の一覧では、関数を最初の列に、結果のタイプ（整数、文字列等）を 2 番目の列に、説明（存在する場合）を 3 番目の列に示しています。たとえば、次に **rem** 関数の説明を示します。

| 関数            | 結果 | 説明  |
|---------------|----|---|
| INT1 rem INT2 | 数値 | INT1 を INT2 で除算した剰余を返しますたとえば、 $INT1 - (INT1 \text{ div } INT2) * INT2$ となります。 |

項目をリストにする方法や、関数内で文字を指定する方法などの使用方法の詳細は、別の場所で説明されています。詳細は、[CLEMデータ型 in IBM SPSS Modeler 15 ユーザー ガイド](#) を参照してください。

## 情報関数

情報関数は、特定のフィールドの値に対する洞察を行うために用いられます。通常これらは、フラグ型フィールドの作成に用いられます。たとえば、**@BLANK** 関数を使って、選択したフィールドに対する値が空白のレコードを示すフラグ型フィールドを作成することができます。同様に、**is\_string** などのストレージタイプ関数を使って、フィールドのストレージタイプを確認することもできます。

| 関数                | 結果  | 説明  |
|-------------------|-----|---|
| @BLANK(FIELD)     | ブール | 上流のデータ型ノードまたは入力ノードで設定された空白処理規則（[データ型] タブ）にしたがって、値が空白のレコードに対して真を返します。この関数は、スクリプトから呼び出すことができません。詳細は、3 章 p.33 スクリプト内の CLEM 式 を参照してください。              |
| @NULL(ITEM)       | ブール | 値が未定義のすべてのレコードに対して真を返します。未定義の値はシステムのヌル値で、IBM® SPSS® Modeler では \$null\$ として表されます。この関数は、スクリプトから呼び出すことができません。詳細は、3 章 p.33 スクリプト内の CLEM 式 を参照してください。 |
| is_date(ITEM)     | ブール | データ型が日付のすべてのレコードに対して真 (true) を返します。   |
| is_datetime(ITEM) | ブール | データ型が日付、時間またはタイムスタンプのすべてのレコードに対して真 (true) を返します。  |
| is_integer(ITEM)  | ブール | データ型が整数のすべてのレコードに対して真 (true) を返します。   |
| is_number(ITEM)   | ブール | データ型が数値のすべてのレコードに対して真 (true) を返します。   |
| is_real(ITEM)     | ブール | データ型が実数のすべてのレコードに対して真 (true) を返します。   |
| is_string(ITEM)   | ブール | データ型が文字列のすべてのレコードに対して真 (true) を返します。  |

| 関数                 | 結果  | 説明                                       |
|--------------------|-----|--|
| is_time(ITEM)      | ブール | データ型が時間のすべてのレコードに対して真 (true) を返します。      |
| is_timestamp(ITEM) | ブール | データ型がタイムスタンプのすべてのレコードに対して真 (true) を返します。 |

## 変換関数

変換関数により、新規フィールドを作成し、既存のファイルのストレージタイプを変換することができます。たとえば、文字列を結合したり、切り離したりして、新しい文字列を生成できます。文字列を結合するには、演算子  $\times$  を使用します。たとえば、フィールド **Site** の値が "BRAMLEY" である場合、" $\text{xx}$ "  $\times$  **Site** は " $\text{xxBRAMLEY}$ " を返します。 $\times$  の結果は、引数が文字列でない場合でも、常に文字列となります。フィールド **V1** が **3** で **V2** が **5** である場合、 $\text{V1} \times \text{V2}$  は "35" を返します (数値ではなく文字列)。

変換の関数と日付や時刻の値のような、入力に特別な型が必要なその他の関数は、[ストリームのオプション] ダイアログ ボックスに指定されている現在のフォーマットに依存します。たとえば、値が Jan 2003、Feb 2003 などの文字列フィールドを日付ストレージへ変換する場合、ストリームのデフォルトの日付フォーマットとして一致する [MON YYYY] を選択します。詳細は、5 章 [ストリームの一般的なオプションの設定 in IBM SPSS Modeler 15 ユーザー ガイド](#) を参照してください。

| 関数                   | 結果        | 説明  |
|----------------------|-----------|---|
| ITEM1 $\times$ ITEM2 | String    | 2 つのフィールドの値を連結し、結果の文字列を ITEM1ITEM2. の形式で返します。 |
| to_integer(ITEM)     | Integer   | 指定されたフィールドのストレージを整数に変換します。                    |
| to_real(ITEM)        | Real      | 指定されたフィールドのストレージを実数に変換します。                    |
| to_number(ITEM)      | 数値        | 指定されたフィールドのストレージを数値に変換します。                    |
| to_string(ITEM)      | String    | 指定されたフィールドのストレージを文字列に変換します。                   |
| to_time(ITEM)        | Time      | 指定されたフィールドのストレージを時間に変換します。                    |
| to_date(ITEM)        | Date      | 指定されたフィールドのストレージを日付に変換します。                    |
| to_timestamp(ITEM)   | Timestamp | 指定されたフィールドのストレージをタイムスタンプに変換します。               |



| 関数                  | 結果   | 説明  |
|---------------------|------|---|
| to_datetime(ITEM)   | 日時   | 指定されたフィールドのストレージを日付、時間またはタイムスタンプ値に変換します。  |
| datetime_date(ITEM) | Date | 数値、文字列またはタイムスタンプの日付値を返します。数値（秒単位）を日付へ変換しなおすことができるのは、この関数だけです。ITEM が文字列の場合は、現在のデータ フォーマットで文字列を解析することにより日付を作成します。この関数が正常に機能するためには、[ストリームのプロパティ] ダイアログ ボックスの [日付のフォーマット] に、正しい値が指定されていなければなりません。ITEM が数値の場合は、基準日（または紀元）からの秒数として解釈します。日付の端数は切り捨てられます。ITEM がタイムスタンプの場合は、日付をタイムスタンプの一部として返します。ITEM が日付の場合は、変更せずに返します。 |

## 比較関数

比較関数は、フィールドの値を互いに比較したり、指定した文字列と比較する場合に用いられます。たとえば、文字列が等しいかどうかは、= を使って確認することができます。次のように、文字列が等しいかどうかを調べます。 **Class = "class 1"**.

数値の比較を目的とする場合、greater（より大きい）は正の無限大に近いことを意味し、lesser（より小さい）は負の無限大に近いことを意味します。つまり、すべての負の数値は、すべての正の数値より小さいこととなります。

| 関数                              | 結果      | 説明   |
|---------------------------------|---------|--|
| count_equal(ITEM1, LIST)        | Integer | フィールドの LIST から ITEM1 と等しい数値を返します。ITEM1 が NULL の場合は、NULL を返します。 <a href="#">詳細は、7 章 複数フィールドの要約 in IBM SPSS Modeler 15 ユーザー ガイド を参照してください。</a> |
| count_greater_than(ITEM1, LIST) | Integer | フィールドの LIST から ITEM1 より大きい数値を返します。ITEM1 が NULL の場合は、NULL を返します。  |
| count_less_than(ITEM1, LIST)    | Integer | フィールドの LIST から ITEM1 より小さい数値を返します。ITEM1 が NULL の場合は、NULL を返します。  |
| count_not_equal(ITEM1, LIST)    | Integer | フィールドの LIST から ITEM1 と等しくない数値を返します。ITEM1 が NULL の場合は、NULL を返します。   |
| count_nulls(LIST)               | Integer | フィールドの LIST から NULL 値の数を返します。  |
| count_non_nulls(LIST)           | Integer | フィールドの LIST から NULL 値以外の数を返します。  |
| date_before(ITEM1, ITEM2)       | ブール     | 日付値の順序の確認に用いられます。DATE1 が DATE2 より前の場合に真 (true) を返します。  |

| 関数                         | 結果      | 説明   |
|----------------------------|---------|--|
| first_index(ITEM, LIST)    | Integer | フィールドの LIST から ITEMを含む最初のフィールドの索引、または値が見つからないなら 0 を返します。サポート対象は、文字列、整数、実数型のみです。詳細は、7 章 複数レスポンス データの処理 in IBM SPSS Modeler 15 ユーザー ガイドを参照してください。  |
| first_non_null(LIST)       | CLEM 式  | 提供されるフィールド リストの最初の非ヌル値を返します。ストレージ タイプはすべてサポート対象です。   |
| first_non_null_index(LIST) | Integer | 非ヌル値を含む特定の LIST の最初のフィールドの索引、またはすべての値がヌル 0 を返します。ストレージ タイプはすべてサポート対象です。  |
| ITEM1 = ITEM2              | ブール     | ITEM1 が ITEM2 と等しい場合に真 (true) を返します。   |
| ITEM1 /= ITEM2             | ブール     | 2 つの文字列が異なるか、同じでも 0 の場合に真 (true) を返します。  |
| ITEM1 < ITEM2              | ブール     | ITEM1 が ITEM2 より小さい場合に真 (true) を返します。  |
| ITEM1 <= ITEM2             | ブール     | ITEM1 が ITEM2 以下の場合に真 (true) を返します。  |
| ITEM1 > ITEM2              | ブール     | ITEM1 が ITEM2 より大きい場合に真 (true) を返します。  |
| ITEM1 >= ITEM2             | ブール     | ITEM1 が ITEM2 以上の場合に真 (true) を返します。  |
| last_index(ITEM, LIST)     | Integer | フィールドの LIST から ITEMを含む最新のフィールドの索引、または値が見つからないなら 0 を返します。サポート対象は、文字列、整数、実数型のみです。詳細は、7 章 複数レスポンス データの処理 in IBM SPSS Modeler 15 ユーザー ガイドを参照してください。  |
| last_non_null(LIST)        | CLEM 式  | 提供されるフィールド リストの最後の非ヌル値を返します。ストレージ タイプはすべてサポート対象です。   |
| last_non_null_index(LIST)  | Integer | 非ヌル値を含む特定の LIST の最後のフィールドの索引、またはすべての値がヌル 0 を返します。ストレージ タイプはすべてサポート対象です。  |
| max(ITEM1, ITEM2)          | CLEM 式  | ITEM1 または ITEM2 のどちらか大きい方を返します。  |
| max_index(LIST)            | Integer | 数値フィールドの LIST から最大値を含むフィールドの索引、またはすべての値がヌルなら 0 を返します。たとえば、3番目にリストされたフィールドに最大値がある場合は、インデックス値 3 を返します。複数のフィールドに最大値がある場合は、最初にリストされたもの (左端) を返します。詳細は、7 章 複数レスポンス データの処理 in IBM SPSS Modeler 15 ユーザー ガイドを参照してください。 |

| 関数                        | 結果      | 説明  |
|---------------------------|---------|---|
| max_n(LIST)               | 数値      | 数値フィールドの LIST から最大値を返します。フィールドのすべての値が NULL の場合は、NULL を返します。詳細は、7 章 複数フィールドの要約 in IBM SPSS Modeler 15 ユーザー ガイド を参照してください。  |
| member(ITEM, LIST)        | ブール     | ITEM が、指定された LIST のメンバーの場合に真 (true) を返します。それ以外の場合は、偽 (false) の値が返されます。また、フィールド名のリストを定義することもできます。詳細は、7 章 複数フィールドの要約 in IBM SPSS Modeler 15 ユーザー ガイド を参照してください。   |
| min(ITEM1, ITEM2)         | CLEM 式  | ITEM1 または ITEM2 のどちらか小さい方を返します。   |
| min_index(LIST)           | Integer | 数値フィールドの LIST から最小値を含むフィールドの索引、またはすべての値がヌルなら 0 を返します。たとえば、3番目にリストされたフィールドに最小値がある場合は、インデックス値 3 を返します。複数のフィールドに最小値がある場合は、最初にリストされたもの (左端) を返します。詳細は、7 章 複数レスポンス データの処理 in IBM SPSS Modeler 15 ユーザー ガイド を参照してください。 |
| min_n(LIST)               | 数値      | 数値フィールドの LIST から最小値を返します。フィールドのすべての値が NULL の場合は、NULL を返します。   |
| time_before(TIME1, TIME2) | ブール     | 時間値の順序の確認に用いられます。TIME1 が TIME2 より前の場合に真 (true) が返されます。  |
| value_at(INT, LIST)       |         | オフセットが有効値の範囲外の場合 (つまり 1 以上またはリストされたフィールドの数値以下)、オフセット NT または NULL でそれぞれリストされたフィールドの値を返します。ストレージ タイプはすべてサポート対象です。   |

## 論理関数

CLEM 式を使って論理演算を行うことができます。

| 関数              | 結果  | 説明   |
|-----------------|-----|--|
| COND1 and COND2 | ブール | この演算は論理積で、COND1 と COND2 の両方が真 (true) の場合に真 (true) の値を返します。COND1 が偽 (false) の場合、COND2 は評価されません。こうすることによって、COND2 の演算が正しいかどうかを COND1 で判断することができます。たとえば、 <code>length(Label) &gt;= 6</code> や <code>Label(6) = 'x'</code> となります。 |
| COND1 or COND2  | ブール | この演算は、(包括的) 論理和で、COND1 と COND2 のどちらかが真 (true) の場合、または両方とも真 (true) の場合に、真 (true) の値を返します。COND1 が真 (true) の場合、COND2 は評価されません。  |

| 関数  | 結果     | 説明   |
|---|--------|--|
| not(COND)   | ブール    | この演算は論理否定で、COND が偽 (false) の場合に真 (true) の値を返します。それ以外の場合、この演算は 0 の値を返します。   |
| if COND then EXPR1 else EXPR2 endif                           | CLEM 式 | この演算は条件評価です。COND が真 (true) の場合、この演算は EXPR1 の結果を返します。それ以外の場合は、EXPR2 を評価した結果を返します。   |
| if COND1 then EXPR1 elseif COND2 then EXPR2 else EXPR_N endif | CLEM 式 | この演算は複数の分岐を持つ条件評価です。COND1 が真 (true) の場合、この演算は EXPR1 の結果を返します。それ以外の場合で、COND2 が真ならば、この演算は EXPR2 を評価した結果を返します。それ以外の場合は、EXPR_N を評価した結果を返します。 |

## 数値関数

CLEM には、一般的に使われるさまざまな数値関数が用意されています。

| 関数            | 結果   | 説明  |
|---------------|------|---|
| -NUM          | 数値   | NUM を否定する場合に用いられます。対応する数値の符号を逆にした値を返します。  |
| NUM1 + NUM2   | 数値   | NUM1 と NUM2 を合計した値を返します。  |
| code -NUM2    | 数値   | NUM1 から NUM2 を減算した値を返します。   |
| NUM1 * NUM2   | 数値   | NUM1 を NUM2 で乗算した値を返します。  |
| NUM1 / NUM2   | 数値   | NUM1 を NUM2 で除算した値を返します。  |
| INT1 div INT2 | 数値   | 整数の除算を行うために用いられます。INT1 を INT2 で除算された値を返します。   |
| INT1 rem INT2 | 数値   | INT1 を INT2 で除算した剰余を返しますたとえば、 $INT1 - (INT1 \text{ div } INT2) * INT2$ となります。   |
| INT1 mod INT2 | 数値   | この関数は廃止されました。代わりに rem 関数を使用します。   |
| BASE ** POWER | 数値   | POWER 乗までべき乗した BASE を返します。BASE と POWER はどちらも任意の数値です (ただし、POWER が整数の 0 以外のいずれかのデータ型のゼロの場合、BASE はゼロ以外である必要があります)。POWER が整数の場合は、BASE のべき乗を順次掛けていくことによって計算されます。したがって、BASE が整数の場合、結果は整数になります。POWER が整数の 0 の場合、結果は常に BASE と同じデータ型の 1 になります。POWER が整数ではない場合、結果は $\exp(\text{POWER} * \log(\text{BASE}))$ のように計算されます。 |
| abs(NUM)      | 数値   | NUM の絶対値を返します。この値は常に、同じデータ型の数値になります。  |
| exp(NUM)      | Real | NUM 乗までべき乗した e を返します。この e は自然対数の底です。  |
| fracof(NUM)   | Real | NUM-intof(NUM) として定義される NUM の小数部を返します。  |

| 関数           | 結果      | 説明   |
|--------------|---------|--|
| intof(NUM)   | Integer | 引数を切り捨てて整数にします。NUM と同じ符号で、 $\text{abs(INT)} \leq \text{abs(NUM)}$ のような最大の絶対値を持つ整数を返します。                                  |
| log(NUM)     | Real    | NUM の自然対数 (底 e) を返します。NUM は、ゼロ以外でなければなりません。  |
| log10(NUM)   | Real    | NUM の常用対数を返します。NUM は、ゼロ以外でなければなりません。この確率は $\log(\text{NUM}) / \log(10)$ のように定義されます。                                      |
| negate(NUM)  | 数値      | NUM を否定する場合に用いられます。対応する数値の符号を逆にした値を返します。   |
| round(NUM)   | Integer | NUM 外政の数の場合は $\text{intof}(\text{NUM}+0.5)$ または NUM が負の数の場合は $\text{intof}(\text{NUM}-0.5)$ を指定して NUM を整数に丸めます。          |
| sign(NUM)    | 数値      | NUM の符号を判断するために用いられます。NUM が整数の場合、この演算は -1, 0, または 1 を返します。NUM が実数の場合、NUM が負、0、または正の値かによって、-1.0、0.0、または 1.0 を返します。        |
| sqrt(NUM)    | Real    | NUM の平方根を返します。NUM は正でなければなりません。  |
| sum_n(LIST)  | 数値      | 数値フィールドの LIST から合計値を返します。フィールドのすべての値が NULL の場合は、NULL を返します。詳細は、7 章 複数フィールドの要約 in IBM SPSS Modeler 15 ユーザー ガイド を参照してください。 |
| mean_n(LIST) | 数値      | 数値フィールドの LIST から平均値を返します。フィールドのすべての値が NULL の場合は、NULL を返します。  |
| sdev_n(LIST) | 数値      | 数値フィールドの LIST から標準偏差を返します。フィールドのすべての値が NULL の場合は、NULL を返します。   |

## 三角関数

この項の関数は、すべて引数として角度を取るかまたは、結果として角度を返します。どちらの場合も、角度の単位 (ラジアンまたは度) は関連するストリーム オプションの設定によって制御されます。

| 関数           | 結果   | 説明                      |
|--------------|------|-------------------------|
| arccos(NUM)  | Real | 指定した角度のアーコサインを計算します。    |
| arccosh(NUM) | Real | 指定した角度の双曲線アーコサインを計算します。 |
| arcsin(NUM)  | Real | 指定した角度のアークサインを計算します。    |
| arcsinh(NUM) | Real | 指定した角度の双曲線アークサインを計算します。 |
| arctan(NUM)  | Real | 指定した角度のアークタンジェントを計算します。 |

| 関数                    | 結果   | 説明  |
|-----------------------|------|---|
| arctan2(NUM_Y, NUM_X) | Real | NUM_Y / NUM_X のアークタンジェントを計算し、2つの数値の符号を使用して、象限情報を派生させます。結果は、 $-\pi < \text{ANGLE} \leq \pi$ (radians) – $-180 < \text{ANGLE} \leq 180$ (degrees) の範囲内の実数になります。 |
| arctanh(NUM)          | Real | 指定した角度の双曲線アークタンジェントを計算します。  |
| cos(NUM)              | Real | 指定した角度のコサインを計算します。  |
| cosh(NUM)             | Real | 指定した角度の双曲線コサインを計算します。   |
| pi                    | Real | この定数は、パイに最も近い値の実数です。  |
| sin(NUM)              | Real | 指定した角度のサインを計算します。   |
| sinh(NUM)             | Real | 指定した角度の双曲線サインを計算します。  |
| tan(NUM)              | Real | 指定した角度のタンジェントを計算します。  |
| tanh(NUM)             | Real | 指定した角度の双曲線タンジェントを計算します。   |

## 確率関数

確率分布で、学生からの t 分布値が特定値に満たなくなる確率など、さまざまな分布を基準にして確立が返されます。

| 関数                            | 結果   | 説明   |
|-------------------------------|------|--|
| cdf_chisq(NUM, DF)            | Real | 指定した自由度のカイ 2 乗分布からの値が特定の数字より小さくなる確率を返します。      |
| cdf_f(NUM, DF1, DF2)          | Real | DF1 と DF2 の自由度の F 分布からの値が指定した数字より小さくなる確率を返します。 |
| cdf_normal(NUM, MEAN, STDDEV) | Real | 指定した平均と標準偏差の正規分布からの値が指定した数字より小さくなる確率を返します。     |
| cdf_t(NUM, DF)                | Real | 指定した自由度の t 分布からの値が特定の数字より小さくなる確率を返します。         |

## ビット単位の整数演算

これらの演算を使用すると、2 の補数値を表すビット パターンとして整数を操作できます。この場合、ビット位置 N は  $2^{**}N$  の重みを持ちます。ビットは 0 から上方向に番号が付けられます。これらの演算は、整数の符号ビットが左方向に無限に拡張されているかのように処理します。つまり、最上位ビットを超えたすべての位置で、正の整数は 0 のビットを持ち、負の整数は 1 のビットを持ちます。

注： Bitwise 関数はスクリプトから呼び出せません。 詳細は、 3 章 p.33 スクリプト内の CLEM 式 を参照してください。

| 関数                                    | 結果      | 説明   |
|---------------------------------------|---------|--|
| <code>~~ INT1</code>                  | Integer | INT1 のビット単位の補数を生成します。つまり、INT1 の 0 がある各ビットが 1 になります。 <code>~~ INT = -(INT + 1)</code> は常に true となります。この関数は、スクリプトから呼び出すことができません。 詳細は、 3 章 p.33 スクリプト内の CLEM 式 を参照してください。   |
| <code>INT1    INT2</code>             | Integer | この演算の結果は、INT1 と INT2 のビット単位の「包括的論理和」またはです。つまり、INT1 と INT2 のどちらかまたは両方に 1 がある各ビットが 1 になります。  |
| <code>INT1   &amp; INT2</code>        | Integer | この演算の結果は、INT1 と INT2 のビット単位の「排他的論理和」またはです。つまり、INT1 と INT2 のどちらかにだけ（両方ではない）1 がある各ビットが 1 になります。  |
| <code>INT1 &amp;&amp; INT2</code>     | Integer | INT1 と INT2 のビット単位の「論理積」を生成します。つまり、INT1 と INT2 の両方に 1 がある各ビットが 1 になります。  |
| <code>INT1 &amp;&amp;~~ INT2</code>   | Integer | この演算の結果は、INT1 と、INT2 のビット単位の補数との、ビット単位の「論理積」です。つまり、INT1 のあるビット位置の値が 1 で、INT2 の同じビット位置の値が 0 の場合に、結果は 1 になります。これは <code>INT1&amp;&amp; (~INT2)</code> と同じで、INT2 の INT1 セットのビットを消去する場合に役に立ちます。  |
| <code>INT &lt;&lt; N</code>           | Integer | N の数だけ位置を左にシフトした INT1 のビットパターンを生成します。N の値が負の場合は、右にシフトします。  |
| <code>INT &gt;&gt; N</code>           | Integer | N の数だけ位置を右にシフトした INT1 のビットパターンを生成します。N の値が負の場合は、左にシフトします。  |
| <code>INT1 &amp;&amp;=_0 INT2</code>  | ブール     | ブール式 <code>INT1 &amp;&amp; INT2 != 0</code> と同じですが、より効率的です。  |
| <code>INT1 &amp;&amp;/=_0 INT2</code> | ブール     | ブール式 <code>INT1 &amp;&amp; INT2 == 0</code> と同じですが、より効率的です。  |
| <code>integer_bitcount(INT)</code>    | Integer | INT の 2 の補数 による結果の、1 または 0 のビットの数をカウントします。INT が負ではない場合、N は 1 のビット数になります。INT が負の場合、N は 0 のビット数になります。符号の拡張のため、負ではない整数には無限大の数の 0 のビットがあります。また、負の整数には無限大の数の 1 のビットがあります。 <code>integer_bitcount(INT) = integer_bitcount(-(INT+1))</code> は常に true となります。 |
| <code>integer_leastbit(INT)</code>    | Integer | 整数 INT の最下位ビット セットのビット位置 N を返します。N は、最も大きな 2 のべき乗です。N によって INT が正確に割られます。  |

| 関数                  | 結果      | 説明  |
|---------------------|---------|---|
| integer_length(INT) | Integer | INT の長さのビット数を、2 の補数の整数として返します。つまり、次のように、N は $INT < (1 \ll N)$ if $INT \geq 0$ $INT \geq (-1 \ll N)$ if $INT < 0$ のような最も小さい整数です。INT が負ではない場合、符号なしの整数として INT を表すには、少なくとも N ビットのフィールドが必要です。または、INT の符号にかかわらず、INT を符号付きの整数として表すには、少なくとも N+1 ビットが必要です。 |
| testbit(INT, N)     | ブール     | 整数 INT の N の位置にあるビットを検定し、ビット N の状態をブール値として返します。このブール値は、1 の場合は真 (true)、0 の場合は偽 (false) になります。  |

## 乱数関数

次の関数は、無作為に項目を選択したり、乱数を生成する場合に用いられます。

| 関数           | 結果     | 説明  |
|--------------|--------|---|
| oneof(LIST)  | CLEM 式 | 無作為 (ランダム) に選択された LIST の要素を返します。LIST 項目は [ITEM1,ITEM2,...,ITEM_N] のように入力する必要があります: フィールド名のリストも定義できることに注意してください。詳細は、7 章 <a href="#">複数フィールドの要約 in IBM SPSS Modeler 15 ユーザーガイド</a> を参照してください。 |
| random(NUM)  | 数値     | 1~NUM の範囲の、同じデータ型 (INT または REAL) の一様に分布した乱数を返します。整数を使用する場合は、整数だけが返されます。実数 (10 進数) を使用する場合は、実数値が返されます (精度はストリーム オプションによって決まります)。この関数で返される可能性がある最大の乱数は、NUM になります。                             |
| random0(NUM) | 数値     | random(NUM) と同じ性質を持ちますが、値の範囲が 0 から始まります。関数により返される可能性がある最大の乱数値が X と等しくなることはありません。  |

## 文字列関数

CLEM では、文字列に対して次の操作を行うことができます。

- 文字列の比較
- 文字列の生成
- 文字へのアクセス

CLEM で文字列とは、"string quotes" のように、二重引用符で囲まれた文字のことを表しています。任意の単一の英数字が、文字 (CHAR) になります。これらは、`z`、`A`、または `2` のように `<文字>` の形式で、単一後方引用



符を使用して CLEM 式内で宣言します。範囲外の文字、または文字列に対する逆索引文字は、未定義の動作が生じます。

注：SQL プッシュバックを使用する文字列と使用しない文字列と比較すると、接尾空白を含むさまざまな結果を生成する場合があります。

| 関数  | 結果      | 説明  |
|---|---------|---|
| <code>allbutfirst(N, STRING)</code>             | String  | STRING の先頭 N 文字、 $\delta$ 削除した文字列を返します。   |
| <code>allbutlast(N, STRING)</code>              | String  | STRING の最後の文字を削除した文字列を返します。   |
| <code>alphabefore(STRING1, STRING2)</code>      | ブール     | 文字列のアルファベット順を確認するために用いられます。STRING1 が STRING2 より前にある場合に真を返します。   |
| <code>endstring(LENGTH, STRING)</code>          | String  | 指定した文字列から最後の N 文字を抽出します。文字列の長さが指定した長さよりも短い、またはそれに等しい場合は、変更されません。  |
| <code>hasendstring(STRING, SUBSTRING)</code>    | Integer | この関数は、 <code>isendstring(SUBSTRING, STRING)</code> と同じです。   |
| <code>hasmidstring(STRING, SUBSTRING)</code>    | Integer | この関数は、 <code>ismidstring(SUBSTRING, STRING)</code> (埋め込みサブ文字列) と同じです。   |
| <code>hasstartstring(STRING, SUBSTRING)</code>  | Integer | この関数は、 <code>isstartstring(SUBSTRING, STRING)</code> と同じです。   |
| <code>hassubstring(STRING, N, SUBSTRING)</code> | Integer | この関数は、 <code>issubstring(SUBSTRING, N, STRING)</code> と同じです。N のデフォルトは 1 です。   |
| <code>count_substring(STRING, SUBSTRING)</code> | Integer | 指定したサブ文字列が文字列内に発生する回数を返します。例をあげると、次のようになります。<br><code>count_substring("foooo.txt", "oo")</code> は 3 を返します。                            |
| <code>hassubstring(STRING, SUBSTRING)</code>    | Integer | この関数は、 <code>issubstring(SUBSTRING, 1, STRING)</code> と同じです。N のデフォルトは 1 です。   |
| <code>isalphacode(CHAR)</code>                  | ブール     | CHAR が、文字コードが文字である指定された文字列 (通常フィールド名) 中の文字の場合に真を返します。それ以外の場合、この関数は 0 の値を返します。たとえば、 <code>isalphacode(produce_num(1))</code> のようになります。 |
| <code>isendstring(SUBSTRING, STRING)</code>     | Integer | 文字列 STRING がサブ文字列 SUBSTRING で終わる場合、この関数は、STRING 内の SUBSTRING の整数の添字を返します。それ以外の場合、この関数は 0 の値を返します。                                     |

| 関数                                       | 結果      | 説明   |
|--|---------|--|
| islowercode(CHAR)                        | ブール     | CHAR が指定された文字列（通常フィールド名）の小文字の場合に、真（true）の値を返します。それ以外の場合、この関数は 0 の値を返します。たとえば、islowercode('')や islowercode(country_name(2)) などが有効な式になります。                           |
| ismidstring(SUBSTRING, STRING)           | Integer | SUBSTRING が STRING の部分文字列で、STRING の初めの文字から始まっていないか、または最後の文字で終わっていない場合、この関数は部分文字列が始まる位置の添字を返します。それ以外の場合、この関数は 0 の値を返します。  |
| isnumbercode(CHAR)                       | ブール     | 指定された文字列（通常フィールド名）の CHAR が、文字コードが数字である文字の場合に真を返します。それ以外の場合、この関数は 0 の値を返します。たとえば、isnumbercode(product_id(2)) のようになります。  |
| isstartstring(SUBSTRING, STRING)         | Integer | 文字列 STRING がサブ文字列 SUBSTRING から始まる場合、この関数は添字 1 を返します。そうでない場合は、この関数は 0 の値を返します。  |
| issubstring(SUBSTRING, N, STRING)        | Integer | この関数は、文字列 STRING の N 番目の文字から始めて、文字列 SUBSTRING と等しいサブ文字列を検索します。文字列が見つかった場合、一致する部分文字列が始まる位置の添字 M（整数）を返します。それ以外の場合、この関数は 0 の値を返します。N が与えられていない場合、この関数はデフォルトで 1 になります。 |
| issubstring(SUBSTRING, STRING)           | Integer | この関数は、文字列 STRING の N 番目の文字から始めて、文字列 SUBSTRING と等しいサブ文字列を検索します。文字列が見つかった場合、一致する部分文字列が始まる位置の添字 M（整数）を返します。それ以外の場合、この関数は 0 の値を返します。N が与えられていない場合、この関数はデフォルトで 1 になります。 |
| issubstring_count(SUBSTRING, N, STRING): | Integer | 指定した STRING 内で N 番目に発生した SUBSTRING のインデックスを返します。N 番目に発生する SUBSTRING よりも少ない場合、0 を返します。  |

| 関数   | 結果      | 説明   |
|--|---------|--|
| <code>issubstring_lim(SUBSTRING, N, STARTLIM, ENDLIM, STRING)</code> | Integer | この関数は <code>issubstring</code> と同じですが、添字 <code>STARTLIM</code> から、またはその前から始まり、添字 <code>ENDLIM</code> で、またはその前で終わるように、マッチングが制限されます。 <code>STARTLIM</code> 制約または <code>ENDLIM</code> 制約は、どちらかの引数に偽 ( <code>false</code> ) の値を指定することによって無効にできます。たとえば、 <code>issubstring_lim(SUBSTRING, N, false, false, STRING)</code> は <code>issubstring</code> と同じです。  |
| <code>isuppercode(CHAR)</code>                                       | ブール     | この関数は <code>CHAR</code> が大文字の場合に、真 ( <code>true</code> ) の値を返します。それ以外の場合、この関数は 0 の値を返します。たとえば、 <code>isuppercode('')</code> や <code>isuppercode(country_name(2))</code> などが有効な式になります。  |
| <code>last(CHAR)</code>  | String  | この関数は、 <code>STRING</code> の最後の文字 <code>CHAR</code> を返します (少なくとも 1 文字以上の長さがなければなりません)。   |
| <code>length(STRING)</code>  | Integer | 文字列 <code>STRING</code> の長さ (つまり文字列内の半角文字数) を返します。   |
| <code>locchar(CHAR, N, STRING)</code>                                | Integer | シンボル値フィールド中の文字の位置を識別するために用いられます。この関数は、文字列 <code>STRING</code> 中の <code>N</code> 番目の文字から、文字 <code>CHAR</code> の検索を開始します。この関数は、文字が見つかった ( <code>N</code> から始まる) 位置を示す値を返します。文字が見つからない場合は 0 を返します。関数のオフセット ( <code>N</code> ) が無効な場合 (たとえばオフセットが文字列の長さを超えているなど)、この関数は <code>\$null\$</code> を返します。<br>たとえば、 <code>locchar('n', 2, web_page)</code> と指定すると、フィールド <code>web_page</code> 中の 2 番目の文字から、文字 <code>'n'</code> を検索します。<br>注: 指定する文字を、忘れずに、単一逆引用符で囲むようにしてください。 |
| <code>locchar_back(CHAR, N, STRING)</code>                           | Integer | <code>locchar</code> に似ていますが、 <code>N</code> 番目の文字から前方向に検索される点が異なります。たとえば、 <code>locchar_back('n', 9, web_page)</code> と指定すると、フィールド <code>web_page</code> の 9 番目の文字から、文字列の先頭方向に向かって検索が開始されます。関数のオフセットが無効な場合 (たとえばオフセットが文字列の長さを超えているなど)、この関数は <code>\$null\$</code> を返します。できる限り、 <code>locchar_back</code> とともに関数 <code>length(&lt;field&gt;)</code> を使用して、フィールドの現在の値の長さを動的に使用することをお勧めします。たとえ   |

| 関数   | 結果          | 説明  |
|--|-------------|---|
|  |             | ば、 <code>locchar_back('n', (length(web_page)), web_page)</code> となります。  |
| <code>lowertoupper(CHAR)</code><br><code>lowertoupper(String)</code> | CHAR または文字列 | 文字または文字列を入力にすることができ、同じデータ型の新しい項目を返すために用いられます。その際小文字はすべて同じ文字の 大文字に変換されます。たとえば、 <code>lowertoupper('a')</code> 、 <code>lowertoupper("My string")</code> 、および <code>lowertoupper(field_name(2))</code> となります。  |
| <code>matches</code>   | ブール         | 文字列が指定したパターンに一致する場合、真を返します。パターンは文字列リテラルにする必要があり、パターンを含むフィールド名にしてはなりません。クエスチョン マーク (?) をパターンに含めて正確に 1 つの文字に一致させることができます。アスタリスク (*) は 0 かそれ以上の文字数に一致します。リテラル クエスチョン マークまたはアスタリスクを (むしろ、ワイルドカードとして使用しないで) 一致させるために、バックslashをエスケープ文字として使用することができます。   |
| <code>replace(SUBSTRING, NEWSUBSTRING, String)</code>                | String      | 指定した String 内で、SUBSTRING のすべてのインスタンスを NEWSUBSTRING を使って置き換えます。  |
| <code>replicate(COUNT, String)</code>                                | String      | 指定した回数だけコピーされた元の文字列を含む文字列を返します。   |
| <code>stripchar(CHAR,String)</code>                                  | String      | 文字列またはフィールドから、指定した文字を削除します。この関数を利用すれば、データから通貨表記などの余分な記号を削除して、単純な数字または名前を取得できます。たとえば、シンタックス <code>stripchar('\$', 'Cost')</code> を使用すると、すべての値からドル記号を除去した新しいフィールドが返されます。<br>注：指定する文字を、忘れずに、単一逆引用符で囲むようにしてください。  |
| <code>skipchar(CHAR, N, String)</code>                               | Integer     | 文字列 String の N 文字目から、CHAR 以外の文字を検索します。この関数は、見つかった文字の位置を示す整数サブ文字列を返します。N 番目以降のすべての文字が CHAR の場合は、0 を返します。関数のオフセットが無効な場合 (たとえばオフセットが文字列の長さを超えているなど)、この関数は <code>\$null\$</code> を返します。<br><code>locchar</code> は、よく関数 <code>skipchar</code> と一緒に、N (文字列の検索開始点) の値を判断するために用いられます。たとえば、 <code>skipchar('s', (locchar('s', 1, "MyString")), "MyString")</code> となります。 |

| 関数  | 結果          | 説明  |
|---|-------------|---|
| skipchar_back(CHAR, N, STRING)              | Integer     | skipchar に似ていますが、N 番目の文字から前に戻る方向に検索される点が異なります。  |
| startstring(LENGTH, STRING)                 | String      | 指定した文字列から最初のN 文字を抽出します。文字列の長さが指定した長さよりも短いか、またはそれに等しい場合は、変更されません。  |
| strmember(CHAR, STRING)                     | Integer     | locchar(CHAR, 1, STRING) への接続CHAR が最初に発生する、または 0 の地点を示す整数の部分文字列を返します。関数に無効なオフセットがある場合（たとえば、オフセットが文字列の長さを超えている）、この関数は \$null\$ を返します。            |
| subscrs(N, STRING)                          | CHAR        | 入力文字列 STRING の N 番目の文字 CHAR を返します。この関数は、STRING(N) という短い形式で記述することもできます。たとえば、lowertoupper("name"(1)) が有効な式となります。                                  |
| substring(N, LEN, STRING)                   | String      | 文字列 SUBSTRING を返します。この文字列は、文字列 STRING の添字 N から LEN 文字分の文字列で構成されています。  |
| substring_between(N1, N2, STRING)           | String      | 添字 N1 から始まり、添字 N2 で終わる STRING のサブ文字列を返します。  |
| trim(STRING)                                | String      | 指定した文字列から、文字列の前後の空白文字を削除します。  |
| trim_start(STRING)                          | String      | 指定した文字列から、文字列の前の空白文字を削除します。   |
| trimend(STRING)                             | String      | 指定した文字列から、文字列の後の空白文字を削除します。   |
| unicode_char(NUM)                           | CHAR        | NUMの Unicode 値を返します。  |
| unicode_value(CHAR)                         | NUM         | CHAR の Unicode 値を返します。  |
| uppertolower(CHAR)<br>uppertolower (STRING) | CHAR または文字列 | 文字または文字列を入力にすることができ、同じデータ型の新しい項目を返すために用いられます。その際、大文字はすべて同じ文字の小文字に変換されます。<br>注：文字列は二重引用符で、文字は単一の逆引用符で忘れずに指定するようにしてください。単純なフィールド名の場合は、引用符は使用しません。 |

## SoundEx 関数

SoundEx は、サウンドは分かっているにもかかわらず正しいスペルが分からない場合に、文字列を検索するために用いられる方法です。1918年に開発されたこの方法では、特定文字の発音方法についての音声的な仮定を基準にする、似通ったサウンドの単語が検索されます。この方法は、たとえば、似通った名前のスペルや発音がさまざまに異なる場合に、データベースで名前検索を行うために使用されます。基本的な SoundEx アルゴリズムはさまざまな文献で引用されており、また、(ph や f のように、文字列の前の文字の組み合わせが、同じサウンドを持つにもかかわらず一致しないことなど) 制約があることは知られていますが、ほとんどのデータベースで何らかの形でサポートされています。

| 関数                                   | 結果      | 説明   |
|--------------------------------------|---------|--|
| soundex(String)                      | Integer | 指定した String の 4 文字の SoundEx コードを返します。  |
| soundex_difference(String1, String2) | Integer | 2 つの文字列で SoundEx エンコードが同じ文字数を示す 0 から 4 の整数を返します。ここで、0 は類似性がないこと、また、4 は強い類似性があること、または同じ文字列であることを示します。 |

## 日付および時刻の関数

CLEM には、日付や時間を表す文字列変数の日付と時間ストレージのフィールドを操作する関数が用意されています。使用する日付と時間の書式は、ストリームごとに異なり、[ストリームのプロパティ] ダイアログ ボックスで指定します。日付と時間の関数は、現在選択されているフォーマットに従って、日付と時間の文字列を解析します。

日付に 2 桁だけを使用する (世紀を指定しない) 年を指定すると、IBM® SPSS® Modeler では、[ストリームのプロパティ] ダイアログ ボックスで指定されているデフォルトの世紀が使用されます。

注：日付と時間の関数はスクリプトから呼び出せません。詳細は、3章 p.33 スクリプト内の CLEM 式 を参照してください。

| 関数                                 | 結果        | 説明  |
|------------------------------------|-----------|---|
| @TODAY                             | String    | [ストリームのプロパティ] ダイアログ ボックスで [日/分をロールオーバー] を選択している場合、この関数は現在の日付形式を使用して、現在の日付を文字列として返します。2桁の日付形式を使用しており、[日/分をロールオーバー]を選択していない場合は、現在のサーバーの \$null\$ を返します。この関数は、スクリプトから呼び出すことができません。詳細は、3章 p.33 スクリプト内の CLEM 式 を参照してください。  |
| to_time(ITEM)                      | Time      | 指定されたフィールドのストレージを時間に変換します。  |
| to_date(ITEM)                      | Date      | 指定されたフィールドのストレージを日付に変換します。  |
| to_timestamp(ITEM)                 | Timestamp | 指定されたフィールドのストレージをタイムスタンプに変換します。   |
| to_datetime(ITEM)                  | 日時        | 指定されたフィールドのストレージを日付、時間またはタイムスタンプ値に変換します。  |
| datetime_date(ITEM)                | Date      | 数値、文字列またはタイムスタンプの日付値を返します数値（秒単位）を日付をへ変換しなおすことができるのは、この関数だけです。ITEM が文字列の場合は、現在のデータ フォーマットで文字列を解析することにより日付を作成します。この関数が正常に機能するためには、[ストリームのプロパティ] ダイアログ ボックスの [日付のフォーマット] に、正しい値が指定されていなければなりません。ITEM が数値の場合は、基準日（または紀元）からの秒数として解釈します。日付の端数は切り捨てられます。ITEM がタイムスタンプの場合は、日付をタイムスタンプの一部として返します。ITEM が日付の場合は、変更せずに返します。 |
| date_before(DATE1, DATE2)          | ブール       | DATE1 が DATE2 より前の日付またはタイムスタンプの場合に true を返します。それ以外の場合、この関数は 0 の値を返します。  |
| date_days_difference(DATE1, DATE2) | Integer   | 日付またはタイムスタンプ DATE1 から日付またはタイムスタンプ DATE2 までの日数を整数で返します。DATE2 が DATE1 よりも前の場合、この関数は負の数値を返します。   |
| date_in_days(DATE)                 | Integer   | 基準日から DATE で表される日付またはタイムスタンプまでの日数を整数で返します。DATE が基準日より前の場合、この関数は負の数値を返します。計算を適切に行うには、有効な日付を指定する必要があります。たとえば、日付として 2001 年 2 月 29 日を指定することはできません。2001 年は閏年ではないので、この日付は存在しません。  |

| 関数  | 結果      | 説明  |
|---|---------|---|
| date_in_months(DATE)  | Real    | 基準日から DATE で表される日付またはタイムスタンプまでの月数を実数で返します。これは、1 か月を 30.4375 日と仮定した近似値になります。DATE が基準日より前の場合、この関数は負の数値を返します。計算を適切に行うには、有効な日付を指定する必要があります。たとえば、日付として 2001 年 2 月 29 日を指定することはできません。2001 年は閏年ではないので、この日付は存在しません。 |
| date_in_weeks(DATE)   | Real    | 基準日から DATE で表される日付またはタイムスタンプまでの週数を実数で返します。ここでは、1 週間を 7.0 日と仮定しています。DATE が基準日より前の場合、この関数は負の数値を返します。計算を適切に行うには、有効な日付を指定する必要があります。たとえば、日付として 2001 年 2 月 29 日を指定することはできません。2001 年は閏年ではないので、この日付は存在しません。         |
| date_in_years(DATE)   | Real    | 基準日から DATE で表される日付またはタイムスタンプまでの年数を実数で返します。これは、1 年 365.25 日を基準とした近似値になります。DATE が基準日より前の場合、この関数は負の数値を返します。計算を適切に行うには、有効な日付を指定する必要があります。たとえば、日付として 2001 年 2 月 29 日を指定することはできません。2001 年は閏年ではないので、この日付は存在しません。   |
| date_months_difference (DATE1, DATE2)                       | Real    | 日付またはタイムスタンプ DATE1 から日付またはタイムスタンプ DATE2 までの月数を実数で返します。これは、1 か月を 30.4375 日と仮定した近似値になります。DATE2 が DATE1 よりも前の場合、この関数は負の数値を返します。  |
| datetime_date(YEAR, MONTH, DAY)                             | Date    | YEAR、MONTH、および DAY の日付値を作成します。引数は整数でなければなりません。  |
| datetime_day(DATE)  | Integer | 指定されたDATE またはタイムスタンプから、日付を返します。結果は 1～31 の範囲の整数になります。  |
| datetime_day_name(DAY)                                      | String  | 指定された DAY のフルネームを返します。引数は、1 (日曜)～7 (土曜) の範囲の整数でなければなりません。   |
| datetime_hour(TIME)   | Integer | TIME またはタイムスタンプから時間を返します。結果は 0～23 の範囲の整数になります。  |
| datetime_in_seconds(TIME)                                   | Real    | TIME に保存された秒の部分の部分を返します。  |
| datetime_in_seconds(DATE),<br>datetime_in_seconds(DATETIME) | Real    | 現在の DATE または DATETIME と基準日の間の差 (1900-01-01) から集計した数値を秒に変換して返します。  |
| datetime_minute(TIME)                                       | Integer | TIME またはタイムスタンプから分を返します。結果は 0～59 の範囲の整数になります。   |
| datetime_month(DATE)  | Integer | DATE またはタイムスタンプから月を返します。結果は 1～12 の範囲の整数になります。   |



| 関数   | 結果        | 説明   |
|--|-----------|--|
| datetime_month_name (MONTH)                                | String    | 指定された DAY のフルネームを返します。引数は、1～12 の範囲の整数でなければなりません。   |
| datetime_now   | Timestamp | 現在の時刻をタイムスタンプとして返します。  |
| datetime_second(TIME)                                      | Integer   | TIME またはタイムスタンプから秒を返します。結果は 0～59 の範囲の整数になります。  |
| datetime_day_short_name(DAY)                               | String    | DAY の名前を省略形で返します。引数は、1 (日曜)～7 (土曜) の範囲の整数でなければなりません。   |
| datetime_month_short_name(MONTH)                           | String    | MONTH の名前を省略形で返します。引数は、1～12 の範囲の整数でなければなりません。  |
| datetime_time(HOUR, MINUTE, SECOND)                        | Time      | 指定された HOUR、MINUTE、および SECOND の時間値を返します。引数は整数でなければなりません。  |
| datetime_time(ITEM)  | Time      | ITEM の時間値を返します。  |
| datetime_timestamp(YEAR, MONTH, DAY, HOUR, MINUTE, SECOND) | Timestamp | 与えられた YEAR、MONTH、DAY、HOUR、MINUTE、および SECOND のタイムスタンプ値を返します。  |
| datetime_timestamp(DATE, TIME)                             | Timestamp | DATE および TIME のタイムスタンプ値を返します。  |
| datetime_timestamp (NUMBER)                                | Timestamp | 与えられた秒数のタイムスタンプ値を返します。   |
| datetime_weekday(DATE)                                     | Integer   | 指定されたDATE またはタイムスタンプから、曜日を返します。  |
| datetime_year(DATE)  | Integer   | DATE またはタイムスタンプから年を返します。結果は 2002 のような整数になります。  |
| date_weeks_difference (DATE1, DATE2)                       | Real      | 日付またはタイムスタンプ DATE1 から日付またはタイムスタンプ DATE2 までの週数を実数で返します。ここでは、1 週間を 7.0 日と仮定しています。DATE2 が DATE1 よりも前の場合、この関数は負の数値を返します。   |
| date_years_difference (DATE1, DATE2)                       | Real      | 日付またはタイムスタンプ DATE1 から日付またはタイムスタンプ DATE2 までの年数を実数で返します。これは、1 年 365.25 日を基準とした近似値になります。DATE2 が DATE1 よりも前の場合、この関数は負の数値を返します。   |
| time_before(TIME1, TIME2)                                  | ブール       | TIME1 が TIME2 より前の時間またはタイムスタンプの場合に真を返します。それ以外の場合、この関数は 0 の値を返します。  |
| time_hours_difference (TIME1, TIME2)                       | Real      | 時間またはタイムスタンプ TIME1 と TIME2 間の時間差 (時間) を実数で返します。[ストリームのプロパティ] ダイアログ ボックスで [日/分をロールオーバー] を選択している場合、TIME1 の値の方が大きいと、その値は前の日付を参照します。ロールオーバー オプションをオンにしていない場合、TIME1 の値の方が大きいと、返される値は負になります。 |

| 関数                                 | 結果      | 説明   |
|------------------------------------|---------|--|
| time_in_hours(TIME)                | Real    | TIME で表される時間を実数で返します。たとえば、時間のフォーマット HHMM では、式 <code>time_in_hours('0130')</code> は 1.5 に評価されます。TIME は時間またはタイムスタンプを示します。  |
| time_in_mins(TIME)                 | Real    | TIME で表される分を実数で返します。TIME は時間またはタイムスタンプを示します。   |
| time_in_secs(TIME)                 | Integer | TIME で表される秒を整数で返します。TIME は時間またはタイムスタンプを示します。   |
| time_mins_difference(TIME1, TIME2) | Real    | 時間またはタイムスタンプ TIME1 と TIME2 間の時間差 (分) を実数で返します。[ストリームのプロパティ] ダイアログ ボックスで [日/分をロールオーバー] を選択している場合、TIME1 の値の方が大きいと、その値は前の日 (または、現在のフォーマットで分と秒だけが指定されている場合は前の時間) を参照します。ロールオーバー オプションをオンにしていない場合、TIME1 の値の方が大きいと、返される値は負になります。 |
| time_secs_difference(TIME1, TIME2) | Integer | TIME1 と TIME2 の時間またはタイムスタンプの差異を秒で返します。[ストリームのプロパティ] ダイアログ ボックスで [日/分をロールオーバー] を選択している場合、TIME1 の値の方が大きいと、その値は前の日 (または、現在のフォーマットで分と秒だけが指定されている場合は前の時間) を参照します。ロールオーバー オプションをオンにしていない場合、TIME1 の値の方が大きいと、返される値は負になります。         |

### 日付と時刻の値の変換:

変換関数および日付や時刻の値のような、入力に特別な型が必要なその他の関数は、[ストリームのオプション] ダイアログ ボックスに指定されている現在のフォーマットに依存します。たとえば、DATE という名前のフィールド名がある場合は、Jan 2003、Feb 2003 などの値で文字列として保存されており、次のように日付ストレージへ変換できます。

`to_date(DATE)`

この返還を行うには、ストリームのデフォルト日付フォーマットとして、一致する日付フォーマット `MON YYYY` を選択します。詳細は、[5 章 ストリームの一般的なオプションの設定 in IBM SPSS Modeler 15 ユーザー ガイド](#) を参照してください。

フィルター ノードを使用して文字列値を日付へ変換する例については、streams サブフォルダ内の ¥Demos フォルダにインストールされている、ストリーム `broadband_create_models.str` を参照してください。詳細は、

14 章 時系列ノードによる予測 in IBM SPSS Modeler 15 アプリケーション ガイド を参照してください。

**数値として保存される日付**：上記例の DATE はフィールド名、to\_date は CLEM 関数です。数値として保存された日付がある場合は、数値が基準日（または紀元）からの 秒数として解釈される datetime\_date 関数を用いることでそれらを変換できます。

```
datetime_date(DATE)
```

日付を秒数へ（および逆）変換することで、次のように現在の日付に一定の日数をプラス、マイナスするといった計算を実行できます。

```
datetime_date((date_in_days(DATE)-7)*60*60*24)
```

## シーケンス関数

一部の演算子では、イベントのシーケンス（順序）が重要になります。アプリケーションで使用できるレコード シーケンスは、以下のとおりです。

- シーケンスと時系列
- シーケンス関数
- レコード インデックスの作成
- 値の平均、合計、および比較
- 変化の把握（差分）
- @SINCE
- オフセット値
- その他のシーケンス機能

多くのアプリケーションでは、ストリームを通過している各レコードは、それぞれ個別で、他のすべてのレコードから独立したものと見なされます。通常、このような場合は、レコードの順序は重要ではありません。

ただし、問題によっては、レコード シーケンスが非常に重要になります。特に時系列の場合がそうで、レコードのシーケンスは、イベントまたは発生の順序、すなわちシーケンスを表します。各レコードは、特定の瞬間のスナップショットを示します。しかし、最も重要な情報は、瞬間的な値にあるのではなく、このような値が時間の経過に伴ってどのように変化し、動いていくのかということにあるのです。

もちろん、該当するパラメータが時間以外のものであってもかまいません。たとえば、レコードが、線からの距離について実行される分析を示している場合でも、同じ原則が適用されます。

シーケンスおよび特殊関数は、次の特徴によってすぐに判別できます。

- 関数名の最初に @ が付いている。
- 関数名が大文字である。

シーケンス関数は、ノードによって現在処理中のレコード、すでにノードを通過したレコード、あるいはまだノードに到達していないレコードを参照します。シーケンス関数は、CLEM 式の他の要素と自由に組み合わせて使用できますが、引数としての使用を制約されているものもあります。

## 例

ある事象が発生してから、またはある条件が真になってからの長さを知りたい場合があります。その場合は、次のように **@SINCE** を使用します。

**@SINCE(Income > Outgoings)**

この関数は、指定した条件が真 (true) であった最後のレコードのオフセットを返します。つまり、指定した条件が真 (true) であった最後のレコード以前のレコード数を返します。指定した条件が真にならなかった場合、**@SINCE** は **@INDEX + 1** を返します。

**@SINCE** で使用される式の現在のレコードの値を参照したいこともあるでしょう。関数 **@THIS** を使用して、フィールド名が常に現在のレコードに適用されるように指定します。**Concentration** フィールドの値が、現在のレコードの 2 倍より大きい最後のレコードのオフセットを調べるには、次のように記述します。

**@SINCE(Concentration > 2 \* @THIS(Concentration))**

定義により、現在のレコードに対して真 (true) である条件を **@SINCE** に指定する場合があります。次に例を示します。

**@SINCE(ID == @THIS(ID))**

この場合、**@SINCE** は現在のレコードに対して条件を評価しません。前のレコードと現在のレコードに対して条件を評価する場合は、同様な関数 **@SINCEO** を使用します。現在のレコードで条件が真 (true) の場合は、**@SINCEO** は **0** を返します。

注：@ 関数はスクリプトから呼び出せません。詳細は、3 章 p.33 スクリプト内の CLEM 式を参照してください。

| 関数                        | 結果   | 説明  |
|---------------------------|------|---|
| <b>MEAN(FIELD)</b>        | Real | 指定された FIELD または FIELDS に対して、値の平均値を返します。   |
| <b>@MEAN(FIELD, EXPR)</b> | Real | 現在のレコードを含めて、現在のノードが受け取った最後の EXPR レコードまでの、FIELD の値の平均値を返します。FIELD は数値型フィールドの名前でなければなりません。EXPR は、0 より大きい整数を評価する任意の式を使用できます。EXPR を省略した場合、または EXPR の値が、現在までに受け取ったレコード数を超えている場合は、現在までに受け取ったすべてのレコードの平均が返されます。この関 |

## CLEM 言語に関するリファレンス

| 関数                      | 結果      | 説明  |
|-------------------------|---------|---|
|                         |         | 数は、スクリプトから呼び出すことができません。詳細は、3 章 p.33 スクリプト内の CLEM 式を参照してください。  |
| @MEAN(FIELD, EXPR, INT) | Real    | 現在のレコードを含めて、現在のノードが受け取った最後の EXPR レコードまでの、FIELD の値の平均値を返します。FIELD は数値型フィールドの名前でなければなりません。EXPR は、0 より大きい整数を評価する任意の式を使用できます。EXPR を省略した場合、または EXPR の値が、現在までに受け取ったレコード数を超えている場合は、現在までに受け取ったすべてのレコードの平均が返されます。INT には、参照する値の最大数を指定します。この方法は、2 つの引数だけを使用するよりも効率的です。 |
| @DIFF1(FIELD)           | Real    | FIELD1 の最初の差分を返します。したがって、1 つの引数を指定する形式では、単純にフィールドの現在値と前の値の差分を返します。前に関連するレコードが存在しない場合は 0 を返します。  |
| @DIFF1(FIELD1, FIELD2)  | Real    | 2 つの引数を指定する形式では、FIELD2 に関する FIELD1 の最初の差分を返します。前に関連するレコードが存在しない場合は 0 を返します。   |
| @DIFF2(FIELD)           | Real    | FIELD1 の 2 番目の差分を返します。したがって、1 つの引数を指定する形式では、単純にフィールドの現在値と前の値の差分を返します。前に関連するレコードが存在しない場合は 0 を返します。   |
| @DIFF2(FIELD1, FIELD2)  | Real    | 2 つの引数を指定する形式では、FIELD2 に関する FIELD1 の最初の差分を返します。前に関連するレコードが存在しない場合は 0 を返します。   |
| @INDEX                  | Integer | 現在のレコードのインデックスを返します。インデックスは、レコードが現在のノードに到達したときにレコードに対して割り当てられます。最初のレコードにはインデックス 1 が与えられます。インデックスは、その後の各レコードに対して 1 ずつ増やされます。   |
| @LAST_NON_BLANK(FIELD)  | CLEM 式  | 上流の入力ノードまたはデータ型ノードで定義されるように、空白でない FIELD の最後の値を返します。それまでに読み込んだレコードの FIELD の値がすべて空白である場合は、\$nullS を返します。ユーザー欠損値とも呼ばれる空白値は、各フィールドに個別に定義することができますことに注意してください。   |
| @MAX(FIELD)             | 数値      | 指定された FIELD の最大値を返します。  |

| 関数                     | 結果     | 説明  |
|------------------------|--------|---|
| @MAX(FIELD, EXPR)      | 数値     | 現在のレコードを含めて、現在までに受け取った過去 EXPR レコードの FIELD の最大値を返します。FIELD は数値型フィールドの名前でなければなりません。EXPR は、0 より大きい整数を評価する任意の式を使用できます。  |
| @MAX(FIELD, EXPR, INT) | 数値     | 現在のレコードを含めて、現在までに受け取った過去 EXPR レコードの FIELD の最大値を返します。FIELD は数値型フィールドの名前でなければなりません。EXPR は、0 より大きい整数を評価する任意の式を使用できます。EXPR を省略した場合、または EXPR の値が、現在までに受け取ったレコード数を超えている場合は、現在までに受け取ったすべてのレコードの最大値が返されます。INT には、参照する値の最大数を指定します。この方法は、2 つの引数だけを使用するよりも効率的です。   |
| @MIN(FIELD)            | 数値     | 指定された FIELD の最小値を返します。  |
| @MIN(FIELD, EXPR)      | 数値     | 現在のレコードを含めて、現在までに受け取った過去 EXPR レコードの FIELD の最小値を返します。FIELD は数値型フィールドの名前でなければなりません。EXPR は、0 より大きい整数を評価する任意の式を使用できます。  |
| @MIN(FIELD, EXPR, INT) | 数値     | 現在のレコードを含めて、現在までに受け取った過去 EXPR レコードの FIELD の最小値を返します。FIELD は数値型フィールドの名前でなければなりません。EXPR は、0 より大きい整数を評価する任意の式を使用できます。EXPR を省略した場合、または EXPR の値が、現在までに受け取ったレコード数を超えている場合は、現在までに受け取ったすべてのレコードの最小値が返されます。INT には、参照する値の最大数を指定します。この方法は、2 つの引数だけを使用するよりも効率的です。   |
| @OFFSET(FIELD, EXPR)   | CLEM 式 | 現在のレコードから EXPR で指定された値のオフセットにあるレコードの FIELD の値を返します。正のオフセットがすでにパスしたレコードを参照するのに対し、負のオフセットはまだ到着していないレコードに「先読み」を指定します。たとえば、@OFFSET(Status, 1) と指定すると、前のレコードの Status フィールドの値が返されます。一方、@OFFSET(Status, -4) では、値を取得するためにシーケンス内で 4 個先のレコードを「先読み (look ahead)」(つまり、このノードをまだ通過していないレコードまで) します。負の (先読み) オフセットは、定数として指定する必要があります。負のオフセットに限っては、EXPR も任意の CLEM 式であり、現在のレコードに対してオフセットを与えるために評価されます。この場合、性能を改善するために、この関数の引数が 3 個のバージョンを使用する必要があります (次の関数を参照)。この式が負ではない整数以外の値 |

## CLEM 言語に関するリファレンス

| 関数                        | 結果     | 説明   |
|---------------------------|--------|--|
|                           |        | を返す場合、エラーになります。つまり、計算された後方参照を指定することは不正です。<br>注：自己参照の @OFFSET 関数ではリテラルの先読みを使用できません。たとえば、置換ノードでは、field1 の値を @OFFSET(field1,-2) のような式を使用して置換できません。  |
| @OFFSET(FIELD, EXPR, INT) | CLEM 式 | @OFFSET 関数と同じ演算を行います。3 番目の引数 INT に、前方参照する値の最大数を指定することができます。オフセットを式から計算することができる場合、性能を改善するために、この 3 番目の引数を使用する必要があります。<br>たとえば、@OFFSET(Foo, Month, 12) のような式では、システムが Foo の最後の 12 個の値だけを保持していればよいと判断できます。さもなくば、安全のため、全ての値を保存しておく必要があります。オフセットの値が定数の場合-これには負数の「先読み」用オフセットの値が定数である場合も含まれます-、3 番目の引数は無意味で、この関数の引数が 2 個のバージョンを使用する必要があります。前述の 2 引数バージョンの自己参照関数に関する注意事項を参照してください。 |
| @SDEV(FIELD)              | Real   | 指定された FIELD または FIELDS に対して、値の平均値を返します。  |
| @SDEV(FIELD, EXPR)        | Real   | 現在のレコードを含めて、現在のノードが受け取った最後の EXPR レコードまでの、FIELD の値の標準偏差を返します。FIELD は数値型フィールドの名前でなければなりません。EXPR は、0 より大きい整数を評価する任意の式を使用できます。EXPR を省略した場合、または EXPR の値が、現在までに受け取ったレコード数を超えている場合は、現在までに受け取ったすべてのレコードの標準偏差を返します。   |
| @SDEV(FIELD, EXPR, INT)   | Real   | 現在のレコードを含めて、現在のノードが受け取った最後の EXPR レコードまでの、FIELD の値の標準偏差を返します。FIELD は数値型フィールドの名前でなければなりません。EXPR は、0 より大きい整数を評価する任意の式を使用できます。EXPR を省略した場合、または EXPR の値が、現在までに受け取ったレコード数を超えている場合は、現在までに受け取ったすべてのレコードの標準偏差を返します。INT には、参照する値の最大数を指定します。この方法は、2 つの引数だけを使用するよりも効率的です。  |
| @SINCE(EXPR)              | CLEM 式 | 任意の CLEM 式が真 (true) の場合に、EXPR から過ぎたレコード数を返します。   |
| @SINCE(EXPR, INT)         | CLEM 式 | 2 番目の引数 INT には、前方参照するレコードの最大数を指定します。EXPR が真になっていない場合、INT は @INDEX+1 になります。   |

| 関数                     | 結果     | 説明  |
|------------------------|--------|---|
| @SINCE0(EXPR)          | CLEM 式 | 現在のレコードも考慮します。一方、@SINCE は現在のレコードは考慮しません。つまり @SINCE0 は、現在のレコードについて EXPR が真 (true) の場合に 0 を返します。  |
| @SINCE0(EXPR, INT)     | CLEM 式 | 2 番目の引数 INT には、前方参照するレコードの最大数を指定します。  |
| @SUM(FIELD)            | 数値     | 指定された FIELD または FIELDS に対して、値の合計値を返します。   |
| @SUM(FIELD, EXPR)      | 数値     | 現在のレコードを含めて、現在のノードが受け取った最後の EXPR レコードまでの、FIELD の値の合計値を返します。FIELD は数値型フィールドの名前でなければなりません。EXPR は、0 より大きい整数を評価する任意の式を使用できます。EXPR を省略した場合、または EXPR の値が、現在までに受け取ったレコード数を超えている場合は、現在までに受け取ったすべてのレコードの合計が返されます。  |
| @SUM(FIELD, EXPR, INT) | 数値     | 現在のレコードを含めて、現在のノードが受け取った最後の EXPR レコードまでの、FIELD の値の合計値を返します。FIELD は数値型フィールドの名前でなければなりません。EXPR は、0 より大きい整数を評価する任意の式を使用できます。EXPR を省略した場合、または EXPR の値が、現在までに受け取ったレコード数を超えている場合は、現在までに受け取ったすべてのレコードの合計が返されます。INT には、参照する値の最大数を指定します。この方法は、2 つの引数だけを使用するよりも効率的です。 |
| @THIS(FIELD)           | CLEM 式 | 現在のレコードの FIELD で指定された名前のフィールドの値を返します。@SINCE 式でだけ使用されます。   |

## グローバル関数

関数 @MEAN、@SUM、@MIN、@MAX、および @SDEV は、最大すべてのレコードおよび現在のレコードに機能します。しかし、現在のレコードの値とデータセット全体での値とを比較できると便利な場合もあります。グローバル ノードを使ってデータ セット全体の値を生成したら、CLEM 式でグローバル関数を使ってこれらの値にアクセスすることができます。

例をあげると、次のようになります。

@GLOBAL\_MAX(Age)

は、データセット内で最も大きい Age の値を返します。一方、

(Value - @GLOBAL\_MEAN(Value)) / @GLOBAL\_SDEV(Value)



は、このレコードの **Value** とグローバル平均との差を標準偏差として示します。グローバル ノードによりグローバル値が算出されないと、グローバル値を使用することはできません。現在のすべてのグローバル値は、[ストリームのプロパティ] ダイアログ ボックスの [グローバル] タブにある [グローバル値の消去] ボタンをクリックしてキャンセルすることができます。

注：@ 関数はスクリプトから呼び出せません。詳細は、3 章 p.33 スクリプト内の CLEM 式 を参照してください。

| 関数                  | 結果 | 説明  |
|---------------------|----|---|
| @GLOBAL_MAX(FIELD)  | 数値 | 以前にグローバル ノードで生成されたように、データ セット全体の FIELD の最大値を返します。FIELD は数値型フィールドの名前でなければなりません。対応するグローバル値が設定されていない場合は、エラーが発生します。この関数は、スクリプトから呼び出すことができません。詳細は、3 章 p.33 スクリプト内の CLEM 式 を参照してください。 |
| @GLOBAL_MIN(FIELD)  | 数値 | 以前にグローバル ノードで生成されたように、データ セット全体の FIELD の最小値を返します。FIELD は数値型フィールドの名前でなければなりません。対応するグローバル値が設定されていない場合は、エラーが発生します。   |
| @GLOBAL_SDEV(FIELD) | 数値 | 以前にグローバル ノードで生成されたように、データ セット全体の FIELD の値の標準偏差を返します。FIELD は数値型フィールドの名前でなければなりません。対応するグローバル値が設定されていない場合は、エラーが発生します。  |
| @GLOBAL_MEAN(FIELD) | 数値 | 以前にグローバル ノードで生成されたように、データ セット全体の FIELD の平均値を返します。FIELD は数値型フィールドの名前でなければなりません。対応するグローバル値が設定されていない場合は、エラーが発生します。   |
| @GLOBAL_SUM(FIELD)  | 数値 | 以前にグローバル ノードで生成されたように、データ セット全体の FIELD の値の合計を返します。FIELD は数値型フィールドの名前でなければなりません。対応するグローバル値が設定されていない場合は、エラーが発生します。  |

## 空白値とヌル値処理関数

CLEM を使って、フィールド内の特定の値を「空白」、つまり欠損値と見なすように指定することができます。空白値を処理する関数を次に示します。

注：@ 関数はスクリプトから呼び出せません。詳細は、3 章 p.33 スクリプト内の CLEM 式を参照してください。

| 関数                     | 結果     | 説明  |
|------------------------|--------|---|
| @BLANK(FIELD)          | ブール    | 上流のデータ型ノードまたは入力ノードで設定された空白処理規則（[データ型] タブ）にしたがって、値が空白のレコードに対して真を返します。この関数は、スクリプトから呼び出すことができません。詳細は、3 章 p.33 スクリプト内の CLEM 式を参照してください。                       |
| @LAST_NON_BLANK(FIELD) | CLEM 式 | 上流の入力ノードまたはデータ型ノードで定義されるように、空白でない FIELD の最後の値を返します。それまでに読み込んだレコードの FIELD の値がすべて空白である場合は、\$null\$ を返します。ユーザー欠損値とも呼ばれる空白値は、各フィールドに個別に定義することができることに注意してください。 |
| @NULL(FIELD)           | ブール    | FIELD の値がシステム欠損値 \$null\$ の場合 true を返し、ユーザー定義の空白地など、他のすべての値については false を返します。これらを確認する場合は、@BLANK(FIELD) および @NULL(FIELD) を使用します。                           |
| undef                  | CLEM 式 | 一般的に CLEM で \$null\$ 値を入力するために用いられます。たとえば、置換ノードで空白値にヌルを挿入するために用いられます。   |

空白フィールドが、置換ノードで書き込まれる場合もあります。置換ノードおよびフィールド作成ノード（複数モードの場合）の両方で、特殊 CLEM 関数の @FIELD は、調査対象の現在のフィールドを表します。

## 特殊フィールド

特殊関数は、調査対象の特定のフィールドを表したり、フィールドのリストを入力として生成したりするために用いられます。たとえば、複数のフィールドを一度に作成する場合、@FIELD を使って「このフィールド作成操作を選択したフィールドに対して行う」ことを指示します。式 log(@FIELD) を使用すると、選択した各フィールドに対して、新しいログフィールドが作成されます。

注：@ 関数はスクリプトから呼び出せません。詳細は、3 章 p.33 スクリプト内の CLEM 式を参照してください。

| 関数                          | 結果     | 説明   |
|-----------------------------|--------|--|
| @FIELD                      | CLEM 式 | 式のコンテキスト中に指定されているすべてのフィールドに対して処理を行います。この関数は、スクリプトから呼び出すことができません。詳細は、3 章 p.33 スクリプト内の CLEM 式を参照してください。  |
| @TARGET                     | CLEM 式 | ユーザー定義の分析関数で CLEM 式を使用する場合、@TARGET は対象のフィールドを表すか、または分析される対象と予測のペアに対する「正しい値」を表します。通常この関数は、精度分析ノードで使用されます。   |
| @PREDICTED                  | CLEM 式 | ユーザー定義の分析関数で CLEM 式を使用する場合、@PREDICTED は、分析される対象と予測のペアに対して、予測される値を表します。通常この関数は、精度分析ノードで使用されます。  |
| @PARTITION_FIELD            | CLEM 式 | 現在のデータ区分フィールドの名前を置き換えます。   |
| @TRAINING_PARTITION         | CLEM 式 | 現在の学習用データ区分の値を返します。たとえば、データ選択ノードを使用して学習レコードを選択する場合、次の CLEM 式を使用します。<br><b>@PARTITION_FIELD = @TRAINING_PARTITION</b><br>この方法は、データ内の各データ区分を表すためにどのデータが使用されているか関係なく、常に選択ノードが正しく動作することを保証します。 |
| @TESTING_PARTITION          | CLEM 式 | 現在のテスト用データ区分の値を返します。   |
| @VALIDATION_PARTITION       | CLEM 式 | 現在の検証用データ区分の値、0返します。   |
| @FIELDS_BETWEEN(start, end) | CLEM 式 | データ中のフィールドの普通の順序（つまり、挿入）にもとづく、指定された開始フィールドと最終フィールドの間（開始、最終フィールドを含む）のフィールド名のリストを返します。詳細は、7 章 複数フィールドの要約 in IBM SPSS Modeler 15 ユーザー ガイドを参照してください。   |

| 関数                        | 結果     | 説明  |
|---------------------------|--------|---|
| @FIELDS_MATCHING(pattern) | CLEM 式 | 指定したパターンに一致するフィールド名のリストを返します。クエスチョン マーク (?) をパターンに含めて正確に 1 つの文字に一致させることができます。アスタリスク (*) は 0 かそれ以上の文字数に一致します。リテラル クエスチョン マークまたはアスタリスクを (むしろ、ワイルドカードとして使用しないで) 一致させるために、バックスラッシュをエスケープ文字として使用することができます。 <a href="#">詳細は、7 章 複数フィールドの要約 in IBM SPSS Modeler 15 ユーザー ガイド を参照してください。</a> |
| @MULTI_RESPONSE_SET       | CLEM 式 | 名前の付いた複数レスポンス セットでフィールドのリストを返します。 <a href="#">詳細は、7 章 複数レスポンス データの処理 in IBM SPSS Modeler 15 ユーザー ガイド を参照してください。</a>   |

# パート II: プロパティ参照

# プロパティ参照

## プロパティ参照の概要

ノード、ストリーム、スーパーノード、プロジェクトに対して、数多くのさまざまなプロパティを指定できます。名前、注釈、およびツールヒントなど、すべてのノードに共通のプロパティもありますが、その一方で、ノードのタイプに固有なプロパティもあります。キャッシングやスーパーノードの動作などの高レベルなストリーム操作を参照するプロパティもあります。プロパティは、標準のユーザー インターフェイスからアクセスでき（ノードのオプションを編集するダイアログ ボックスをオープンする場合など）、また、多くの標準とは異なる方法でも使用できます。

- プロパティは、このセクションで説明されているように、スクリプトからアクセスできます。詳細は [プロパティのシンタックス](#) 下記を参照してください。
- ノードのプロパティは、スーパーノード パラメータ中で使用することができます。詳細は、[9 章 スーパーノード パラメータを使ったノードのプロパティへのアクセス in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。
- ノードのプロパティは、IBM® SPSS® Modeler の起動時にコマンド ライン オプションの一部として使用することもできます (-P フラグを使用)。

SPSS Modeler のスクリプトでは、ノードおよびストリームのプロパティは、よく **スロット パラメータ** と呼ばれます。このガイドでは、スロット パラメータをノードまたはストリームのプロパティと記載しています。

スクリプト言語の詳細は、3 章を参照してください。

## プロパティのシンタックス

プロパティは、次のシンタックス構造で使用する必要があります。

NAME:TYPE.PROPERTY

NAME はノードの名前で、TYPE はノードの種類です (multiplotnode または derivenode など)。NAME と TYPE は、どちらか片方だけなら省略できます。PROPERTY は式の参照先であるノード パラメータまたはストリーム パラメータの名前です。たとえば、フィールド Age のデータをフィルタリングして、下流に流さないシンタックスの例を次に示します。

```
set mynode:filternode.include.Age = false
```

パラメータのいずれかのカスタム値 (NAME、TYPE、または PROPERTY) を使用するには、まず `set derive.new_name = mynewfield` のように文で値を設定します。値を設定したら、その値 (先ほどの例では `mynewfield`) をパラメータとして使用することができます。値を使用する場合は、値の前に `^` 記号を付けてください。たとえば、上記の名前を持つフィールド作成ノードにデータ型を設定するには、次のシンタックスを使用します。

```
set ^mynewfield.result_type = "Conditional"
```

IBM® SPSS® Modeler で使用されるすべてのノードは、シンタックス `NAME:TYPE.PROPERTY` の `TYPE` パラメータで指定できます。

## 構造化プロパティ

スクリプト解析時の明確性を向上するために構造化プロパティを使用するには、次の 2 種類の方法があります。

- データ型、フィルタ、またはバランス ノードなどの、複雑なノードのプロパティ名を構造化する。
- 複数のプロパティを同時に指定するフォーマットを提供する。

### 複雑なインターフェイスの構造化

テーブルや他の複雑なインターフェイスがあるノード、たとえば、データ型、フィルタ、およびバランス ノードなどを対象とするスクリプトは、正しく解析されるために一定の構造を遵守する必要があります。これらの構造化プロパティには、1 つの識別子名と比べてより複雑な名前が必要です。たとえばフィルタ ノードでは、使用可能な各フィールド (上流側) が On (オン) または Off (オフ) に切り替えられます。この情報を参照するため、フィルタ ノードではフィールドごとに 1 つの情報項目 (各フィールドが真か偽か) が保存されます。またこれら複数の項目には、**field** と呼ばれる 1 つのプロパティを使用してアクセスし、これらを更新します。このプロパティには、真 (**true**) または偽 (**false**) の値があるか、または与えられることがあります。 `mynode` というフィルタ ノード (上流側) に、`Age` というフィールドがある場合を考えてみましょう。これをオフにするには、プロパティ `mynode.include.Age` に値 **false** を次のように設定します。

```
set mynode.include.Age = false
```

### 複数のプロパティの設定構造

多数のノードに対して、複数のノードおよびストリームのプロパティを同時に割り当てることができます。これは、**multiset** コマンドまたは**セット ブロック**と呼ばれています。詳細は、4 章 p.39 `set` コマンドを参照してください。

場合によっては、構造化プロパティがきわめて複雑なこともあります。引数を明確に記述するために、行継続文字として円記号 (¥) を使用できます。一例を以下に挙げます。

```
mynode:sortnode.keys = [{ 'K' Descending } \
                        { 'Age' Ascending } \
                        { 'Na' Descending } ]
```

構造化プロパティのもう 1 つの利点は、ノードが安定していなくてもそのノード上に複数のプロパティが設定できることです。デフォルトでは、multiset はブロック内のすべてのプロパティを設定してから、個別のプロパティ設定に基づいてアクションを実行します。たとえば固定長ノードを定義するとき、フィールド プロパティを 2 ステップに分けて設定するとエラーが生じます。これは、両方の設定が有効になるまでノードが一貫しないためです。プロパティを multiset として定義すれば、データ モデルを更新する前に両方のプロパティが設定でき、エラーが回避されます。

## 省略形

ノードのプロパティのシンタックスでは、標準省略形が使用されています。省略形を覚えておけば、スクリプトの作成に役立ちます。

| 略語     | 意味               |
|--------|------------------|
| abs    | 絶対値              |
| len    | 長さ               |
| min    | 最小               |
| max    | 最大               |
| correl | Correlation      |
| covar  | Covariance       |
| num    | 数字または数値          |
| pct    | パーセントまたは割合       |
| transp | 透過度              |
| xval   | 交差検証             |
| var    | 分散または変数 (入力ノードで) |

## ノードおよびストリームのプロパティの例

ノードおよびストリームのプロパティは、IBM® SPSS® Modeler のさまざまな場面で使用されます。一般的にこれらのプロパティは、複数のストリームや操作を自動化するために用いられる **スタンドアロン スクリプト**、または単一のストリーム内のプロセスの自動化に用いられる **ストリーム スクリプト** など、スクリプトの一部として使われます。スーパーノード内で、ノードのプロパティを使ってノード パラメータを指定することもできま



す。もっとも基本的なレベルで、SPSS Modeler の起動時にコマンド ライン オプションとしてプロパティを指定することもできます。コマンド ラインの起動時に、**-p** 引数を指定すれば、ストリーム プロパティを使ってストリームの設定を変更することができます。

|                                    |  |
|------------------------------------|--|
| <code>s.max_size</code>            | ノード <code>s</code> のプロパティ <code>max_size</code> を表します。                               |
| <code>s:samplename.max_size</code> | ノード <code>s</code> のプロパティ <code>max_size</code> を表します。このノードは、サンプリングノードでなければなりません。    |
| <code>:samplename.max_size</code>  | 現在のストリーム中のサンプリングノードの、プロパティ <code>max_size</code> を表します (サンプリングノードは 1 つだけでなければなりません)。 |
| <code>s:sample.max_size</code>     | ノード <code>s</code> のプロパティ <code>max_size</code> を表します。このノードは、サンプリングノードでなければなりません。    |
| <code>t.direction.Age</code>       | データ型ノード <code>t</code> の Age フィールドの役割を表します。  |
| <code>:.max_size</code>            | *** 無効 *** ノード名またはノードの種類を指定する必要があります。  |

`s:sample.max_size` の例は、ノードの種類を完全に記述する必要がないことを示しています。

`t.direction.Age` の例は、1 つのノードの属性が個別の値を持つ単純な個々のスロットよりも複雑な場合に、一部のスロット名を構造化できることを示しています。このようなスロットは、**構造化**または**複雑なプロパティ**と呼ばれます。

## ノードのプロパティの概要

ノードの種類ごとに、独自の有効なプロパティのセットが用意されています。また、各プロパティにはデータ型があります。一般的なデータ型の数値、フラグ、または文字列の場合、プロパティの設定は強制的に正しいデータ型に設定されます。強制的に設定できない場合はエラーが発生します。それに対し、プロパティ参照が、**Discard**、**PairAndDiscard**、および **IncludeAsText** のような有効な値の範囲を指定していることもあります。この場合、範囲外の値が使われた場合にエラーになります。フラグ型プロパティは、**true** および **false** の値を使用して読み込まれるか、設定される必要があります (**Off**、**OFF**、**off**、**No**、**NO**、**no**、**n**、**N**、**f**、**F**、**false**、**False**、**FALSE**、または **0** など) 値の設定時に認識されますが、プロパティ値の読み込み時にエラーが発生する場合があります。その他の値はすべて真と見なされます。**true** と **false** を使用すると、こうした混乱が避けられます)。このガイドにある参照テーブルでは、構造化プロパティはそのまま「プロパティの説明」欄に、使用フォーマットとともに記載されています。

## 共通のノード プロパティ

数多くのプロパティが、IBM® SPSS® Modeler 中のすべてのノード（スーパーノードも含む）で共通に使われています。

| プロパティ名          | データ型   | プロパティの説明  |
|-----------------|--|---|
| use_custom_name | フラグ型   |   |
| name            | string   | ストリーム領域上のノード名を対象とする読み込み専用プロパティです（自動またはユーザー設定）。  |
| custom_name     | string   | ノードのカスタム（ユーザー設定）名を指定します。  |
| tooltip         | string   |   |
| annotation      | string   |   |
| keywords        | string   | オブジェクトに関連付けられているキーワードのリストを指定する構造化スロットです（例：["Keyword1" "Keyword2"]）。   |
| cache_enabled   | フラグ型   |   |
| node_type       | source_supernode<br>process_supernode<br>terminal_supernode<br>スクリプトに指定するすべてのノード名。 | ノードをタイプごとに参照するために使用される読み込み専用プロパティ。たとえば、ノードを <code>real_income</code> のような名前だけで参照する代わりに、 <code>userinputnode</code> または <code>filternode</code> のようなタイプで指定することもできます。 |

スーパーノード固有のプロパティは、他のノードと同様に、個別に説明します。詳細は、[22 章 p.347 スーパーノードのプロパティ](#) を参照してください。

# ストリームのプロパティ

スクリプトにより、さまざまなストリームのプロパティを制御することができます。ストリームのプロパティを参照するには、特殊なストリーム変数を使用する必要があります。この変数は、ストリームの先頭に `^` を付けて表されます。

```
set ^stream.execute_method = Script
```

## 例

`nodes` プロパティは、現在のストリーム中のノードを参照するために使用されます。次のストリーム スクリプトに、その例を示します。

```
var listofnodes
var thenode
set listofnodes = ^stream.nodes

set ^stream.annotation = ^stream.annotation << "\n\nThis stream is called \"\" << ^stream.name >
< \"\" and contains/ the following nodes\n"

for thenode in listofnodes
set ^stream.annotation = ^stream.annotation << "\n" << ^thenode.node_type
endfor
```

上記の例では、ノードのプロパティを使って、ストリーム中のすべてのノードの一覧を作成し、そのリストをストリームの注釈に書き込みます。この注釈は、次のようになります。

This stream is called "druglearn" and contains the following nodes

```
derivenode
neuralnetworknode
variablefilenode
typenode
c50node
filternode
```

ストリームのプロパティを次の表に示します。

| プロパティ名                    | データ型  | プロパティの説明 |
|---------------------------|---|----------|
| execute_method            | Normal Script   |          |
| date_format               | "DDMMYY"<br>"MMDDYY"<br>"YYMMDD"<br>"YYYYMMDD"<br>"YYYYDDD"<br>DAY<br>MONTH<br>"DD-MM-YY"<br>"DD-MM-YYYY"<br>"MM-DD-YY"<br>"MM-DD-YYYY"<br>"DD-MON-YY"<br>"DD-MON-YYYY"<br>"YYYY-MM-DD"<br>"DD.MM.YY"<br>"DD.MM.YYYY"<br>"MM.DD.YY"<br>"MM.DD.YYYY"<br>"DD.MON.YY"<br>"DD.MON.YYYY"<br>"DD/MM/YY"<br>"DD/MM/YYYY"<br>"MM/DD/YY"<br>"MM/DD/YYYY"<br>"DD/MON/YY"<br>"DD/MON/YYYY"<br>MON YYYY<br>q Q YYYY<br>ww WK YYYY |          |
| date_baseline             | number  |          |
| date_2digit_baseline      | number  |          |
| time_format               | "HHMMSS"<br>"HHMM"<br>"MMSS"<br>"HH:MM:SS"<br>"HH:MM"<br>"MM:SS"<br>"(H)H:(M)M:(S)S"<br>"(H)H:(M)M"<br>"(M)M:(S)S"<br>"HH.MM.SS"<br>"HH.MM"<br>"MM.SS"<br>"(H)H.(M)M.(S)S"<br>"(H)H.(M)M"<br>"(M)M.(S)S"  |          |
| time_rollover             | フラグ型  |          |
| import_datetime_as_string | フラグ型  |          |

| プロパティ名                                     | データ型                       | プロパティの説明  |
|--|----------------------------|---|
| decimal_places                             | number                     |   |
| decimal_symbol                             | Default<br>Period<br>Comma |   |
| angles_in_radians                          | フラグ型                       |   |
| use_max_set_size                           | フラグ型                       |   |
| max_set_size                               | number                     |   |
| ruleset_evaluation                         | Voting<br>FirstHit         |   |
| refresh_source_nodes                       | フラグ型                       | ストリーム実行時に、入力ノードを自動的にリフレッシュするために使用します。   |
| script                                     | string                     |   |
| annotation                                 | string                     | 例:<br><code>set ^stream.annotation = "something interesting"</code>   |
| name                                       | string                     | 例:<br><code>set x = ^stream.name</code><br>注：このプロパティは読み取り専用です。<br>ストリーム名を変更する場合は、別名で保存する必要があります。            |
| parameters                                 |                            | スタンドアロン スクリプト内からストリーム パラメータを更新する場合に、このプロパティを使用します。<br>例:<br><code>set ^stream.parameters.height = 23</code> |
| nodes                                      |                            | 詳細は以下を参照してください。   |
| encoding                                   | SystemDefault<br>"UTF-8"   |   |
| stream_rewriting                           | boolean                    |   |
| stream_rewriting_maximise_sql              | boolean                    |   |
| stream_rewriting_optimise_clem_execution   | boolean                    |   |
| stream_rewriting_optimise_syntax_execution | boolean                    |   |
| enable_parallelism                         | boolean                    |   |
| sql_generation                             | boolean                    |   |
| database_caching                           | boolean                    |   |
| sql_logging                                | boolean                    |   |
| sql_generation_logging                     | boolean                    |   |
| sql_log_native                             | boolean                    |   |
| sql_log_prettyprint                        | boolean                    |   |
| record_count_suppress_input                | boolean                    |   |
| record_count_feedback_interval             | integer                    |   |

# プロジェクトのプロパティ

プロジェクトのスクリプトに、多くのプロパティが利用できます。

## 例

```
load project "C:/clemdata/DrugData.cpj"
set ^project.summary="Initial modeling work on the latest drug data."
set ^project.ordering=NameAddedType
execute_project
```

| プロパティ名            | データ型   | プロパティの説明  |
|-------------------|--|---|
| summary           | string   | プロジェクトの要約、通常は注釈の簡略版。  |
| title             | string   | レポートのタイトル。  |
| author            | string   | レポートの作成者。   |
| structure         | Phase<br>Class   | プロジェクトがどのように編成されるか、つまりデータマイニングの段階 (Phase) によってか、オブジェクトタイプ (Class) によってかを決めます。 |
| include_mode      | IncludedItems<br>ExcludedItems<br>AllItems                       | プロジェクト レポートに含める項目を決めます。   |
| select_mode       | AllItems<br>RecentItems<br>OldItems                              | プロジェクト レポートに含める項目を作成時間順に決めます。   |
| recent_item_limit | integer  | <b>select_mode</b> が <b>RecentItems</b> の場合に使用。                               |
| old_item_limit    | integer  | <b>select_mode</b> が <b>OldItems</b> の場合に使用。                                  |
| ordering          | TypeNameAdded<br>TypeAddedName<br>NameAddedType<br>AddedNameType | プロジェクト レポートに表示される項目の順序を決めます。  |

# 入カノードのプロパティ

## 入カノードの共通プロパティ

すべての入カノードに共通するプロパティを次に一覧にします。その後、特定のノードに関する情報が続きます。

### 例

```
create variablefilenode
set :variablefilenode.full_filename = "$CLEO_DEMOS/DRUG4n"
set :variablefilenode.use_custom_values.Age = True
set :variablefilenode.direction.Age = Input
set :variablefilenode.type.Age = Range
#storage is read only
set :variablefilenode.check.Age = None
set :variablefilenode.values.Age = [1 100]
```

| プロパティ名    | データ型   | プロパティの説明  |
|-----------|--|---|
| direction | Input<br>Target<br>Both<br>None<br>Partition<br>Split<br>Frequency<br>RecordID | フィールドの役割のキープロパティ。<br>使用フォーマット：<br>NODE.direction.FIELDNAME<br>注：値 In および Out は廃止されています。今後のリリースではサポートが中断される場合があります。  |
| type      | Range<br>Flag<br>Set<br>Typeless<br>Discrete<br>Ordered Set<br>Default         | フィールドのデータ型。このプロパティを Default に設定すると、values プロパティに関するすべての値は消去され、value_mode を Specify に設定すると、それが Read にリセットされます。value_mode が Pass または Read がすでに設定されている場合、type の設定によって影響を受けることはありません。<br>使用フォーマット：<br>NODE.type.FIELDNAME |
| storage   | Unknown<br>String<br>Integer<br>Real<br>Time<br>Date<br>Timestamp              | フィールドのストレージタイプ用読み込み専用キープロパティ。<br>使用フォーマット：<br>NODE.storage.FIELDNAME  |

| プロパティ名             | データ型  | プロパティの説明   |
|--------------------|---|--|
| check              | None<br>Nullify<br>Coerce<br>Discard<br>Warn<br>Abort | フィールド タイプと範囲の検査用のキー プロパティ。<br>使用フォーマット：<br>NODE.check.FIELDNAME  |
| values             | [値 値]   | 連続型（範囲）フィールドの場合、最初の値が最小値で最後の値が最大値になります。名義型（セット型）フィールドの場合、すべての値を指定します。フラグ型の場合、最初の値が false（偽）を、最後の値が true（真）を表します。このプロパティを設定すると、 <b>value_mode</b> プロパティの値が自動的に Specify に設定されます。<br>使用フォーマット：<br>NODE.values.FIELDNAME |
| value_mode         | Read<br>Pass<br>Read+<br>Current<br>Specify           | 次のデータの受け渡し時にフィールドに値を設定する方法を決定します。<br>使用フォーマット：<br>NODE.value_mode.FIELDNAME<br>このプロパティに Specify を直接には設定できないことに注意してください。特定の値を使用するには、 <b>values</b> プロパティを設定します。   |
| default_value_mode | Read<br>Pass  | すべてのフィールドに値を設定するためのデフォルトの方法を指定します。<br>使用フォーマット：<br>NODE.default_value_mode<br>例：<br>set mynode.default_value_mode = Pass<br>この設定による特定のフィールドの設定は、 <b>value_mode</b> プロパティを使用するとオーバーライドされることがあります。                     |
| extend_values      | フラグ型  | <b>value_mode</b> が Read に設定された場合に適用されます。新しく読み込んだ値を、フィールドの既存の値に追加する場合は、T を設定します。新しく読み込んだ値を優先して、既存の値を破棄する場合は、F を設定します。<br>使用フォーマット：<br>NODE.extend_values.FIELDNAME   |
| value_labels       | string  | 値ラベルの指定に使用します。例：<br>set :varfilenode.value_labels.Age = [{3 three}{5 five}]<br>数値を先に指定します。   |
| enable_missing     | フラグ型  | T を設定した場合、フィールドの欠損値の追跡が有効になります。<br>使用フォーマット：<br>NODE.enable_missing.FIELDNAME  |
| missing_values     | [値 値 ...]   | 欠損データを示すデータ値を指定します。<br>使用フォーマット：<br>NODE.missing_values.FIELDNAME  |



## 入カノードのプロパティ

| プロパティ名             | データ型   | プロパティの説明  |
|--------------------|--------|---|
| range_missing      | フラグ型   | プロパティが T に設定されている場合、フィールドに欠損値（空白）の範囲が定義されているかどうかを指定します。<br>使用フォーマット：<br>NODE.range_missing.FIELDNAME                          |
| missing_lower      | string | range_missing が真（true）の場合、欠損値範囲の下限値を指定します。<br>使用フォーマット：<br>NODE.missing_lower.FIELDNAME                                       |
| missing_upper      | string | range_missing が真（true）の場合、欠損値範囲の上限値を指定します。<br>使用フォーマット：<br>NODE.missing_upper.FIELDNAME                                       |
| null_missing       | フラグ型   | このプロパティが T に設定されていると、ヌル（ソフトウェアでは \$null\$ として表示される未定義値）は欠損値と見なされます。<br>使用フォーマット：<br>NODE.null_missing.FIELDNAME               |
| whitespace_missing | フラグ型   | このプロパティが T に設定されていると、空白値（スペース、タブ、および改行）だけを含まれる値は欠損値とみなされます。<br>使用フォーマット：<br>NODE.whitespace_missing.FIELDNAME                 |
| description        | string | フィールドのラベルまたは説明の指定に使用します。  |
| default_include    | フラグ型   | デフォルトの処理としてフィールドを通過させるかフィルタをかけるかの指定をするキープロパティ。<br>NODE.default_include<br>例：<br>set mynode:filternode.default_include = false |
| include            | フラグ型   | 各フィールドを適用するかフィルタをかけるかを決定するキープロパティ：<br>NODE.include.FIELDNAME.<br>例：<br>set mynode:filternode.include.Age = true               |
| new_name           | string | 例：<br>set mynode:filternode.new_name.'Age' = "years"  |

## cognosimportnode のプロパティ



IBM Cognos BI 入カノードは、Cognos BI データベースからデータをインポートします。詳細は、2 章 Cognos データのインポート in IBM SPSS Modeler 15 入カノード、プロセス ノード、出力ノード を参照してください。

## 例

```
create cognosimportnode
set :cognosimportnode.cognos_connection = {'http://mycogsvr1:9300/p2pd/servlet/dispatch', true, "", "", ""}
set :cognosimportnode.cognos_package_name = '/Public Folders/GOSALES'
set :cognosimportnode.cognos_items = {'[GreatOutdoors].[BRANCH].[BRANCH_CODE]',
'[GreatOutdoors].[BRANCH].[COUNTRY_CODE]'}
```

| cognosimportnode プロパティ | データ型                                | プロパティの説明   |
|------------------------|-------------------------------------|--|
| mode                   | Data Report                         | Cognos BI データ (デフォルト) またはレポートをインポートするかどうかを指定します。   |
| cognos_connection      | { "フィールド", "フィールド", ... , "フィールド" } | Cognos サーバーの接続の詳細を含むリストのプロパティ。形式は次のとおりです。<br>{ "Cognos_server_URL", login_mode, "namespace", "username", "password" }<br>ここでの意味は次の通りです。<br>Cognos_server_URL は、ソースを含む Cognos サーバーの URL です。<br>login_mode は匿名ログインが使用されるかどうかを示し、true または false となります。true の場合、次のフィールドは "" に設定されます。<br>namespace はサーバーへのログインに使用するセキュリティ認証プロバイダを指定します。<br>username および password は Cognos サーバーにログインする際に使用するユーザー名とパスワードです。   |
| cognos_package_name    | string                              | データ オブジェクトをインポートしている Cognos データ ソース (通常はデータベース) のパスおよび名前。次に例を示します。<br><b>/Public Folders/GOSALES</b><br>注 : フォワード スラッシュのみが有効です。  |
| cognos_items           | { "フィールド", "フィールド", ... , "フィールド" } | インポートする 1 つまたは複数のデータ オブジェクトの名前。フィールドの形式は [namespace].[query_subject].[query_item] となります。<br>例:<br>set :cognosimport.cognos_items = {'[Inventory (query)].[Inventory].[Opening inventory]', "[Inventory (query)].[Inventory].[Quantity shipped]", "[Inventory (query)].[Inventory].[Additions]", "[Inventory (query)].[Inventory].[Unit cost]", "[Inventory (query)].[Inventory].[Closing inventory]", "[Inventory (query)].[Inventory].[Average unit cost]'} |

| cognosimportnode プロパティ   | データ型  | プロパティの説明  |
|--------------------------|-------|---|
| cognos_filters           | フィールド | データをインポートする前に適用するフィルタの名前。<br>例:<br><code>set :cognosimport.cognos_filters = [{"Inventory"}, {"Filter"}, {"MyFilter"}]</code>  |
| cognos_data_parameters   | リスト   | データのプロンプト パラメータの値。名前と値のペアはかっこで囲み、複数のペアはカンマで区切り、文字列全体は角かっこで囲みます。<br>書式:<br>[{" param1" ,<br>"value "}, ..., {" paramN" ,<br>"value "}]<br>例:<br><code>set :cognosimport.cognos_data_parameters = [{"SexValue","F"}, {"a","1"}, {"b","1"}]</code> |
| cognos_report_location   | フィールド | レポートをインポートするフォルダまたはパッケージの Cognos パス。次に例を示します。<br><code>/Public Folders/GOSALES</code><br>注 : フォワード スラッシュのみが有効です。  |
| cognos_report_name       | フィールド | インポートするレポートのレポートの位置内にあるパスと名前。次に例を示します。<br><code>set :cognosimport.cognos_report_name = /Jimmy/Package/Drug4nPackage/3columns</code>   |
| cognos_report_parameters | リスト   | レポート パラメータの値。名前と値のペアはかっこで囲み、複数のペアはカンマで区切り、文字列全体は角かっこで囲みます。<br>書式:<br>[{" param1" ,<br>"value "}, ..., {" paramN" ,<br>"value "}]<br>例:<br><code>set :cognosimport.cognos_report_parameters = [{"SexValue","F"}, {"a","1"}, {"b","1"}]</code>    |

## datasourcenode のプロパティ



データベース ノードは、Microsoft SQL Server、DB2、Oracle など ODBC（開放型データベース接続）を使用するさまざまなパッケージからデータをインポートするのに使用できます。詳細は、[2 章 データベース入力ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

## 例

```

create databasenode
set :databasenode.mode = Table
set :databasenode.query = "SELECT * FROM drug4n"
set :databasenode.datasource = "Drug4n_db"
set :databasenode.username = "spss"
set :databasenode.password = "spss"
var test_e
set test_e = :databasenode.epassword
set :databasenode.tablename = ".Drug4n"

```

| databasenodeプロパティ | データ型                          | プロパティの説明   |
|-------------------|-------------------------------|--|
| mode              | Table<br>Query                | ダイアログ ボックスのコントロールを使用してデータベースに接続するには、Table を指定します。SQL を使用して選択されたデータベースにクエリーを行うには、Query を指定します。                                    |
| datasource        | string                        | データベース名（下記の注意を参照）。   |
| username          | string                        | データベース接続の詳細（下記の注意を参照）。   |
| password          | string                        |  |
| epassword         | string                        | スクリプト内でパスワードをハードコード化する代わりに、エンコードされたパスワードを指定します。<br><a href="#">詳細は、5 章 p.69 暗号化パスワードの生成 を参照してください。</a> このプロパティは、実行時に読み取り専用になります。 |
| tablename         | string                        | アクセスするテーブルの名前。   |
| strip_spaces      | None<br>Left<br>Right<br>Both | 文字列の前後のスペースを破棄するためのオプションです。  |
| use_quotes        | AsNeeded<br>Always<br>Never   | クエリーをデータベースに送信するときにテーブル名と列名を引用符で囲むかどうかを指定します（たとえば、テーブル名と列名にスペースや句読点が含まれているような場合）。  |
| query             | string                        | 送信するクエリーを表す SQL コードを指定します。   |

注：データベース名（datasource プロパティ）ニスペースがある場合、datasource、username および password の代わりに、次の形式で単一のデータソース プロパティを使用します。

| datenodeプロパティ | データ型   | プロパティの説明  |
|---------------|--------|---|
| datasource    | string | 書式：<br>{database_name,username,password[,true   false]}<br>暗号化パスワードと使用しないパラメータです。true に設定すると、パスワードが使用前に復号化されます。 |

### 例

```
create datanode
set :datanode.mode = Table
set :datanode.query = "SELECT * FROM drug4n"
set :datanode.datasource = {"ORA 10gR2", user1, mypsw, true}
var test_e
set test_e = :datanode.epassword
set :datanode.tablename = ".Drug4n"
```

データ ソースを変更する場合、この形式を使用します。ただし、ユーザー名またはパスワードを変更する場合、username プロパティまたは password プロパティを使用できます。

## datacollectionimportnode のプロパティ



IBM® SPSS® Data Collection データ インポート ノードで、IBM Corp. 市場調査製品によって使用される Data Collection Data Model に基づいた調査データをインポートします。このノードを使用するには、Data Collection Data Library がインストールされている必要があります。詳細は、2 章 [Data Collection ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

### 例

```
create datacollectionimportnode
set :datacollectionimportnode.metadata_name="mrQvDsc"
set :datacollectionimportnode.metadata_file="C:/Program Files/IBM/SPSS/DataCollection/DDL/Data/Quanvert/Museum/museum.pkd"
set :datacollectionimportnode.casedata_name="mrQvDsc"
set :datacollectionimportnode.casedata_source_type=File
set :datacollectionimportnode.casedata_file="C:/Program Files/IBM/SPSS/DataCollection/DDL/Data/Quanvert/Museum/museum.pkd"
set :datacollectionimportnode.import_system_variables = Common
```

set :datacollectionimportnode.import\_multi\_response = MultipleFlags

| datacollectionimportnode プロパティ | データ型                                    | プロパティの説明   |
|--------------------------------|---|--|
| metadata_name                  | string                                  | MDSC の名前。特殊な値の <b>DimensionsMDD</b> は、標準的な Data Collection メタデータ ドキュメントが使用される必要のあることを示します。ほかに、次の値を指定できます。<br><b>mrADODsc</b><br><b>mrI2dDsc</b><br><b>mrLogDsc</b><br><b>mrQdiDrsDsc</b><br><b>mrQvDsc</b><br><b>mrSampleReportingMDSC</b><br><b>mrSavDsc</b><br><b>mrSCDsc</b><br><b>mrScriptMDSC</b><br>特殊な値の <b>none</b> は、MDSC がないことを示します。 |
| metadata_file                  | string                                  | メタデータが格納されるファイルの名前。  |
| casedata_name                  | string                                  | CDSC の名前。ほかに、次の値を指定できます。<br><b>mrADODsc</b><br><b>mrI2dDsc</b><br><b>mrLogDsc</b><br><b>mrPunchDSC</b><br><b>mrQdiDrsDsc</b><br><b>mrQvDsc</b><br><b>mrRdbDsc2</b><br><b>mrSavDsc</b><br><b>mrScDSC</b><br><b>mrXmlDsc</b><br>特殊な値の <b>none</b> は、CDSC がないことを示します。   |
| casedata_source_type           | Unknown<br>File<br>Folder<br>UDL<br>DSN | CDSC のソース タイプを示します。  |
| casedata_file                  | string                                  | <b>casedata_source_type</b> が <b>File</b> のときに、ケース データが含まれるファイルを指定します。   |
| casedata_folder                | string                                  | <b>casedata_source_type</b> が <b>Folder</b> のときに、ケース データが含まれるフォルダを指定します。   |
| casedata_udl_string            | string                                  | <b>casedata_source_type</b> が <b>UDL</b> のときに、ケース データが含まれるデータ ソースのための OLD-DB 接続文字列を指定します。  |

## 入力ノードのプロパティ

| datacollectionimportnodeプロパティ | データ型                     | プロパティの説明   |
|-------------------------------|--------------------------|--|
| casedata_dsn_string           | string                   | <b>casedata_source_type</b> が DSN のときに、データ ソースのための ODBC 接続文字列を指定します。                                   |
| casedata_project              | string                   | Data Collection データベースからケース データを読み込むときに、プロジェクトの名前を入力できます。その他のケース データのデータ型については、この設定を空白のままにしておく必要があります。 |
| version_import_mode           | All<br>Latest<br>Specify | 各バージョンの取り扱い方法を定義します。   |
| specific_version              | string                   | <b>version_import_mode</b> が Specify のときに、インポートされるケース データのバージョンを定義します。                                 |
| use_language                  | string                   | 特定言語のラベルが使用される必要があるかどうかを定義します。   |
| language                      | string                   | <b>use_language</b> が真 (true) の場合、入力に使用する言語コードを定義します。言語コードは、ケース データ内で利用できる中の 1 つにする必要があります。            |
| use_context                   | string                   | 特定のコンテキストが入力される必要があるかどうかを定義します。コンテキストは、応答に関連する説明を多様化させるために使用されます。                                      |
| context                       | string                   | <b>use_context</b> が真 (true) の場合、入力するコンテキストを定義します。コンテキストは、ケース データ内で利用できる中の 1 つにする必要があります。              |
| use_label_type                | string                   | 特定のラベル タイプが入力される必要があるかどうかを定義します。   |
| label_type                    | string                   | <b>use_label_type</b> が真 (true) の場合、入力するラベル タイプを定義します。ラベル タイプは、ケース データ内で利用できる中の 1 つにする必要があります。         |
| user_id                       | string                   | 明示的なログインが必要なデータベースの場合、データ ソースにアクセスするためのユーザー ID とパスワードを提供できます。  |
| password                      | string                   |  |
| import_system_variables       | Common<br>None<br>All    | インポートされるシステム変数を指定します。  |
| import_codes_variables        | フラグ型                     |  |

| datacollectionimportnode プロパティ | データ型                    | プロパティの説明 |
|--------------------------------|-------------------------|----------|
| import_sourcefile_variables    | フラグ型                    |          |
| import_multi_response          | MultipleFlags<br>Single |          |

## excelimportnod のプロパティ



Excel インポート ノードで、Microsoft Excel の各バージョンからデータをインポートします。ODBC データ ソースは不要です。詳細は、2 章 Excel 入力ノード in IBM SPSS Modeler 15 入力ノード、プロセスノード、出力ノード を参照してください。

### 例

```
#To use a named range:
create excelimportnode
set :excelimportnode.excel_file_type = Excel2007
set :excelimportnode.full_filename = "C:/drug.xls"
set :excelimportnode.use_named_range = True
set :excelimportnode.named_range = "DRUG"
set :excelimportnode.read_field_names = True
```

```
#To use an explicit range:
create excelimportnode
set :excelimportnode.excel_file_type = Excel2007
set :excelimportnode.full_filename = "C:/drug.xls"
set :excelimportnode.worksheet_mode = Name
set :excelimportnode.worksheet_name = "Drug"
set :excelimportnode.explicit_range_start = A1
set :excelimportnode.explicit_range_end = F300
```

| excelimportnode プロパティ | データ型                   | プロパティの説明  |
|-----------------------|------------------------|---|
| excel_file_type       | Excel2003<br>Excel2007 |   |
| full_filename         | string                 | パスを含む、完全なファイル名。   |
| use_named_range       | ブール                    | 名前付けられた範囲を使用するかどうかを指定します。真の場合、読み込む範囲を指定するのに <b>named_range</b> プロパティが使用され、その他のワークシートとデータ範囲の設定は無視されます。 |
| named_range           | string                 |   |
| worksheet_mode        | Index<br>Name          | ワークシートがインデックスで定義されているのか (Index)、または名前で定義されているのか (Name) を指定します。  |



## 入力ノードのプロパティ

| excelimportnode プロパティ | データ型                           | プロパティの説明  |
|-----------------------|--------------------------------|---|
| worksheet_index       | integer                        | 読み込むべきワークシートのインデックス。最初のワークシートは 0、2 番目は 1、というようにインデックスが指します。         |
| worksheet_name        | string                         | 読み込むべきワークシートの名前。  |
| data_range_mode       | FirstNonBlank<br>ExplicitRange | 範囲の決定方法を指定します。  |
| blank_rows            | StopReading<br>ReturnBlankRows | <code>data_range_mode</code> が FirstNonBlank のときに、空白行の処理方法を指定します。   |
| explicit_range_start  | string                         | <code>data_range_mode</code> が ExplicitRange のときに、読み込む範囲の開始点を指定します。 |
| explicit_range_end    | string                         |   |
| read_field_names      | ブール                            | 指定された範囲の最初の行がフィールド（列）名として使用されるかどうかを指定します。                           |

## evimportnode のプロパティ



Enterprise View ノードは、IBM SPSS Collaboration and Deployment Services Repository への接続を作成し、Enterprise View のデータをストリームに読み込み、他のユーザーがレポジトリからアクセスできるシナリオにモデルをパッケージ化できます。詳細は、[2 章 Enterprise View ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

## 例

```
create evimportnode
set :evimportnode.connection = ['Training data','/Application views/Marketing','LATEST','Analytic',
'/Data Providers/Marketing']
set :evimportnode.tablename = "cust1"
```

| evimportnode プロパティ | データ型   | プロパティの説明   |
|--------------------|--------|--|
| connection         | リスト    | 構造化プロパティ - エンタープライズ ビューの接続を作成するパラメータのリスト。<br>使用フォーマット：<br><code>evimportnode.connection = [description,app_view_path,app_view_version_label,environment,DPD_path]</code> |
| tablename          | string | アプリケーション ビューのテーブル名。  |

## fixedfilenode のプロパティ



固定長ノードで、固定長フィールド テキスト ファイルからデータをインポートします。ここで、ファイルのフィールドは区切られていませんが、同じ位置から始まって長さは固定されています。コンピュータ生成のデータや、旧来のシステムのデータなどは、しばしば固定長フィールド形式で保存されています。詳細は、2 章 固定長ファイル ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード を参照してください。

### 例

```
create fixedfilenode
set :fixedfilenode.full_filename = "$CLEO_DEMOS/DRUG4n"
set :fixedfilenode.record_len = 32
set :fixedfilenode.skip_header = 1
set :fixedfilenode.fields = [{ 'Age' 13 } { 'Sex' 5 7 } { 'BP' 9 10 } { 'Cholesterol' 12 22 } { 'Na' 24 25 } { 'K' 27 27 } { 'Drug' 29 32 } ]
set :fixedfilenode.decimal_symbol = Period
set :fixedfilenode.lines_to_scan = 30
```

| fixedfilenode プロパティ     | データ型                          | プロパティの説明  |
|-------------------------|-------------------------------|---|
| record_len              | number                        | 各レコードの文字数を指定します。  |
| line_oriented           | フラグ型                          | 各レコードの末尾の改行文字をスキップします。  |
| decimal_symbol          | Default<br>Comma<br>Period    | データ ソースで使われている小数点記号。例：<br>set :fixedfilenode.decimal_symbol = Period                            |
| skip_header             | number                        | 最初のレコードの先頭で無視する行数を指定します。列見出しを無視する場合などに役立ちます。  |
| auto_recognize_datetime | フラグ型                          | 入力データの日付または時刻を自動的に特定するかどうかを指定します。   |
| lines_to_scan           | number                        | 例：<br>set :fixedfilenode.lines_to_scan = 50.  |
| fields                  | リスト                           | 構造化プロパティ。<br>使用フォーマット：<br>fixedfilenode.fields = [{field start length}<br>{field start length}] |
| full_filename           | string                        | 読み込みファイルのディレクトリを含む完全な名前。  |
| strip_spaces            | None<br>Left<br>Right<br>Both | インポート時に文字列の前後のスペースを破棄します。   |

## 入カノードのプロパティ

| fixedfilenodeプロパティ       | データ型  | プロパティの説明  |
|--------------------------|---|---|
| invalid_char_mode        | Discard<br>Replace  | データ入力から不正な文字（ヌル、0、または現在のエンコード中に存在していない文字）をデータ入力から削除するか（Discard）、指定された 1 文字の記号で不正な文字を置き換えます（Replace）。  |
| invalid_char_replacement | string  |   |
| use_custom_values        | フラグ型  | 次のフォーマットのキー スロット：<br>set :varfilenode.use_custom_values.Age = true  |
| custom_storage           | Unknown<br>String<br>Integer<br>Real<br>Time<br>Date<br>Timestamp   | 次のフォーマットのキー スロット：<br>set :varfilenode.custom_storage.'Age' = "Real"   |
| custom_date_format       | "DDMMYY"<br>"MMDDYY"<br>"YYMMDD"<br>"YYYYMMDD"<br>"YYYYDDD"<br>DAY<br>MONTH<br>"DD-MM-YY"<br>"DD-MM-YYYY"<br>"MM-DD-YY"<br>"MM-DD-YYYY"<br>"DD-MON-YY"<br>"DD-MON-YYYY"<br>"YYYY-MM-DD"<br>"DD.MM.YY"<br>"DD.MM.YYYY"<br>"MM.DD.YY"<br>"MM.DD.YYYY"<br>"DD.MON.YY"<br>"DD.MON.YYYY"<br>"DD/MM/YY"<br>"DD/MM/YYYY"<br>"MM/DD/YY"<br>"MM/DD/YYYY"<br>"DD/MON/YY"<br>"DD/MON/YYYY"<br>MON YYYY<br>q Q YYYY<br>ww WK YYYY | このプロパティは、カスタム（ユーザー設定）ストレージが指定される場合のみ適用されます。<br>例：<br>set:varfilenode.custom<br>次のフォーマットのキー スロット：<br>set :varfilenode.custom_date_format.'LaunchDate' = "DDMMYY" |

| fixedfilenode プロパティ   | データ型   | プロパティの説明   |
|-----------------------|--|--|
| custom_time_format    | "HHMMSS"<br>"HHMM"<br>"MMSS"<br>"HH:MM:SS"<br>"HH:MM"<br>"MM:SS"<br>"(H)H:(M)M:(S)S"<br>"(H)H:(M)M"<br>"(M)M:(S)S"<br>"HH.MM.SS"<br>"HH.MM"<br>"MM.SS"<br>"(H)H.(M)M.(S)S"<br>"(H)H.(M)M"<br>"(M)M.(S)S" | このプロパティは、カスタム（ユーザー設定）ストレージが指定される場合のみ適用されます。<br>次のフォーマットのキー スロット :<br>set :varfilenode.custom_time_format.<br>'Initialize' = "HHMM" |
| custom_decimal_symbol | フィールド  | カスタム（ユーザー設定）ストレージが指定される場合のみ適用されます。<br>次のフォーマットのキー スロット :<br>set :varfilenode.custom_decimal_symbol.<br>'Revenue' = "Comma"         |
| encoding              | StreamDefault<br>SystemDefault<br>"UTF-8"  | テキストのエンコード方法を指定します。  |

## sasimportnode のプロパティ



SAS インポート ノードで、SAS データを IBM® SPSS® Modeler へインポートします。詳細は、[2 章 SAS 入力ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

### 例

```
create sasimportnode
set :sasimportnode.format = Windows
set :sasimportnode.full_filename = "C:/data/retail.sas7bdat"
set :sasimportnode.member_name = "Test"
set :sasimportnode.read_formats = False
set :sasimportnode.full_format_filename = "Test"
```

```
set :sasimportnode.import_names = True
```

| sasimportnode プロパティ  | データ型   | プロパティの説明   |
|----------------------|--|--|
| format               | Windows<br>UNIX<br>Transport<br>SAS7<br>SAS8<br>SAS9 | インポートするファイルのフォーマット。                              |
| full_filename        | string   | パスも含めた、完全なファイル名。この名前を入力します。                      |
| member_name          | string   | 指定した SAS トランスポート ファイルからインポートするメンバーを指定します。        |
| read_formats         | フラグ型   | 指定されたフォーマット ファイルから、データ フォーマット (変数ラベルなど) を読み込みます。 |
| full_format_filename | string   |  |
| import_names         | NamesAndLabels<br>LabelsasNames                      | インポート時に変数名と変数ラベルをマッピングする方法を指定します。                |

## statisticsimportnode のプロパティ



IBM® SPSS® Statistics ファイル ノードは、同じフォーマットを使用する SPSS Statistics で使用される .sav ファイル形式のデータおよび IBM® SPSS® Modeler に保存されたキャッシュ ファイルを読み込みます。詳細は、[8 章 Statistics ファイル ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

このノードのプロパティについては、「[statisticsimportnode のプロパティ](#)」( p. 343 ) に記載されています。

## userinputnode のプロパティ



ユーザー入力ノードで、最初から、または既存のデータを変更して、合成データを作成する簡単な方法が提供されます。これは、モデル作成用の検定データセットを作成する場合などに役立ちます。詳細は、[2 章 ユーザー入力ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

### 例

```
create userinputnode
set :userinputnode.data.test1 = "2, 4, 8"
set :userinputnode.names = [test1 test2]
set :userinputnode.custom_storage.test1 = Integer
```

set :userinputnode.data\_mode = "Ordered"

| userinputnodeプロパティ | データ型  | プロパティの説明  |
|--------------------|---|---|
| data               |   | 次のフォーマットのキープロパティ：<br>set :userinputnode.data.Age = "1 2 3 4"<br>または、低、高、およびサイズの各値をカンマで区切った文字列で指定することもできます。例：<br>set :userinputnode.data.Age = "10, 70, 5" 各フィールドのデータの長さは異なる場合がありますが、フィールドのストレージについて一貫している必要があります。存在していないフィールドに値を設定すると、そのフィールドが作成されます。また、フィールドの値として空文字列 (" ") を設定すると、指定したフィールドが削除されます。 |
| names              |   | ノードにより生成されたフィールド名のリストを設定または返す構造化スロット。<br>例：<br>['Field1' 'Field2']  |
| custom_storage     | Unknown<br>String<br>Integer<br>Real<br>Time<br>Date<br>Timestamp | フィールドのストレージを設定するか返す、キー スロット。<br>例：<br>set :userinputnode.custom_storage.'Age' = "Real"  |
| data_mode          | Combined<br>Ordered   | Combined が指定された場合、レコードは、セット値と最小/最大値のそれぞれ組み合わせについて生成されます。生成されたレコード数は、それぞれのフィールドの値の数値の積に等しくなります。Ordered が指定された場合、データ行を生成するために、各レコードの各列から 1 個の値が取られます。生成されるレコード数は、フィールドに関連付けられている最大の値に等しくなります。より小さいデータ値を持つフィールドは、ヌル値で埋められます。   |
| values             |   | このプロパティは、userinputnode.dataにより廃止されるため、サポートされていません。  |

## variablefilenode のプロパティ



可変長ノードで、可変長フィールド テキスト ファイル、つまりフィールド数は一定でも各フィールド内の文字数が異なるレコードを含むファイルから、データを読み込みます。このノードは、固定長のヘッダー テキストやある種の注釈があるファイルにも使用できます。詳細は、[2 章 可変長ファイル ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

### 例

```
create variablefilenode
set :variablefilenode.full_filename = "$CLEO_DEMOS/DRUG4n"
set :variablefilenode.read_field_names = True
set :variablefilenode.delimit_other = True
set :variablefilenode.other = ','
set :variablefilenode.quotes_1 = Discard
set :variablefilenode.decimal_symbol = Comma
set :variablefilenode.invalid_char_mode = "Replace"
set :variablefilenode.invalid_char_replacement = ""
set :variablefilenode.use_custom_values.Age = True
set :variablefilenode.direction.Age = Input
set :variablefilenode.type.Age = Range
set :variablefilenode.values.Age = [1 100]
```

| variablefilenode プロパティ | データ型   | プロパティの説明  |
|------------------------|--------|---|
| skip_header            | number | 最初のレコードの先頭で無視する文字数を指定します。<br>使用フォーマット：<br><b>variablefilenode:skip_header = 3</b>                                       |
| num_fields_auto        | フラグ型   | 各レコードのフィールドの数を自動的に決定します。レコードは、改行文字で終わる必要があります。<br>使用フォーマット：<br><b>variablefilenode:num_fields_auto</b>                  |
| num_fields             | number | 各レコードのフィールドの数を手動で指定します。   |
| delimit_space          | フラグ型   | ファイルのフィールドを区切る文字を指定します。   |
| delimit_tab            | フラグ型   |   |
| delimit_new_line       | フラグ型   |   |
| delimit_non_printing   | フラグ型   |   |
| delimit_comma          | フラグ型   | この場合、カンマはストリーム内でフィールドの区切り文字と桁区切り記号の両方であるため、 <b>delimit_other</b> を true に設定し、 <b>other</b> プロパティを使用し、カンマを区切り記号として指定します。 |

| variablefilenodeプロパティ    | データ型  | プロパティの説明   |
|--------------------------|---|--|
| delimit_other            | フラグ型  | <b>other</b> プロパティを使用して、カスタム区切り記号をユーザーが指定できます。   |
| other                    | string  | <b>delimit_other</b> が true に設定されているときに使用される区切り記号を指定します。   |
| decimal_symbol           | Default<br>Comma<br>Period  | データ ソースで使われている小数点記号を指定します。   |
| multi_blank              | フラグ型  | 複数の隣接する空白区切り文字を 1 つの区切り文字として扱います。  |
| read_field_names         | フラグ型  | データ ファイル中の最初の行を列のラベルとして取り扱います。   |
| strip_spaces             | None<br>Left<br>Right<br>Both                                     | インポート時に文字列の前後のスペースを破棄します。  |
| invalid_char_mode        | Discard<br>Replace  | データ入力から不正な文字（ヌル、0、または現在のエンコード中に存在していない文字）をデータ入力から削除するか（Discard）、指定された 1 文字の記号で不正な文字を置き換えます（Replace）。 |
| invalid_char_replacement | string  |  |
| lines_to_scan            | number  | 指定したデータ型をスキャンする行数を指定します。   |
| auto_recognize_datetime  | フラグ型  | 入力データの日付または時刻を自動的に特定するかどうかを指定します。  |
| quotes_1                 | Discard<br>PairAndDiscard<br>IncludeAsText                        | インポートでの単一引用符の処理方法を指定します。   |
| quotes_2                 | Discard<br>PairAndDiscard<br>IncludeAsText                        | インポートでの二重引用符の処理方法を指定します。   |
| full_filename            | string  | 読み込みファイルのディレクトリを含む完全な名前。   |
| use_custom_values        | フラグ型  | 次のフォーマットのキー スロット：<br>set :varfilenode.use_custom_values.Age = true                                   |
| custom_storage           | Unknown<br>String<br>Integer<br>Real<br>Time<br>Date<br>Timestamp | 次のフォーマットのキー スロット：<br>set :varfilenode.custom_storage.'Age' = "Real"                                  |



## 入カノードのプロパティ

| variablefilenodeプロパティ | データ型  | プロパティの説明  |
|-----------------------|---|---|
| custom_date_format    | "DDMMYY"<br>"MMDDYY"<br>"YYMMDD"<br>"YYYYMMDD"<br>"YYYYDDD"<br>DAY<br>MONTH<br>"DD-MM-YY"<br>"DD-MM-YYYY"<br>"MM-DD-YY"<br>"MM-DD-YYYY"<br>"DD-MON-YY"<br>"DD-MON-YYYY"<br>"YYYY-MM-DD"<br>"DD.MM.YY"<br>"DD.MM.YYYY"<br>"MM.DD.YY"<br>"MM.DD.YYYY"<br>"DD.MON.YY"<br>"DD.MON.YYYY"<br>"DD/MM/YY"<br>"DD/MM/YYYY"<br>"MM/DD/YY"<br>"MM/DD/YYYY"<br>"DD/MON/YY"<br>"DD/MON/YYYY"<br>MON YYYY<br>q Q YYYY<br>ww WK YYYY | カスタム（ユーザー設定）ストレージが指定される場合のみ適用されます。<br>例：<br>set:varfilenode.custom<br>次のフォーマットのキー スロット：<br>set<br>:varfilenode.custom_date_format.<br>'LaunchDate' = "DDMMYY" |
| custom_time_format    | "HHMMSS"<br>"HHMM"<br>"MMSS"<br>"HH:MM:SS"<br>"HH:MM"<br>"MM:SS"<br>"(H)H:(M)M:(S)S"<br>"(H)H:(M)M"<br>"(M)M:(S)S"<br>"HH.MM.SS"<br>"HH.MM"<br>"MM.SS"<br>"(H)H.(M)M.(S)S"<br>"(H)H.(M)M"<br>"(M)M.(S)S"  | カスタム（ユーザー設定）ストレージが指定される場合のみ適用されます。<br>次のフォーマットのキー スロット：<br>set<br>:varfilenode.custom_time_format.<br>'Initialize' = "HHMM"                                   |
| custom_decimal_symbol | フィールド   | カスタム（ユーザー設定）ストレージが指定される場合のみ適用されます。<br>次のフォーマットのキー スロット：<br>set :varfilenode.custom_decimal_symbol.'Revenue' = "Comma"   |
| encoding              | StreamDefault<br>SystemDefault<br>"UTF-8"   | テキストのエンコード方法を指定します。   |

## xmlimportnode のプロパティ



XML 入力ノードを使用して、XML 形式のデータをストリームにインポートできます。ディレクトリの 1 つのファイルまたはすべてのファイルをインポートできます。オプションで、XML 構造を読み込むスキーマ ファイルを指定できます。詳細は、2 章 XML 入力ノード in IBM SPSS Modeler 15 入力ノード、プロセスノード、出力ノード を参照してください。

### 例

```
create xmlimportnode
set :xmlimportnode.full_filename = "c:\import\ ebooks.xml"
set :xmlimportnode.records = "/author/name"
```

| xmlimportnode プロパティ  | データ型                | プロパティの説明  |
|----------------------|---------------------|---|
| read                 | single<br>directory | 単独のデータ ファイルを読み込む (デフォルト) か、ディレクトリ内のすべての XML ファイルを読み込みます。                            |
| recurse              | フラグ型                | 指定したディレクトリのすべてのサブディレクトリから XML ファイルを追加で読み込むかどうかを指定します。                               |
| full_filename        | string              | (必須) インポートする XML ファイルの完全パスおよびファイル名 (read = single の場合)。                             |
| directory_name       | string              | (必須) XML ファイルをインポートするディレクトリの完全パスおよび名前 (read = directory の場合)。                       |
| full_schema_filename | string              | XML 構造を読み込む XSD ファイルまたは DTD ファイルの完全パスおよびファイル名。このパラメータを使用すると、構造を XML 入力ファイルから読み込みます。 |
| records              | string              | レコードの境界を定義する XPath 式 (例: /author/name)。入力ファイルにこの要素が出現するごとに、新しいレコードが作成されます。          |
| mode                 | read<br>specify     | すべてのデータを読み込む (デフォルト) か、読み込む項目を指定します。  |
| fields               |                     | インポートする項目 (要素と属性) のリスト。リスト内の各アイテムは XPath 式です。                                       |

# レコード設定ノードのプロパティ

## appendnode のプロパティ



レコード追加ノードで、レコードのセットを連結します。レコード追加ノードは、構造が似ていながらデータが異なるデータセットを組み合わせる場合に役立ちます。詳細は、[3 章 レコード追加ノード in IBM SPSS Modeler 15 入力ノード、プロセスノード、出力ノード](#) を参照してください。

### 例

```
create appendnode
set :appendnode.match_by = Name
set :appendnode.match_case = True
set :appendnode.include_fields_from = All
set :appendnode.create_tag_field = True
set :appendnode.tag_field_name = "Append_Flag"
```

| appendnode プロパティ    | データ型             | プロパティの説明   |
|---------------------|------------------|--|
| match_by            | Position<br>Name | メイン データ ソース中のフィールドの位置 (Position) 、または入力データセット中のフィールド名 (Name) を基準にして、データセットを追加できます。 |
| match_case          | フラグ型             | フィールド名を比較するときに大文字と小文字の区別を有効にします。   |
| include_fields_from | Main<br>All      |  |
| create_tag_field    | フラグ型             |  |
| tag_field_name      | string           |  |

## aggregatenode のプロパティ



レコード集計ノードで、一連の入力レコードを要約集計された出力レコードに置き換えます。詳細は、[3 章 レコード集計ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

### 例

```
create aggregatenode
connect :databasenode to :aggregatenode
```

```

set :aggregatenode.contiguous = True
set :aggregatenode.keys = ['Drug']
set :aggregatenode.aggregates.Age = [Sum Mean]
set :aggregatenode.inc_record_count = True
set :aggregatenode.count_field = "index"
set :aggregatenode.extension = "Aggregated_"
set :aggregatenode.add_as = Prefix

```

| aggregatenodeプロパティ | データ型                    | プロパティの説明   |
|--------------------|-------------------------|--|
| keys               | [フィールド フィールド ... フィールド] | 集計にキーとして使用できるフィールドが一覧表示されます。たとえば、 <b>Sex</b> や <b>Region</b> がキー フィールドの場合、 <b>M</b> と <b>F</b> および地域 <b>N</b> と地域 <b>S</b> の一意の組み合わせ (4 通り) ごとに集計レコードが生成されます。                          |
| contiguous         | フラグ型                    | 同じキー値を持つすべてのレコードが入力にグループ化されている場合 (たとえば、入力がキー フィールドにソートされる場合)、このオプションを選択します。このオプションを選択すると、パフォーマンスが向上します。  |
| aggregates         |                         | 集計する数値フィールド、および選択されている集計モードを表示する構造化プロパティ。例:<br>set :aggregatenode.<br>aggregates.Age = [Sum Mean Min Max SDev Median Count Variance Firstquartile Thirdquartile]。指定された集計方法がリスト内に含まれます。 |
| extension          | string                  | 重複集計フィールドに対応させる接頭辞または接尾辞を指定します (下の例を参照)。   |
| add_as             | Suffix<br>Prefix        |  |
| inc_record_count   | フラグ型                    | 各集計レコードを作成するために集計された入力レコード数を指定する追加フィールドを作成します。   |
| count_field        | string                  | レコード度数フィールドの名前を指定します。  |

## balancenode のプロパティ



バランス ノードで、データ セットが指定した条件に合うように、データ セットの不均衡を修正します。バランス式で、指定した比率によって条件が真 (true) の場合に、レコードの比率を調整します。詳細は、[3 章 バランス ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード を参照してください。](#)

**例**

```
create balancenode
set :balancenode.training_data_only = true
set :balancenode.directives = \
  [{1.3 "Age > 60"}{1.5 "Na > 0.5"}]
```

| balancenode プロパティ  | データ型 | プロパティの説明   |
|--------------------|------|--|
| directives         |      | 指定された数値に基づいてフィールド値の割合を均衡にするための構造化プロパティ（次の例を参照してください）。                |
| training_data_only | フラグ型 | 学習データのみがバランス化されるよう指定します。データ区分フィールドがストリーム中で指定されていない場合、このオプションは無視されます。 |

**例**

```
create balancenode
set :balancenode.directives = \
  [{1.3 "Age > 60"}{1.5 "Na > 0.5"}]
```

このノードのプロパティは次のフォーマットを使用します。

[{ 数値文字列 } ¥ { 数値文字列 } ¥ .... { 数値文字列 }].

注： 文字列を式に埋め込む場合（二重引用符を使用）、その先頭にエスケープ文字 "\" を指定する必要があります。文字 "\" は、引数を明確に記述するための行継続文字としても使われます。

## distinctnode のプロパティ



重複レコード ノードで、重複レコードを削除します。その場合、最初の重複するレコードをデータ ストリームに渡すか、または、最初のレコードを破棄して、その後の重複レコードをデータ ストリームに渡します。 [詳細は、3 章 重複レコード ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード を参照してください。](#)

**例**

```
create distinctnode
set :distinctnode.mode = Include
set :distinctnode.fields = ['Age' 'Sex']
set :distinctnode.keys_pre_sorted = True
```

| distinctnode プロパティ     | データ型                   | プロパティの説明  |
|------------------------|------------------------|---|
| mode                   | Include<br>Discard     | データ ストリームに最初の重複レコードを含めるか、最初の重複レコードを破棄して、代わりにすべての重複レコードをデータ ストリームに渡すことができます。 |
| fields                 | [フィールド フィールド<br>フィールド] | レコードが同一であるかどうかを判断するために使われるフィールドを表示します。                                      |
| low_distinct_key_count | フラグ型                   | キー フィールドに少ないレコードまたは少ない一意の値を持つよう指定します。                                       |
| keys_pre_sorted        | フラグ型                   | 同じキー値を持つすべてのレコードが入力で一緒にグループ化されるよう指定します。                                     |
|                        |                        |   |

## mergenode のプロパティ



レコード結合ノードは、複数の入力レコードを取得し、入力フィールドの全部または一部を含む 1 つの出力レコードを作成します。この機能は、内部顧客データと購入人口データのような、異なるソースからのデータを結合する場合に役立ちます。詳細は、[3 章 レコード結合ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

### 例

```
create mergenode
connect customerdata to :mergenode
connect salesdata to :mergenode
set :mergenode.method = Keys
set :mergenode.key_fields = ['id']
set :mergenode.common_keys = true
set :mergenode.join = PartialOuter
set :mergenode.outer_join_tag.2 = true
set :mergenode.outer_join_tag.4 = true
set :mergenode.single_large_input = true
set :mergenode.single_large_input_tag = '2'
set :mergenode.use_existing_sort_keys = true
```

## レコード設定ノードのプロパティ

```
set :mergenode.existing_sort_keys = [{id' Ascending}]
```

| mergenodeプロパティ         | データ型  | プロパティの説明   |
|------------------------|---|--|
| method                 | Order<br>Keys<br>Condition                    | データ ファイル内の順序に応じてレコードが結合されるか (Order)、またはキー フィールド内が同じ値のレコードを結合するのにキー フィールドが使用されるかどうか (Keys)、または指定された条件を満たした場合にレコードが結合されるかどうか (Condition) を指定します。         |
| condition              | string  | <b>method</b> が <b>Condition</b> に設定されている場合、レコードを含めるまたは破棄する条件を指定します。   |
| key_fields             | [フィールド フィールド<br>フィールド フィールド]                  |  |
| common_keys            | フラグ型  |  |
| join                   | Inner<br>FullOuter<br>PartialOuter<br>Anti    | 一例を以下に挙げます。<br><b>set :merge.join = FullOuter</b>  |
| outer_join_tag.n       | フラグ型  | このプロパティでは、n は [データセットの選択] ダイアログ ボックスに表示されるタグ名です。どのようなデータセット数であっても不完全なレコードを作成する可能性があるため、複数のタグ名を指定できます。  |
| single_large_input     | フラグ型  | ほかの入力と比べて比較的大きな入力を指定し最適化を行うかどうかを指定します。   |
| single_large_input_tag | string  | [ラージ データセットの選択] ダイアログ ボックスに表示されるタグ名を指定します。このプロパティの用途は、1 つの入力データセットしか指定できないという点で、 <b>outer_join_tag</b> プロパティとは若干異なることに注意してください (データ型がフラグと文字列という違いもあり)。 |
| use_existing_sort_keys | フラグ型  | 入力がすでにキー フィールドでソート済みかどうかを指定します。  |
| existing_sort_keys     | [[string Ascending] ¥<br>{string Descending}] | すでにソートされたフィールドとソート方向を指定します。  |

## rfmaggregatenode のプロパティ



リーセンシ、フリクエンシ、マネタリー（RFM）のレコード集計ノードを使用すると、顧客の過去のトランザクション データを取得、未使用のデータを削除、残りのトランザクション データをすべて単一行に結合することができます。これにより、最後のトランザクションの時期、トランザクション数、これらのトランザクションの合計金額が一覧表示されます。詳細は、3 章 RFM レコード集計ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード を参照してください。

### 例

```
create rfmaggregatenode
connect :fillernode to :rfmaggregatenode
set :rfmaggregatenode.relative_to = Fixed
set :rfmaggregatenode.reference_date = "2007-10-12"
set :rfmaggregatenode.id_field = "CardID"
set :rfmaggregatenode.date_field = "Date"
set :rfmaggregatenode.value_field = "Amount"
set :rfmaggregatenode.only_recent_transactions = True
set :rfmaggregatenode.transaction_date_after = "2000-10-01"
```

| rfmaggregatenodeプロパティ     | データ型             | プロパティの説明  |
|---------------------------|------------------|---|
| relative_to               | Fixed<br>Today   | トランザクションのリーセンシが計算される日付を指定します。   |
| reference_date            | 日付               | Fixed が relative_to に設定されている場合にのみ使用できます。  |
| contiguous                | フラグ型             | データ ストリーム中で同じ ID を持つすべてのレコードが一緒に表示されるようにデータをソートしている場合、このオプションを選択すると処理を高速化することができます。 |
| id_field                  | フィールド            | 顧客およびトランザクションを識別するために使用するフィールドを指定します。   |
| date_field                | フィールド            | リーセンシを計算するために使用される日付フィールドを選択します。  |
| value_field               | フィールド            | マネタリー値を計算するために使用するフィールドを指定します。  |
| extension                 | string           | 重複集計フィールドに対応させる接頭辞または接尾辞を指定します。   |
| add_as                    | Suffix<br>Prefix | extension を接尾辞として追加するか、または接頭辞として追加するかを指定します。  |
| discard_low_value_records | フラグ型             | discard_records_below 設定の使用を有効にします。   |



## レコード設定ノードのプロパティ

| rfmaggregatenodeプロパティ    | データ型                             | プロパティの説明  |
|--------------------------|----------------------------------|---|
| discard_records_below    | number                           | RFM の合計を計算する場合に使用されないトランザクションの詳細の最小値を指定することができます。値の単位は、選択された [value] フィールドに関連します。                                   |
| only_recent_transactions | フラグ型                             | specify_transaction_date または transaction_within_last 設定の使用を有効にします。  |
| specify_transaction_date | フラグ型                             |   |
| transaction_date_after   | 日付                               | specify_transaction_date が選択されている場合にのみ使用できます。データが分析に含まれた後のトランザクションの日付を指定します。  |
| transaction_within_last  | number                           | transaction_within_last が選択されている場合にのみ使用できます。レコードが分析に含まれる後の [リーセンシの相対値を計算] の日付からさかのぼった期間の数および種類 (日、週、月または年数) を指定します。 |
| transaction_scale        | Days<br>Weeks<br>Months<br>Years | transaction_within_last が選択されている場合にのみ使用できます。レコードが分析に含まれる後の [リーセンシの相対値を計算] の日付からさかのぼった期間の数および種類 (日、週、月または年数) を指定します。 |
| save_r2                  | フラグ型                             | 各顧客の 2 番目に最近のトランザクションの日付を表示します。   |
| save_r3                  | フラグ型                             | save_r2 が選択されている場合にのみ使用できます。各顧客の 3 番目に最近のトランザクションの日付を表示します。   |

## samplenode のプロパティ



サンプル ノードでは、レコードのサブセットを選択します。層化サンプル、クラスタ サンプル、非無作為 (構造化) サンプルなど、さまざまなサンプルの種類がサポートされています。サンプリングは、パフォーマンスの向上、および分析のための関連するレコードまたはトランザクションのグループの選択に役に立ちます。詳細は、3 章 サンプル ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード を参照してください。

## 例

```
/* Create two Sample nodes to extract
different samples from the same data */
```

```
create variablefilenode
```

```
set :variablefilenode.full_filename = "$CLEO_DEMOS/DRUG1n"
```

```
set node = create samplenode at 300 100
rename ^node as 'First 500'
connect :variablefilenode to 'First 500'
set 'First 500':samplenode.method = Simple
set 'First 500':samplenode.mode = Include
set 'First 500':samplenode.sample_type = First
set 'First 500':samplenode.first_n = 500
```

```
set node = create samplenode at 300 200
rename ^node as 'Custom Strata'
connect :variablefilenode to 'Custom Strata'
set 'Custom Strata':samplenode.method = Complex
set 'Custom Strata':samplenode.stratify_by = ['Sex' 'Cholesterol']
set 'Custom Strata':samplenode.sample_units = Proportions
set 'Custom Strata':samplenode.sample_size_proportions = Custom
set 'Custom Strata':samplenode.sizes_proportions= \
  [{"M" "High" "Default"}{"M" "Normal" "Default"}\
  {"F" "High" "0.3"}{"F" "Normal" "0.3"}]
```

| samplenodeプロパティ | データ型                         | プロパティの説明   |
|-----------------|------------------------------|--|
| method          | Simple<br>Complex            |  |
| mode            | Include<br>Discard           | 指定された条件を満たすレコードを含めるか (Include)、破棄 (Discard) します。   |
| sample_type     | First<br>OneInN<br>RandomPct | サンプリング方法を指定します。一例を以下に挙げます。<br><b>set :samplenode.sample_type = First</b><br><b>set :samplenode.first_n = 100</b>   |
| first_n         | integer                      | 指定された分割点までのレコードを含めるか破棄します。   |
| one_in_n        | number                       | n 番目ごとにレコードを含めるか破棄します。   |
| rand_pct        | number                       | 含めるか破棄するレコードのパーセンテージを指定します。  |
| use_max_size    | フラグ型                         | <b>maximum_size</b> 設定の使用を有効にします。  |
| maximum_size    | integer                      | データ ストリームに入れるまたはデータ ストリームから破棄するサンプルの最大数を指定します。このオプションは冗長であり、そのため、 <b>First</b> と <b>Include</b> が指定されているときは破棄されます。 |
| set_random_seed | フラグ型                         | ランダム シード設定の使用を有効にします。  |

## レコード設定ノードのプロパティ

| samplenodeプロパティ         | データ型  | プロパティの説明  |
|-------------------------|---|---|
| random_seed             | integer   | ランダム シードとして使用する値を指定します。   |
| complex_sample_type     | Random<br>Systematic                                  |   |
| sample_units            | Proportions<br>Counts                                 |   |
| sample_size_proportions | Fixed<br>Custom<br>Variable                           |   |
| sample_size_counts      | Fixed<br>Custom<br>Variable                           |   |
| fixed_proportions       | number  |   |
| fixed_counts            | integer   |   |
| variable_proportions    | フィールド   |   |
| variable_counts         | フィールド   |   |
| use_min_stratum_size    | フラグ型  |   |
| minimum_stratum_size    | integer   | このオプションは、 <b>Sample units=Proportions</b> によって複雑なサンプルが作成された場合にのみ適用されます。                                     |
| use_max_stratum_size    | フラグ型  |   |
| maximum_stratum_size    | integer   | このオプションは、 <b>Sample units=Proportions</b> によって複雑なサンプルが作成された場合にのみ適用されます。                                     |
| clusters                | フィールド   |   |
| stratify_by             | [フィールド1 ...<br>フィールドN]                                |   |
| specify_input_weight    | フラグ型  |   |
| input_weight            | フィールド   |   |
| new_output_weight       | string  |   |
| sizes_proportions       | [{stringstring<br>value} {stringstring<br>value} ...] | <b>sample_units=proportions</b> および <b>sample_size_proportions=Custom</b> の場合、層化フィールドの値の考えられる組み合わせの値を指定します。 |
| default_proportion      | number  |   |
| sizes_counts            | [{stringstring<br>value} {stringstring<br>value} ...] | 層化フィールドの値の考えられる組み合わせの値を指定します。使用方法は <b>sizes_proportions</b> と似ていますが、割合ではなく整数を指定します。                         |
| default_count           | number  |   |

## selectnode のプロパティ



条件抽出ノードで、特定の条件に基づいて、データ ストリームからレコードのサブセットを選択したり破棄したりできます。たとえば、特定の営業地域に関連するレコードを選択できます。詳細は、[3 章 条件抽出ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

### 例

```
create selectnode
set :selectnode.mode = Include
set :selectnode.condition = "Age < 18"
```

| selectnode プロパティ | データ型               | プロパティの説明                      |
|------------------|--------------------|-------------------------------|
| mode             | Include<br>Discard | 選択したレコードを含めるか、または破棄するかを指定します。 |
| condition        | string             | レコードを含めるか、または破棄かの条件。          |

## sortnode のプロパティ



ソート ノードで、1 つまたは複数のフィールド値に基づいて、レコードを昇順または降順にソートします。詳細は、[3 章 ソート ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

### 例

```
create sortnode
set :sortnode.keys = [{"Age" Ascending} {"Sex" Descending}]
set :sortnode.default_ascending = False
set :sortnode.use_existing_keys = True
set :sortnode.existing_keys = [{"Age" Ascending}]
```

| sortnode プロパティ    | データ型  | プロパティの説明  |
|-------------------|---|---|
| keys              | [{"string Ascending"} ¥<br>{"string Descending"}] | ソートの対象となるフィールドを指定します (下の例を参照)。ソートの方向が指定されていない場合、デフォルトが使用されます。 |
| default_ascending | フラグ型  | デフォルトのソート順を指定します。   |
| use_existing_keys | フラグ型  | 前に使用されたフィールドのソート順を使用してソートを最適化するかどうかを指定します。                    |
| existing_keys     |   | すでにソートされたフィールドとソート方向を指定します。keys プロパティと同じフォーマットを使用します。         |

# フィールド設定ノードのプロパティ

## anonymizenode のプロパティ



匿名化ノードは、フィールド名や値の下流の表示方法を変換し、元のデータを隠します。これは、他のユーザーが顧客名やその他の詳細情報をなどの重要情報を使用してモデルを構築できるようにする場合に有用です。詳細は、4章 [匿名化ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

### 例

```
create anonymizenode
set:anonymizenode.enable_anonymize = age
set:anonymizenode.use_prefix = true
set:anonymizenode.prefix = "myprefix"
set:anonymizenode.transformation = Random
set:anonymizenode.set_random_seed = true
set:anonymizenode.random_seed = "123"
```

| anonymizenodeプロパティ | データ型            | プロパティの説明  |
|--------------------|-----------------|---|
| enable_anonymize   | フラグ型            | T に設定したときは、フィールドの値の匿名化をアクティブ化します（[値を匿名化] 列でそのフィールドのために [はい] を選択することと同等です）。  |
| use_prefix         | フラグ型            | T に設定したときは、ユーザー指定接頭辞が指定されている場合に、そのユーザー指定接頭辞が使用されます。ハッシュ メソッドによって匿名化されるフィールドに適用され、そのフィールドの [値を置換] ダイアログの [ユーザー設定] ラジオ ボタンを選択することと同等です。 |
| prefix             | string          | [値を置換] ダイアログ ボックスのテキスト ボックスに接頭辞を入力することと同等です。デフォルトの接頭辞は、何も他に指定されていない場合は、デフォルト値です。  |
| transformation     | Random<br>Fixed | Transform メソッドにより匿名化されたフィールドの変換パラメータが無作為 (Random) か固定 (Fixed) かを決定します。  |
| set_random_seed    | フラグ型            | T に設定したときは、指定されたシード値を使用します（変換も Random に設定されている場合）。  |
| random_seed        | integer         | set_random_seed を設定したときは、これが乱数のシードになります。  |

| anonymizenode プロパティ | データ型   | プロパティの説明  |
|---------------------|--------|---|
| scale               | number | 変換を Fixed に設定したときに、この値がスケールに使用されます。最大スケール値は通常 10 ですが、あふれを防止するために減少できます。 |
| translate           | number | 変換を Fixed に設定したときに、この値が変換に使用されます。最大変換値は通常 1000 ですが、あふれを防止するために減少できます。   |

## autodataprenode のプロパティ



自動データ準備 (ADP) ノードでは、データ分析、固定値の識別、問題のあるまたは役に立たない可能性のあるフィールドのスクリーニング、必要に応じた新しい属性の取得、詳細なスクリーニングおよびサンプリング手法を使用したパフォーマンスの向上などを行うことができます。完全に自動化された方法でノードを使用し、ノードで固定値を選択および適用できます。または必要に応じて変更の作成および承認、拒否または修正の前に変更をプレビューできます。 [詳細は、4 章 自動データ準備 in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード を参照してください。](#)

### 例

```
create autodataprenode
set:autodataprenode.objective = Balanced
set:autodataprenode.excluded_fields = Filter
set:autodataprenode.prepare_dates_and_times = true
set:autodataprenode.compute_time_until_date = true
set:autodataprenode.reference_date = Today
set:autodataprenode.units_for_date_durations = Automatic
```

| autodataprenode プロパティ | データ型                                    | プロパティの説明  |
|-----------------------|---|---|
| objective             | Balanced<br>Speed<br>Accuracy<br>Custom |   |
| custom_fields         | フラグ型                                    | 真 (true) の場合は、現在のノードのターゲット、入力、その他フィールドなどを指定することができます。偽 (false) の場合は、上流のデータ型ノードから現在の設定が使用されます。 |
| target                | フィールド                                   | 1 つの対象フィールドを指定します。  |
| inputs                | [フィールド 1 ...<br>フィールド N]                | モデルで使用される入力または予測変数フィールド。  |
| use_frequency         | フラグ型                                    |   |

## フィールド設定ノードのプロパティ

| autodatapreinodeプロパティ       | データ型                           | プロパティの説明                    |
|-----------------------------|--------------------------------|-----------------------------|
| frequency_field             | フィールド                          |                             |
| use_weight                  | フラグ型                           |                             |
| weight_field                | フィールド                          |                             |
| excluded_fields             | Filter<br>None                 |                             |
| if_fields_do_not_match      | StopExecution<br>ClearAnalysis |                             |
| prepare_dates_and_times     | フラグ型                           | すべての日付/時間フィールドへのアクセスを制御します。 |
| compute_time_until_date     | フラグ型                           |                             |
| reference_date              | Today<br>Fixed                 |                             |
| fixed_date                  | 日付                             |                             |
| units_for_date_durations    | Automatic<br>Fixed             |                             |
| fixed_date_units            | Years<br>Months<br>Days        |                             |
| compute_time_until_time     | フラグ型                           |                             |
| reference_time              | CurrentTime<br>Fixed           |                             |
| fixed_time                  | 時間                             |                             |
| units_for_time_durations    | Automatic<br>Fixed             |                             |
| fixed_date_units            | Hours<br>Minutes<br>Seconds    |                             |
| extract_year_from_date      | フラグ型                           |                             |
| extract_month_from_date     | フラグ型                           |                             |
| extract_day_from_date       | フラグ型                           |                             |
| extract_hour_from_time      | フラグ型                           |                             |
| extract_minute_from_time    | フラグ型                           |                             |
| extract_second_from_time    | フラグ型                           |                             |
| exclude_low_quality_inputs  | フラグ型                           |                             |
| exclude_too_many_missing    | フラグ型                           |                             |
| maximum_percentage_missing  | number                         |                             |
| exclude_too_many_categories | フラグ型                           |                             |
| maximum_number_categories   | number                         |                             |
| exclude_if_large_category   | フラグ型                           |                             |
| maximum_percentage_category | number                         |                             |
| prepare_inputs_and_target   | フラグ型                           |                             |
| adjust_type_inputs          | フラグ型                           |                             |

| autodatapreinodeプロパティ             | データ型              | プロパティの説明 |
|-----------------------------------|-------------------|----------|
| adjust_type_target                | フラグ型              |          |
| reorder_nominal_inputs            | フラグ型              |          |
| reorder_nominal_target            | フラグ型              |          |
| replace_outliers_inputs           | フラグ型              |          |
| replace_outliers_target           | フラグ型              |          |
| replace_missing_continuous_inputs | フラグ型              |          |
| replace_missing_continuous_target | フラグ型              |          |
| replace_missing_nominal_inputs    | フラグ型              |          |
| replace_missing_nominal_target    | フラグ型              |          |
| replace_missing_ordinal_inputs    | フラグ型              |          |
| replace_missing_ordinal_target    | フラグ型              |          |
| maximum_values_for_ordinal        | number            |          |
| minimum_values_for_continuous     | number            |          |
| outlier_cutoff_value              | number            |          |
| outlier_method                    | Replace<br>Delete |          |
| rescale_continuous_inputs         | フラグ型              |          |
| rescaling_method                  | MinMax<br>ZScore  |          |
| min_max_minimum                   | number            |          |
| min_max_maximum                   | number            |          |
| z_score_final_mean                | number            |          |
| z_score_final_sd                  | number            |          |
| rescale_continuous_target         | フラグ型              |          |
| target_final_mean                 | number            |          |
| target_final_sd                   | number            |          |
| transform_select_input_fields     | フラグ型              |          |
| maximize_association_with_target  | フラグ型              |          |
| p_value_for_merging               | number            |          |
| merge_ordinal_features            | フラグ型              |          |
| merge_nominal_features            | フラグ型              |          |
| minimum_cases_in_category         | number            |          |
| bin_continuous_fields             | フラグ型              |          |
| p_value_for_binning               | number            |          |
| perform_feature_selection         | フラグ型              |          |
| p_value_for_selection             | number            |          |
| perform_feature_construction      | フラグ型              |          |
| transformed_target_name_extension | string            |          |



| autodataprenodeプロパティ              | データ型   | プロパティの説明 |
|-----------------------------------|--------|----------|
| transformed_inputs_name_extension | string |          |
| constructed_features_root_name    | string |          |
| years_duration_name_extension     | string |          |
| months_duration_name_extension    | string |          |
| days_duration_name_extension      | string |          |
| hours_duration_name_extension     | string |          |
| minutes_duration_name_extension   | string |          |
| seconds_duration_name_extension   | string |          |
| year_cyclical_name_extension      | string |          |
| month_cyclical_name_extension     | string |          |
| day_cyclical_name_extension       | string |          |
| hour_cyclical_name_extension      | string |          |
| minute_cyclical_name_extension    | string |          |
| second_cyclical_name_extension    | string |          |

## binningnode のプロパティ



データ分割ノードで、既存の 1 つまたは複数の連続型（数値範囲）フィールドの値に基づいて、自動的に新しい名義型（セット型）フィールドを作成します。たとえば、連続型収入フィールドを、平均からの偏差による収入グループを含む、新しいカテゴリ フィールドに変換することができます。新規フィールドのビンを作成すると、分割点に基づいてフィールド作成ノードを生成することができます。詳細は、[4 章 データ分割ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

### 例

```
create binningnode
set :binningnode.fields = [Na K]
set :binningnode.method = Rank
set :binningnode.fixed_width_name_extension = "_binned"
set :binningnode.fixed_width_add_as = Suffix
set :binningnode.fixed_bin_method = Count
set :binningnode.fixed_bin_count = 10
set :binningnode.fixed_bin_width = 3.5
```

set :binningnode.tile10 = true

| binningnode プロパティ          | データ型  | プロパティの説明  |
|----------------------------|---|---|
| fields                     | [フィールド1<br>フィールド2 ...<br>フィールドn]                    | 変換保留中の連続型（数値範囲）フィールド。複数のフィールドを同時にビンに分割できます。   |
| method                     | FixedWidth<br>EqualCount<br>Rank<br>SDev<br>Optimal | 新規フィールドのビン（カテゴリ）の分割点を決める方法。   |
| rcalculate_bins            | Always<br>IfNecessary                               | ノードが実行されるごとに、ビンが再計算され、適切なビンの中にデータが配置されるか、またはデータが既存のビンおよび追加された新規のビンに追加されるだけかを指定します。                          |
| fixed_width_name_extension | string  | デフォルトの拡張子は _BIN です。   |
| fixed_width_add_as         | Suffix<br>Prefix                                    | 拡張子をフィールド名の最後に追加するか（Suffix）、または先頭に追加するか（Prefix）を指定します。デフォルトの拡張子は income_BIN です。                             |
| fixed_bin_method           | Width<br>Count                                      |   |
| fixed_bin_count            | integer   | 新規フィールドの固定幅ビン（カテゴリ）数を決定するのに使用する整数を指定します。  |
| fixed_bin_width            | real  | ビンの幅を算出するために使用する値（整数または実数）。   |
| equal_count_name_extension | string  | デフォルトの拡張子は _TILE です。  |
| equal_count_add_as         | Suffix<br>Prefix                                    | 標準の分位を使って生成されるフィールドに対して使用される拡張子が、Suffix（接頭辞）か Prefix（接尾辞）かを指定します。デフォルトの拡張子は、_TILE に N を付けたものになります。N は分位数です。 |
| tile4                      | フラグ型  | それぞれが 25 % のケースを含む、4 分位のビンを生成します。   |
| tile5                      | フラグ型  | 5 つの 5 分位ビンを生成します。  |
| tile10                     | フラグ型  | 10 個の十分位（デシル）ビンを生成します。  |
| tile20                     | フラグ型  | 20 個の二十分位ビンを生成します。  |
| tile100                    | フラグ型  | 100 個の百分位（パーセンタイル）ビンを生成します。   |
| use_custom_tile            | フラグ型  |   |
| custom_tile_name_extension | string  | デフォルトの拡張子は _TILEN です。   |

## フィールド設定ノードのプロパティ

| binningnodeプロパティ               | データ型                      | プロパティの説明  |
|--------------------------------|---------------------------|---|
| custom_tile_add_as             | Suffix<br>Prefix          |   |
| custom_tile                    | integer                   |   |
| equal_count_method             | RecordCount<br>ValueSum   | RecordCount の方法は、同じ数のレコードを各ビンに割り当てます。一方、ValueSum では、各ビンの値の合計が同じになるようにレコードを割り当てます。 |
| tied_values_method             | Next<br>Current<br>Random | 可否同数の値のデータに配置されるビンを指定。  |
| rank_order                     | Ascending<br>Descending   | このプロパティには、Ascending (もっとも小さい値が 1 となる) または Descending (もっとも大きい値が 1 となる) が含まれます。    |
| rank_add_as                    | Suffix<br>Prefix          | このオプションは、ランク、ランクの比率、およびランクのパーセンテージに適用されます。  |
| rank                           | フラグ型                      |   |
| rank_name_extension            | string                    | デフォルトの拡張子は _RANK です。  |
| rank_fractional                | フラグ型                      | 新規フィールドの値が、ランクを非欠損ケースの重みの合計で除算した値になるように、ケースをランク付けします。ランクの比率は 0 - 1 の範囲の値になります。    |
| rank_fractional_name_extension | string                    | デフォルトの拡張子は _F_RANK です。  |
| rank_pct                       | フラグ型                      | 各ランクが、有効な値を持つレコード数で除算された後、100 倍されます。ランクのパーセンテージは、1 - 100 の範囲の値になります。              |
| rank_pct_name_extension        | string                    | デフォルトの拡張子は _P_RANK です。  |
| sdev_name_extension            | string                    |   |
| sdev_add_as                    | Suffix<br>Prefix          |   |
| sdev_count                     | One<br>Two<br>Three       |   |
| optimal_name_extension         | string                    | デフォルトの拡張子は _OPTIMAL です。   |
| optimal_add_as                 | Suffix<br>Prefix          |   |
| optimal_supervisor_field       | フィールド                     | データ分割のために選択されたフィールドが関係する監督フィールドとして選ばれたフィールド。                                      |

| binningnode プロパティ           | データ型                   | プロパティの説明                            |
|-----------------------------|------------------------|-------------------------------------|
| optimal_merge_bins          | フラグ型                   | ケース度数が小さいビンをより大きな隣接ビンに追加することを指定します。 |
| optimal_small_bin_threshold | integer                |                                     |
| optimal_pre_bin             | フラグ型                   | データセットの事前データ分割を実行することを示します。         |
| optimal_max_bins            | integer                | 過度に多数のビンを作成しないように、上限を指定します。         |
| optimal_lower_end_point     | Inclusive<br>Exclusive |                                     |
| optimal_first_bin           | Unbounded<br>Bounded   |                                     |
| optimal_last_bin            | Unbounded<br>Bounded   |                                     |

## derivenode のプロパティ



フィールド作成ノードで、1 つまたは複数の既存フィールドから、データ値を変更するか、新しいフィールドを作成します。これで、タイプ式、フラグ、名義、ステート、カウント、および条件式の各フィールドが作成されます。 [詳細は、4 章 フィールド作成ノード in IBM SPSS Modeler 15 入力ノード、プロセスノード、出力ノード を参照してください。](#)

### 例

```
# Create and configure a Flag Derive field node
create divenode
rename derive:derivenode as "Flag"
set Flag:derivenode.new_name = "DrugX_Flag"
set Flag:derivenode.result_type = Flag
set Flag:derivenode.flag_true = 1
set Flag:derivenode.flag_false = 0
set Flag:derivenode.flag_expr = "Drug = X"
# Create and configure a Conditional Derive field node
create divenode
rename derive:derivenode as "Conditional"
set Conditional:derivenode.result_type = Conditional
set Conditional:derivenode.cond_if_cond = "@OFFSET('\Age\', 1) = '\Age'"
set Conditional:derivenode.cond_then_expr = "(!@OFFSET('\Age\', 1) = '\Age') >< @INDEX"
```

## フィールド設定ノードのプロパティ

set Conditional:derivenode.cond\_else\_expr = "\Age\"

| derivenodeプロパティ       | データ型  | プロパティの説明  |
|-----------------------|---|---|
| new_name              | string  | 新しいフィールド名。  |
| mode                  | Single<br>Multiple                                      | 1つのフィールドか (Single)、または複数フィールドか (Multiple) を指定します。   |
| fields                | [フィールド<br>フィールド<br>フィールド]                               | 複数フィールドを選択する場合にだけ、Multiple モードで使用。  |
| name_extension        | string  | 新しいフィールド名に使用する拡張子を指定します。  |
| add_as                | Suffix<br>Prefix  | 拡張子をフィールド名の Prefix (先頭、接頭辞)、または Suffix (最後、接尾辞) として追加します。   |
| result_type           | Formula<br>Flag<br>Set<br>State<br>Count<br>Conditional | 作成可能な新しいフィールドの 6 つの種類。  |
| formula_expr          | string  | フィールド作成ノードの新しいフィールド値を計算する式。   |
| flag_expr             | string  |   |
| flag_true             | string  |   |
| flag_false            | string  |   |
| set_default           | string  |   |
| set_value_cond        | string  | 特定の値に関連付けられた条件を提供するように構造化プロパティ。<br>使用フォーマット：<br>set :derivenode.<br>set_value_cond.<br>Retired = 'age > 65' |
| state_on_val          | string  | オン (On) の条件を満たす場合の新規フィールドの値を指定します。  |
| state_off_val         | string  | オフ (Off) の条件を満たす場合の新規フィールドの値を指定します。   |
| state_on_expression   | string  |   |
| state_off_expression  | string  |   |
| state_initial         | On<br>Off   | 各レコードで新しいフィールドの初期値として On または Off を割り当てます。この値は、それぞれの条件が満たされるごとに変化します。  |
| count_initial_val     | string  |   |
| count_inc_condition   | string  |   |
| count_inc_expression  | string  |   |
| count_reset_condition | string  |   |

| derivenodeプロパティ | データ型   | プロパティの説明 |
|-----------------|--------|----------|
| cond_if_cond    | string |          |
| cond_then_expr  | string |          |
| cond_else_expr  | string |          |

## ensemblenode のプロパティ



アンサンブル ノードでは、2 つまたはそれ以上のモデル ナゲットを組み合わせ、1 つのモデルよりもより正確な予測を取得します。詳細は、4 章 [アンサンブル ノード in IBM SPSS Modeler 15](#) 入力ノード、プロセス ノード、出力ノード を参照してください。

### 例

```
# Create and configure an Ensemble node
# Use this node with the models in demos\streams\pm_binaryclassifier.str
create ensemblenode
set :ensemblenode.ensemble_target_field = response
set :ensemblenode.filter_individual_model_output = false
set :ensemblenode.flag_ensemble_method = ConfidenceWeightedVoting
set :ensemblenode.flag_voting_tie_selection = HighestConfidence
```

| ensemblenodeプロパティ              | データ型  | プロパティの説明   |
|--------------------------------|---|--|
| ensemble_target_field          | フィールド   | アンサンブルで使用されるすべてのモデルの対象フィールドを指定します。                                   |
| filter_individual_model_output | フラグ型  | 個々のモデルのスコアリング結果を抑制するかどうかを指定します。                                      |
| flag_ensemble_method           | Voting<br>ConfidenceWeighted-Voting<br>RawPropensity-WeightedVoting<br>AdjustedPropensity-WeightedVoting<br>HighestConfidence<br>AverageRawPropensity<br>AverageAdjusted-Propensity | アンサンブル スコアを決定するために使用する方法を指定します。この設定は、選択された対象がフラグ型フィールドである場合のみ適用されます。 |
| set_ensemble_method            | Voting<br>ConfidenceWeighted-Voting<br>HighestConfidence  | アンサンブル スコアを決定するために使用する方法を指定します。この設定は、選択された対象が名義型フィールドである場合のみ適用されます。  |

## フィールド設定ノードのプロパティ

| ensembledenodeプロパティ       | データ型   | プロパティの説明   |
|---------------------------|--|--|
| flag_voting_tie_selection | Random<br>HighestConfidence<br>RawPropensity<br>AdjustedPropensity | 票決方法が選択された場合、可否同数の解決方法を指定しますこの設定は、選択された対象がフラグ型フィールドである場合にのみ適用されます。                 |
| set_voting_tie_selection  | Random<br>HighestConfidence  | 票決方法が選択された場合、可否同数の解決方法を指定しますこの設定は、選択された対象が名義型フィールドである場合にのみ適用されます。                  |
| calculate_standard_error  | フラグ型   | 対象フィールドが連続型の場合、標準誤差の計算がデフォルトで実施され、測定された値または推定された値と真の値との差異を計算し、それらの推定がどれほど近いかを示します。 |

## fillernode のプロパティ



置換ノードで、フィールド値の置換やストレージの変更を行います。[@BLANK\(@FIELD\)](#) のような、CLEM 条件に基づいて値を置換することができます。また、すべての空白値やヌル値を特定の値に置換することもできます。置換ノードは、データ型ノードと一緒に使用される場合が多く、欠損値の置き換えが行われます。詳細は、[4 章 置換ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

## 例

```
create fillernode
set :fillernode.fields = ['Age']
set :fillernode.replace_mode = Always
set :fillernode.condition = "(\Age\ > 60) and (\Sex\ = \M\)"
set :fillernode.replace_with = "\old man\"
```

| fillernodeプロパティ | データ型   | プロパティの説明  |
|-----------------|--|---|
| fields          | [フィールド<br>フィールド<br>フィールド]                              | 検査されて置換される値のデータセットのフィールド群。                      |
| replace_mode    | Always<br>Conditional<br>Blank<br>Null<br>BlankAndNull | すべての値、空白値、またはヌル値を置換できます。または、指定した条件に基づいて、置換できます。 |
| condition       | string   |   |
| replace_with    | string   |   |

## filternode のプロパティ



フィルタ ノードで、1 つの入力ノードから他の 1 つの入力ノードへ、フィールドをフィルタリング（破棄）し、フィールド名を変更し、また、フィールドを関連付けます。詳細は、4 章 フィールドのフィルタリングまたは名前の変更 in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード を参照してください。

### 例

```
create filternode
set :filternode.default_include = True
set :filternode.new_name.'Drug' = 'Chemical'
set :filternode.include.'Drug' = off
```

**default\_include プロパティを使用** :default\_include プロパティの値を設定しても、すべてのフィールドが自動的に取り込まれたり除外されたりするわけではありません。単に、現在の選択フィールドに対するデフォルトを決めるだけです。これは、[フィルタ ノード] ダイアログ ボックスで [デフォルトでフィールドを含める] をクリックすることと、機能的に同じです。たとえば、次のスクリプトを実行すると想定します。

```
set Filter.default_include=False
# Include only fields in the list
for f in Age Sex
  set Filter.include.^f=True
endfor
```

これにより、Age（年齢）フィールドと Sex（性別）フィールドがノードを通過し、その他はすべて除外されます。次に、同じスクリプトを再び実行しますが、2 つの異なるフィールドを指定します。

```
set Filter.default_include=False
# Include only fields in the list
for f in BP Na
  set Filter.include.^f=True
endfor
```

これにより、さらに 2 つのフィールドがフィルタに追加されたので、合計 4 フィールド（Age（年齢）、Sex（性別）、BP（血圧）、Na（ナトリウム値））がフィルタを通過します。つまり、default\_include の値を False にリセットしても、すべてのフィールドが自動的にリセットされるわけではありません。

その代わりに、スクリプトを使用するか [フィルタ ノード] ダイアログ ボックス内で default\_include を True にこの時点で変更すると、動作が反対になり、上記の 4 フィールドは上記の 4 フィールドは除外されます。



[フィルタ ノード] ダイアログ ボックス内のコントロールで実験することが、この相互関係を理解するうえで役に立ちます。

| filternode プロパティ | データ型   | プロパティの説明  |
|------------------|--------|---|
| default_include  | フラグ型   | デフォルトの処理としてフィールドを通過させるかフィルタをかけるかの指定をするキー プロパティ。<br><b>NODE.include.FIELDNAME</b><br>一例を以下に挙げます。<br><b>set mynode:filternode.default_include = false</b><br>このプロパティを設定しても、すべてのフィールドが自動的に取り込まれたり除外されたりするわけではありません。選択したフィールドが、デフォルトでは取り込まれるか除外されるかを決めるだけです。詳細は、下の例を参照してください。 |
| include          | フラグ型   | フィールドを取り込むか除外するかのキー プロパティ。<br>使用フォーマット：<br><b>NODE.include.FIELDNAME</b><br>一例を以下に挙げます。<br><b>set mynode:filternode.include.Age = false</b>   |
| new_name         | string | 一例を以下に挙げます。<br><b>set mynode:filternode.new_name.Age = "age"</b>  |

## historynode のプロパティ



時系列ノードにより、以前レコードのフィールドのデータを含む、新規フィールドが作成されます。時系列ノードは、多くの場合、時系列データなどの継続的なデータに使用されます。時系列ノードを使用する前に、ソート ノードを使用して、データをソートしておくこともできます。詳細は、4 章 [時系列ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

### 例

```
create historynode
set :historynode.fields = ['Drug']
set :historynode.offset = 1
set :historynode.span = 3
set :historynode.unavailable = Discard
```

```
set :historynode.fill_with = "undef"
```

| historynode プロパティ | データ型                     | プロパティの説明  |
|-------------------|--------------------------|---|
| fields            | [フィールド<br>フィールド フィールド]   | 履歴の対象となるフィールド。  |
| offset            | number                   | 時系列フィールド値を抽出する最新レコードが、現在のレコードのいくつ前にあるかを指定します。                           |
| span              | number                   | 値を抽出する元になるレコードの前にあるレコード数を指定します。   |
| unavailable       | Discard<br>Leave<br>Fill | 時系列として使用する前のレコードがないデータセットの先頭の数レコードを通常は指しますが、その時系列値がないレコードの取り扱い方法を指定します。 |
| fill_with         | String<br>Number         | 時系列値が利用できないレコードを充填するのに使用する値 (Number) または文字列 (String) を指定します。            |

## partitionnode のプロパティ



データ区分ノードで、モデル構築の学習、テスト、および検証の各ステージ用に、データを独立したサブセットに分割するデータ区分フィールドが生成されます。詳細は、4 章 [データ区分ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

### 例

```
create partitionnode
set :partitionnode.create_validation = True
set :partitionnode.training_size = 33
set :partitionnode.testing_size = 33
set :partitionnode.validation_size = 33
set :partitionnode.set_random_seed = True
set :partitionnode.random_seed = "123"
set :partitionnode.value_mode = System
```

| partitionnode プロパティ | データ型    | プロパティの説明                          |
|---------------------|---------|-----------------------------------|
| new_name            | string  | ノードにより生成されたデータ区分フィールドの名前です。       |
| create_validation   | フラグ型    | 検証用のデータ区分を作成するかどうかを指定します。         |
| training_size       | integer | 学習用区分に割り当てるレコード数のパーセンテージ (0-100)。 |

## フィールド設定ノードのプロパティ

| partitionnodeプロパティ    | データ型                              | プロパティの説明   |
|-----------------------|-----------------------------------|--|
| testing_size          | integer                           | テスト用区分に割り当てるレコード数のパーセンテージ (0-100)。   |
| validation_size       | integer                           | 検証用区分に割り当てるレコード数のパーセンテージ (0-100)。検証用データ区分を生成しない場合は無視されます。  |
| training_label        | string                            | 学習用データ区分のラベル。  |
| testing_label         | string                            | テスト用データ区分のラベル。   |
| validation_label      | string                            | 検証用データ区分のラベル。検証用データ区分を生成しない場合は無視されます。  |
| value_mode            | System<br>SystemAndLabel<br>Label | データ中の各データ区分を表すために使用される値を指定します。たとえば、学習用サンプルは、システム整数 1、ラベル Training、またはこの 2 つを組み合わせた 1_Training のように表されます。   |
| set_random_seed       | ブール                               | ユーザー指定のランダム シードを使用するかどうかを指定します。  |
| random_seed           | integer                           | ユーザー定義のランダム シードの値。この値が使用されるようにするには、set_random_seed を True に設定する必要があります。                                    |
| enable_sql_generation | ブール                               | SQL プッシュバックを使用してレコードをデータ区分に割り当てるかどうかを指定します。  |
| unique_field          |                                   | レコードが無作為で繰り返し可能な方法でデータ区分に割り当てるよう、入力フィールドを指定します。この値が使用されるようにするには、enable_sql_generation を True に設定する必要があります。 |

## reclassifynode のプロパティ



データ分類ノードにより、あるカテゴリ値のセットが別のセットに変換されます。データ分類ノードは、カテゴリを再編成したり、分析用のデータをグループ化しなおす場合に役立ちます。詳細は、[4 章 データ分類ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

## 例

```
create reclassifynode
set :reclassifynode.mode = Multiple
set :reclassifynode.replace_field = true
set :reclassifynode.field = "Drug"
```

```

set :reclassifynode.new_name = "Chemical"
set :reclassifynode.fields = [Drug, BP]
set :reclassifynode.name_extension = "reclassified"
set :reclassifynode.add_as = Prefix
set :reclassifynode.reclassify.'drugA' = 'Yes'
set :reclassifynode.use_default = True
set :reclassifynode.default = "BrandX"
set :reclassifynode.pick_list = [BrandX, Placebo, Generic]

```

| reclassifynodeプロパティ | データ型                            | プロパティの説明   |
|---------------------|---------------------------------|--|
| mode                | Single<br>Multiple              | 1つのフィールドのカテゴリを再分類する場合、 <b>Single</b> を使用します。 <b>Multiple</b> (複数)を使用すると、一度に複数のフィールドを同時に変換できます。   |
| replace_field       | フラグ型                            |  |
| field               | string                          | Single モードでしか使用できません。  |
| new_name            | string                          | Single モードでしか使用できません。  |
| fields              | [フィールド1<br>フィールド2...<br>フィールドn] | Multiple モードでしか使用できません。  |
| name_extension      | string                          | Multiple モードでしか使用できません。  |
| add_as              | Suffix<br>Prefix                | Multiple モードでしか使用できません。  |
| reclassify          | string                          | フィールド値用構造化プロパティ。<br>使用フォーマット：<br><b>NODE.reclassify.<br/>OLD_VALUE</b><br>一例を以下に挙げます。<br><b>set :reclassifynode.reclassify.'drugB' = 'Yes'</b> |
| use_default         | フラグ型                            | デフォルト値を使用します。  |
| default             | string                          | デフォルト値を指定します。  |
| pick_list           | [文字列 文字列<br>... 文字列]            | ユーザーが、既知の新しい値をインポートしてテーブル内のドロップダウン リストをデータで埋めることができるようにします。<br>一例を以下に挙げます。<br><b>set :reclassify.pick_list = [fruit dairy cereals]</b>         |

## reordernode のプロパティ



フィールド順序ノードで、下流のフィールド表示に使用する順序を定義します。この順序は、テーブル、リスト、およびフィールドピッカーなど、さまざまな場所のフィールドの表示に適用されます。この操作は、さまざまなデータセットにおいて、特定のフィールドをより参照しやすくする場合に役立ちます。詳細は、[4 章 フィールド順序ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

### 例

```
create reordernode
set :reordernode.mode = Custom
set :reordernode.sort_by = Storage
set :reordernode.ascending = "false"
set :reordernode.start_fields = [Age Cholesterol]
set :reordernode.end_fields = [Drug]
```

| reordernodeプロパティ | データ型                             | プロパティの説明                            |
|------------------|----------------------------------|-------------------------------------|
| mode             | Custom<br>Auto                   | 値を自動的に並び替えたり、ユーザー指定の順序を指定することができます。 |
| sort_by          | Name<br>Type<br>Storage          |                                     |
| ascending        | フラグ型                             |                                     |
| start_fields     | [フィールド1<br>フィールド2 ...<br>フィールドn] | 新規フィールドは、これらのフィールドの後に挿入されます。        |
| end_fields       | [フィールド1<br>フィールド2 ...<br>フィールドn] | 新規フィールドは、これらのフィールドの前に挿入されます。        |

## restructurenode のプロパティ



再構成ノードで、名義型またはグラフ型フィールドを、これから別のフィールドの値で埋めることができるフィールドのグループへ変換します。たとえば、credit、cash、および debit の値の payment type という名前のフィールドがある場合、3 つの新しいフィールド (credit、cash、debit) が作成されます。その各々には、実際の支払の値を含めることができます。詳細は、[4 章 再構成ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

## 例

```
create restructurenode
connect :typenode to :restructurenode
set :restructurenode.fields_from.Drug = ["drugA" "drugX"]
set :restructurenode.include_field_name = "True"
set :restructurenode.value_mode = "OtherFields"
set :restructurenode.value_fields = ["Age" "BP"]
```

| restructurenodeプロパティ | データ型                             | プロパティの説明   |
|----------------------|----------------------------------|--|
| fields_from          | [カテゴリ カテゴリ<br>カテゴリ カテゴリ ]<br>all | 例をあげると、次のようになります。<br><code>set :restructurenode.fields_from.Drug = [drugA drugB]</code> は <code>Drug_drugA</code> および <code>Drug_drugB</code> というフィールドを作成します。<br>特定フィールドのすべてのカテゴリを使用するには<br><code>set :restructurenode.fields_from.Drug = all</code> |
| include_field_name   | フラグ型                             | 再構成されるフィールド名に元のフィールド名を使用するかどうかを示します。   |
| value_mode           | OtherFields<br>Flags             | 再構成されるフィールドの値を指定するためのモードを示します。<br><code>OtherFields</code> を指定すると、使用するフィールドを指定する必要があります(下を参照)。 <code>Flags</code> を指定する場合、値は数値のフラグです。  |
| value_fields         | [フィールド<br>フィールド<br>フィールド]        | <code>value_mode</code> が <code>OtherFields</code> の場合は必須です。値のフィールドとして使用するフィールドを指定します。   |

## rfmanalysisnode のプロパティ



リーセンシ、フリクエンシ、マネタリー (RFM) の分析ノードを使用すると、最後に購入したのがどのくらい最近か (リーセンシ)、どのくらい頻繁に購入するか (フリクエンシ)、トランザクション全体でいくら消費したか (マネタリー) を検証することによって、最も良い顧客となると考えられるのはどの顧客かを量的に決定することができます。詳細は、[4 章 RFM 分析ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

## 例

```
create rfmanalysisnode
connect :rfmaggregatenode to :rfmanalysisnode
set :rfmanalysisnode.recency = Recency
set :rfmanalysisnode.frequency = Frequency
set :rfmanalysisnode.monetary = Monetary
```

## フィールド設定ノードのプロパティ

```
set :rfmanalysisnode.tied_values_method = Next
set :rfmanalysisnode.recalculate_bins = IfNecessary
set :rfmanalysisnode.recency_thresholds = [1, 500, 800, 1500, 2000, 2500]
```

| rfmanalysisnode プロパティ | データ型                             | プロパティの説明  |
|-----------------------|----------------------------------|---|
| recency               | フィールド                            | リーセンシ フィールドを指定します。このフィールドは日付、タイムスタンプまたは単純な数値です。   |
| frequency             | フィールド                            | フリクエンシ フィールドを指定します。   |
| monetary              | フィールド                            | マネタリー フィールドを指定します。  |
| recency_bins          | integer                          | 生成されるリーセンシ ビンの数を指定します。  |
| recency_weight        | number                           | リーセンシ データに適用される重みを指定します。デフォルトは 100 です。  |
| frequency_bins        | integer                          | 生成されるフリクエンシ ビンの数を指定します。   |
| frequency_weight      | number                           | フリクエンシ データに適用される重みを指定します。デフォルトは 10 です。  |
| monetary_bins         | integer                          | 生成されるマネタリー ビンの数を指定します。  |
| monetary_weight       | number                           | マネタリー データに適用される重みを指定します。デフォルトは 1 です。  |
| tied_values_method    | Next<br>Current                  | 可否同数の値のデータに配置されるビンを選択。  |
| recalculate_bins      | Always<br>IfNecessary            |   |
| add_outliers          | フラグ型                             | recalculate_bins が IfNecessary に設定されている場合使用できます。設定されると、下限のビンの下にあるレコードが下限のビンに追加され、上限のビンの上にあるレコードが上限のビンに追加されます。   |
| binned_field          | Recency<br>Frequency<br>Monetary |   |
| recency_thresholds    | 値 値                              | recalculate_bins が Always に設定されている場合使用できます。リーセンシ ビンの上限および下限の閾値を指定します。あるビンの上限の閾値が次のビンの下限の閾値として使用されます。たとえば、[10 30 60] は、最初のビンに 10 および 30 の上限および下限の閾値があり、2 番目のビンには 30 および 60 の閾値があると定義します。 |

| rfanalysisnode プロパティ | データ型 | プロパティの説明                                    |
|----------------------|------|---|
| frequency_thresholds | 値 値  | recalculate_bins が Always に設定されている場合使用できます。 |
| monetary_thresholds  | 値 値  | recalculate_bins が Always に設定されている場合使用できます。 |

## settoflagnode のプロパティ



フラグ設定ノードで、1 つ以上の名義型フィールドに定義されたカテゴリ値に基づいた、複数のフラグ型フィールドが派生します。詳細は、4 章 フラグ設定ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード を参照してください。

### 例

```
create settoflagnode
connect :typenode to :settoflag
set :settoflagnode.fields_from.Drug = ["drugA" "drugX"]
set :settoflagnode.true_value = "1"
set :settoflagnode.false_value = "0"
set :settoflagnode.use_extension = "True"
set :settoflagnode.extension = "Drug_Flag"
set :settoflagnode.add_as = Suffix
set :settoflagnode.aggregate = True
set :settoflagnode.keys = ['Cholesterol']
```

| settoflagnode プロパティ | データ型                        | プロパティの説明   |
|---------------------|-----------------------------|--|
| fields_from         | [カテゴリ カテゴリ<br>カテゴリ ]<br>all | 例をあげると、次のようになります。<br><code>set :settoflagnode.fields_from.Drug = [drugA drugB]</code> は Drug_drugA および Drug_drugB というフラグ型フィールドを作成します。<br>特定フィールドのすべてのカテゴリを使用するには<br><code>set :settoflagnode.fields_from.Drug = all</code> |
| true_value          | string                      | フラグを設定するときにノードが使用する真 (true) の値を指定します。デフォルトは T です。  |
| false_value         | string                      | フラグを設定するときにノードが使用する偽 (false) の値を指定します。デフォルトは F です。   |
| use_extension       | フラグ型                        | 新規フラグ型フィールドの接尾辞または接頭辞として、拡張子を使用します。  |
| extension           | string                      |  |



| settoflagnode プロパティ | データ型                      | プロパティの説明  |
|---------------------|---------------------------|---|
| add_as              | Suffix<br>Prefix          | 拡張子が接尾辞 (Suffix) または接頭辞 (Prefix) として追加されることを指定します。                                    |
| aggregate           | フラグ型                      | キー フィールドに基づいてレコードをグループ化します。真 (true) に設定されたレコードが 1 つでもあると、グループ内のすべてのフラグ型フィールドが有効になります。 |
| keys                | [フィールド<br>フィールド<br>フィールド] | キー フィールド。   |

## statisticstransformnode プロパティ



Statistics 変換ノードは、IBM® SPSS® Modeler のデータ ソースに対する IBM® SPSS® Statistics シンタックス コマンドの選択を行います。このノードは、ライセンスが与えられた SPSS Statistics のコピーが必要です。詳細は、[8 章 Statistics 変換ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

このノードのプロパティについては、「[statisticstransformnode プロパティ](#)」( p. 343 ) に記載されています。

## timeintervalsnode のプロパティ



時間区分ノードで、時系列データのモデル作成用に区分を指定し、必要に応じてラベルを作成します。値の間隔が均等に空けられていない場合は、レコード間に一律の間隔をとる必要に応じて、値を充填したり集計したりできます。詳細は、[4 章 時間区分ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

### 例

```
create timeintervalsnode
set :timeintervalsnode.interval_type=SecondsPerDay
set :timeintervalsnode.days_per_week=4
set :timeintervalsnode.week_begins_on=Tuesday
set :timeintervalsnode.hours_per_day=10
set :timeintervalsnode.day_begins_hour=7
set :timeintervalsnode.day_begins_minute=5
set :timeintervalsnode.day_begins_second=17
set :timeintervalsnode.mode=Label
set :timeintervalsnode.year_start=2005
set :timeintervalsnode.month_start=January
set :timeintervalsnode.day_start=4
```

```

set :timeintervalsnode.pad.AGE=MeanOfRecentPoints
set :timeintervalsnode.agg_mode=Specify
set :timeintervalsnode.agg_set_default=Last

```

| timeintervalsnodeプロパティ | データ型   | プロパティの説明   |
|------------------------|--|--|
| interval_type          | None<br>Periods<br>CyclicPeriods<br>Years<br>Quarters<br>Months<br>DaysPerWeek<br>DaysNonPeriodic<br>HoursPerDay<br>HoursNonPeriodic<br>MinutesPerDay<br>MinutesNonPeriodic<br>SecondsPerDay<br>SecondsNonPeriodic |  |
| mode                   | Label<br>Create  | レコードに連続してラベルを付ける (Label) か、または指定された日付、タイムスタンプ、または時間フィールドに基づいて系列を構築するか (Create) を指定します。 |
| field                  | フィールド  | データから系列を構築する場合は、各レコードの日付または時刻を示すフィールドを指定します。   |
| period_start           | integer  | 期間または循環する期間の開始期間を指定します。  |
| cycle_start            | integer  | 循環する期間の開始サイクル。   |
| year_start             | integer  | 適用可能な区分タイプの、最初の区分が入る年。   |
| quarter_start          | integer  | 適用可能な区分タイプの、最初の区分が入る四半期。   |
| month_start            | January<br>February<br>March<br>April<br>May<br>June<br>July<br>August<br>September<br>October<br>November<br>December   |  |
| day_start              | integer  |  |
| hour_start             | integer  |  |
| minute_start           | integer  |  |
| second_start           | integer  |  |

## フィールド設定ノードのプロパティ

| timeintervalsnode プロパティ | データ型   | プロパティの説明  |
|-------------------------|--|---|
| periods_per_cycle       | integer  | 循環する期間の、各サイクル内の期間数。   |
| fiscal_year_begins      | January<br>February<br>March<br>April<br>May<br>June<br>July<br>August<br>September<br>October<br>November<br>December | 四半期単位の区分の場合、会計年度が始まる月を指定します。  |
| week_begins_on          | Sunday<br>Monday<br>Tuesday<br>Wednesday<br>Thursday<br>Friday<br>Saturday<br>Sunday                                   | 定期的な区分（週当たりの日数、日当たりの時間数、日当たりの分数、日当たりの秒数）の、週が始まる曜日を指定します。  |
| day_begins_hour         | integer  | 定期的な区分（日当たりの時間数、日当たりの分数、日当たりの秒数）の、日が始まる時間を指定します。day_begins_minute と day_begins_second を組み合わせて 8:05:01 のように、正確な時刻を指定できます。下の使用例を参照してください。 |
| day_begins_minute       | integer  | 定期的な区分（日当たりの時間数、日当たりの分数、日当たりの秒数）の、日が始まる時間の分を指定します（たとえば 8:05 の 5）。   |
| day_begins_second       | integer  | 定期的な区分（日当たりの時間数、日当たりの分数、日当たりの秒数）の、日が始まる時間の秒を指定します（たとえば 08:05:17 の 17）。  |
| days_per_week           | integer  | 定期的な区分（週当たりの日数、日当たりの時間数、日当たりの分数、日当たりの秒数）の、週当たりの日数を指定します。  |
| hours_per_day           | integer  | 定期的な区分（日当たりの時間数、日当たりの分数、日当たりの秒数）の、1 日の時間数を指定します。  |

| timeintervalsnode プロパティ        | データ型  | プロパティの説明                                    |
|--------------------------------|---|---|
| interval_increment             | 1<br>2<br>3<br>4<br>5<br>6<br>10<br>15<br>20<br>30  | 日当たりの分数と日当たりの秒数について、各レコード用増分の分数または秒数を指定します。 |
| field_name_extension           | string  |   |
| field_name_extension_as_prefix | フラグ型  |   |
| date_format                    | "DDMMYY"<br>"MMDDYY"<br>"YYMMDD"<br>"YYYYMMDD"<br>"YYYYDDD"<br>DAY<br>MONTH<br>"DD-MM-YY"<br>"DD-MM-YYYY"<br>"MM-DD-YY"<br>"MM-DD-YYYY"<br>"DD-MON-YY"<br>"DD-MON-YYYY"<br>"YYYY-MM-DD"<br>"DD.MM.YY"<br>"DD.MM.YYYY"<br>"MM.DD.YY"<br>"MM.DD.YYYY"<br>"DD.MON.YY"<br>"DD.MON.YYYY"<br>"DD/MM/YY"<br>"DD/MM/YYYY"<br>"MM/DD/YY"<br>"MM/DD/YYYY"<br>"DD/MON/YY"<br>"DD/MON/YYYY"<br>MON YYYY<br>q Q YYYY<br>ww WK YYYY |   |

## フィールド設定ノードのプロパティ

| timeintervalsnodeプロパティ | データ型   | プロパティの説明   |
|------------------------|--|--|
| time_format            | "HHMMSS"<br>"HHMM"<br>"MMSS"<br>"HH:MM:SS"<br>"HH:MM"<br>"MM:SS"<br>"(H)H:(M)M:(S)S"<br>"(H)H:(M)M"<br>"(M)M:(S)S"<br>"HH.MM.SS"<br>"HH.MM"<br>"MM.SS"<br>"(H)H.(M)M.(S)S"<br>"(H)H.(M)M"<br>"(M)M.(S)S" |  |
| aggregate              | Mean<br>Sum<br>Mode<br>Min<br>Max<br>First<br>Last<br>TrueIfAnyTrue  | フィールドに対して集計方法を指定します (たとえば、 <b>aggregate.AGE=Mean</b> ) .             |
| pad                    | Blank<br>MeanOfRecentPoints<br>True<br>False   | フィールドを埋める方法を指定します (たとえば、 <b>pad.AGE=MeanOfRecentPoints</b> ) .       |
| agg_mode               | All<br>Specify   | 必要に応じてデフォルトの関数ですべてのフィールドを集計または充填するかどうかを指定します。または、使用するフィールドと関数を指定します。 |
| agg_range_default      | Mean<br>Sum<br>Mode<br>Min<br>Max  | 連続型フィールドを集計するときに使用するデフォルトの関数を指定します。                                  |
| agg_set_default        | Mode<br>First<br>Last  | 名義型フィールドを集計するときに使用するデフォルトの関数を指定します。                                  |
| agg_flag_default       | TrueIfAnyTrue<br>Mode<br>First<br>Last   |  |
| pad_range_default      | Blank<br>MeanOfRecentPoints  | 連続型フィールドをパディングするときに使用するデフォルトの関数を指定します。                               |
| pad_set_default        | Blank<br>MostRecentValue   |  |

| timeintervalsnode プロパティ   | データ型                   | プロパティの説明                     |
|---------------------------|------------------------|------------------------------|
| pad_flag_default          | Blank<br>True<br>False |                              |
| max_records_to_create     | integer                | 系列を充填するときに作成する最大レコード数を指定します。 |
| estimation_from_beginning | フラグ型                   |                              |
| estimation_to_end         | フラグ型                   |                              |
| estimation_start_offset   | integer                |                              |
| estimation_num_holdouts   | integer                |                              |
| create_future_records     | フラグ型                   |                              |
| num_future_records        | integer                |                              |
| create_future_field       | フラグ型                   |                              |
| future_field_name         | string                 |                              |

## transposenode のプロパティ



行列入替ノードで、レコードがフィールドになり、フィールドがレコードになるように、行内と列内のデータを交換します。詳細は、4 章 行列入替ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード を参照してください。

### 例

```
create transposenode
set :transposenode.transposed_names=Read
set :transposenode.read_from_field="TimeLabel"
set :transposenode.max_num_fields="1000"
set :transposenode.id_field_name="ID"
```

| transposenode プロパティ | データ型           | プロパティの説明   |
|---------------------|----------------|--|
| transposed_names    | Prefix<br>Read | 新しいフィールド名は、指定された接頭辞 (Prefix) に基づいて自動的に作成できます。または、既存のデータ内のフィールドからフィールド名を読み込むことができます (Read)。 |
| prefix              | string         |  |
| num_new_fields      | integer        | 接頭辞を使用する場合は、作成する新しいフィールドの最大数を指定します。  |
| read_from_field     | フィールド          | 名前が読み込まれるフィールド。これはインスタンス化されたフィールドである必要があります。そうでない場合は、ノードが実行されるときにエラーが発生します。                |

| transposenode プロパティ | データ型                        | プロパティの説明   |
|---------------------|-----------------------------|--|
| max_num_fields      | integer                     | フィールドから名前を読み込む場合は、異常に大量のフィールドを作成しないように、フィールド数の上限を指定します。  |
| transpose_type      | Numeric<br>String<br>Custom | デフォルトでは連続型のフィールドのみの行列が入れ替えられますが、代わりに、数値フィールドのカスタム（ユーザー設定）サブセットを選択またはすべての文字列フィールドを入れ替えることもできます。 |
| transpose_fields    | [フィールド フィールド<br>フィールド]      | Custom（ユーザー設定）オプションを使用するときに、行列を入れ替えるフィールドを指定します。   |
| id_field_name       | フィールド                       |  |

## typenode のプロパティ



データ型ノードで、フィールドのメタデータとプロパティを指定します。たとえば、各フィールドに、測定レベル（連続型、名義型、順序型、またはフラグ）を指定し、欠損値とシステムヌルの処理のためのオプションを設定し、モデル作成の目的に対するフィールドの役割を設定し、フィールドと値のラベルを指定し、フィールドの値を指定します。詳細は、[4 章 データ型ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

### 例

```
create typenode
connect :variablefilenode to :typenode
set :typenode.check.'Cholesterol' = Coerce
set :typenode.direction.'Drug' = Input
set :typenode.type.K = Range
set :typenode.values.Drug = [drugA drugB drugC drugD drugX drugY drugZ]
set :typenode.null_missing.BP = false
set :typenode.whitespace_missing.BP = "false"
set :typenode.description.BP = "Blood Pressure"
set :typenode.value_labels.BP = [{HIGH 'High Blood Pressure'}{NORMAL 'normal blood pressure'}]
set :typenode.display_places.K = 5
set :typenode.export_places.K = 2
set :typenode.grouping_symbol.Drug = None
set :typenode.column_width.Cholesterol = 25
set :typenode.justify.Cholesterol = Right
```

ある種の場合、ほかのノードが正しく機能するように、フラグ設定ノードの `fields from` プロパティのように、データ型ノードを完全にインスタンス化する必要がある場合があります。フィールドをインスタンス化するには、次のように、テーブル ノードを接続して実行するだけです。

```
create tablenode
connect :typenode to :tablenode
execute :tablenode
delete :tablenode
```

| typenodeプロパティ | データ型   | プロパティの説明  |
|---------------|--|---|
| direction     | Input<br>Target<br>Both<br>None<br>Partition<br>Split<br>Frequency<br>RecordID | フィールドの役割のキープロパティ。<br>使用フォーマット：<br>NODE.direction.FIELDNAME<br>注：値 In および Out は廃止されています。今後のリリースではサポートが中断される場合があります。  |
| type          | Range<br>Flag<br>Set<br>Typeless<br>Discrete<br>Ordered Set<br>Default         | フィールドのデータ型。type を Default に設定すると values パラメータ設定をクリアします。value_mode の値が Specify の場合、Read にリセットします。value_mode が Pass または Read に設定される場合、type を設定しても value_mode には影響ありません。<br>使用フォーマット：<br>NODE.type.FIELDNAME |
| storage       | Unknown<br>String<br>Integer<br>Real<br>Time<br>Date<br>Timestamp              | フィールドのストレージタイプ用読み込み専用キー プロパティ。<br>使用フォーマット：<br>NODE.storage.FIELDNAME   |
| check         | None<br>Nullify<br>Coerce<br>Discard<br>Warn<br>Abort                          | フィールドタイプと範囲の検査用のキー プロパティ。<br>使用フォーマット：<br>NODE.check.FIELDNAME  |



## フィールド設定ノードのプロパティ

| typenodeプロパティ  | データ型  | プロパティの説明  |
|----------------|---|---|
| values         | [値 値]                                       | 連続型フィールドの場合、最初の値が最小値で最後の値が最大値になります。名義型フィールドの場合、すべての値を指定します。フラグ型の場合、最初の値が false (偽) を、最後の値が true (真) を表します。このプロパティを設定すると、 <b>value_mode</b> プロパティの値が自動的に <b>Specify</b> に設定されます。<br>使用フォーマット：<br>NODE.values.FIELDNAME |
| value_mode     | Read<br>Pass<br>Read+<br>Current<br>Specify | 値の設定方法を決定します。このプロパティに <b>Specify</b> を直接には設定できないことに注意してください。特定の値を使用するには、 <b>values</b> プロパティを設定します。<br>使用フォーマット：<br>NODE.value_mode.FIELDNAME   |
| extend_values  | フラグ型  | <b>value_mode</b> に <b>Read</b> が設定された場合に適用されます。新しく読み込んだ値を、フィールドの既存の値に追加する場合は、 <b>T</b> を設定します。新しく読み込んだ値を優先して、既存の値を破棄する場合は、 <b>F</b> を設定します。<br>使用フォーマット：<br>NODE.extend_values.FIELDNAME                             |
| enable_missing | フラグ型  | <b>T</b> を設定した場合、フィールドの欠損値の追跡が有効になります。<br>使用フォーマット：<br>NODE.enable_missing.FIELDNAME  |
| missing_values | [値 値 ...]                                   | 欠損データを示すデータ値を指定します。<br>使用フォーマット：<br>NODE.missing_values.FIELDNAME   |
| range_missing  | フラグ型  | フィールドに欠損値 (空白) の範囲が定義されているかどうかを指定します。   |
| missing_lower  | string                                      | <b>range_missing</b> が真 (true) の場合、欠損値範囲の下限值を指定します。   |
| missing_upper  | string                                      | <b>range_missing</b> が真 (true) の場合、欠損値範囲の上限値を指定します。   |
| null_missing   | フラグ型  | <b>T</b> を設定した場合、ヌル値 (ソフトウェアでは \$null\$ として表示される未定義値) は欠損値と見なされます。<br>使用フォーマット：<br>NODE.null_missing.FIELDNAME  |

| typenodeプロパティ      | データ型  | プロパティの説明  |
|--------------------|---|---|
| whitespace_missing | フラグ型  | T を設定した場合、空白類（スペース、タブ、および改行）だけを含む値が欠損値と見なされます。<br>使用フォーマット：<br>NODE.whitespace_missing.FIELDNAME  |
| description        | string  | フィールドの説明を指定します。   |
| value_labels       | [ {Value<br>LabelString}<br>{ Value<br>LabelString}<br>... ]  | 値のペアのためのラベルを指定します。<br>一例を以下に挙げます。<br>set :typenode.value_labels.'Drug'={ {drugA<br>label1} {drugB label2} }   |
| display_places     | integer   | フィールドが表示される時の小数部の桁数を設定します (REAL ストレージのフィールドにのみ適用)。-1 を設定すると、ストリームのデフォルトが使用されます。<br>使用フォーマット：<br>NODE.display_places.FIELDNAME   |
| export_places      | integer   | フィールドが表示される時の小数部の桁数を設定します (REAL ストレージのフィールドにのみ適用)。-1 を設定すると、ストリームのデフォルトが使用されます。<br>使用フォーマット：<br>NODE.export_places.FIELDNAME  |
| decimal_separator  | DEFAULT<br>PERIOD<br>COMMA  | フィールドの小数点記号を指定します (REAL ストレージのフィールドにのみ適用)。<br>使用フォーマット：<br>NODE.decimal_separator.FIELDNAME   |
| date_format        | "DDMMYY"<br>"MMDDYY"<br>"YYMMDD"<br>"YYYYMMDD"<br>"YYYYDDD"<br>DAY<br>MONTH<br>"DD-MM-YY"<br>"DD-MM-YYYY"<br>"MM-DD-YY"<br>"MM-DD-YYYY"<br>"DD-MON-YY"<br>"DD-MON-YYYY"<br>"YYYY-MM-DD"<br>"DD.MM.YY"<br>"DD.MM.YYYY"<br>"MM.DD.YY"<br>"MM.DD.YYYY"<br>"DD.MON.YY"<br>"DD.MON.YYYY"<br>"DD/MM/YY" | フィールドの日付フォーマットを設定します (DATE または TIMESTAMP ストレージのフィールドにのみ適用されます)。<br>使用フォーマット：<br>NODE.date_format.FIELDNAME<br>一例を以下に挙げます。<br>set :tablenode.date_format.'LaunchDate' = "DDMMYY" |

## フィールド設定ノードのプロパティ

| typenodeプロパティ     | データ型   | プロパティの説明  |
|-------------------|--|---|
|                   | "DD/MM/YYYY"<br>"MM/DD/YY"<br>"MM/DD/YYYY"<br>"DD/MON/YY"<br>"DD/MON/YYYY"<br>MON YYYY<br>q Q YYYY<br>ww WK YYYY   |   |
| time_format       | "HHMMSS"<br>"HHMM"<br>"MMSS"<br>"HH:MM:SS"<br>"HH:MM"<br>"MM:SS"<br>"(H)H:(M)M:(S)S"<br>"(H)H:(M)M"<br>"(M)M:(S)S"<br>"HH.MM.SS"<br>"HH.MM"<br>"MM.SS"<br>"(H)H.(M)M.(S)S"<br>"(H)H.(M)M"<br>"(M)M.(S)S" | フィールドの日付フォーマットを設定します (TIME または TIMESTAMP ストレージのフィールドにのみ適用されます)。<br>使用フォーマット :<br>NODE.time_format.FIELDNAME<br>一例を以下に挙げます。<br>set :tablenode.time_format.'BOF_enter' = "HHMMSS" |
| number_format     | DEFAULT<br>STANDARD<br>SCIENTIFIC<br>CURRENCY  | フィールドに数値の表示フォーマットを設定します。<br>使用フォーマット :<br>NODE.number_format.FIELDNAME  |
| standard_places   | integer  | フィールドが標準フォーマットで表示されるときに小数点以下の桁数を指定します。-1 を設定すると、ストリームのデフォルトが使用されます。既存の display_places スロットでもこの設定が変更されますが、現在は廃止されています。<br>使用フォーマット :<br>NODE.standard_places.FIELDNAME           |
| scientific_places | integer  | フィールドが科学系のフォーマットで表示されるときに小数点以下の桁数を設定します。-1 を設定すると、ストリームのデフォルトが使用されます。<br>使用フォーマット :<br>NODE.scientific_places.FIELDNAME   |
| currency_places   | integer  | フィールドが通貨のフォーマットで表示されるときにフィールドの小数点以下の桁数を設定します。-1 を設定すると、ストリームのデフォルトが使用されます。<br>使用フォーマット :<br>NODE.currency_places.FIELDNAME  |

| typenodeプロパティ   | データ型  | プロパティの説明   |
|-----------------|---|--|
| grouping_symbol | DEFAULT<br>NONE<br>LOCALE<br>PERIOD<br>COMMA<br>SPACE | フィールドにグループ化シンボルを設定します。<br>使用フォーマット：<br>NODE.grouping_symbol.FIELDNAME                            |
| column_width    | integer   | フィールドに列幅を設定します。-1<br>という値を指定すると、列幅は Auto<br>に設定されます。<br>使用フォーマット：<br>NODE.column_width.FIELDNAME |
| justify         | AUTO<br>CENTER<br>LEFT<br>RIGHT                       | フィールドに列調整を設定します。<br>使用フォーマット：<br>NODE.justify.FIELDNAME  |

# グラフ作成ノードのプロパティ

## グラフ作成ノードの共通のプロパティ

このセクションでは、グラフ作成ノードで使用できるプロパティについて、共通なプロパティとノードタイプ固有のプロパティも含めて説明します。

| グラフ作成ノードの共通プロパティ | データ型  | プロパティの説明   |
|------------------|---|--|
| title            | string                                      | タイトルを指定します。例:” This is a title.”   |
| caption          | string                                      | 解説を指定します。例:” This is a caption.”   |
| output_mode      | Screen<br>File                              | グラフ作成ノードからの出力が表示されるか、ファイルへ書き込まれるかを指定します。   |
| output_format    | BMP<br>JPEG<br>PNG<br>HTML<br>output (.cou) | 出力のタイプを指定します。出力可能なタイプは、各ノードに応じて変化します。  |
| full_filename    | string                                      | グラフ作成ノードから生成されたグラフの、出力先のパスとファイル名を指定します。  |
| use_graph_size   | フラグ型  | 下に説明する幅と高さのプロパティを使用してグラフのサイズが明示して設定されるかどうかを制御します。画面に出力されるグラフにだけ影響します。棒グラフノードには使用できません。 |
| graph_width      | number                                      | use_graph_size が True の場合、グラフの幅をピクセル数で指定します。   |
| graph_height     | number                                      | use_graph_size が True の場合、グラフの高さをピクセル数で指定します。  |

### メモ

**オプションフィールドの無効化：** 散布図のオーバーレイフィールドなどのオプションフィールドは、次の例のようにプロパティ値に "" (空文字列) を設定することにより、無効化することができます。

```
set :plotnode.color_field = ""
```

**色の指定：** タイトル、解説、背景、およびラベルの色は、ハッシュ記号 (#) で始まる 16進文字列で指定することができます。たとえば、グラフの背景を空色にするには、次の文を指定します。

```
set mygraph.graph_background="#87CEEB"
```

ここで、最初の 2 桁 87 は赤色の量を、次の 2 桁 CE は緑の量を、最後の 2 桁 EB は青の量を示します。各桁は、0 ~ 9 または A ~ F の範囲の値になります。これらの値を使って、赤-緑-青 (RGB) の色を指定します。

注： 色を RGB で指定する場合、フィールド ピッカーを使って正しい色コードを決定することができます。ピッカーを目的の色の上にかざせば、その色コードがツールヒントに表示されます。

## collectionnode のプロパティ



集計棒グラフ ノードで、他の数値フィールドの値に相対的な数値フィールドの値の棒グラフを表示します (集計棒グラフ ノードでは、ヒストグラムに似たグラフが作成されます)。集計棒グラフは、値が時間の経過とともに変化する変数やフィールドを表示する場合に役立ちます。3 次元グラフを使って、分布をカテゴリ別に表示するシンボル値軸を追加することもできます。詳細は、[5 章 集計棒グラフの \[プロット\] タブ in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

### 例

```
create collectionnode
position :collectionnode at ^posX ^posY
# "Plot" tab
set :collectionnode.three_D = True
set :collectionnode.collect_field = 'Drug'
set :collectionnode.over_field = 'Age'
set :collectionnode.by_field = 'BP'
set :collectionnode.operation = Sum
# "Overlay" section
set :collectionnode.color_field = 'Drug'
set :collectionnode.panel_field = 'Sex'
set :collectionnode.animation_field = ""
# "Options" tab
set :collectionnode.range_mode = Automatic
set :collectionnode.range_min = 1
set :collectionnode.range_max = 100
set :collectionnode.bins = ByNumber
set :collectionnode.num_bins = 10
```

```
set :collectionnode.bin_width = 5
```

| collectionnodeプロパティ | データ型                              | プロパティの説明                      |
|---------------------|-----------------------------------|-------------------------------|
| over_field          | フィールド                             |                               |
| over_label_auto     | フラグ型                              |                               |
| over_label          | string                            |                               |
| collect_field       | フィールド                             |                               |
| collect_label_auto  | フラグ型                              |                               |
| collect_label       | string                            |                               |
| three_D             | フラグ型                              |                               |
| by_field            | フィールド                             |                               |
| by_label_auto       | フラグ型                              |                               |
| by_label            | string                            |                               |
| operation           | Sum<br>Mean<br>Min<br>Max<br>SDev |                               |
| color_field         | string                            |                               |
| panel_field         | string                            |                               |
| animation_field     | string                            |                               |
| range_mode          | Automatic<br>UserDefined          |                               |
| range_min           | number                            |                               |
| range_max           | number                            |                               |
| bins                | ByNumber<br>ByWidth               |                               |
| num_bins            | number                            |                               |
| bin_width           | number                            |                               |
| use_grid            | フラグ型                              |                               |
| graph_background    | 色                                 | 標準のグラフ色は、このセクションの最初に説明されています。 |
| page_background     | 色                                 | 標準のグラフ色は、このセクションの最初に説明されています。 |

## distributionnode のプロパティ



棒グラフ ノードで、ローンの種類や性別など、シンボル値（カテゴリ）の出現頻度を表示します。通常、棒グラフ ノードを使用してデータの不均衡を表示しますが、そのデータはモデルの作成前にバランス ノードを使って修正できます。詳細は、5 章 [棒グラフ ノード in IBM SPSS Modeler 15 入力ノード、プロセスノード、出力ノード](#) を参照してください。

**例**

```
create distributionnode
# "Plot" tab
set :distributionnode.plot = Flags
set :distributionnode.x_field = 'Age'
set :distributionnode.color_field = 'Drug'
set :distributionnode.normalize = True
set :distributionnode.sort_mode = ByOccurence
set :distributionnode.use_proportional_scale = True
```

| distributionnode プロパティ | データ型                      | プロパティの説明      |
|------------------------|---------------------------|---------------|
| plot                   | SelectedFields<br>Flags   |               |
| x_field                | フィールド                     |               |
| color_field            | フィールド                     | オーバーレイ フィールド。 |
| normalize              | フラグ型                      |               |
| sort_mode              | ByOccurence<br>Alphabetic |               |
| use_proportional_scale | フラグ型                      |               |

**evaluationnode のプロパティ**

評価ノードは、予測モデルの評価と比較に用いられます。評価グラフで、モデルが特定の結果をどの程度予測するかを表示します。それによって、予測値と予測の信頼度に基づいたレコードがソートされます。そして、レコードが等サイズ（分位）のグループに分割され、各分位のビジネスに関する基準の値が、高い方から降順でプロットされます。プロットには、複数のモデルが異なる線で示されます。 [詳細は、5 章 評価ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノードを参照してください。](#)

**例**

```
create evaluationnode
position :evaluationnode at ^posX ^posY
# "Plot" tab
set :evaluationnode.chart_type = Gains
set :evaluationnode.cumulative = False
set :evaluationnode.field_detection_method = Name
set :evaluationnode.inc_baseline = True
set :evaluationnode.n_tile = Deciles
set :evaluationnode.style = Point
set :evaluationnode.point_type = Dot
set :evaluationnode.use_fixed_cost = True
set :evaluationnode.cost_value = 5.0
set :evaluationnode.cost_field = 'Na'
```



## グラフ作成ノードのプロパティ

```

set :evaluationnode.use_fixed_revenue = True
set :evaluationnode.revenue_value = 30.0
set :evaluationnode.revenue_field = 'Age'
set :evaluationnode.use_fixed_weight = True
set :evaluationnode.weight_value = 2.0
set :evaluationnode.weight_field = 'K'

```

| evaluationnodeプロパティ    | データ型   | プロパティの説明 |
|------------------------|--|----------|
| chart_type             | Gains<br>Response<br>Lift<br>Profit<br>ROI   |          |
| inc_baseline           | フラグ型   |          |
| field_detection_method | Metadata<br>Name   |          |
| use_fixed_cost         | フラグ型   |          |
| cost_value             | number   |          |
| cost_field             | string   |          |
| use_fixed_revenue      | フラグ型   |          |
| revenue_value          | number   |          |
| revenue_field          | string   |          |
| use_fixed_weight       | フラグ型   |          |
| weight_value           | number   |          |
| weight_field           | フィールド  |          |
| n_tile                 | Quartiles<br>Quintles<br>Deciles<br>Vingtiles<br>Percentiles<br>1000-tiles   |          |
| cumulative             | フラグ型   |          |
| style                  | Line<br>Point  |          |
| point_type             | Rectangle<br>Dot<br>Triangle<br>Hexagon<br>Plus<br>Pentagon<br>Star<br>BowTie<br>HorizontalDash<br>VerticalDash<br>IronCross<br>Factory<br>House<br>Cathedral<br>OnionDome |          |

| evaluationnodeプロパティ     | データ型   | プロパティの説明 |
|-------------------------|--|----------|
|                         | ConcaveTriangle<br>OblateGlobe<br>CatEye<br>FourSidedPillow<br>RoundRectangle<br>Fan |          |
| export_data             | フラグ型   |          |
| data_filename           | string   |          |
| delimiter               | string   |          |
| new_line                | フラグ型   |          |
| inc_field_names         | フラグ型   |          |
| inc_best_line           | フラグ型   |          |
| inc_business_rule       | フラグ型   |          |
| business_rule_condition | string   |          |
| plot_score_fields       | フラグ型   |          |
| score_fields            | [フィールド1 ...<br>フィールドN]   |          |
| target_field            | フィールド  |          |
| use_hit_condition       | フラグ型   |          |
| hit_condition           | string   |          |
| use_score_expression    | フラグ型   |          |
| score_expression        | string   |          |
| caption_auto            | フラグ型   |          |

## graphboardnode のプロパティ



グラフボード ノードでは、単一のノードにさまざまな種類のグラフを提供しています。このノードを使用して、検証するデータフィールドを選択肢、選択したデータに使用できるグラフを選択できます。選択したフィールドに適していないグラフの種類は、自動的に除外されます。詳細は、[5 章 グラフボード ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

注： グラフ タイプに有効でないプロパティを設定した場合（ヒストグラムに `y_field` を指定するなど）、そのプロパティは無視されます。

### 例

```
create graphboardnode
connect DRUG4n to :graphboardnode
set :graphboardnode.graph_type="Line"
set :graphboardnode.x_field = "K"
set :graphboardnode.y_field = "Na"
```

execute :graphboardnode

| graphboard プロパティ | データ型   | プロパティの説明                                  |
|------------------|--|---|
| graph_type       | 2DDotplot<br>3DArea<br>3DBar<br>3DDensity<br>3DHistogram<br>3DPie<br>3DScatterplot<br>Area<br>ArrowMap<br>Bar<br>BarCounts<br>BarCountsMap<br>BarMap<br>BinnedScatter<br>Boxplot<br>Bubble<br>ChoroplethMeans<br>ChoroplethMedians<br>ChoroplethSums<br>ChoroplethValues<br>ChoroplethCounts<br>CoordinateMap<br>CoordinateChoroplethMeans<br>CoordinateChoroplethMedians<br>CoordinateChoroplethSums<br>CoordinateChoroplethValues<br>CoordinateChoroplethCounts<br>Dotplot<br>Heatmap<br>HexBinScatter<br>Histogram<br>Line<br>LineChartMap<br>LineOverlayMap<br>Parallel<br>Path<br>Pie<br>PieCountMap<br>PieCounts<br>PieMap<br>PointsOverlayMap<br>PolygonOverlayMap<br>Ribbon<br>Scatterplot<br>SPLOM<br>Surface | グラフの種類を特定します。                             |
| x_field          | フィールド  | x 軸のカスタム (ユーザー設定) ラベルを指定します。ラベルでのみ使用できます。 |

| graphboard プロパティ | データ型   | プロパティの説明                                  |
|------------------|--------|---|
| y_field          | フィールド  | y 軸のカスタム (ユーザー設定) ラベルを指定します。ラベルでのみ使用できます。 |
| z_field          | フィールド  | 3 次元グラフの一部で使用します。                         |
| color_field      | フィールド  | ヒート マップで使用します。                            |
| size_field       | フィールド  | バブル プロットで使用します。                           |
| categories_field | フィールド  |   |
| values_field     | フィールド  |   |
| rows_field       | フィールド  |   |
| columns_field    | フィールド  |   |
| fields           | フィールド  |   |
| start_longitude  | フィールド  | 参照マップの矢印で使用します。                           |
| end_longitude    | フィールド  |   |
| start_latitude   | フィールド  |   |
| end_latitude     | フィールド  |   |
| data_key_field   | フィールド  | さまざまなマップで使用します。                           |
| panelrow_field   | string |   |
| panelcol_field   | string |   |
| animation_field  | string |   |
| longitude        | フィールド  | マップ上の座標で使用します。                            |
| latitude         | フィールド  |   |
| map_color_field  | フィールド  |   |

## histogramnode のプロパティ



ヒストグラム ノードでは、数値フィールドの値の出現頻度が示されます。多くの場合、ヒストグラム ノードは、操作やモデルの構築前にデータを調べるために使用されます。棒グラフ ノードと同様、ヒストグラム ノードにより、データ内の不均衡がしばしば明らかになります。詳細は、[5 章 ヒストグラムの \[プロット\] タブ in IBM SPSS Modeler 15 入力ノード、プロセスノード、出力ノード](#) を参照してください。

### 例

```
create histogramnode
position :histogramnode at ^posX ^posY
# "Plot" tab
set :histogramnode.field = 'Drug'
set :histogramnode.color_field = 'Drug'
set :histogramnode.panel_field = 'Sex'
```

```

set :histogramnode.animation_field = ""
#"Options" tab
set :histogramnode.range_mode = Automatic
set :histogramnode.range_min = 1.0
set :histogramnode.range_max = 100.0
set :histogramnode.num_bins = 10
set :histogramnode.bin_width = 10
set :histogramnode.normalize = True
set :histogramnode.separate_bands = False

```

| histogramnodeプロパティ | データ型                     | プロパティの説明                      |
|--------------------|--------------------------|-------------------------------|
| field              | フィールド                    |                               |
| color_field        | フィールド                    |                               |
| panel_field        | フィールド                    |                               |
| animation_field    | フィールド                    |                               |
| range_mode         | Automatic<br>UserDefined |                               |
| range_min          | number                   |                               |
| range_max          | number                   |                               |
| bins               | ByNumber<br>ByWidth      |                               |
| num_bins           | number                   |                               |
| bin_width          | number                   |                               |
| normalize          | フラグ型                     |                               |
| separate_bands     | フラグ型                     |                               |
| x_label_auto       | フラグ型                     |                               |
| x_label            | string                   |                               |
| y_label_auto       | フラグ型                     |                               |
| y_label            | string                   |                               |
| use_grid           | フラグ型                     |                               |
| graph_background   | 色                        | 標準のグラフ色は、このセクションの最初に説明されています。 |
| page_background    | 色                        | 標準のグラフ色は、このセクションの最初に説明されています。 |
| normal_curve       | フラグ型                     | 正規分布のカーブを出力に表示するかどうかを指定します。   |

## multiplotnode のプロパティ



線グラフ ノードでは、1 つの X フィールドに対して複数の Y フィールドを表示するプロットが作成されます。Y フィールドは色付きの線でプロットされ、それぞれ [スタイル] フィールドを [ライン] に、[X モード] フィールドを [ソート] に設定した散布図ノードに相当します。線グラフは、複数の変数の変動を長期にわたって調査するときに役立ちます。詳細は、[5 章 線グラフ ノード in IBM SPSS Modeler 15 入力ノード、プロセスノード、出力ノード](#) を参照してください。

### 例

```
create multiplotnode
# "Plot" tab
set :multiplotnode.x_field = 'Age'
set :multiplotnode.y_fields = ['Drug' 'BP']
set :multiplotnode.panel_field = 'Sex'
# "Overlay" section
set :multiplotnode.animation_field = ''
set :multiplotnode.tooltip = "test"
set :multiplotnode.normalize = True
set :multiplotnode.use_overlay_expr = False
set :multiplotnode.overlay_expression = "test"
set :multiplotnode.records_limit = 500
set :multiplotnode.if_over_limit = PlotSample
```

| multiplotnode プロパティ | データ型                              | プロパティの説明 |
|---------------------|-----------------------------------|----------|
| x_field             | フィールド                             |          |
| y_fields            | [フィールド<br>フィールド フィールド]            |          |
| panel_field         | フィールド                             |          |
| animation_field     | フィールド                             |          |
| normalize           | フラグ型                              |          |
| use_overlay_expr    | フラグ型                              |          |
| overlay_expression  | string                            |          |
| records_limit       | number                            |          |
| if_over_limit       | PlotBins<br>PlotSample<br>PlotAll |          |
| x_label_auto        | フラグ型                              |          |
| x_label             | string                            |          |
| y_label_auto        | フラグ型                              |          |
| y_label             | string                            |          |
| use_grid            | フラグ型                              |          |

| multiplotnode プロパティ | データ型 | プロパティの説明                      |
|---------------------|------|-------------------------------|
| graph_background    | 色    | 標準のグラフ色は、このセクションの最初に説明されています。 |
| page_background     | 色    | 標準のグラフ色は、このセクションの最初に説明されています。 |

## plotnode のプロパティ



散布図ノードで、数値フィールド間の関係が示されます。プロットは、点（散布図）または折れ線を使用して作成できます。詳細は、[5 章 散布図ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

### 例

```
create plotnode
# "Plot" tab
set :plotnode.three_D = True
set :plotnode.x_field = 'BP'
set :plotnode.y_field = 'Cholesterol'
set :plotnode.z_field = 'Drug'
# "Overlay" section
set :plotnode.color_field = 'Drug'
set :plotnode.size_field = 'Age'
set :plotnode.shape_field = ''
set :plotnode.panel_field = 'Sex'
set :plotnode.animation_field = 'BP'
set :plotnode.transp_field = ''
set :plotnode.style = Point
# "Output" tab
set :plotnode.output_mode =
set :plotnode.output_format = JPEG
set :plotnode.full_filename = "C:/Temp/Graph_Output/plot_output.jpeg"
```

| plotnode プロパティ | データ型  | プロパティの説明  |
|----------------|-------|---|
| x_field        | フィールド | x 軸のカスタム（ユーザー設定）ラベルを指定します。ラベルでのみ使用できます。         |
| y_field        | フィールド | y 軸のカスタム（ユーザー設定）ラベルを指定します。ラベルでのみ使用できます。         |
| three_D        | フラグ型  | y 軸のカスタム（ユーザー設定）ラベルを指定します。3-D グラフのラベルでのみ使用できます。 |
| z_field        | フィールド |   |
| color_field    | フィールド | オーバーレイ フィールド。                                   |

| plotnodeプロパティ      | データ型   | プロパティの説明  |
|--------------------|--|---|
| size_field         | フィールド  |   |
| shape_field        | フィールド  |   |
| panel_field        | フィールド  | 各カテゴリ個別のグラフの作成に使用する名義型またはフラグ型フィールドを指定します。グラフは「パネル化」され、複数のグラフが 1 つの出力ウィンドウに表示されます。 |
| animation_field    | フィールド  | アニメーションを使って順番に表示する一連のグラフを作成してデータ値のカテゴリを描画する、名義型またはフラグ型フィールドを指定します。                |
| transp_field       | フィールド  | カテゴリごとに異なるレベルの透過度を使用して、データ値のカテゴリを表すフィールドを指定します。折れ線グラフでは使用できません。                   |
| overlay_type       | None<br>Smoother<br>Function   | オーバーレイ関数が表示されるか、LOESS平滑化が表示されるかを指定します。  |
| overlay_expression | string   | overlay_type が Function に設定されているときに使用される式を指定します。                                  |
| style              | Point<br>Line  |   |
| point_type         | Rectangle<br>Dot<br>Triangle<br>Hexagon<br>Plus<br>Pentagon<br>Star<br>BowTie<br>HorizontalDash<br>VerticalDash<br>IronCross<br>Factory<br>House<br>Cathedral<br>OnionDome<br>ConcaveTriangle<br>OblateGlobe<br>CatEye<br>FourSidedPillow<br>RoundRectangle<br>Fan |   |
| x_mode             | Sort<br>Overlay<br>AsRead  |   |
| x_range_mode       | Automatic<br>UserDefined   |   |
| x_range_min        | number   |   |
| x_range_max        | number   |   |



| plotnodeプロパティ    | データ型                              | プロパティの説明                      |
|------------------|-----------------------------------|-------------------------------|
| y_range_mode     | Automatic<br>UserDefined          |                               |
| y_range_min      | number                            |                               |
| y_range_max      | number                            |                               |
| z_range_mode     | Automatic<br>UserDefined          |                               |
| z_range_min      | number                            |                               |
| z_range_max      | number                            |                               |
| jitter           | フラグ型                              |                               |
| records_limit    | number                            |                               |
| if_over_limit    | PlotBins<br>PlotSample<br>PlotAll |                               |
| x_label_auto     | フラグ型                              |                               |
| x_label          | string                            |                               |
| y_label_auto     | フラグ型                              |                               |
| y_label          | string                            |                               |
| z_label_auto     | フラグ型                              |                               |
| z_label          | string                            |                               |
| use_grid         | フラグ型                              |                               |
| graph_background | 色                                 | 標準のグラフ色は、このセクションの最初に説明されています。 |
| page_background  | 色                                 | 標準のグラフ色は、このセクションの最初に説明されています。 |
| use_overlay_expr | フラグ型                              | overlay_type の代わりに廃止される予定。    |

## timeplotnode のプロパティ



時系列ノードで、時系列データの 1 つ以上のセットを表示します。通常、最初に時間区分ノードを使用して TimeLabel フィールドを作成します。このフィールドは、x 軸にラベルを付けるために使用されます。詳細は、[5 章 時系列ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#)を参照してください。

### 例

```
create timeplotnode
set :timeplotnode.y_fields = ['sales' 'men' 'women']
set :timeplotnode.panel = True
set :timeplotnode.normalize = True
set :timeplotnode.line = True
set :timeplotnode.smoother = True
set :timeplotnode.use_records_limit = True
```

```

set :timeplotnode.records_limit = 2000
# Appearance settings
set :timeplotnode.symbol_size = 2.0

```

| timeplotnodeプロパティ  | データ型   | プロパティの説明   |
|--------------------|--|--|
| plot_series        | Series<br>Models   |  |
| use_custom_x_field | フラグ型   |  |
| x_field            | フィールド  |  |
| y_fields           | [フィールド<br>フィールド フィールド]   |  |
| panel              | フラグ型   |  |
| normalize          | フラグ型   |  |
| line               | フラグ型   |  |
| points             | フラグ型   |  |
| point_type         | Rectangle<br>Dot<br>Triangle<br>Hexagon<br>Plus<br>Pentagon<br>Star<br>BowTie<br>HorizontalDash<br>VerticalDash<br>IronCross<br>Factory<br>House<br>Cathedral<br>OnionDome<br>ConcaveTriangle<br>OblateGlobe<br>CatEye<br>FourSidedPillow<br>RoundRectangle<br>Fan |  |
| smoother           | フラグ型   | panel を True に設定した場合にのみ、平滑化を散布図に追加できます。  |
| use_records_limit  | フラグ型   |  |
| records_limit      | integer  |  |
| symbol_size        | number   | マーカー サイズを指定します。<br>例をあげると、次のようになります。<br><b>set :webnode.symbol_size = 5.5</b><br>これで、デフォルトよりも大きいマーカー サイズが作成されます。 |
| panel_layout       | Horizontal<br>Vertical   |  |

## webnode のプロパティ



Web グラフ ノードで、複数のシンボル値（カテゴリ）フィールドの値の関係の強さが示されます。このグラフでは、接続の強さを示すためにさまざまな幅の線が使用されます。Web グラフ ノードを使用して、たとえば、E コマース サイトで購入されたさまざまな商品の関係を調査できます。詳細は、[5 章 Web グラフ ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

### 例

```
create webnode
# "Plot" tab
set :webnode.use_directed_web = True
set :webnode.to_field = 'Drug'
set :webnode.fields = ['BP' 'Cholesterol' 'Sex' 'Drug']
set :webnode.from_fields = ['BP' 'Cholesterol' 'Sex']
set :webnode.true_flags_only = False
set :webnode.line_values = Absolute
set :webnode.strong_links_heavier = True
# "Options" tab
set :webnode.max_num_links = 300
set :webnode.links_above = 10
set :webnode.num_links = ShowAll
set :webnode.discard_links_min = True
set :webnode.links_min_records = 5
set :webnode.discard_links_max = True
set :webnode.weak_below = 10
set :webnode.strong_above = 19
set :webnode.link_size_continuous = True
set :webnode.web_display = Circular
```

| webnodeプロパティ         | データ型  | プロパティの説明 |
|----------------------|---|----------|
| use_directed_web     | フラグ型  |          |
| fields               | [フィールド<br>フィールド フィールド]                            |          |
| to_field             | フィールド   |          |
| from_fields          | [フィールド<br>フィールド フィールド]                            |          |
| true_flags_only      | フラグ型  |          |
| line_values          | Absolute<br>OverallPct<br>PctLarger<br>PctSmaller |          |
| strong_links_heavier | フラグ型  |          |

| webnodeプロパティ         | データ型                                     | プロパティの説明   |
|----------------------|--|--|
| num_links            | ShowMaximum<br>ShowLinksAbove<br>ShowAll |  |
| max_num_links        | number                                   |  |
| links_above          | number                                   |  |
| discard_links_min    | フラグ型                                     |  |
| links_min_records    | number                                   |  |
| discard_links_max    | フラグ型                                     |  |
| links_max_records    | number                                   |  |
| weak_below           | number                                   |  |
| strong_above         | number                                   |  |
| link_size_continuous | フラグ型                                     |  |
| web_display          | Circular<br>Network<br>Directed<br>Grid  |  |
| graph_background     | 色  | 標準のグラフ色は、このセクションの最初に説明されています。  |
| symbol_size          | number                                   | マーカー サイズを指定します。<br>例をあげると、次のようになります。<br><b>set :webnode.symbol_size = 5.5</b><br>これで、デフォルトよりも大きいマーカー サイズが作成されます。 |

# モデル作成ノードのプロパティ

## 一般的なモデル作成ノードのプロパティ

次のプロパティは、複数またはすべてのデータベース モデル作成ノードに共通です。個別のモデル作成ノードに関しては、必要に応じてドキュメント内に例外を記載しています。

| プロパティ                   | 値                                       | プロパティの説明  |
|-------------------------|---|---|
| custom_fields           | フラグ型                                    | 真 (true) の場合は、現在のノードのターゲット、入力、その他フィールドなどを指定することができます。偽 (false) の場合は、上流のデータ型ノードから現在の設定が使用されます。 |
| target<br>or<br>targets | フィールド<br>or<br>[フィールド 1 ...<br>フィールド N] | モデルのタイプによって、単一の対象フィールドまたは複数の対象フィールドを指定します。  |
| inputs                  | [フィールド 1 ...<br>フィールド N]                | モデルで使用される入力または予測変数フィールド。  |
| partition               | フィールド                                   |   |
| use_partitioned_data    | フラグ型                                    | 区分フィールドが定義される場合、このオプションは学習データ区分からのデータのみがモデル構築に使用されるようにします。                                    |
| use_split_data          | フラグ型                                    |   |
| splits                  | [フィールド1 ... フィールドN]                     | 分割モデル作成に使用する、フィールドを選択します。use_split_data が True に設定されている場合にのみ有効です。                             |
| use_frequency           | フラグ型                                    | 各モデル タイプで言及するとおり、重みフィールドおよび度数フィールドが特定のモデルで使用されます。   |
| frequency_field         | フィールド                                   |   |
| use_weight              | フラグ型                                    |   |
| weight_field            | フィールド                                   |   |
| use_model_name          | フラグ型                                    |   |
| model_name              | string                                  | ユーザーが指定する新規モデル名。  |
| mode                    | Simple<br>Expert                        |   |

## anomalydetectionnode のプロパティ



異常値検出ノードで、「正常な」データのパターンに合致しない異常ケースや外れ値を識別します。このノードで、外れ値が既知のパターンに当てはまらなかったり、何を探しているのかはつきりしなかったりする場合でも、外れ値を識別できます。詳細は、4章 異常値検出ノード in IBM SPSS Modeler 15 モデル作成ノード を参照してください。

### 例

```
create anomalydetectionnode
set :anomalydetectionnode.anomaly_method=PerRecords
set :anomalydetectionnode.percent_records=95
set :anomalydetectionnode.mode=Expert
set :anomalydetectionnode.peer_group_num_auto=true
set :anomalydetectionnode.min_num_peer_groups=3
set :anomalydetectionnode.max_num_peer_groups=10
```

| anomalydetectionnode のプロパティ | 値                                      | プロパティの説明  |
|-----------------------------|--|---|
| inputs                      | [フィールド1 ... フィールドN]                    | 異常値検出モデルは、指定の入力フィールドに基づいてレコードをスクリーニングします。ターゲット フィールドは使用しません。重みフィールドおよび度数フィールドも使用しません。詳細は、 p. 213 一般的なモデル作成ノードのプロパティを参照してください。 |
| mode                        | Expert<br>Simple                       |   |
| anomaly_method              | IndexLevel<br>PerRecords<br>NumRecords | レコードに異常としてフラグを設定するための、分割値を決めるのに使用される方法を指定します。   |
| index_level                 | number                                 | 異常としてフラグを設定するための最小分割値を指定します。  |
| percent_records             | number                                 | 学習データ内のレコードの割合 (%) に基づいてレコードにフラグを設定するための、閾値を設定します。  |
| num_records                 | number                                 | 学習データ内のレコードの数に基づいてレコードにフラグを設定するための、閾値を設定します。  |
| num_fields                  | integer                                | 各異常レコードに報告するフィールド数。   |
| impute_missing_values       | フラグ型                                   |   |

## モデル作成ノードのプロパティ

| anomalydetectionnode のプロパティ | 値       | プロパティの説明  |
|-----------------------------|---------|---|
| adjustment_coeff            | number  | 距離の計算時、連続型とカテゴリフィールド間に指定された関連の重みのバランスをとるために使用される値。                      |
| peer_group_num_auto         | フラグ型    | ピア グループ数を自動的に計算します。   |
| min_num_peer_groups         | integer | peer_group_num_auto が True に設定されている場合に使用されるピア グループの最小数を指定します。           |
| max_num_per_groups          | integer | ピア グループの最大数を指定します。  |
| num_peer_groups             | integer | peer_group_num_auto が False に設定されている場合に使用されるピア グループの数を指定します。            |
| noise_level                 | number  | クラスタリング中の外れ値の処理方法を決定します。0 から 0.5 までの値を指定してください。                         |
| noise_ratio                 | number  | ノイズのバッファリングに使用されるコンポーネントに割り当てられる、メモリーの“.”を指定します。0 から 0.5 までの値を指定してください。 |

## apriorinode のプロパティ



Apriori ノードで、データからルール セットを抽出し、情報内容が最も充実したルールを引き出します。Apriori には、5 種類のルール選択方法があり、高度なインデックス作成方法を使用して、大きなデータ セットが効率的に処理されます。大きな問題の場合は、一般に、Apriori の方が高速に学習できます。保持できるルール数に特に制限はありません。また、最大 32 の前提条件を持つルールを処理できます。Apriori では、入力フィールドと出力フィールドのすべてがカテゴリであることが必要ですが、この種類のデータに合わせて最適化されているので、よりよいパフォーマンスを実現します。詳細は、12 章 Apriori ノード in IBM SPSS Modeler 15 モデル作成ノード を参照してください。

## 例

```
create apriorinode
# "Fields" tab
set :apriorinode.custom_fields = True
set :apriorinode.use_transactional_data = True
set :apriorinode.id_field = 'Age'
set :apriorinode.contiguous = True
```

```

set :apriorinode.content_field = 'Drug'
# These seem to have changed, used to be:
#help set :apriorinode.consequents = ['Age']
#help set :apriorinode.antecedents = ['BP' 'Cholesterol' 'Drug']
# now it seems we have;
#help set :apriorinode.content = ['Age']
set :apriorinode.partition = Test
# "Model" tab
set :apriorinode.use_model_name = False
set :apriorinode.model_name = "Apriori_bp_choles_drug"
set :apriorinode.min_supp = 7.0
set :apriorinode.min_conf = 30.0
set :apriorinode.max_antecedents = 7
set :apriorinode.true_flags = False
set :apriorinode.optimize = Memory
# "Expert" tab
set :apriorinode.mode = Expert
set :apriorinode.evaluation = ConfidenceRatio
set :apriorinode.lower_bound = 7

```

| apriorinode のプロパティ     | 値  | プロパティの説明  |
|------------------------|--|---|
| consequents            | フィールド  | Apriori モデルは標準的な対象フィールドおよび入力フィールドの結果と条件を使用します。重みフィールドおよび度数フィールドは使用しません。詳細は、 <a href="#">p.213 一般的なモデル作成ノードのプロパティ</a> を参照してください。 |
| antecedents            | [フィールド 1 ...<br>フィールド N]   |   |
| min_supp               | number   |   |
| min_conf               | number   |   |
| max_antecedents        | number   |   |
| true_flags             | フラグ型   |   |
| optimize               | Speed<br>Memory  |   |
| use_transactional_data | フラグ型   |   |
| contiguous             | フラグ型   |   |
| id_field               | string   |   |
| content_field          | string   |   |
| mode                   | Simple<br>Expert   |   |
| evaluation             | RuleConfidence<br>DifferenceToPrior<br>ConfidenceRatio<br>InformationDifference<br>NormalizedChiSquare |   |



| apriorinode のプロパティ | 値               | プロパティの説明                           |
|--------------------|-----------------|------------------------------------|
| lower_bound        | number          |                                    |
| optimize           | Speed<br>Memory | モデル作成が速度とメモリーのどちらにより最適化されるかを指定します。 |

## autoclassifiernode のプロパティ



自動分類ノードは、2種類の結果 (yes/no、churn/don't churn など) を生じる多くの異なるモデルを作成および比較し、与えられた分析への最善のアプローチを選ぶことができるようになります。多くのモデル作成アルゴリズムに対応し、希望する方法、各特定のオプション、そして結果を比較するための基準を選択することができます。このノードで、指定されたオプションに基づいてモデルのセットが生成され、指定された基準に基づいて最善の候補がランク付けされます。詳細は、[5 章 自動分類ノード in IBM SPSS Modeler 15 モデル作成ノード](#) を参照してください。

### 例

```
create autoclassifiernode
set :autoclassifiernode.ranking_measure=Accuracy
set :autoclassifiernode.ranking_dataset=Training
set :autoclassifiernode.enable_accuracy_limit=true
set :autoclassifiernode.accuracy_limit=0.9
set :autoclassifiernode.calculate_variable_importance=true
set :autoclassifiernode.use_costs=true
set :autoclassifiernode.svm=false
```

| autoclassifiernode のプロパティ | 値   | プロパティの説明  |
|---------------------------|---|---|
| target                    | フィールド   | フラグ型対照の場合、自動分類ノードは 1 つの対象フィールドおよび 1 つ以上の入力フィールドを使用します。重みフィールドおよび度数フィールドも指定することができます。詳細は、 <a href="#">p. 213 一般的なモデル作成ノードのプロパティ</a> を参照してください。 |
| ranking_measure           | Accuracy<br>Area_under_curve<br>Profit<br>Lift<br>Num_variables |   |
| ranking_dataset           | Training<br>Test  |   |
| number_of_models          | integer   | モデル ナゲットに含まれるモデルの数。1 と 100の間の整数を指定します。  |

| autoclassifiernode のプロパティ              | 値       | プロパティの説明   |
|--|---------|--|
| calculate_variable_importance          | フラグ型    |  |
| enable_accuracy_limit                  | フラグ型    |  |
| accuracy_limit                         | integer | 0 と 100 の間の整数です。   |
| enable_area_under_curve_limit          | フラグ型    |  |
| area_under_curve_limit                 | number  | 0.0 と 1.0 の間の実数。   |
| enable_profit_limit                    | フラグ型    |  |
| profit_limit                           | number  | 1 以上の整数。   |
| enable_lift_limit                      | フラグ型    |  |
| lift_limit                             | number  | 1.0 を超える実数。  |
| enable_number_of_variables_limit       | フラグ型    |  |
| number_of_variables_limit              | number  | 1 以上の整数。   |
| use_fixed_cost                         | フラグ型    |  |
| fixed_cost                             | number  | 0.0 を超える実数。  |
| variable_cost                          | フィールド   |  |
| use_fixed_revenue                      | フラグ型    |  |
| fixed_revenue                          | number  | 0.0 を超える実数。  |
| variable_revenue                       | フィールド   |  |
| use_fixed_weight                       | フラグ型    |  |
| fixed_weight                           | number  | 0.0 を超える実数。  |
| variable_weight                        | フィールド   |  |
| lift_percentile                        | number  | 0 と 100 の間の整数です。   |
| enable_model_build_time_limit          | フラグ型    |  |
| model_build_time_limit                 | number  | 個々のモデルのそれぞれを構築するためにかかる時間を制限するために分数を設定する整数。   |
| enable_stop_after_time_limit           | フラグ型    |  |
| stop_after_time_limit                  | number  | 自動分類の実行のための全体経過時間を制限するために時間数を設定する実数。   |
| enable_stop_after_valid_model_produced | フラグ型    |  |
| use_costs                              | フラグ型    |  |
| <algorithm>                            | フラグ型    | 次のように特定のアルゴリズムの使用の有効、無効を切り替えます。<br><code>set :autoclassifiernode.chaid=true</code> |
| <algorithm>.<property>                 | string  | 特定のアルゴリズムのプロパティ値を設定します。詳細は、 <a href="#">p.219 アルゴリズム プロパティの設定</a> を参照してください。       |

## アルゴリズム プロパティの設定

自動分類ノード、自動数値ノード、自動クラスタ ノードについては、ノードが使用する特定のアルゴリズムのプロパティは、次の一般フォーマットを使用して設定できます。

```
set :autoclassifiernode.<algorithm>.<property> = <value>
```

```
set :autonumericnode.<algorithm>.<property> = <value>
```

```
set :autoclusternode.<algorithm>.<property> = <value>
```

次に例を示します。

```
set :autoclassifiernode.neuralnetwork.method = MultilayerPerceptron
```

自動分類ノードのアルゴリズム名は、`cart`、`chaid`、`quest`、`c50`、`logreg`、`decisionlist`、`bayesnet`、`discriminant`、`svm` および `knn` です。

自動数値ノードのアルゴリズム名は、`cart`、`chaid`、`neuralnetwork`、`genlin`、`svm`、`regression`、`linear` および `knn` です。

自動クラスタ ノードのアルゴリズム名は、`twostep`、`k-means`、および `kohonen` です。

プロパティ名は、各アルゴリズムノードのために文書化されている標準です。

ピリオドなどの句読点を含むアルゴリズム プロパティは、次のように一重引用符で囲む必要があります。

```
set :autoclassifiernode.logreg.tolerance = '1.0E-5'
```

次のように、複数の値をプロパティに割り当てることもできます。

```
set :autoclassifiernode.decisionlist.search_direction = [Up Down]
```

特定のアルゴリズムの使用の有効、無効を切り替えるには、次のようにします。

```
set :autoclassifiernode.chaid=true
```

注:

- `true` および `false` の値を設定するときは、小文字を使用します (`False` のように大文字は使用しません)。
- 自動分類ノードで特定のアルゴリズム オプションが使用可能でない場合、または値の範囲ではなく、1 つの値だけを指定できるときは、標準の方法でノードにアクセスするときと同じ制限が、スクリプトにも適用されます。

## autoclusternode のプロパティ



自動クラスタ ノードは、同様の特性を持つレコードのグループを識別するクラスタリング モデルを推定し、比較します。ノードは他の自動化モデル作成ノードと同じように動作し、複数の組み合わせのオプションを単一のモデル作成の実行で検証できます。モデルは、クラスタ モデルの有用性をフィルタリングおよびランク付けする基本的な指標を使用して比較し、特定のフィールドの重要度に基づいて指標を提供します。詳細は、[5 章 自動クラスタ ノード in IBM SPSS Modeler 15 モデル作成ノード](#) を参照してください。

### 例

```
create autoclusternode
set :autoclusternode.ranking_measure=Silhouette
set :autoclusternode.ranking_dataset=Training
set :autoclusternode.enable_silhouette_limit=true
set :autoclusternode.silhouette_limit=5
```

| autoclusternode のプロパティ        | 値  | プロパティの説明  |
|-------------------------------|--|---|
| evaluation                    | フィールド  | 注：自動クラスタ ノードのみ。重要度の値を計算するフィールドを識別します。また、どれだけクラスタがフィールドの値を区別するか、どれだけ正確にモデルがこのフィールドを予測するかを識別するために使用することができます。 |
| ranking_measure               | Silhouette<br>Num_clusters<br>Size_smallest_cluster<br>Size_largest_cluster<br>Smallest_to_largest<br>Importance |   |
| ranking_dataset               | Training<br>Test   |   |
| summary_limit                 | integer  | レポートに一覧するモデルの数。1 と 100の間の整数を指定します。  |
| enable_silhouette_limit       | フラグ型   |   |
| silhouette_limit              | integer  | 0 と 100 の間の整数です。  |
| enable_number_less_limit      | フラグ型   |   |
| number_less_limit             | number   | 0.0 と 1.0 の間の実数。  |
| enable_number_greater_limit   | フラグ型   |   |
| number_greater_limit          | number   | 1 以上の整数。  |
| enable_smallest_cluster_limit | フラグ型   |   |
| smallest_cluster_units        | Percentage<br>Counts   |   |

| autoclusternode のプロパティ            | 値                         | プロパティの説明  |
|-----------------------------------|---------------------------|---|
| smallest_cluster_limit_percentage | number                    |   |
| smallest_cluster_limit_count      | integer                   | 1 以上の整数。  |
| enable_largest_cluster_limit      | フラグ型                      |   |
| largest_cluster_units             | Percentage<br>Counts      |   |
| largest_cluster_limit_percentage  | number                    |   |
| largest_cluster_limit_count       | integer                   |   |
| enable_smallest_largest_limit     | フラグ型                      |   |
| smallest_largest_limit            | number                    |   |
| enable_importance_limit           | フラグ型                      |   |
| importance_limit_condition        | Greater_than<br>Less_than |   |
| importance_limit_greater_than     | number                    | 0 と 100 の間の整数です。  |
| importance_limit_less_than        | number                    | 0 と 100 の間の整数です。  |
| <algorithm>                       | フラグ型                      | 次のように特定のアルゴリズムの使用の有効、無効を切り替えます。<br><b>set :autoclusternode.kohonen=true</b>   |
| <algorithm>.<property>            | string                    | 特定のアルゴリズムのプロパティ値を設定します。 <a href="#">詳細は、p. 219 アルゴリズム プロパティの設定</a> を参照してください。 |

## autonumericnode のプロパティ



自動数値ノードでは、多くのさまざまな方法を使用し、連続する数値範囲の結果を求めてモデルを推定し比較します。このノードは、自動分類ノードと同じ方法で動作し、1 回のモデル作成のパスで、複数の組み合わせのオプションを使用し試すアルゴリズムを選択することができます。使用できるアルゴリズムには、ニューラル ネットワーク、C&R Tree、CHAID、線型回帰、一般化線型回帰、サポート ベクトル マシン (SVM) が含まれています。モデルは、相関、相対エラー、または使用された変数の数に基づいて比較できます。 [詳細は、5 章 自動数値ノード in IBM SPSS Modeler 15 モデル作成ノード](#) を参照してください。

### 例

```
create autonumericnode
set :autonumericnode.ranking_measure=Correlation
set :autonumericnode.ranking_dataset=Training
set :autonumericnode.enable_correlation_limit=true
set :autonumericnode.correlation_limit=0.8
set :autonumericnode.calculate_variable_importance=true
set :autonumericnode.neuralnetwork=true
```

```
set :autonumericnode.chaid=false
```

| autonumericnode のプロパティ        | 値                             | プロパティの説明   |
|-------------------------------|-------------------------------|--|
| custom_fields                 | フラグ型                          | 真 (True) の場合、データ型ノード設定の代わりにカスタム フィールド設定が使用されます。  |
| target                        | フィールド                         | 自動数値ノードは 1 つの対象フィールドおよび 1 つ以上の入力フィールドを使用します。重みフィールドおよび度数フィールドも指定することができます。詳細は、 <a href="#">p.213 一般的なモデル作成ノードのプロパティ</a> を参照してください。 |
| inputs                        | [フィールド1 ... フィールド2]           |  |
| partition                     | フィールド                         |  |
| use_frequency                 | フラグ型                          |  |
| frequency_field               | フィールド                         |  |
| use_weight                    | フラグ型                          |  |
| weight_field                  | フィールド                         |  |
| use_partitioned_data          | フラグ型                          | データ区分フィールドが定義されている場合、学習データだけがモデルの構築に使用されます。  |
| ranking_measure               | Correlation<br>NumberOfFields |  |
| ranking_dataset               | Test<br>Training              |  |
| number_of_models              | integer                       | モデル ナゲットに含まれるモデルの数。1 と 100の間の整数を指定します。   |
| calculate_variable_importance | フラグ型                          |  |
| enable_correlation_limit      | フラグ型                          |  |
| correlation_limit             | integer                       |  |
| enable_number_of_fields_limit | フラグ型                          |  |
| number_of_fields_limit        | integer                       |  |
| enable_relative_error_limit   | フラグ型                          |  |
| relative_error_limit          | integer                       |  |
| enable_model_build_time_limit | フラグ型                          |  |
| model_build_time_limit        | integer                       |  |
| enable_stop_after_time_limit  | フラグ型                          |  |
| stop_after_time_limit         | integer                       |  |
| stop_if_valid_model           | フラグ型                          |  |

| autonumericnode のプロパティ | 値      | プロパティの説明  |
|------------------------|--------|---|
| <algorithm>            | フラグ型   | 次のように特定のアルゴリズムの使用の有効、無効を切り替えます。<br><b>set :autonumericnode.chaid=true</b>     |
| <algorithm>.<property> | string | 特定のアルゴリズムのプロパティ値を設定します。 <a href="#">詳細は、p. 219 アルゴリズム プロパティの設定</a> を参照してください。 |

## bayesnetnode プロパティ



ベイズ ネットワーク ノードを使用すると、観測された情報および記録された情報を実際の知識を組み合わせることによって確率モデルを作成し、発生の尤度を確立できます。ノードは主に分類に使用される Tree Augmented Naïve Bayes (TAN) および Markov Blanket ネットワークに焦点を当てています。 [詳細は、7 章 Bayesian Network ノード in IBM SPSS Modeler 15 モデル作成 ノード](#) を参照してください。

### 例

```
create bayesnetnode
set :bayesnetnode.continue_training_existing_model = True
set :bayesnetnode.structure_type = MarkovBlanket
set :bayesnetnode.use_feature_selection = True
# Expert tab
set :bayesnetnode.mode = Expert
set :bayesnetnode.all_probabilities = True
set :bayesnetnode.independence = Pearson
```

| bayesnetnode のプロパティ              | 値                    | プロパティの説明  |
|----------------------------------|----------------------|---|
| inputs                           | [フィールド1 ... フィールドN]  | ベイズ ネットワーク モデルは単一の対象フィールドおよび 1 つ以上の入力フィールドを使用します。連続フィールドは自動的に分割されます。 <a href="#">詳細は、p. 213 一般的なモデル作成ノードのプロパティ</a> を参照してください。 |
| continue_training_existing_model | フラグ型                 |   |
| structure_type                   | TAN<br>MarkovBlanket | Bayesian ネットワークを構築時に使用する構造を選択します。   |
| use_feature_selection            | フラグ型                 |   |
| parameter_learning_method        | Likelihood<br>Bayes  | 親の値が認識されるノード間の条件確率テーブルを推定するために用いる方法を指定します。  |

| bayesnetnode のプロパティ             | 値                      | プロパティの説明   |
|---------------------------------|------------------------|--|
| mode                            | Expert<br>Simple       |  |
| missing_values                  | フラグ型                   |  |
| all_probabilities               | フラグ型                   |  |
| independence                    | Likelihood<br>Pearson  | 2 つの変数のペアの観測がお互いに独立しているかどうかを評価するために用いる方法を指定します。                    |
| significance_level              | number                 | 独立性を判断するための分割値を指定します。  |
| maximal_conditioning_set        | number                 | 独立性検定に使用する条件変数の最大数を指定します。  |
| inputs_always_selected          | [フィールド1 ...<br>フィールドN] | バイズ ネットワーク構築時にデータセットのどのフィールドを常に使用するかを指定します。<br>注：対象フィールドは常に選択されます。 |
| maximum_number_inputs           | number                 | バイズ ネットワーク構築で使用する入力フィールドの最大数を指定します。                                |
| calculate_variable_importance   | フラグ型                   |  |
| calculate_raw_propensities      | フラグ型                   |  |
| calculate_adjusted_propensities | フラグ型                   |  |
| adjusted_propensity_partition   | Test<br>Validation     |  |

## c50node のプロパティ



C5.0 ノードは、ディシジョン ツリーとルール セットのどちらかを構築します。このモデルは、各レベルで最大の情報の対応をもたらすフィールドに基づいてサンプルを分割します。対象フィールドは、カテゴリでなければなりません。複数の分割を 2 つ以上のサブグループに分割できます。詳細は、6 章 C5.0 ノード in IBM SPSS Modeler 15 モデル作成ノード を参照してください。

### 例

```
create c50node
# "Model" tab
set :c50node.use_model_name = False
set :c50node.model_name = "C5_Drug"
set :c50node.use_partitioned_data = True
set :c50node.output_type = DecisionTree
set :c50node.use_xval = True
set :c50node.xval_num_folds = 3
```



## モデル作成ノードのプロパティ

```

set :c50node.mode = Expert
set :c50node.favor = Generality
set :c50node.min_child_records = 3
# "Costs" tab
set :c50node.use_costs = True
set :c50node.costs = [{"drugA" "drugX" 2}]

```

| c50node のプロパティ                  | 値                       | プロパティの説明   |
|---------------------------------|-------------------------|--|
| target                          | フィールド                   | C50 モデルは単一の対象フィールドおよび 1 つ以上の入力フィールドを使用します。重みフィールドも指定できます。詳細は、 <a href="#">p. 213 一般的なモデル作成ノードのプロパティ</a> を参照してください。 |
| output_type                     | DecisionTree<br>RuleSet |  |
| group_symbolics                 | フラグ型                    |  |
| use_boost                       | フラグ型                    |  |
| boost_num_trials                | number                  |  |
| use_xval                        | フラグ型                    |  |
| xval_num_folds                  | number                  |  |
| mode                            | Simple<br>Expert        |  |
| favor                           | Accuracy<br>Generality  | 精度 (Accuracy) または一般化 (Generality) を選択。   |
| expected_noise                  | number                  |  |
| min_child_records               | number                  |  |
| pruning_severity                | number                  |  |
| use_costs                       | フラグ型                    |  |
| costs                           | 構造化                     | これは構造化されたプロパティです。  |
| use_winning                     | フラグ型                    |  |
| use_global_pruning              | フラグ型                    | デフォルトではオン (True)。  |
| calculate_variable_importance   | フラグ型                    |  |
| calculate_raw_propensities      | フラグ型                    |  |
| calculate_adjusted_propensities | フラグ型                    |  |
| adjusted_propensity_partition   | Test<br>Validation      |  |

## carmanode のプロパティ



CARMA モデルは、入力または対象フィールドを指定しなくても、データからルールセットを抽出します。Apriori とは対照的に、CARMA ノードは、前提条件サポートではなく、ルールサポート（前提条件と結果の両方のサポート）の構築の設定ができます。これは、生成されたルールをさまざまなアプリケーションで活用できることを意味します。たとえば、この休暇シーズンに販売促進する項目を結果とする、商品またはサービス（前提条件）のリストを調べることができます。詳細は、[12 章 CARMA ノード in IBM SPSS Modeler 15 モデル作成ノード](#) を参照してください。

### 例

```
create carmanode
# "Fields" tab
set :carmanode.custom_fields = True
set :carmanode.use_transactional_data = True
set :carmanode.inputs = ['BP' 'Cholesterol' 'Drug']
set :carmanode.partition = Test
# "Model" tab
set :carmanode.use_model_name = False
set :carmanode.model_name = "age_bp_drug"
set :carmanode.use_partitioned_data = False
set :carmanode.min_supp = 10.0
set :carmanode.min_conf = 30.0
set :carmanode.max_size = 5
# Expert Options
set :carmanode.mode = Expert
#help set :carmanode.exclude_simple = True
set :carmanode.use_pruning = True
set :carmanode.pruning_value = 300
set :carmanode.vary_support = True
set :carmanode.estimated_transactions = 30
set :carmanode.rules_without_antecedents = True
```

| carmanode のプロパティ       | 値                   | プロパティの説明   |
|------------------------|---------------------|--|
| inputs                 | [フィールド1 ... フィールドn] | CARMA モデルは対象フィールドではなく、入力フィールドのリストを使用します。重みフィールドおよび度数フィールドは使用しません。詳細は、 <a href="#">p. 213 一般的なモデル作成ノードのプロパティ</a> を参照してください。 |
| id_field               | フィールド               | モデル作成の ID フィールドとして使用するフィールド。   |
| contiguous             | フラグ型                | ID フィールドの ID が連続するかどうかを指定します。  |
| use_transactional_data | フラグ型                |  |

| carmanode のプロパティ          | 値                | プロパティの説明                                     |
|---------------------------|------------------|--|
| content_field             | フィールド            |  |
| min_supp                  | 数値 (パーセント)       | 前提条件範囲(サポート)ではなく、ルール範囲に関連します。デフォルト値は 20% です。 |
| min_conf                  | 数値 (パーセント)       | デフォルト値は 20% です。                              |
| max_size                  | number           | デフォルトは 10 です。                                |
| mode                      | Simple<br>Expert | デフォルトは Simple です。                            |
| exclude_multiple          | フラグ型             | 複数の結果を持つルールを除外します。デフォルトは False です。           |
| use_pruning               | フラグ型             | デフォルトは False です。                             |
| pruning_value             | number           | デフォルトは 500 です。                               |
| vary_support              | フラグ型             |  |
| estimated_transactions    | integer          |  |
| rules_without_antecedents | フラグ型             |  |

## cartnode のプロパティ



C&R Tree (分類と回帰ツリー) ノードは、ディシジョン ツリーを生成し、将来の観測値を予測または分類できるようにします。この方法は再帰的なデータ区分を使用して学習レコードを複数のセグメントに分割し、各ステップで不純性を最小限に抑えます。ツリーのノードが「純粹」であると考えられるのは、ノード中にあるケースの 100% が、対象フィールドのある特定のカテゴリに分類される場合です。対象フィールドおよび入力フィールドは、数値範囲またはカテゴリ (名義型、順序型、フラグ) が使用できます。すべての分岐は 2 分割です (2 つのサブグループのみ)。詳細は、[6 章 C&R ツリー ノード in IBM SPSS Modeler 15 モデル作成ノード](#) を参照してください。

### 例

```
create cartnode
# "Fields" tab
set :cartnode.custom_fields = True
set :cartnode.target = 'Drug'
set :cartnode.inputs = ['Age' 'BP' 'Cholesterol']
# "Build Options" tab, 'Objective' panel
set :cartnode.model_output_type = InteractiveBuilder
set :cartnode.use_tree_directives = True
set :cartnode.tree_directives = ""Grow Node Index 0 Children 1 2
Grow Node Index 2 Children 3 4""
# "Build Options" tab, 'Basics' panel
set :cartnode.prune_tree = False
set :cartnode.use_std_err_rule = True
set :cartnode.std_err_multiplier = 3.0
```

```

set :cartnode.max_surrogates = 7
#"Build Options" tab, 'Stopping Rules' panel
set :cartnode.use_percentage = True
set :cartnode.min_parent_records_pc = 5
set :cartnode.min_child_records_pc = 3
#"Build Options" tab, 'Costs & Priors' panel
set :cartnode.use_costs = True
set :cartnode.costs = [{"drugA" "drugX" 2}]
set :cartnode.priors = Custom
# custom priors must add to 1
set :cartnode.custom_priors = [{"drugA" 0.3}{drugX" 0.7}]
set :cartnode.adjust_priors = True
#"Build Options" tab, 'Advanced' panel
set :cartnode.min_impurity = 0.0003
set :cartnode.impurity_measure = Twoing
#"Model Options" tab
set :cartnode.use_model_name = False
set :cartnode.model_name = "Cart_Drug"

```

| cartnode のプロパティ                  | 値                                      | プロパティの説明   |
|----------------------------------|--|--|
| target                           | フィールド                                  | C&R Tree モデルは 1 つの対象フィールドおよび 1 つ以上の入力フィールドを使用します。度数フィールドも指定できます。 <a href="#">詳細は、 p.213 一般的なモデル作成ノードのプロパティを参照してください。</a>   |
| continue_training_existing_model | フラグ型                                   |  |
| objective                        | Standard<br>Boosting<br>Bagging<br>psm | psm は非常に大きいデータセットに使用され、Server の接続が必要です。  |
| model_output_type                | Single<br>InteractiveBuilder           |  |
| use_tree_directives              | フラグ型                                   |  |
| tree_directives                  | string                                 | ツリーの成長のためのディレクティブ (式) を指定します。ディレクティブ (式) は、改行や引用符のエスケープ処理を回避するために、三重の引用符で囲むことができます。ディレクティブは、データやモデルリング オプションの些細な変更に依存するため、他のデータセットに対しては一般化できません。 <a href="#">詳細は、 6 章 ツリー成長ディレクティブ in IBM SPSS Modeler 15 モデル作成ノードを参照してください。</a> |
| use_max_depth                    | Default<br>Custom                      |  |

## モデル作成ノードのプロパティ

| cartnode のプロパティ        | 値  | プロパティの説明   |
|------------------------|--|--|
| max_depth              | integer  | 最大ツリー深さ (0 ~ 1000)。<br>use_max_depth = Custom の場合にのみ使用します。                                     |
| prune_tree             | フラグ型   | オーバーフィットしないようにツリーを剪定します。   |
| use_std_err            | フラグ型   | リスクにおける最大差 (標準誤差) を使用します。  |
| std_err_multiplier     | number   | 最大差。   |
| max_surrogates         | number   | 最大代理フィールド :  |
| use_percentage         | フラグ型   |  |
| min_parent_records_pc  | number   |  |
| min_child_records_pc   | number   |  |
| min_parent_records_abs | number   |  |
| min_child_records_abs  | number   |  |
| use_costs              | フラグ型   |  |
| costs                  | 構造化  | 次のフォーマットを使用した構造化プロパティ。<br>[[drugA drugB 1.5] {drugA drugC 2.1}].<br>{ } 内の引数は実際の予測コストです。       |
| priors                 | Data<br>Equal<br>Custom                                |  |
| custom_priors          | 構造化  | 次のフォーマットを使用した構造化プロパティ。<br>set :cartnode.<br>custom_priors =<br>[ { drugA 0.3 } { drugB 0.6 } ] |
| adjust_priors          | フラグ型   |  |
| trails                 | number   | ブーストまたはバグのコンポーネント モデル数。  |
| set_ensemble_method    | Voting<br>HighestProbability<br>HighestMeanProbability | カテゴリ型対象のデフォルト結合ルール。  |
| range_ensemble_method  | Mean<br>Median   | 連続型対象のデフォルト結合ルール。  |
| large_boost            | フラグ型   | 特に大きなデータセットのブースティングを適用します。   |
| min_impurity           | number   |  |
| impurity_measure       | Gini<br>Twoing<br>Ordered                              |  |
| train_pct              | number   | オーバーフィット防止セット。   |
| set_random_seed        | フラグ型   | 結果の複製オプション。  |
| seed                   | number   |  |

| cartnode のプロパティ                 | 値                  | プロパティの説明 |
|---------------------------------|--------------------|----------|
| calculate_variable_importance   | フラグ型               |          |
| calculate_raw_propensities      | フラグ型               |          |
| calculate_adjusted_propensities | フラグ型               |          |
| adjusted_propensity_partition   | Test<br>Validation |          |

## chaidnode のプロパティ



CHAID ノードはディビジョン ツリーを生成し、カイ二乗統計値を使用して最適な分割を識別します。C&RT Tree および QUEST ノードと異なり、CHAID は、非 2 分岐ツリーを生成できます。これは、ある分岐が 3 個以上の枝葉を持つことを意味します。対象フィールドおよび入力フィールドは、数値範囲（連続型）またはカテゴリとなります。Exhaustive CHAID は CHAID の修正版で、可能性のある分割すべてを調べることで、よりよい結果を得られますが、計算時間も長くなります。詳細は、[6 章 CHAID ノード in IBM SPSS Modeler 15 モデル作成ノード](#) を参照してください。

### 例

```
create chaidnode
set :chaidnode.custom_fields = True
set :chaidnode.target = Drug
set :chaidnode.inputs = [Age Na K Cholesterol BP]
set :chaidnode.use_model_name = true
set :chaidnode.model_name = "CHAID"
set :chaidnode.method = Chaid
set :chaidnode.model_output_type = InteractiveBuilder
set :chaidnode.use_tree_directives = True
set :chaidnode.tree_directives = "Test"
set :chaidnode.mode = Expert
set :chaidnode.split_alpha = 0.03
set :chaidnode.merge_alpha = 0.04
set :chaidnode.chi_square = Pearson
set :chaidnode.use_percentage = True
set :chaidnode.min_parent_records_abs = 40
set :chaidnode.min_child_records_abs = 30
set :chaidnode.epsilon = 0.003
set :chaidnode.max_iterations = 75
set :chaidnode.split_merged_categories = true
```

## モデル作成ノードのプロパティ

set :chaidnode.bonferroni\_adjustment = true

| chaidnode のプロパティ                 | 値  | プロパティの説明   |
|----------------------------------|--|--|
| target                           | フィールド  | CHAID モデルは単一の対象フィールドおよび 1 つ以上の入力フィールドを使用します。度数フィールドも指定できます。 <a href="#">詳細は、p. 213 一般的なモデル作成ノードのプロパティ</a> を参照してください。 |
| continue_training_existing_model | フラグ型   |  |
| objective                        | Standard<br>Boosting<br>Bagging<br>psm                 | psm は非常に大きいデータセットに使用され、Server の接続が必要です。  |
| model_output_type                | Single<br>InteractiveBuilder                           |  |
| use_tree_directives              | フラグ型   |  |
| tree_directives                  | string   |  |
| method                           | Chaid<br>ExhaustiveChaid                               |  |
| use_max_depth                    | Default<br>Custom                                      |  |
| max_depth                        | integer  | 最大ツリー深さ (0 ~ 1000)。<br>use_max_depth = Custom の場合にのみ使用します。   |
| use_percentage                   | フラグ型   |  |
| min_parent_records_pc            | number   |  |
| min_child_records_pc             | number   |  |
| min_parent_records_abs           | number   |  |
| min_child_records_abs            | number   |  |
| use_costs                        | フラグ型   |  |
| costs                            | 構造化  | 次のフォーマットを使用した構造化プロパティ。<br>[{drugA drugB 1.5} {drugA drugC 2.1}].<br>{ } 内の引数は実際の予測コストです。                             |
| trails                           | number   | ブーストまたはバグのコンポーネント モデル数。  |
| set_ensemble_method              | Voting<br>HighestProbability<br>HighestMeanProbability | カテゴリ型対象のデフォルト結合ルール。  |
| range_ensemble_method            | Mean<br>Median   | 連続型対象のデフォルト結合ルール。  |
| large_boost                      | フラグ型   | 特に大きなデータセットのブースティングを適用します。   |
| split_alpha                      | number   | 分割の有意水準 :  |

| chaidnode のプロパティ                | 値                  | プロパティの説明                             |
|---------------------------------|--------------------|--------------------------------------|
| merge_alpha                     | number             | 結合の有意水準。                             |
| bonferroni_adjustment           | フラグ型               | Bonferroni メソッドを使用して有意確率値を調整。        |
| split_merged_categories         | フラグ型               | マージしたカテゴリの再分割を許可。                    |
| chi_square                      | Pearson<br>LR      | カイ 2 乗統計の計算に使用される方法 (Pearson または尤度比) |
| epsilon                         | number             | 期待されるセル度数の最小変化。                      |
| max_iterations                  | number             | 収束のための最大反復回数。                        |
| set_random_seed                 | integer            |                                      |
| seed                            | number             |                                      |
| calculate_variable_importance   | フラグ型               |                                      |
| calculate_raw_propensities      | フラグ型               |                                      |
| calculate_adjusted_propensities | フラグ型               |                                      |
| adjusted_propensity_partition   | Test<br>Validation |                                      |
| maximum_number_of_models        | integer            |                                      |

## coxregnode のプロパティ



Cox 回帰ノードを使用すると、打ち切りレコードの存在下でイベントまでの時間のデータの生存モデルを構築します。モデルは、対象のイベントが入力変数の指定の値で指定の時間 (t) に発生する確率を予測する生存関数を作成します。詳細は、[10 章 Cox ノード in IBM SPSS Modeler 15 モデル作成ノード](#)を参照してください。

### 例

```
create coxregnode
set :coxregnode.survival_time = tenure
set :coxregnode.method = BackwardsStepwise
# Expert tab
set :coxregnode.mode = Expert
set :coxregnode.removal_criterion = Conditional
```



## モデル作成ノードのプロパティ

```
set :coxregnode.survival = True
```

| coxregnode のプロパティ   | 値   | プロパティの説明   |
|---------------------|---|--|
| survival_time       | フィールド   | Cox回帰モデルは 生存時間のある 1 つのフィールドを使用します。   |
| target              | フィールド   | Cox 回帰モデルは 1 つの対象フィールドおよび 1 つ以上の入力フィールドを使用します。 <a href="#">詳細は、 p. 213 一般的なモデル作成ノードのプロパティ を参照してください。</a> |
| method              | Enter<br>Stepwise<br>BackwardsStepwise              |  |
| groups              | フィールド   |  |
| model_type          | MainEffects<br>Custom                               |  |
| custom_terms        | [ “BP*Sex”<br>“BP*Age” ]                            | 例：<br>set :coxregnode. custom_terms =<br>["BP*Sex" "BP" "Age"]   |
| mode                | Expert<br>Simple                                    |  |
| max_iterations      | number  |  |
| p_converge          | 1.0E-4<br>1.0E-5<br>1.0E-6<br>1.0E-7<br>1.0E-8<br>0 |  |
| p_converge          | 1.0E-4<br>1.0E-5<br>1.0E-6<br>1.0E-7<br>1.0E-8<br>0 |  |
| l_converge          | 1.0E-1<br>1.0E-2<br>1.0E-3<br>1.0E-4<br>1.0E-5<br>0 |  |
| removal_criterion   | LR<br>Wald<br>Conditional                           |  |
| probability_entry   | number  |  |
| probability_removal | number  |  |
| output_display      | EachStep<br>LastStep                                |  |
| ci_enable           | フラグ型  |  |

| coxregnode のプロパティ  | 値              | プロパティの説明  |
|--------------------|----------------|---|
| ci_value           | 90<br>95<br>99 |   |
| correlation        | フラグ型           |   |
| display_baseline   | フラグ型           |   |
| survival           | フラグ型           |   |
| hazard             | フラグ型           |   |
| log_minus_log      | フラグ型           |   |
| one_minus_survival | フラグ型           |   |
| separate_line      | フィールド          |   |
| value              | 数値型 または 文字列    | フィールドに対して値の指定がない場合、デフォルト オプションの「Mean」をそのフィールドで使用します。<br>数値型フィールドの使用<br>:coxnode.value = [{"age" "35.8"}]<br>カテゴリ フィールドの使用 :<br>coxnode.value = [{"color" "pink"}] |

## decisionlistnode のプロパティ



ディシジョン リスト ノードは、母集団に関連する与えられた 2 値の結果の高いもしくは低い尤度を示すサブグループまたはセグメントを識別します。たとえば、離れる可能性の少ないもしくはキャンペーンに好意的に答える可能性のある顧客を探することができます。顧客区分を追加し、結果を比較するために他のモデルを並べて表示することによって、ビジネスに関する知識をモデルに導入することができます。ディシジョン リスト モデルは、ルールのリストから構成され、各ルールには条件と結果が含まれます。ルールは順番に適用され、一致する最初のルールで、結果が決まります。詳細は、[9 章 ディシジョン リスト in IBM SPSS Modeler 15 モデル作成ノード](#) を参照してください。

### 例

```
create decisionlistnode
set :decisionlistnode.search_direction=Down
set :decisionlistnode.target_value=1
set :decisionlistnode.max_rules=4
```

## モデル作成ノードのプロパティ

set :decisionlistnode.min\_group\_size\_pct = 15

| decisionlistnode のプロパティ         | 値                           | プロパティの説明   |
|---------------------------------|-----------------------------|--|
| target                          | フィールド                       | ディシジョン リスト モデルは 1 つの対象フィールドおよび 1 つ以上の入力フィールドを使用します。度数フィールドも指定できません。詳細は、p. 213 一般的なモデル作成ノードのプロパティを参照してください。 |
| model_output_type               | Model<br>InteractiveBuilder |  |
| search_direction                | Up<br>Down                  | セグメントの検索に関連します。Up は、高い確率の検索、Down は低い確率の検索と同じです。  |
| target_value                    | string                      | 指定しない場合は、フラグには真の値が想定されます。  |
| max_rules                       | integer                     | 残りを除外するセグメントの最大数   |
| min_group_size                  | integer                     | 最小セグメント サイズ :  |
| min_group_size_pct              | number                      | 最小セグメント サイズ (パーセントとして)。  |
| confidence_level                | number                      | セグメント定義に追加するためにふさわしくするために、応答の尤度を向上するために入力フィールドが持つ最小しきい値。   |
| max_segments_per_rule           | integer                     |  |
| mode                            | Simple<br>Expert            |  |
| bin_method                      | EqualWidth<br>EqualCount    |  |
| bin_count                       | number                      |  |
| max_models_per_cycle            | integer                     | リストの検索幅。   |
| max_rules_per_cycle             | integer                     | セグメント ルールの検索幅。   |
| segment_growth                  | number                      |  |
| include_missing                 | フラグ型                        |  |
| final_results_only              | フラグ型                        |  |
| reuse_fields                    | フラグ型                        | 属性 (ルールに表示される入力フィールド) の再使用を許可します。  |
| max_alternatives                | integer                     |  |
| calculate_raw_propensities      | フラグ型                        |  |
| calculate_adjusted_propensities | フラグ型                        |  |
| adjusted_propensity_partition   | Test<br>Validation          |  |

## discriminantnode のプロパティ



判別分析によって、ロジスティック回帰より厳密な仮説を立てることができますが、これらの仮説が一致した場合、ロジスティック回帰分析に対する様々な代替あるいは補足になります。詳細は、10 章 判別分析ノード in IBM SPSS Modeler 15 モデル作成ノード を参照してください。

### 例

```
create discriminantnode
set :discriminantnode.target = custcat
set :discriminantnode.use_partitioned_data = False
set :discriminantnode.method = Stepwise
```

| discriminantnode のプロパティ    | 値                              | プロパティの説明   |
|----------------------------|--------------------------------|--|
| target                     | フィールド                          | 判別分析 モデルは単一の対象フィールドおよび 1 つ以上の入力フィールドを使用します。重みフィールドおよび度数フィールドは使用しません。詳細は、p. 213 一般的なモデル作成ノードのプロパティ を参照してください。 |
| method                     | Enter<br>Stepwise              |  |
| mode                       | Simple<br>Expert               |  |
| prior_probabilities        | AllEqual<br>ComputeFromSizes   |  |
| covariance_matrix          | WithinGroups<br>SeparateGroups |  |
| means                      | フラグ型                           | [詳細出力] ダイアログ ボックスの統計オプション  |
| univariate_anovas          | フラグ型                           |  |
| box_m                      | フラグ型                           |  |
| within_group_covariance    | フラグ型                           |  |
| within_groups_correlation  | フラグ型                           |  |
| separate_groups_covariance | フラグ型                           |  |
| total_covariance           | フラグ型                           |  |
| fishers                    | フラグ型                           |  |
| unstandardized             | フラグ型                           |  |
| casewise_results           | フラグ型                           | [詳細出力] ダイアログ ボックスの統計オプション  |
| limit_to_first             | number                         | デフォルト値は 10 です。   |
| summary_table              | フラグ型                           |  |

| discriminantnode のプロパティ         | 値   | プロパティの説明          |
|---------------------------------|---|-------------------|
| leave_one_classification        | フラグ型  |                   |
| combined_groups                 | フラグ型  |                   |
| separate_groups_covariance      | フラグ型  | 個別グループ共分散 行列オプション |
| territorial_map                 | フラグ型  |                   |
| combined_groups                 | フラグ型  | 結合グループ散布図オプション    |
| separate_groups                 | フラグ型  | 個別グループ散布図オプション    |
| summary_of_steps                | フラグ型  |                   |
| F_pairwise                      | フラグ型  |                   |
| stepwise_method                 | WilksLambda<br>UnexplainedVariance<br>MahalanobisDistance<br>SmallestF<br>RaosV |                   |
| V_to_enter                      | number  |                   |
| criteria                        | UseValue<br>UseProbability  |                   |
| F_value_entry                   | number  | デフォルト値は 3.84 です。  |
| F_value_removal                 | number  | デフォルト値は 2.71 です。  |
| probability_entry               | number  | デフォルト値は 0.05 です。  |
| probability_removal             | number  | デフォルト値は 0.10 です。  |
| calculate_variable_importance   | フラグ型  |                   |
| calculate_raw_propensities      | フラグ型  |                   |
| calculate_adjusted_propensities | フラグ型  |                   |
| adjusted_propensity_partition   | Test<br>Validation  |                   |

## factornode のプロパティ



因子分析ノードには、データの複雑性を整理する強力なデータ分解手法が 2 種類あります。主成分分析 (PCA) : 入力フィールドの線型結合が検出されます。成分が互いに直交する (直角に交わる) 場合に、フィールドのセット全体の分散を把握するのに役立ちます。因子分析 : 一連の観測フィールド内の相関パターンを説明する基本因子が識別されます。どちらの手法でも、元のフィールド セットの情報を効果的に要約する少数の派生フィールドの検出が目標です。詳細は、[10 章 因子分析ノード in IBM SPSS Modeler 15 モデル作成ノード](#) を参照してください。

### 例

```
create factornode
# "Fields" tab
set :factornode.custom_fields = True
set :factornode.inputs = ['BP' 'Na' 'K']
```

```

set :factornode.partition = Test
# "Model" tab
set :factornode.use_model_name = True
set :factornode.model_name = "Factor_Age"
set :factornode.use_partitioned_data = False
set :factornode.method = GLS
# Expert options
set :factornode.mode = Expert
set :factornode.complete_records = true
set :factornode.matrix = Covariance
set :factornode.max_iterations = 30
set :factornode.extract_factors = ByFactors
set :factornode.min_eigenvalue = 3.0
set :factornode.max_factor = 7
set :factornode.sort_values = True
set :factornode.hide_values = True
set :factornode.hide_below = 0.7
# "Rotation" section
set :factornode.rotation = DirectOblimin
set :factornode.delta = 0.3
set :factornode.kappa = 7.0

```

| factornode のプロパティ | 値   | プロパティの説明  |
|-------------------|---|---|
| inputs            | [フィールド 1 ...<br>フィールド N]                        | 主成分分析/因子モデルは対象フィールドでなく、入力フィールドのリストを使用します。重みフィールドおよび度数フィールドは使用しません。詳細は、 <a href="#">p. 213 一般的なモデル作成ノードのプロパティ</a> を参照してください。 |
| method            | PC<br>ULS<br>GLS<br>ML<br>PAF<br>Alpha<br>Image |   |
| mode              | Simple<br>Expert                                |   |
| max_iterations    | number  |   |
| complete_records  | フラグ型  |   |
| matrix            | Correlation<br>Covariance                       |   |
| extract_factors   | ByEigenvalues<br>ByFactors                      |   |
| min_eigenvalue    | number  |   |
| max_factor        | number  |   |

| factornode のプロパティ | 値  | プロパティの説明   |
|-------------------|--|--|
| rotation          | None<br>Varimax<br>DirectOblimin<br>Equamax<br>Quartimax<br>Promax |  |
| delta             | number   | rotation でDirectObliminを選択した場合、delta の値を指定できる。値を指定しない場合は、delta のデフォルト値を使用。 |
| kappa             | number   | rotation でPromaxを選択した場合、kappa の値を指定できる。値を指定しない場合は、kappa のデフォルト値を使用。        |
| sort_values       | フラグ型   |  |
| hide_values       | フラグ型   |  |
| hide_below        | number   |  |

## featureselectionnode のプロパティ



フィールド選択ノードで、(欠損値の割合などの) 諸基準に基づいて入力フィールドをスクリーニングして除去にかけ、指定した目標に相対的な残りの入力フィールドの重要度をランク付けします。たとえば、数百の潜在的入力フィールドを含むデータセットがあるとして、患者予後のモデリングにはどれが役に立つのでしょうか？ 詳細は、[4 章 フィールド選択ノード in IBM SPSS Modeler 15 モデル作成ノード](#) を参照してください。

### 例

```
create featureselectionnode
set :featureselectionnode.screen_single_category=true
set :featureselectionnode.max_single_category=95
set :featureselectionnode.screen_missing_values=true
set :featureselectionnode.max_missing_values=80
set :featureselectionnode.criteria = Likelihood
set :featureselectionnode.unimportant_below = 0.8
set :featureselectionnode.important_above = 0.9
set :featureselectionnode.important_label = "Check Me Out!"
set :featureselectionnode.selection_mode = TopN
set :featureselectionnode.top_n = 15
```

フィールド選択モデルを作成して適用する詳細な例は、2 章（ p.16 ）の  
 スタンドアロン スクリプトの例：フィールド選択モデルの生成 を参照  
 してください。

| featureselectionnode のプロパティ | 値   | プロパティの説明   |
|-----------------------------|---|--|
| target                      | フィールド                                       | フィールド選択モデルは指定対象に関連した予測フィールドをランク付けします。重みフィールドおよび度数フィールドは使用しません。詳細は、 p.213 一般的なモデル作成ノードのプロパティ を参照してください。 |
| screen_single_category      | フラグ型  | True の場合、総レコード数に比べ同じカテゴリに多くかたよったレコードを持つフィールドを選別します。  |
| max_single_category         | number                                      | screen_single_category が True の場合に使用される閾値を指定します。   |
| screen_missing_values       | フラグ型  | True の場合、レコードの総数のパーセントで表すレコード数になるまで、多すぎる欠損値フィールドをスクリーニング(選別)します。                                       |
| max_missing_values          | number                                      |  |
| screen_num_categories       | フラグ型  | True の場合、レコードの総数に対して多すぎるカテゴリを減らす目的で、フィールドをスクリーニング(選別)します。  |
| max_num_categories          | number                                      |  |
| screen_std_dev              | フラグ型  | True の場合、指定された最小値以下の標準偏差で、フィールドをスクリーニング (選別) します。  |
| min_std_dev                 | number                                      |  |
| screen_coeff_of_var         | フラグ型  | True の場合、指定された最小値以下の分散係数で、フィールドをスクリーニング (選別) します。  |
| min_coeff_of_var            | number                                      |  |
| criteria                    | Pearson<br>Likelihood<br>CramersV<br>Lambda | カテゴリ対象に対するカテゴリ予測値のランク付けのときに、重要な値が基準とする測定単位を指定します。  |
| unimportant_below           | number                                      | 重要、境界、非重要として変数をランク付けするときに使用される閾値 p を指定します。0.0 ~ 1.0 の値を指定します。  |
| important_above             | number                                      | 0.0 ~ 1.0 の値を指定します。  |



## モデル作成ノードのプロパティ

| featureselectionnode のプロパティ | 値  | プロパティの説明   |
|-----------------------------|--|--|
| unimportant_label           | string                                     | 非重要ランクのラベルを指定します。  |
| marginal_label              | string                                     |  |
| important_label             | string                                     |  |
| selection_mode              | ImportanceLevel<br>ImportanceValue<br>TopN |  |
| select_important            | フラグ型                                       | selection_mode が ImportanceLevel に設定されているときに、重要なフィールドを選択するかどうかを指定します。        |
| select_marginal             | フラグ型                                       | selection_mode が ImportanceLevel に設定されているときに、境界フィールドを選択するかどうかを指定します。         |
| select_unimportant          | フラグ型                                       | selection_mode が ImportanceLevel に設定されているときに、重要でないフィールドを選択するかどうかを指定します。      |
| importance_value            | number                                     | selection_mode が ImportanceValue に設定されているときに、使用する分割値を指定します。0 ~ 100 の値を指定します。 |
| top_n                       | integer                                    | selection_mode が TopN に設定されているときに、使用する分割値を指定します。0 ~ 1000 の値を指定します。           |

## genlinnode のプロパティ



一般化線型モデルは、指定したリンク関数によって従属変数が因子および共変量と線型関係になるよう、一般線型モデルを拡張したものです。さらにこのモデルでは、非正規分布の従属変数を使用することができます。線型回帰、ロジスティック回帰、カウント データに関するログ線型モデル、そして区間打ち切り生存モデルなど、統計モデルの機能が数多く含まれています。詳細は、10 章 GenLin ノード in IBM SPSS Modeler 15 モデル作成ノードを参照してください。

## 例

```
create genlinnode
set :genlinnode.model_type = MainAndAllTwoWayEffects
set :genlinnode.offset_type = Variable
```

set :genlinnode.offset\_field = Claimant

| genlinnode のプロパティ        | 値  | プロパティの説明   |
|--------------------------|--|--|
| target                   | フィールド  | 一般化線型モデルは、名義型またはフラグ型の 1 つの対象フィールドおよび 1 つ以上の入力フィールドが必要です。重みフィールドも指定できます。詳細は、 <a href="#">p. 213 一般的なモデル作成ノードのプロパティ</a> を参照してください。 |
| use_weight               | フラグ型   |  |
| weight_field             | フィールド  | フィールドのデータ型は連続型だけです。  |
| target_represents_trials | フラグ型   |  |
| trials_type              | Variable<br>FixedValue   |  |
| trials_field             | フィールド  | フィールドのデータ型はフラグ型または順序型です。   |
| trials_number            | number   | デフォルト値は 10 です。   |
| model_type               | MainEffects<br>MainAndAllTwoWayEffects   |  |
| offset_type              | Variable<br>FixedValue   |  |
| offset_field             | フィールド  | フィールドのデータ型は連続型だけです。  |
| offset_value             | number   | 実数である必要があります。  |
| base_category            | Last<br>First  |  |
| include_intercept        | フラグ型   |  |
| mode                     | Simple<br>Expert   |  |
| distribution             | BINOMIAL<br>GAMMA<br>IGAUSS<br>NEGBIN<br>NORMAL<br>POISSON<br>TWEEDIE<br>MULTINOMIAL | IGAUSS: 逆ガウス。<br>NEGBIN: 負の 2 項分布。   |
| negbin_para_type         | Specify<br>Estimate  |  |
| negbin_parameter         | number   | デフォルト値は 1 で、負でない実数を含む必要があります。  |
| tweedie_parameter        | number   |  |

## モデル作成ノードのプロパティ

| genlinnode のプロパティ         | 値   | プロパティの説明  |
|---------------------------|---|---|
| link_function             | IDENTITY<br>CLOGLOG<br>LOG<br>LOGC<br>LOGIT<br>NEGBIN<br>NLOGLOG<br>ODDSPOWER<br>PROBIT<br>POWER<br>CUMCAUCHIT<br>CUMCLOGLOG<br>CUMLOGIT<br>CUMNLOGLOG<br>CUMPROBIT | CLOGLOG: 補ログ・マイナス・ログ。<br>LOGC: 対数の補数。<br>NEGBIN: 負の 2 項分布。<br>NLOGLOG: 負ログ・マイナス・ログ。<br>CUMCAUCHIT: 累積コーチット。<br>CUMCLOGLOG: 累積補ログ マイナス ログ。<br>CUMLOGIT: 累積ロジット。<br>CUMNLOGLOG: 累積負ログ マイナス ログ。<br>CUMPROBIT: 累積プロビット。 |
| power                     | number  | 値は 0 でない実数である必要があります。   |
| method                    | Hybrid<br>Fisher<br>NewtonRaphson   |   |
| max_fisher_iterations     | number  | デフォルト値は 1 です。正の整数値だけが使用できます。  |
| scale_method              | MaxLikelihoodEstimate<br>Deviance<br>PearsonChiSquare<br>FixedValue   |   |
| scale_value               | number  | デフォルト値は 1 です。0 を超える必要があります。   |
| covariance_matrix         | ModelEstimator<br>RobustEstimator   |   |
| max_iterations            | number  | デフォルト値は 100 です。0 以上の整数値だけを使用できます。   |
| max_step_halving          | number  | デフォルト値は 5 です。正の整数値だけが使用できます。  |
| check_separation          | フラグ型  |   |
| start_iteration           | number  | デフォルト値は 20 です。正の整数値だけが使用できます。   |
| estimates_change          | フラグ型  |   |
| estimates_change_min      | number  | デフォルト値は 1E-006 です。正の数値だけが使用できます。  |
| estimates_change_type     | Absolute<br>Relative  |   |
| loglikelihood_change      | フラグ型  |   |
| loglikelihood_change_min  | number  | 正の数値だけが使用できます。  |
| loglikelihood_change_type | Absolute<br>Relative  |   |
| hessian_convergence       | フラグ型  |   |

| genlinnode のプロパティ               | 値  | プロパティの説明                     |
|---------------------------------|--|------------------------------|
| hessian_convergence_min         | number   | 正の数値だけが使用できます。               |
| hessian_convergence_type        | Absolute<br>Relative                                     |                              |
| case_summary                    | フラグ型   |                              |
| contrast_matrices               | フラグ型   |                              |
| descriptive_statistics          | フラグ型   |                              |
| estimable_functions             | フラグ型   |                              |
| model_info                      | フラグ型   |                              |
| iteration_history               | フラグ型   |                              |
| goodness_of_fit                 | フラグ型   |                              |
| print_interval                  | number   | デフォルト値は 1 です。正の整数である必要があります。 |
| model_summary                   | フラグ型   |                              |
| lagrange_multiplier             | フラグ型   |                              |
| parameter_estimates             | フラグ型   |                              |
| include_exponential             | フラグ型   |                              |
| covariance_estimates            | フラグ型   |                              |
| correlation_estimates           | フラグ型   |                              |
| analysis_type                   | TypeI<br>TypeIII<br>TypeIAndTypeIII                      |                              |
| statistics                      | Wald<br>LR   |                              |
| citype                          | Wald<br>Profile  |                              |
| tolerancelevel                  | number   | デフォルト値は 0.0001 です。           |
| confidence_interval             | number   | デフォルト値は 95 です。               |
| loglikelihood_function          | Full<br>Kernel   |                              |
| singularity_tolerance           | 1E-007<br>1E-008<br>1E-009<br>1E-010<br>1E-011<br>1E-012 |                              |
| value_order                     | Ascending<br>Descending<br>DataOrder                     |                              |
| calculate_variable_importance   | フラグ型   |                              |
| calculate_raw_propensities      | フラグ型   |                              |
| calculate_adjusted_propensities | フラグ型   |                              |
| adjusted_propensity_partition   | Test<br>Validation                                       |                              |

## glmmnode のプロパティ



一般化線形混合モデル (GLMN) は線形モデルを拡張したため、対象が非正規分布となる場合があります。指定されたリンク関数を介して因子および共変量に線形に関連し、観測が関連できるようになりました。一般化線形混合モデルは、単純な線形回帰から非正常な時系列データの複雑なマルチレベルに至るまで、さまざまなモデルをカバーします。詳細は、10 章 GLMM ノード in IBM SPSS Modeler 15 モデル作成ノード を参照してください。

| glmmnode のプロパティ             | 値   | プロパティの説明  |
|-----------------------------|---|---|
| residual_subject_spec       | 構造化   | 指定したカテゴリ型フィールドの組み合わせにより、データセット内の被験者が一意に定義されることが必要です。  |
| repeated_measures           | 構造化   | 反復する観察の特定に使用されるフィールド。   |
| residual_group_spec         | [フィールド 1 ...<br>フィールド N]  | 反復効果共変量パラメータの独立セットを定義するフィールド。   |
| residual_covariance_type    | Diagonal<br>AR1<br>ARMA11<br>COMPOUND_SYMMETRY<br>IDENTITY<br>TOEPLITZ<br>UNSTRUCTURED<br>VARIANCE_COMPONENTS | 残差の共変量構造を指定します。   |
| custom_target               | フラグ型  | 上流のノードで定義された対象を使用するか ( <b>false</b> ) または <b>target_field</b> によって指定されたカスタム対象を使用するか ( <b>true</b> ) を定義します。 |
| target_field                | フィールド   | <b>custom_target</b> が <b>true</b> の場合対象として使用するフィールド。   |
| use_trials                  | フラグ型  | 試行回数を指定する追加フィールド又は値を、対象フィールドが一連の試行が発生する様々なイベントである場合に使用するかどうかを示します。デフォルトは <b>false</b> です。                   |
| use_field_or_value          | Field<br>Value  | フィールドまたは値を使用して試行回数を指定するかどうかを示します。   |
| trials_field                | フィールド   | 試行回数の指定に使用するフィールド。  |
| trials_value                | integer   | 試行回数の指定に使用する値。指定する場合、最小値は 1 です。   |
| use_custom_target_reference | フラグ型  | カスタム参照カテゴリをカテゴリ型対象に使用するかどうかを示します。デフォルトは <b>false</b> です。  |

| glmmnode のプロパティ         | 値  | プロパティの説明   |
|-------------------------|--|--|
| target_reference_value  | string   | use_custom_target_reference が true の場合使用する参照カテゴリ。  |
| dist_link_combination   | Nominal<br>Logit<br>GammaLog<br>BinomialLogit<br>PoissonLog<br>BinomialProbit<br>NegbinLog<br>BinomialLogC<br>Custom | 対象の値の分布に関する一般モデル。Custom を選択して、target_distribution で提供されたリストから分布を指定します。                                     |
| target_distribution     | Normal<br>Binomial<br>Multinomial<br>Gamma<br>Inverse<br>NegativeBinomial<br>Poisson                                 | dist_link_combination が Custom の場合の対象の値の分布。  |
| link_function_type      | Identity<br>LogC<br>Log<br>CLOGLOG<br>Logit<br>NLOGLOG<br>PROBIT<br>POWER<br>CAUCHIT                                 | 対象値を予測値に関連付けるリンク関数。  |
| link_function_param     | number   | 使用するリンク関数パラメータ値。link_function が POWER の場合のみ適用されます。   |
| use_predefined_inputs   | フラグ型   | 固定効果フィールドを入力フィールドとして上流で定義されたフィールドとするか (true) fixed_effects_list のフィールドとするか (false) を指定します。デフォルトは false です。 |
| fixed_effects_list      | 構造化  | use_predefined_inputs が false の場合、固定効果フィールドとして使用する入力フィールドを指定します。   |
| use_intercept           | フラグ型   | true (デフォルト) の場合、モデルに定数項を含みます。   |
| random_effects_list     | 構造化  | ランダム効果として指定するフィールドのリスト。  |
| regression_weight_field | フィールド  | 分析の重みフィールドとして使用するフィールド。  |
| use_offset              | None<br>offset_value<br>offset_field   | オフセットを指定する方法を示します。値 None は、オフセットが使用されないことを意味します。   |

## モデル作成ノードのプロパティ

| glmmnode のプロパティ                | 値                               | プロパティの説明   |
|--------------------------------|---------------------------------|--|
| offset_value                   | number                          | use_offset が offset_value の場合オフセットに使用する値。                                    |
| offset_field                   | フィールド                           | use_offset が offset_field の場合オフセット値に使用する値。                                   |
| target_category_order          | Ascending<br>Descending<br>Data | カテゴリ型対象のソート順。値 Data は、データ内のソート順を使用するよう指定します。デフォルトは Ascending です。             |
| inputs_category_order          | Ascending<br>Descending<br>Data | カテゴリ型予測値のソート順。値 Data は、データ内のソート順を使用するよう指定します。デフォルトは Ascending です。            |
| max_iterations                 | integer                         | アルゴリズムで実行される反復の最大回数です。負の数ではない整数。デフォルト値は 100 です。                              |
| confidence_level               | integer                         | モデル係数の区間推定の計算に使用する確信度。負の数ではない整数。最小値は 100、デフォルト値は 95 です。                      |
| degrees_of_freedom_method      | Fixed<br>Varied                 | 自由度が有意性検定に計算される方法を指定します。   |
| test_fixed_effects_coeficients | Model<br>Robust                 | パラメータ推定共変量マトリクスを計算する方法。  |
| use_model_name                 | フラグ型                            | モデルのカスタム名を使用するか (true) システムによって生成された名前を使用するか (false) を指定します。デフォルトは false です。 |
| model_name                     | string                          | use_model_name が true のときに、使用するモデルを指定します。                                    |
| confidence                     | highest<br>difference           | スコアリングの確信度を計算する基準 (最も高い予測確率、または最も高い予測確率と 2 番目に高い予測確率との差)。                    |
| score_category_probabilities   | フラグ型                            | true の場合、カテゴリ型対象の予測確率を生成します。デフォルトは false です。                                 |
| max_categories                 | integer                         | score_category_probabilities が true のときに、使用するカテゴリの最大数を指定します。                 |
| score_propensity               | フラグ型                            | true の場合、フィールドの「true」の結果の確率を示すフラグ型対象フィールドの傾向スコアを生成します。                       |
| emeans                         | structure                       | 固定効果リストの各カテゴリ型フィールドについて、推定周辺平均を生成するかどうかを指定します。                               |

| glimmnode のプロパティ             | 値                                | プロパティの説明  |
|------------------------------|----------------------------------|---|
| covariance_list              | structure                        | 固定効果リストの各カテゴリ型フィールドについて、推定周辺平均を計算する場合に平均値を使用するかカスタム値を使用するかを指定します。 |
| mean_scale                   | Original<br>Transformed          | 対象の元の尺度に基づいて（デフォルト）、またはリンク関数変換に基づいて推定周辺平均を計算するかどうかを指定します。         |
| comparison_adjustment_method | LSD<br>SEQBONFERRONI<br>SEQSIDAK | 複数の対比で仮定検定を実行する場合に使用する調整方法。                                       |

## kmeansnode のプロパティ



K-Means ノードで、データ セットが異なるグループ（つまりクラスター）へ、クラスタリングされます。この方法で、固定数のクラスターを定義し、クラスターにレコードを繰り返し割り当てて、これ以上調整してもモデルが改善されなくなるまで、クラスターの中心を調整します。K-means では、結果を予測するのではなく、「非監視学習」として知られるプロセスが使用されます。詳細は、11 章 [K-Means ノード in IBM SPSS Modeler 15 モデル作成ノード](#) を参照してください。

### 例

```
create kmeansnode
# "Fields" tab
set :kmeansnode.custom_fields = True
set :kmeansnode.inputs = ['Cholesterol' 'BP' 'Drug' 'Na' 'K' 'Age']
# "Model" tab
set :kmeansnode.use_model_name = False
set :kmeansnode.model_name = "Kmeans_allinputs"
set :kmeansnode.num_clusters = 9
set :kmeansnode.gen_distance = True
set :kmeansnode.cluster_label = "Number"
set :kmeansnode.label_prefix = "Kmeans_"
set :kmeansnode.optimize = Speed
# "Expert" tab
set :kmeansnode.mode = Expert
set :kmeansnode.stop_on = Custom
set :kmeansnode.max_iterations = 10
set :kmeansnode.tolerance = 3.0
```



```
set :kmeansnode.encoding_value = 0.3
```

| kmeansnode のプロパティ | 値                        | プロパティの説明  |
|-------------------|--------------------------|---|
| inputs            | [フィールド 1 ...<br>フィールド N] | K-means モデルは入力フィールドのセットでクラスタ分析を行います。対象フィールドは使用しません。重みフィールドおよび度数フィールドは使用しません。詳細は、 <a href="#">p. 213 一般的なモデル作成ノードのプロパティ</a> を参照してください。 |
| num_clusters      | number                   |   |
| gen_distance      | フラグ型                     |   |
| cluster_label     | String<br>Number         |   |
| label_prefix      | string                   |   |
| mode              | Simple<br>Expert         |   |
| stop_on           | Default<br>Custom        |   |
| max_iterations    | number                   |   |
| tolerance         | number                   |   |
| encoding_value    | number                   |   |
| optimize          | Speed<br>Memory          | モデル作成が速度とメモリーのどちらにより最適化されるかを指定します。  |

## knnnode のプロパティ



k が整数である場合、k 最近隣 (KNN) ノードは、新しいケースを、予測領域の新しいケースに最も近い k 個のオブジェクトのカテゴリまたは値と関連付けます。類似したケースはお互いに近く、類似していないケースはお互いに離れています。詳細は、[16 章 KNN ノード in IBM SPSS Modeler 15 モデル作成ノード](#)を参照してください。

### 例

```
create knnnode
# Objectives tab
set:knnnode.objective = Custom
# Settings tab - Neighbors panel
set:knnnode.automatic_k_selection = false
set:knnnode.fixed_k = 2
set:knnnode.weight_by_importance = True
# Settings tab - Analyze panel
```

```
set:knnnode.save_distances = True
```

| knnnode のプロパティ                  | 値                                      | プロパティの説明   |
|---------------------------------|--|--|
| analysis                        | PredictTarget<br>IdentifyNeighbors     |  |
| objective                       | Balance<br>Speed<br>Accuracy<br>Custom |  |
| normalize_ranges                | フラグ型                                   |  |
| use_case_labels                 | フラグ型                                   | 次のオプションを有効化する<br>チェック ボックス。                            |
| case_labels_field               | フィールド                                  |  |
| identify_focal_cases            | フラグ型                                   | 次のオプションを有効化する<br>チェック ボックス。                            |
| focal_cases_field               | フィールド                                  |  |
| automatic_k_selection           | フラグ型                                   |  |
| fixed_k                         | integer                                | automatic_k_selectio が False の場合<br>にのみ有効です。           |
| minimum_k                       | integer                                | automatic_k_selectio が True の場合<br>にのみ有効です。            |
| maximum_k                       | integer                                |  |
| distance_computation            | Euclidean<br>CityBlock                 |  |
| weight_by_importance            | フラグ型                                   |  |
| range_predictions               | Mean<br>Median                         |  |
| perform_feature_selection       | フラグ型                                   |  |
| forced_entry_inputs             | [フィールド 1 ...<br>フィールド N]               |  |
| stop_on_error_ratio             | フラグ型                                   |  |
| number_to_select                | integer                                |  |
| minimum_change                  | number                                 |  |
| validation_fold_assign_by_field | フラグ型                                   |  |
| number_of_folds                 | integer                                | validation_fold_assign_by_field が<br>False の場合にのみ有効です。 |
| set_random_seed                 | フラグ型                                   |  |
| random_seed                     | number                                 |  |
| folds_field                     | フィールド                                  | validation_fold_assign_by_field が True<br>の場合にのみ有効です。  |
| all_probabilities               | フラグ型                                   |  |
| save_distances                  | フラグ型                                   |  |
| calculate_raw_propensities      | フラグ型                                   |  |

| knnnode のプロパティ                  | 値                  | プロパティの説明 |
|---------------------------------|--------------------|----------|
| calculate_adjusted_propensities | フラグ型               |          |
| adjusted_propensity_partition   | Test<br>Validation |          |

## kohonennode のプロパティ



Kohonen ノードは、ニューラル ネットワークの一種であり、データ セットをクラスタ化して異なるグループを形成する目的で使用できます。ネットワークの学習が完了すると、類似のレコードは出力マップで互い近くに表示され、違いの大きいレコードほど離れたところに表示されます。強度の高いユニットを識別するために生成されたモデル内で、各ユニットが獲得した観察の数値を調べることができます。これは、適切なクラスタ数についてのヒントになる場合があります。詳細は、[11 章 Kohonen ノード in IBM SPSS Modeler 15 モデル作成ノード](#) を参照してください。

### 例

```
create kohonennode
# "Model" tab
set :kohonennode.use_model_name = False
set :kohonennode.model_name = "Symbolic Cluster"
set :kohonennode.stop_on = Time
set :kohonennode.time = 1
set :kohonennode.set_random_seed = True
set :kohonennode.random_seed = 12345
set :kohonennode.optimize = Speed
# "Expert" tab
set :kohonennode.mode = Expert
set :kohonennode.width = 3
set :kohonennode.length = 3
set :kohonennode.decay_style = Exponential
set :kohonennode.phase1_neighborhood = 3
set :kohonennode.phase1_eta = 0.5
set :kohonennode.phase1_cycles = 10
set :kohonennode.phase2_neighborhood = 1
set :kohonennode.phase2_eta = 0.2
```

```
set :kohonennode.phase2_cycles = 75
```

| kohonennode のプロパティ  | 値                        | プロパティの説明  |
|---------------------|--------------------------|---|
| inputs              | [フィールド 1 ...<br>フィールド N] | Kohonen モデルは対象フィールドでなく、入力フィールドのリストを使用します。度数フィールドおよび重みフィールドは使用しません。詳細は、p. 213 一般的なモデル作成ノードのプロパティを参照してください。 |
| continue            | フラグ型                     |   |
| show_feedback       | フラグ型                     |   |
| stop_on             | Default<br>Time          |   |
| time                | number                   |   |
| optimize            | Speed<br>Memory          | モデル作成が速度とメモリーのどちらにより最適化されるかを指定します。  |
| cluster_label       | フラグ型                     |   |
| mode                | Simple<br>Expert         |   |
| width               | number                   |   |
| length              | number                   |   |
| decay_style         | Linear<br>Exponential    |   |
| phase1_neighborhood | number                   |   |
| phase1_eta          | number                   |   |
| phase1_cycles       | number                   |   |
| phase2_neighborhood | number                   |   |
| phase2_eta          | number                   |   |
| phase2_cycles       | number                   |   |

## linearnode プロパティ



線型回帰モデルは、対象と 1 つまたは複数の予測値との線型の関係に基づいて連続型対象を予測します。詳細は、10 章 [線型モデル in IBM SPSS Modeler 15 モデル作成ノード](#) を参照してください。

### 例

```
create linearnode
# [作成オプション] タブ - [目的] パネル
set:linearnode.objective = Standard
# [作成オプション] タブ - [モデル選択] パネル
set:linearnode.model_selection = BestSubsets
```

## モデル作成ノードのプロパティ

```
set:linearnode.criteria_best_subsets = ASE
#[作成オプション]タブ-[アンサンブル]パネル
set:linearnode.combining_rule_categorical = HighestMeanProbability
```

| linearnode プロパティ                 | 値   | プロパティの説明                                |
|----------------------------------|---|---|
| target                           | フィールド   | 1 つの対象フィールドを指定します。                      |
| inputs                           | [フィールド 1 ...<br>フィールド N]                      | モデルで使用される入力または入力または予測変数フィールド。           |
| continue_training_existing_model | フラグ型  |   |
| objective                        | Standard<br>Bagging<br>Boosting<br>psm        | psm は非常に大きいデータセットに使用され、Server の接続が必要です。 |
| use_auto_data_preparation        | フラグ型  |   |
| confidence_level                 | 数値型   |   |
| model_selection                  | ForwardStepwise<br>BestSubsets<br>None        |   |
| criteria_forward_stepwise        | AICC<br>Fstatistics<br>AdjustedRSquare<br>ASE |   |
| probability_entry                | 数値型   |   |
| probability_removal              | 数値型   |   |
| use_max_effects                  | フラグ型  |   |
| max_effects                      | 数値型   |   |
| use_max_steps                    | フラグ型  |   |
| max_steps                        | 数値型   |   |
| criteria_best_subsets            | AICC<br>AdjustedRSquare<br>ASE                |   |
| combining_rule_continuous        | Mean<br>Median                                |   |
| component_models_n               | 数値型   |   |
| use_random_seed                  | フラグ型  |   |
| random_seed                      | 数値型   |   |
| use_custom_model_name            | フラグ型  |   |
| custom_model_name                | 文字列   |   |
| use_custom_name                  | フラグ型  |   |
| custom_name                      | 文字列   |   |
| tooltip                          | 文字列   |   |

| linearnode プロパティ | 値   | プロパティの説明 |
|------------------|-----|----------|
| keywords         | 文字列 |          |
| annotation       | 文字列 |          |

## logregnode のプロパティ



ロジスティック回帰は、入力フィールドの値に基づいてレコードを分類する統計手法です。線型回帰と似ていますが、数値範囲ではなくカテゴリ対象フィールドを使用します。詳細は、10章 [ロジスティック ノード in IBM SPSS Modeler 15 モデル作成 ノード](#) を参照してください。

### Multinomial Example

```

create logregnode
# "Fields" tab
set :logregnode.custom_fields = True
set :logregnode.target = 'Drug'
set :logregnode.inputs = ['BP' 'Cholesterol' 'Age']
set :logregnode.partition = Test
# "Model" tab
set :logregnode.use_model_name = False
set :logregnode.model_name = "Log_reg Drug"
set :logregnode.use_partitioned_data = True
set :logregnode.method = Stepwise
set :logregnode.logistic_procedure = Multinomial
set :logregnode.multinomial_base_category = BP
set :logregnode.model_type = FullFactorial
set :logregnode.custom_terms = {{BP Sex}{Age}{Na K}}
set :logregnode.include_constant = False
# "Expert" tab
set :logregnode.mode = Expert
set :logregnode.scale = Pearson
set :logregnode.scale_value = 3.0
set :logregnode.all_probabilities = True
set :logregnode.tolerance = "1.0E-7"
# "Convergence..." section
set :logregnode.max_iterations = 50
set :logregnode.max_steps = 3
set :logregnode.l_converge = "1.0E-3"
set :logregnode.p_converge = "1.0E-7"
set :logregnode.delta = 0.03
# "Output..." section
set :logregnode.summary = True
set :logregnode.likelihood_ratio = True
set :logregnode.asymptotic_correlation = True
set :logregnode.goodness_fit = True

```

```
set :logregnode.iteration_history = True
set :logregnode.history_steps = 3
set :logregnode.parameters = True
set :logregnode.confidence_interval = 90
set :logregnode.asymptotic_covariance = True
set :logregnode.classification_table = True
# "Stepping" options
set :logregnode.min_terms = 7
set :logregnode.use_max_terms = true
set :logregnode.max_terms = 10
set :logregnode.probability_entry = 3
set :logregnode.probability_removal = 5
set :logregnode.requirements = Containment
```

## 二項式のサンプル

```
create logregnode
# "Fields" tab
set :logregnode.custom_fields = True
set :logregnode.target = 'Cholesterol'
set :logregnode.inputs = ['BP' 'Drug' 'Age']
set :logregnode.partition = Test
# "Model" tab
set :logregnode.use_model_name = False
set :logregnode.model_name = "Log_reg Cholesterol"
set :logregnode.multinomial_base_category = BP
set :logregnode.use_partitioned_data = True
set :logregnode.binomial_method = Forwards
set :logregnode.logistic_procedure = Binomial
set :logregnode.binomial_categorical_input = Sex
set :logregnode.binomial_input_contrast.Sex = Simple
set :logregnode.binomial_input_category.Sex = Last
set :logregnode.include_constant = False
# "Expert" tab
set :logregnode.mode = Expert
set :logregnode.scale = Pearson
set :logregnode.scale_value = 3.0
set :logregnode.all_probabilities = True
set :logregnode.tolerance = "1.0E-7"
# "Convergence..." section
set :logregnode.max_iterations = 50
set :logregnode.l_converge = "1.0E-3"
set :logregnode.p_converge = "1.0E-7"
# "Output..." section
set :logregnode.binomial_output_display = at_each_step
set :logregnode.binomial_goodness_fit = True
set :logregnode.binomial_iteration_history = True
set :logregnode.binomial_parameters = True
set :logregnode.binomial_ci_enable = True
```

```

set :logregnode.binomial_ci = 85
# "Stepping" options
set :logregnode.binomial_removal_criterion = LR
set :logregnode.binomial_probability_removal = 0.2

```

| logregnode のプロパティ          | 値   | プロパティの説明   |
|----------------------------|---|--|
| target                     | フィールド   | ロジスティック回帰モデルは 1 つの対象フィールドおよび 1 つ以上の入力フィールドを使用します。度数フィールドおよび重みフィールドは使用しません。詳細は、 <a href="#">p. 213 一般的なモデル作成ノードのプロパティ</a> を参照してください。                                    |
| logistic_procedure         | Binomial<br>Multinomial   |  |
| include_constant           | フラグ型  |  |
| mode                       | Simple<br>Expert  |  |
| method                     | Enter<br>Stepwise<br>Forwards<br>Backwards<br>BackwardsStepwise                     |  |
| binomial_method            | Enter<br>Forwards<br>Backwards  |  |
| model_type                 | MainEffects<br>FullFactorial<br>Custom  | モデルタイプとして FullFactorial が指定されている場合、ステップ手法が指定されたとしても、実行されません。その代わりに、強制投入法 (Enter) が使用されます。モデルタイプに Custom が設定されてもユーザー設定フィールド (custom fields) が指定されていない場合は、主効果モデルが構築されます。 |
| custom_terms               | [ {BP Sex} {BP} {Age} ]   | 例:<br>set :logregnode. custom_terms =<br>{{Na} {K} {Na K}}   |
| multinomial_base_category  | string  | 参照カテゴリの決定方法を指定します。   |
| binomial_categorical_input | string  |  |
| binomial_input_contrast    | Indicator<br>Simple<br>Difference<br>Helmert<br>Repeated<br>Polynomial<br>Deviation | コントラストを決定する方法を指定するカテゴリ入力用のキープロパティ。<br>使用フォーマット：<br>NODE.binomial_input_contrast.FIELD-NAME   |



## モデル作成ノードのプロパティ

| logregnode のプロパティ            | 値   | プロパティの説明   |
|------------------------------|---|--|
| binomial_input_category      | First<br>Last   | 参照カテゴリを決定する方法を指定するカテゴリ入力用のキープロパティ。<br>使用フォーマット：<br>NODE.binomial_input_category.FIELD-NAME |
| scale                        | None<br>UserDefined<br>Pearson<br>Deviance                |  |
| scale_value                  | number  |  |
| all_probabilities            | フラグ型  |  |
| tolerance                    | 1.0E-5<br>1.0E-6<br>1.0E-7<br>1.0E-8<br>1.0E-9<br>1.0E-10 |  |
| min_terms                    | number  |  |
| use_max_terms                | フラグ型  |  |
| max_terms                    | number  |  |
| entry_criterion              | Score<br>LR   |  |
| removal_criterion            | LR<br>Wald  |  |
| probability_entry            | number  |  |
| probability_removal          | number  |  |
| binomial_probability_entry   | number  |  |
| binomial_probability_removal | number  |  |
| requirements                 | HierarchyDiscrete<br>HierarchyAll<br>Containment<br>None  |  |
| max_iterations               | number  |  |
| max_steps                    | number  |  |
| p_converge                   | 1.0E-4<br>1.0E-5<br>1.0E-6<br>1.0E-7<br>1.0E-8<br>0       |  |
| l_converge                   | 1.0E-1<br>1.0E-2<br>1.0E-3<br>1.0E-4<br>1.0E-5<br>0       |  |
| delta                        | number  |  |

| logregnode のプロパティ              | 値                            | プロパティの説明 |
|--------------------------------|------------------------------|----------|
| iteration_history              | フラグ型                         |          |
| history_steps                  | number                       |          |
| summary                        | フラグ型                         |          |
| likelihood_ratio               | フラグ型                         |          |
| asymptotic_correlation         | フラグ型                         |          |
| goodness_fit                   | フラグ型                         |          |
| parameters                     | フラグ型                         |          |
| confidence_interval            | number                       |          |
| asymptotic_covariance          | フラグ型                         |          |
| classification_table           | フラグ型                         |          |
| stepwise_summary               | フラグ型                         |          |
| info_criteria                  | フラグ型                         |          |
| monotonicity_measures          | フラグ型                         |          |
| binomial_output_display        | at_each_step<br>at_last_step |          |
| binomial_goodness_of_fit       | フラグ型                         |          |
| binomial_parameters            | フラグ型                         |          |
| binomial_iteration_history     | フラグ型                         |          |
| binomial_classification_plots  | フラグ型                         |          |
| binomial_ci_enable             | フラグ型                         |          |
| binomial_ci                    | number                       |          |
| binomial_residual              | outliers<br>all              |          |
| binomial_residual_enable       | フラグ型                         |          |
| binomial_outlier_threshold     | number                       |          |
| binomial_classification_cutoff | number                       |          |
| binomial_removal_criterion     | LR<br>Wald<br>Conditional    |          |
| calculate_variable_importance  | フラグ型                         |          |
| calculate_raw_propensities     | フラグ型                         |          |

## neuralnetnode のプロパティ

**注意:** 機能が拡張された新しいバージョンのニューラル ネットワーク ノードがこのリリースで使用できます。新しいバージョンについては次の項で説明します (neuralnetwork)。旧バージョンでモデルを作成およびスコアリングできますが、新しいバージョンを使用するようスクリプトを更新することをお勧めします。以下は旧バージョンの詳細です。

## 例

```

create neuralnetnode
# "Fields" tab
set :neuralnetnode.custom_fields = True
set :neuralnetnode.targets = ['Drug']
set :neuralnetnode.inputs = ['Age' 'Na' 'K' 'Cholesterol' 'BP']
# "Model" tab
set :neuralnetnode.use_partitioned_data = True
set :neuralnetnode.method = Dynamic
set :neuralnetnode.train_pct = 30
set :neuralnetnode.set_random_seed = True
set :neuralnetnode.random_seed = 12345
set :neuralnetnode.stop_on = Time
set :neuralnetnode.accuracy = 95
set :neuralnetnode.cycles = 200
set :neuralnetnode.time = 3
set :neuralnetnode.optimize = Speed
# "Multiple Method Expert Options" section
set :neuralnetnode.m_topologies = "5 30 5; 2 20 3, 1 10 1"
set :neuralnetnode.m_non_pyramids = False
set :neuralnetnode.m_persistence = 100

```

| neuralnetnode のプロパティ | 値  | プロパティの説明  |
|----------------------|--|---|
| targets              | [フィールド 1 ...<br>フィールド N]   | ニューラル ノードには、1 つ以上の対象フィールドと 1 つ以上の入力フィールドが必要です。度数および重みフィールドは無視されます。詳細は、 <a href="#">p. 213 一般的なモデル作成ノードのプロパティ</a> を参照してください。 |
| method               | Quick<br>Dynamic<br>Multiple<br>Prune<br>ExhaustivePrune<br>RBFN |   |
| prevent_overtrain    | フラグ型   |   |
| train_pct            | number   |   |
| set_random_seed      | フラグ型   |   |
| random_seed          | number   |   |
| mode                 | Simple<br>Expert   |   |
| stop_on              | Default<br>Accuracy<br>Cycles<br>Time                            | 停止モード。  |
| accuracy             | number   | 停止精度。   |

| neuralnetnode のプロパティ | 値                   | プロパティの説明  |
|----------------------|---------------------|-----------|
| cycles               | number              | 学習サイクル。   |
| time                 | number              | 学習時間 (分)。 |
| continue             | フラグ型                |           |
| show_feedback        | フラグ型                |           |
| binary_encode        | フラグ型                |           |
| use_last_model       | フラグ型                |           |
| gen_logfile          | フラグ型                |           |
| logfile_name         | string              |           |
| alpha                | number              |           |
| initial_eta          | number              |           |
| high_eta             | number              |           |
| low_eta              | number              |           |
| eta_decay_cycles     | number              |           |
| hid_layers           | One<br>Two<br>Three |           |
| hl_units_one         | number              |           |
| hl_units_two         | number              |           |
| hl_units_three       | number              |           |
| persistence          | number              |           |
| m_topologies         | string              |           |
| m_non_pyramids       | フラグ型                |           |
| m_persistence        | number              |           |
| p_hid_layers         | One<br>Two<br>Three |           |
| p_hl_units_one       | number              |           |
| p_hl_units_two       | number              |           |
| p_hl_units_three     | number              |           |
| p_persistence        | number              |           |
| p_hid_rate           | number              |           |
| p_hid_pers           | number              |           |
| p_inp_rate           | number              |           |
| p_inp_pers           | number              |           |
| p_overall_pers       | number              |           |
| r_persistence        | number              |           |
| r_num_clusters       | number              |           |
| r_eta_auto           | フラグ型                |           |
| r_alpha              | number              |           |
| r_eta                | number              |           |

| neuralnetnode のプロパティ            | 値                  | プロパティの説明  |
|---------------------------------|--------------------|---|
| optimize                        | Speed<br>Memory    | モデル作成が速度とメモリーのどちらにより最適化されるかを指定します。  |
| calculate_variable_importance   | フラグ型               | 注：前回のリリースで使用した <code>sensitivity_analysis</code> プロパティは、このプロパティにより廃止されます。古いプロパティはまだサポートされますが、 <code>calculate_variable_importance</code> をお勧めします。 |
| calculate_raw_propensities      | フラグ型               |   |
| calculate_adjusted_propensities | フラグ型               |   |
| adjusted_propensity_partition   | Test<br>Validation |   |

## neuralnetworknode プロパティ



ニューラル ネットワーク ノードは、人間の脳が情報を処理する方法を単純化したモデルを使用します。ニューラル ネットワーク ノードは、連係する多数の単純な処理単位をシミュレートします。処理単位は、ニューロンを抽象化したものと表現できます。ニューラル ネットワークは強力な一般関数推定法であり、学習させたり、適用するには、最低限の統計学および数学の知識しか必要ありません。

### 例

```
create neuralnetworknode
#[作成オプション]タブ-[目的]パネル
set:neuralnetworknode.objective = Standard
#[作成オプション]タブ-[停止規則]パネル
set:neuralnetworknode.model_selection = BestSubsets
set:neuralnetworknode.criteria_best_subsets = ASE
#[作成オプション]タブ-[アンサンプル]パネル
set:neuralnetworknode.combining_rule_categorical = HighestMeanProbability
```

| neuralnetworknode プロパティ | 値                        | プロパティの説明                      |
|-------------------------|--------------------------|-------------------------------|
| targets                 | [フィールド 1 ...<br>フィールド N] | 対象フィールドを指定します。                |
| inputs                  | [フィールド 1 ...<br>フィールド N] | モデルで使用される入力または入力または予測変数フィールド。 |
| splits                  | [フィールド1 ...<br>フィールドN]   | 分割モデル作成に使用する、フィールドを選択します。     |

| neuralnetworknode プロパティ      | 値  | プロパティの説明  |
|------------------------------|--|---|
| use_partition                | フラグ型   | 区分フィールドが定義される場合、このオプションは学習データ区分からのデータのみがモデル構築に使用されるようになります。 |
| continue                     | フラグ型   | 既存モデルの学習を継続：  |
| objective                    | Standard<br>Bagging<br>Boosting<br>psm                 | psm は非常に大きいデータセットに使用され、Server の接続が必要です。                     |
| method                       | MultilayerPerceptron<br>RadialBasisFunction            |   |
| use_custom_layers            | フラグ型   |   |
| first_layer_units            | 数値型  |   |
| second_layer_units           | 数値型  |   |
| use_max_time                 | フラグ型   |   |
| max_time                     | 数値型  |   |
| use_max_cycles               | フラグ型   |   |
| max_cycles                   | 数値型  |   |
| use_min_accuracy             | フラグ型   |   |
| min_accuracy                 | 数値型  |   |
| combining_rule_categorical   | Voting<br>HighestProbability<br>HighestMeanProbability |   |
| combining_rule_continuous    | Mean<br>Median   |   |
| component_models_n           | 数値型  |   |
| overfit_prevention_pct       | 数値型  |   |
| use_random_seed              | フラグ型   |   |
| random_seed                  | 数値型  |   |
| missing_values               | listwiseDeletion<br>missingValueImputation             |   |
| use_custom_model_name        | フラグ型   |   |
| custom_model_name            | 文字列  |   |
| confidence                   | onProbability<br>onIncrease                            |   |
| score_category_probabilities | フラグ型   |   |
| max_categories               | 数値型  |   |
| score_propensity             | フラグ型   |   |
| use_custom_name              | フラグ型   |   |
| custom_name                  | 文字列  |   |
| tooltip                      | 文字列  |   |

| neuralnetworknode プロパティ | 値   | プロパティの説明 |
|-------------------------|-----|----------|
| keywords                | 文字列 |          |
| annotation              | 文字列 |          |

## questnode のプロパティ



QUEST ノードには、ディビジョン ツリーの構築用に2 分岐の方法が用意されています。これは、大規模な C&R ツリー分析が必要とする処理時間を短縮すると同時に、より多くの分割を可能にする入力値が優先される分類ツリー内の傾向を低減するように設計されています。入力フィールドは、数値範囲（連続型）にできませんが、対象変数はカテゴリでなければなりません。すべての分割は 2 分岐です。詳細は、6 章 QUEST ノード in IBM SPSS Modeler 15 モデル作成ノード を参照してください。

### 例

```
create questnode
set :questnode.custom_fields = True
set :questnode.target = Drug
set :questnode.inputs = [Age Na K Cholesterol BP]
set :questnode.model_output_type = InteractiveBuilder
set :questnode.use_tree_directives = True
set :questnode.mode = Expert
set :questnode.max_surrogates = 5
set :questnode.split_alpha = 0.03
set :questnode.use_percentage = False
set :questnode.min_parent_records_abs = 40
set :questnode.min_child_records_abs = 30
set :questnode.prune_tree = True
set :questnode.use_std_err = True
set :questnode.std_err_multiplier = 3
set :questnode.priors = Custom
set :questnode.custom_priors = [{drugA 0.3}{drugB 0.4}]
set :questnode.adjust_priors = true
```

| questnode のプロパティ                 | 値     | プロパティの説明  |
|----------------------------------|-------|---|
| target                           | フィールド | QUEST モデルは単一の対象フィールドおよび 1 つ以上の入力フィールドを使用します。度数フィールドも指定できます。詳細は、p. 213 一般的なモデル作成ノードのプロパティ を参照してください。 |
| continue_training_existing_model | フラグ型  |   |

| questnode のプロパティ       | 値  | プロパティの説明   |
|------------------------|--|--|
| objective              | Standard<br>Boosting<br>Bagging<br>psm                 | psm は非常に大きいデータセットに使用され、Server の接続が必要です。  |
| model_output_type      | Single<br>InteractiveBuilder                           |  |
| use_tree_directives    | フラグ型   |  |
| tree_directives        | string   |  |
| use_max_depth          | Default<br>Custom                                      |  |
| max_depth              | integer  | 最大ツリー深さ (0 ~ 1000)。<br>use_max_depth = Custom の場合にのみ使用します。                                     |
| prune_tree             | フラグ型   | オーバーフィットしないようにツリーを剪定します。   |
| use_std_err            | フラグ型   | リスクにおける最大差 (標準誤差) を使用します。  |
| std_err_multiplier     | number   | 最大差。   |
| max_surrogates         | number   | 最大代理フィールド :  |
| use_percentage         | フラグ型   |  |
| min_parent_records_pc  | number   |  |
| min_child_records_pc   | number   |  |
| min_parent_records_abs | number   |  |
| min_child_records_abs  | number   |  |
| use_costs              | フラグ型   |  |
| costs                  | 構造化  | 次のフォーマットを使用した構造化プロパティ。<br>[[drugA drugB 1.5] {drugA drugC 2.1}].<br>{ } 内の引数は実際の予測コストです。       |
| priors                 | Data<br>Equal<br>Custom                                |  |
| custom_priors          | 構造化  | 次のフォーマットを使用した構造化プロパティ。<br>set :cartnode.<br>custom_priors =<br>[ { drugA 0.3 } { drugB 0.6 } ] |
| adjust_priors          | フラグ型   |  |
| trails                 | number   | ブーストまたはバグのコンポーネント モデル数。  |
| set_ensemble_method    | Voting<br>HighestProbability<br>HighestMeanProbability | カテゴリ型対象のデフォルト結合ルール。  |
| range_ensemble_method  | Mean<br>Median   | 連続型対象のデフォルト結合ルール。  |



| questnode のプロパティ                | 値                  | プロパティの説明                   |
|---------------------------------|--------------------|----------------------------|
| large_boost                     | フラグ型               | 特に大きなデータセットのブースティングを適用します。 |
| split_alpha                     | number             | 分割の有意水準：                   |
| train_pct                       | number             | オーバーフィット防止セット。             |
| set_random_seed                 | フラグ型               | 結果の複製オプション。                |
| seed                            | number             |                            |
| calculate_variable_importance   | フラグ型               |                            |
| calculate_raw_propensities      | フラグ型               |                            |
| calculate_adjusted_propensities | フラグ型               |                            |
| adjusted_propensity_partition   | Test<br>Validation |                            |

## regressionnode のプロパティ



線型回帰は、データを要約する一般的な統計手法であり、予測された出力値と実際の出力値の違いを最小限にする直線または面を当てはめることにより予測を行います。

注：回帰ノードは、今後のリリースでは線型ノードに置き換えられます。今後、線型回帰には線型モデルを使用することをお勧めします。

### 例

```
create regressionnode
# "Fields" tab
set :regressionnode.custom_fields = True
set :regressionnode.target = 'Age'
set :regressionnode.inputs = ['Na' 'K']
set :regressionnode.partition = Test
set :regressionnode.use_weight = True
set :regressionnode.weight_field = 'Drug'
# "Model" tab
set :regressionnode.use_model_name = False
set :regressionnode.model_name = "Regression Age"
set :regressionnode.use_partitioned_data = True
set :regressionnode.method = Stepwise
set :regressionnode.include_constant = False
# "Expert" tab
set :regressionnode.mode = Expert
set :regressionnode.complete_records = False
set :regressionnode.tolerance = "1.0E-3"
# "Stepping..." section
set :regressionnode.stepping_method = Probability
set :regressionnode.probability_entry = 0.77
```

```

set :regressionnode.probability_removal = 0.88
set :regressionnode.F_value_entry = 7.0
set :regressionnode.F_value_removal = 8.0
# "Output..." section
set :regressionnode.model_fit = True
set :regressionnode.r_squared_change = True
set :regressionnode.selection_criteria = True
set :regressionnode.descriptives = True
set :regressionnode.p_correlations = True
set :regressionnode.collinearity_diagnostics = True
set :regressionnode.confidence_interval = True
set :regressionnode.covariance_matrix = True
set :regressionnode.durbin_watson = True

```

| regressionnode のプロパティ | 値   | プロパティの説明   |
|-----------------------|---|--|
| target                | フィールド   | 回帰モデルは単一の対象フィールドおよび 1 つ以上の入力フィールドを使用します。重みフィールドも指定できます。詳細は、 <a href="#">p. 213 一般的なモデル作成ノードのプロパティ</a> を参照してください。 |
| method                | Enter<br>Stepwise<br>Backwards<br>Forwards  |  |
| include_constant      | フラグ型  |  |
| use_weight            | フラグ型  |  |
| weight_field          | フィールド   |  |
| mode                  | Simple<br>Expert  |  |
| complete_records      | フラグ型  |  |
| tolerance             | 1.0E-1<br>1.0E-2<br>1.0E-3<br>1.0E-4<br>1.0E-5<br>1.0E-6<br>1.0E-7<br>1.0E-8<br>1.0E-9<br>1.0E-10<br>1.0E-11<br>1.0E-12 | 引数には二重引用符を使用します。   |
| stepping_method       | useP<br>useF  | useP: F 値確率を使用<br>useF: F 値を使用   |
| probability_entry     | number  |  |
| probability_removal   | number  |  |
| F_value_entry         | number  |  |

| regressionnode のプロパティ         | 値      | プロパティの説明 |
|-------------------------------|--------|----------|
| F_value_removal               | number |          |
| selection_criteria            | フラグ型   |          |
| confidence_interval           | フラグ型   |          |
| covariance_matrix             | フラグ型   |          |
| collinearity_diagnostics      | フラグ型   |          |
| regression_coefficients       | フラグ型   |          |
| exclude_fields                | フラグ型   |          |
| durbin_watson                 | フラグ型   |          |
| model_fit                     | フラグ型   |          |
| r_squared_change              | フラグ型   |          |
| p_correlations                | フラグ型   |          |
| descriptives                  | フラグ型   |          |
| calculate_variable_importance | フラグ型   |          |

## sequencenode のプロパティ



シーケンス ノードで、シーケンシャルな、または時間経過が伴うデータ内のアソシエーション ルールを検出します。予測可能な順序で起こる傾向にあるアイテム セットのリストを、シーケンスと呼びます。たとえば、顧客がひげそりとアフター シェーブ ローションを購入した場合、その顧客は次の購入時にシェービング クリームを購入する可能性があります。シーケンス ノードは CARMA アソシエーション ルール アルゴリズムに基づいているため、効率的な 2 段階通過法でシーケンスが検出されます。詳細は、[12 章 シーケンス ノード in IBM SPSS Modeler 15 モデル作成ノード](#) を参照してください。

### 例

```
create sequencenode
connect :databasenode to :sequencenode
# "Fields" tab
set :sequencenode.id_field = 'Age'
set :sequencenode.contiguous = True
set :sequencenode.use_time_field = True
set :sequencenode.time_field = 'Date1'
set :sequencenode.content_fields = ['Drug' 'BP']
set :sequencenode.partition = Test
# "Model" tab
set :sequencenode.use_model_name = True
set :sequencenode.model_name = "Sequence_test"
set :sequencenode.use_partitioned_data = False
set :sequencenode.min_supp = 15.0
set :sequencenode.min_conf = 14.0
```

```

set :sequencenode.max_size = 7
set :sequencenode.max_predictions = 5
# "Expert" tab
set :sequencenode.mode = Expert
set :sequencenode.use_max_duration = True
set :sequencenode.max_duration = 3.0
set :sequencenode.use_pruning = True
set :sequencenode.pruning_value = 4.0
set :sequencenode.set_mem_sequences = True
set :sequencenode.mem_sequences = 5.0
set :sequencenode.use_gaps = True
set :sequencenode.min_item_gap = 20.0
set :sequencenode.max_item_gap = 30.0

```

| sequencenode のプロパティ | 値                   | プロパティの説明   |
|---------------------|---------------------|--|
| id_field            | フィールド               | シーケンス モデルを作成するには、ID フィールドを指定する必要があります。さらにオプションで時間フィールドと 1 つ以上の内容フィールドを指定します。重みフィールドおよび度数フィールドは使用しません。詳細は、 <a href="#">p. 213 一般的なモデル作成ノードのプロパティ</a> を参照してください。 |
| time_field          | フィールド               |  |
| use_time_field      | フラグ型                |  |
| content_fields      | [フィールド1 ... フィールドn] |  |
| contiguous          | フラグ型                |  |
| min_supp            | number              |  |
| min_conf            | number              |  |
| max_size            | number              |  |
| max_predictions     | number              |  |
| mode                | Simple<br>Expert    |  |
| use_max_duration    | フラグ型                |  |
| max_duration        | number              |  |
| use_gaps            | フラグ型                |  |
| min_item_gap        | number              |  |
| max_item_gap        | number              |  |
| use_pruning         | フラグ型                |  |
| pruning_value       | number              |  |
| set_mem_sequences   | フラグ型                |  |
| mem_sequences       | integer             |  |

## slrmnode のプロパティ



SLRM（自己学習応答モデル）ノードを使用するとモデルを構築でき、単一または少数の新しいケースを使用して全データを使用するモデルの保持をすることなく、モデルの再見積もりを行うことができます。詳細は、14章 SLRM ノード in IBM SPSS Modeler 15 モデル作成ノード を参照してください。

### 例

```
create slrmnode
set :slrmnode.target = Offer
set :slrmnode.target_response = Response
set :slrmnode.inputs = ['Cust_ID' 'Age' 'Ave_Bal']
```

| slrmnode のプロパティ                  | 値                       | プロパティの説明   |
|----------------------------------|-------------------------|--|
| target                           | フィールド                   | 対象フィールドは名義型またはフラグ型である必要があります。度数フィールドも指定できます。詳細は、p. 213 一般的なモデル作成ノードのプロパティ を参照してください。 |
| target_response                  | フィールド                   | フラグ型である必要があります。  |
| continue_training_existing_model | フラグ型                    |  |
| target_field_values              | フラグ型                    | すべて使用: ソースのすべての値を使用します。<br>指定: 必要な値を選択します。   |
| target_field_values_specify      | [フィールド1 ... フィールドN]     |  |
| include_model_assessment         | フラグ型                    |  |
| model_assessment_random_seed     | number                  | 実数であることが必要です。  |
| model_assessment_sample_size     | number                  | 実数であることが必要です。  |
| model_assessment_iterations      | number                  | 反復数 :  |
| display_model_evaluation         | フラグ型                    |  |
| max_predictions                  | number                  |  |
| randomization                    | number                  |  |
| scoring_random_seed              | number                  |  |
| sort                             | Ascending<br>Descending | 高いスコアまたは低いスコアのどちらを持つオファーが最初に表示されるかを指定します。  |
| model_reliability                | フラグ型                    |  |
| calculate_variable_importance    | フラグ型                    |  |

## statisticsmodelnode のプロパティ



Statistics モデル ノードを使用すると、PMML を作成する IBM® SPSS® Statistics プロシージャを実行してデータを分析および使用することができます。このノードは、ライセンスが与えられた SPSS Statistics のコピーが必要です。詳細は、8 章 Statistics モデル ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード を参照してください。

このノードのプロパティについては、「statisticsmodelnode のプロパティ」( p.344 ) に記載されています。

## svmnode プロパティ



サポート ベクター マシン (SVM) ノードを使用すると、オーバーフィットすることなく、データを 2 つのグループのいずれかに分類することができます。SVM は、非常に多数の入力フィールドを含むデータセットなど、広範なデータセットを処理することができます。詳細は、15 章 SVM ノード in IBM SPSS Modeler 15 モデル作成ノード を参照してください。

### 例

```
create svmnode
# Expert tab
set :svmnode.mode=Expert
set :svmnode.all_probabilities=True
set :svmnode.kernel=Polynomial
set :svmnode.gamma=1.5
```

| svmnode のプロパティ    | 値  | プロパティの説明                             |
|-------------------|--|--------------------------------------|
| all_probabilities | フラグ型   |                                      |
| stopping_criteria | 1.0E-1<br>1.0E-2<br>1.0E-3 (デフォルト)<br>1.0E-4<br>1.0E-5<br>1.0E-6 | 最適化アルゴリズムをいつ停止するかを決定します。             |
| regularization    | number   | C パラメータとしても知られています。                  |
| precision         | number   | 対象フィールドの尺度が Continuous の場合にのみ使用されます。 |
| kernel            | RBF (デフォルト)<br>Polynomial<br>Sigmoid<br>Linear                   | 変換に使用されるカーネル関数のタイプ。                  |

| svmnode のプロパティ                  | 値               | プロパティの説明                                      |
|---------------------------------|-----------------|---|
| rbf_gamma                       | number          | kernel が RBF の場合にのみ使用されます。                    |
| gamma                           | number          | kernel が Polynomial または Sigmoid の場合にのみ使用されます。 |
| bias                            | number          |   |
| degree                          | number          | kernel が Polynomial の場合にのみ使用されます。             |
| calculate_variable_importance   | フラグ型            |   |
| calculate_raw_propensities      | フラグ型            |   |
| calculate_adjusted_propensities | フラグ型            |   |
| adjusted_propensity_partition   | Test Validation |   |

## timeseriesnode のプロパティ



時系列ノードは、時系列から指数平滑法、1 変量の自己回帰型統合移動平均法 (ARIMA)、および多変量 ARIMA (または転送関数) モデルを推測し、将来のパフォーマンスの予測を作成します。時系列ノードは、時間区分ノードによって常に先行される必要があります。詳細は、[13 章 時系列モデル作成ノード in IBM SPSS Modeler 15 モデル作成ノード](#) を参照してください。

### 例

```
create timeseriesnode
set :timeseriesnode.method = Exsmooth
set :timeseriesnode.exsmooth_model_type = HoltLinearTrend
set :timeseriesnode.exsmooth_transformation_type = None
```

| timeseriesnode のプロパティ | 値     | プロパティの説明  |
|-----------------------|-------|---|
| targets               | フィールド | 時系列ノードは、オプションで 1 つ以上の入力フィールドを予測値として使用しながら、1 つ以上の対象フィールドを予測します。度数フィールドおよび重みフィールドは使用しません。詳細は、 <a href="#">p.213 一般的なモデル作成ノードのプロパティ</a> を参照してください。 |
| continue              | フラグ型  |   |

| timeseriesnode のプロパティ            | 値  | プロパティの説明 |
|----------------------------------|--|----------|
| method                           | ExpertModeler<br>Exsmooth<br>Arima<br>Reuse  |          |
| expert_modeler_method            | フラグ型   |          |
| consider_seasonal                | フラグ型   |          |
| detect_outliers                  | フラグ型   |          |
| expert_outlier_additive          | フラグ型   |          |
| expert_outlier_level_shift       | フラグ型   |          |
| expert_outlier_innovational      | フラグ型   |          |
| expert_outlier_level_shift       | フラグ型   |          |
| expert_outlier_transient         | フラグ型   |          |
| expert_outlier_seasonal_additive | フラグ型   |          |
| expert_outlier_local_trend       | フラグ型   |          |
| expert_outlier_additive_patch    | フラグ型   |          |
| exsmooth_model_type              | Simple<br>HoltsLinearTrend<br>BrownsLinearTrend<br>DampedTrend<br>SimpleSeasonal<br>WintersAdditive<br>WintersMultiplicative |          |
| exsmooth_transformation_type     | None<br>SquareRoot<br>NaturalLog   |          |
| arima_p                          | integer  |          |
| arima_d                          | integer  |          |
| arima_q                          | integer  |          |
| arima_sp                         | integer  |          |
| arima_sd                         | integer  |          |
| arima_sq                         | integer  |          |
| arima_transformation_type        | None<br>SquareRoot<br>NaturalLog   |          |
| arima_include_constant           | フラグ型   |          |
| tf_arima_p.fieldname             | integer  | 転送関数用。   |
| tf_arima_d.fieldname             | integer  | 転送関数用。   |
| tf_arima_q.fieldname             | integer  | 転送関数用。   |
| tf_arima_sp.fieldname            | integer  | 転送関数用。   |
| tf_arima_sd.fieldname            | integer  | 転送関数用。   |
| tf_arima_sq.fieldname            | integer  | 転送関数用。   |
| tf_arima_delay.fieldname         | integer  | 転送関数用。   |



| timeseriesnode のプロパティ                 | 値                                | プロパティの説明                   |
|---------------------------------------|----------------------------------|----------------------------|
| tf_arma_transformation_type.fieldname | None<br>SquareRoot<br>NaturalLog | 転送関数用。                     |
| arma_detect_outlier_mode              | None<br>Automatic                |                            |
| arma_outlier_additive                 | フラグ型                             |                            |
| arma_outlier_level_shift              | フラグ型                             |                            |
| arma_outlier_innovational             | フラグ型                             |                            |
| arma_outlier_transient                | フラグ型                             |                            |
| arma_outlier_seasonal_additive        | フラグ型                             |                            |
| arma_outlier_local_trend              | フラグ型                             |                            |
| arma_outlier_additive_patch           | フラグ型                             |                            |
| conf_limit_pct                        | real                             |                            |
| max_lags                              | integer                          |                            |
| events                                | fields                           |                            |
| scoring_model_only                    | フラグ型                             | 多く (1 万単位) の時系列のモデルに使用します。 |

## twostepnode のプロパティ



TwoStep ノードで、2 段階のクラスタ化手法が使用されます。最初のステップでは、データを 1 度通過させて、未処理の入力データを管理可能な一連のサブクラスタに圧縮します。2 番目のステップでは、階層クラスタ化手法を使用して、サブクラスタをより大きなクラスタに結合させていきます。TwoStep には、学習データに最適なクラスタ数を自動的に推定するという利点があります。また、フィールド タイプの混在や大規模データ セットも効率よく処理できます。詳細は、11 章 [TwoStep クラスタ ノード in IBM SPSS Modeler 15 モデル作成ノード](#) を参照してください。

### 例

```
create twostep
set :twostep.custom_fields = True
set :twostep.inputs = ['Age' 'K' 'Na' 'BP']
set :twostep.partition = Test
set :twostep.use_model_name = False
set :twostep.model_name = "TwoStep_Drug"
set :twostep.use_partitioned_data = True
set :twostep.exclude_outliers = True
set :twostep.cluster_label = "String"
set :twostep.label_prefix = "TwoStep_"
set :twostep.cluster_num_auto = False
set :twostep.max_num_clusters = 9
```

```
set :twostep.min_num_clusters = 3
set :twostep.num_clusters = 7
```

| twostepnode のプロパティ   | 値                          | プロパティの説明  |
|----------------------|----------------------------|---|
| inputs               | [フィールド 1 ...<br>フィールド N]   | TwoStep モデルは対象フィールドでなく、入力フィールドのリストを使用します。重みフィールドおよび度数フィールドは認識されません。詳細は、 <a href="#">p.213 一般的なモデル作成ノードのプロパティ</a> を参照してください。 |
| standardize          | フラグ型                       |   |
| exclude_outliers     | フラグ型                       |   |
| percentage           | number                     |   |
| cluster_num_auto     | フラグ型                       |   |
| min_num_clusters     | number                     |   |
| max_num_clusters     | number                     |   |
| num_clusters         | number                     |   |
| cluster_label        | String<br>Number           |   |
| label_prefix         | string                     |   |
| distance_measure     | Euclidean<br>Loglikelihood |   |
| clustering_criterion | AIC<br>BIC                 |   |

# モデル ナゲット ノードのプロパティ

モデル ナゲット ノードは、他のノードと同じ共通のプロパティを共有しています。詳細は、9 章 p.130 共通のノード プロパティ を参照してください。

## applyanomalydetectionnode のプロパティ

異常値検出モデル作成ノードを使用して、異常値検出モデル ナゲットを生成することができます。このモデル ナゲットのスクリプト名は、`applyanomalydetectionnode` です。モデル作成ノード自体のスクリプトの詳細は、16 章 ( p.214 ) の「`anomalydetectionnode` のプロパティ」を参照してください。

| applyanomalydetectionnode のプロパティ       | 値                                     | プロパティの説明   |
|--|---------------------------------------|--|
| <code>anomaly_score_method</code>      | FlagAndScore<br>FlagOnly<br>ScoreOnly | スコアリング用に、作成される出力を決めます。   |
| <code>num_fields</code>                | 整数                                    | 報告するフィールド数。  |
| <code>discard_records</code>           | フラグ型                                  | レコードが出力から廃棄されるかどうかを示します。   |
| <code>discard_anomalous_records</code> | フラグ型                                  | 異常なレコードを廃棄するか、または異常でないレコードを廃棄するかの標識。デフォルトは、異常でないレコードが廃棄されることを示す <code>off</code> です。それに対し、 <code>on</code> の場合は、異常なレコードが廃棄されます。このプロパティは、 <code>discard_records</code> が有効な場合にだけ、有効になります。 |

## applyapriorinode のプロパティ

Apriori モデル作成ノードを使用して、Apriori モデル ナゲットを生成することができます。このモデル ナゲットのスクリプト名は、`applyapriorinode` です。モデル作成ノード自体のスクリプトの詳細は、16 章 ( p.215 ) の「[apriorinode のプロパティ](#)」を参照してください。

| applyapriorinode のプロパティ         | 値   | プロパティの説明 |
|---------------------------------|---|----------|
| <code>max_predictions</code>    | 数値 (整数)   |          |
| <code>ignore_unmatedched</code> | フラグ型  |          |
| <code>allow_repeats</code>      | フラグ型  |          |
| <code>check_basket</code>       | NoPredictions<br>Predictions<br>NoCheck                       |          |
| <code>criterion</code>          | Confidence<br>Support<br>RuleSupport<br>Lift<br>Deployability |          |

## applyautoclassifiernode のプロパティ

自動分類モデル作成ノードを使用して、自動分類モデル ナゲットを生成することができます。このモデル ナゲットのスクリプト名は、`applyautoclassifiernode` です。モデル作成ノードのスクリプト化の詳細は、16 章 ( p.217 ) の「[autoclassifiernode のプロパティ](#)」を参照してください。

| applyautoclassifiernode のプロパティ         | 値  | プロパティの説明  |
|--|--|---|
| <code>flag_ensemble_method</code>      | Voting<br>ConfidenceWeightedVoting<br>RawPropensityWeightedVoting<br>HighestConfidence<br>AverageRawPropensity | アンサンブル スコアを決定するために使用する方法を指定します。この設定は、選択された対象がフラグ型フィールドである場合にのみ適用されます。 |
| <code>flag_voting_tie_selection</code> | Random<br>HighestConfidence<br>RawPropensity   | 票決方法が選択された場合、可否同数の解決方法を指定しますこの設定は、選択された対象がフラグ型フィールドである場合にのみ適用されます。    |

| applyautoclassifiernode のプロパティ | 値   | プロパティの説明  |
|--------------------------------|---|---|
| set_ensemble_method            | Voting<br>ConfidenceWeightedVoting<br>HighestConfidence | アンサンブル スコアを決定するために使用する方法を指定します。この設定は、選択された対象がセット型フィールドである場合にのみ適用されます。 |
| set_voting_tie_selection       | Random<br>HighestConfidence                             | 票決方法が選択された場合、可否同数の解決方法を指定しますこの設定は、選択された対象が名義型フィールドである場合にのみ適用されます。     |

## applyautoclusternode のプロパティ

自動クラスタ モデル作成ノードを使用して、自動クラスタ モデル ナゲットを生成することができます。このモデル ナゲットのスクリプト名は、applyautoclusternode です。このモデル ナゲットの他のプロパティはありません。モデル作成ノード自体のスクリプトの詳細は、16 章（ p. 220 ）の「[autoclusternode のプロパティ](#)」を参照してください。

## applyautonumericnode プロパティ

自動数値モデル作成ノードを使用して、自動数値モデル ナゲットを生成することができます。このモデル ナゲットのスクリプト名は、applyautonumericnode です。モデル作成ノードのスクリプト化の詳細は、16 章（ p. 221 ）の「[autonumericnode のプロパティ](#)」を参照してください。

| applyautonumericnode プロパティ | 値    | プロパティの説明 |
|----------------------------|------|----------|
| calculate_standard_error   | フラグ型 |          |

## applybayesnetnode のプロパティ

ベイズ ネットワーク モデル作成ノードを使用して、ベイズ ネットワークモデル ナゲットを生成することができます。このモデル ナゲットのスクリプト名は、applybayesnetnode です。モデル作成ノード自体のスク

リプトの詳細は、16 章（ p.223 ）の「bayesnetnode プロパティ」を参照してください。

| applybayesnetnode のプロパティ        | 値    | プロパティの説明 |
|---------------------------------|------|----------|
| all_probabilities               | フラグ型 |          |
| raw_propensity                  | フラグ型 |          |
| adjusted_propensity             | フラグ型 |          |
| calculate_raw_propensities      | フラグ型 |          |
| calculate_adjusted_propensities | フラグ型 |          |

## applyc50node のプロパティ

C5.0 モデル作成ノードを使用して、C5.0 モデル ナゲットを生成することができます。このモデル ナゲットのスクリプト名は、applyc50node です。モデル作成ノード自体のスクリプトの詳細は、16 章（ p.224 ）の「c50node のプロパティ」を参照してください。

| applyc50node のプロパティ             | 値                        | プロパティの説明  |
|---------------------------------|--------------------------|---|
| sql_generate                    | Never<br>NoMissingValues | ルールセット実行時の SQL 生成オプションの設定に使用します。                            |
| calculate_conf                  | フラグ型                     | SQL 生成が有効になっている場合に利用できます。このプロパティには、生成されたツリー中の確信度計算が含まれています。 |
| calculate_raw_propensities      | フラグ型                     |   |
| calculate_adjusted_propensities | フラグ型                     |   |

## applycarmanode のプロパティ

CARMA モデル作成ノードを使用して、CARMA モデル ナゲットを生成することができます。このモデル ナゲットのスクリプト名は、applycarmanode です。このモデル ナゲットの他のプロパティはありません。モデル作成ノード自体のスクリプトの詳細は、16 章（ p.226 ）の「carmanode のプロパティ」を参照してください。

## applycartnode のプロパティ

C&R Tree モデル作成を使用して、C&R Tree モデル ナゲットを生成することができます。このモデル ナゲットのスクリプト名は、applycartnode です。モデル作成ノード自体のスクリプトの詳細は、16 章（ p.227 ）の「cartnode のプロパティ」を参照してください。

| applycartnode のプロパティ            | 値   | プロパティの説明  |
|---------------------------------|---|---|
| sql_generate                    | Never<br>MissingValues<br>NoMissingValues | ルールセット実行時の SQL 生成オプションの設定に使用します。                                  |
| calculate_conf                  | フラグ型                                      | SQL 生成が有効になっている場合に利用できます。このプロパティには、生成されたツリー中の確信度計算が含まれています。       |
| display_rule_id                 | フラグ型                                      | フィールドが 1 つスコアリング出力に追加されますが、これは各レコードを割り当てるターミナルノードに ID を示すためのものです。 |
| calculate_raw_propensities      | フラグ型                                      |   |
| calculate_adjusted_propensities | フラグ型                                      |   |

## applychaidnode のプロパティ

CHAID モデル作成ノードを使用して、CHAID モデル ナゲットを生成することができます。このモデル ナゲットのスクリプト名は、applychaidnode です。モデル作成ノード自体のスクリプトの詳細は、16 章（ p.230 ）の「chaidnode のプロパティ」を参照してください。

| applychaidnode のプロパティ           | 値                      | プロパティの説明  |
|---------------------------------|------------------------|---|
| sql_generate                    | Never<br>MissingValues |   |
| calculate_conf                  | フラグ型                   |   |
| display_rule_id                 | フラグ型                   | フィールドが 1 つスコアリング出力に追加されますが、これは各レコードを割り当てるターミナルノードに ID を示すためのものです。 |
| calculate_raw_propensities      | フラグ型                   |   |
| calculate_adjusted_propensities | フラグ型                   |   |

## applycoxregnode のプロパティ

Cox モデル作成ノードを使用して、Cox モデル ナゲットを生成することができます。このモデル ナゲットのスクリプト名は、applycoxregnode です。モデル作成ノード自体のスクリプトの詳細は、16 章（ p.232 ）の「coxregnode のプロパティ」を参照してください。

| applycoxregnode のプロパティ | 値                   | プロパティの説明 |
|------------------------|---------------------|----------|
| future_time_as         | Intervals<br>Fields |          |
| time_interval          | 数値型                 |          |
| num_future_times       | 整数                  |          |
| time_field             | フィールド               |          |
| past_survival_time     | フィールド               |          |
| all_probabilities      | フラグ型                |          |
| cumulative_hazard      | フラグ型                |          |

## applydecisionlistnode のプロパティ

ディシジョン リスト モデル作成ノードを使用して、ディシジョン リスト モデル ナゲットを生成することができます。このモデル ナゲットのスクリプト名は、applydecisionlistnode です。モデル作成ノード自体のスクリプトの詳細は、16 章（ p.234 ）の「decisionlistnode のプロパティ」を参照してください。

| applydecisionlistnode のプロパティ    | 値    | プロパティの説明  |
|---------------------------------|------|---|
| enable_sql_generation           | フラグ型 | 真に設定したときは、ディシジョン リスト モデルが SQL ヘプッシュバックされるように IBM® SPSS® Modeler が試行します。 |
| calculate_raw_propensities      | フラグ型 |   |
| calculate_adjusted_propensities | フラグ型 |   |

## applydiscriminantnode のプロパティ

判別分析モデル作成ノードを使用して、判別分析モデル ナゲットを生成することができます。このモデル ナゲットのスクリプト名は、applydiscriminantnode です。モデル作成ノード自体のスクリプトの詳細



は、16 章（ p.236 ）の「discriminantnode のプロパティ」を参照してください。

| applydiscriminantnode のプロパティ    | 値    | プロパティの説明 |
|---------------------------------|------|----------|
| calculate_raw_propensities      | フラグ型 |          |
| calculate_adjusted_propensities | フラグ型 |          |

## applyfactornode のプロパティ

因子分析モデル作成ノードを使用して、因子分析モデル ナゲットを生成することができます。このモデル ナゲットのスクリプト名は、applyfactornode です。このモデル ナゲットの他のプロパティはありません。モデル作成ノード自体のスクリプトの詳細は、16 章（ p.237 ）の「factornode のプロパティ」を参照してください。

## applyfeatureselectionnode のプロパティ

フィールド選択モデル作成ノードを使用して、フィールド選択モデル ナゲットを生成することができます。このモデル ナゲットのスクリプト名は、applyfeatureselectionnode です。モデル作成ノード自体のスクリプトの詳細は、16 章（ p.239 ）の「featureselectionnode のプロパティ」を参照してください。

| applyfeatureselectionnode のプロパティ | 値 | プロパティの説明                              |
|----------------------------------|---|---------------------------------------|
| selected_ranked_fields           |   | モデル ブラウザ内で検査されるランク付きのフィールドを指定します。     |
| selected_screened_fields         |   | モデル ブラウザ内で検査されるスクリーニングされたフィールドを指定します。 |

## applygeneralizedlinearnode のプロパティ

一般化線型 (genlin) モデル作成ノードを使用して、一般化線型モデル ナゲットを生成することができます。このモデル ナゲットのスクリプト名は、applygeneralizedlinearnode です。モデル作成ノード自体のスク

リプトの詳細は、16 章（ p.241 ）の「[genlinnode のプロパティ](#)」を参照してください。

| applygeneralizedlinearnode のプロパティ | 値    | プロパティの説明 |
|-----------------------------------|------|----------|
| calculate_raw_propensities        | フラグ型 |          |
| calculate_adjusted_propensities   | フラグ型 |          |

## applykmeansnode のプロパティ

K-means モデル作成ノードを使用して、K-means モデル ナゲットを生成することができます。このモデル ナゲットのスクリプト名は、applykmeansnode です。このモデル ナゲットの他のプロパティはありません。モデル作成ノード自体のスクリプトの詳細は、16 章（ p.248 ）の「[kmeansnode のプロパティ](#)」を参照してください。

## applyknnnode プロパティ

KNN モデル作成ノードを使用して、KNN モデル ナゲットを生成することができます。このモデル ナゲットのスクリプト名は、applyknnnode です。モデル作成ノード自体のスクリプトの詳細は、16 章（ p.249 ）の「[knnnode のプロパティ](#)」を参照してください。

| applyknnnode プロパティ | 値    | プロパティの説明 |
|--------------------|------|----------|
| all_probabilities  | フラグ型 |          |
| save_distances     | フラグ型 |          |

## applykohonenode のプロパティ

Kohonen モデル作成ノードを使用して、Kohonen モデル ナゲットを生成することができます。このモデル ナゲットのスクリプト名は、applykohonenode です。このモデル ナゲットの他のプロパティはありません。モデル作成ノード自体のスクリプトの詳細は、16 章（ p.224 ）の「[c50node のプロパティ](#)」を参照してください。

## applylinearnode プロパティ

線型モデル作成ノードを使用して、線型モデル ナゲットを生成することができます。このモデル ナゲットのスクリプト名は、`applylinearnode` です。モデル作成ノード自体のスクリプトの詳細は、16 章（ p.252 ）の「[linearnode プロパティ](#)」を参照してください。

| linear プロパティ                       | 値    | プロパティの説明 |
|------------------------------------|------|----------|
| <code>use_custom_name</code>       | フラグ型 |          |
| <code>custom_name</code>           | 文字列  |          |
| <code>enable_sql_generation</code> | フラグ型 |          |

## applylogregnode のプロパティ

ロジスティック回帰モデル作成ノードを使用して、ロジスティック回帰モデル ナゲットを生成することができます。このモデル ナゲットのスクリプト名は、`applylogregnode` です。モデル作成ノード自体のスクリプトの詳細は、16 章（ p.254 ）の「[logregnode のプロパティ](#)」を参照してください。

| applylogregnode のプロパティ                  | 値    | プロパティの説明 |
|---|------|----------|
| <code>calculate_raw_propensities</code> | フラグ型 |          |

## applyneuralnetnode のプロパティ

ニューラル ネットワーク モデル作成ノードを使用して、ニューラル ネットワーク モデル ナゲットを生成することができます。このモデル ナゲットのスクリプト名は、`applyneuralnetnode` です。モデル作成ノード自体のスクリプトの詳細は、16 章（ p.258 ）の「[neuralnetnode のプロパティ](#)」を参照してください。

**注意:** 機能が拡張された新しいバージョンのニューラル ネットワーク ナゲットがこのリリースで使用できます。新しいバージョンについては次の項で説明します (`applyneuralnetwork`)。以前のバージョンは現在も使用できますが、スクリプトを更新して新しいバージョンを使用することをお勧めします。

めします。旧バージョンの詳細を参照用に記載しておりますが、それに対するサポートは今後のリリースで廃止されます。

| applyneuralnetnode のプロパティ       | 値                     | プロパティの説明  |
|---------------------------------|-----------------------|---|
| calculate_conf                  | フラグ型                  | SQL 生成が有効になっている場合に利用できます。このプロパティには、生成されたツリー中の確信度計算が含まれています。 |
| enable_sql_generation           | フラグ型                  |   |
| nn_score_method                 | Difference<br>SoftMax |   |
| calculate_raw_propensities      | フラグ型                  |   |
| calculate_adjusted_propensities | フラグ型                  |   |

## applyneuralnetworknode プロパティ

ニューラル ネットワーク モデル作成ノードを使用して、ニューラル ネットワーク モデル ナゲットを生成することができます。このモデル ナゲットのスクリプト名は、applyneuralnetworknode です。モデル作成ノード自体のスクリプトの詳細は、16 章（ p. 261 ）の「[neuralnetworknode プロパティ](#)」を参照してください。

| applyneuralnetworknode プロパティ | 値                           | プロパティの説明 |
|------------------------------|-----------------------------|----------|
| use_custom_name              | フラグ型                        |          |
| custom_name                  | 文字列                         |          |
| confidence                   | onProbability<br>onIncrease |          |
| score_category_probabilities | フラグ型                        |          |
| max_categories               | 数値型                         |          |
| score_propensity             | フラグ型                        |          |

## applyquestnode のプロパティ

QUEST モデル作成ノードを使用して、QUEST モデル ナゲットを生成することができます。このモデル ナゲットのスクリプト名は、applyquestnode です。モデル作成ノード自体のスクリプトの詳細は、16 章（ p. 263 ）の「[questnode のプロパティ](#)」を参照してください。

| applyquestnode のプロパティ | 値   | プロパティの説明 |
|-----------------------|---|----------|
| sql_generate          | Never<br>MissingValues<br>NoMissingValues |          |

| applyquestnode のプロパティ           | 値    | プロパティの説明  |
|---------------------------------|------|---|
| calculate_conf                  | フラグ型 |   |
| display_rule_id                 | フラグ型 | フィールドが 1 つスコアリング出力に追加されますが、これは各レコードを割り当てるターミナルノードに ID を示すためのものです。 |
| calculate_raw_propensities      | フラグ型 |   |
| calculate_adjusted_propensities | フラグ型 |   |

## applyregressionnode のプロパティ

線型回帰モデル作成ノードを使用して、線型回帰モデル ナゲットを生成することができます。このモデル ナゲットのスクリプト名は、applyregressionnode です。このモデル ナゲットの他のプロパティはありません。モデル作成ノード自体のスクリプトの詳細は、16 章 ( p.265 ) の「[regressionnode のプロパティ](#)」を参照してください。

## applyselflearningnode のプロパティ

自己学習応答モデル (SLRM) モデル作成ノードを使用して、SLRM モデル ナゲットを生成することができます。このモデル ナゲットのスクリプト名は、applyselflearningnode です。モデル作成ノード自体のスクリプトの詳細は、16 章 ( p.269 ) の「[slrmnode のプロパティ](#)」を参照してください。

| applyselflearningnode のプロパティ | 値                       | プロパティの説明                                  |
|------------------------------|-------------------------|---|
| max_predictions              | 数値型                     |   |
| randomization                | 数値型                     |   |
| scoring_random_seed          | 数値型                     |   |
| sort                         | ascending<br>descending | 高いスコアまたは低いスコアのどちらを持つオファーが最初に表示されるかを指定します。 |
| model_reliability            | フラグ型                    | [設定] タブでモデルの信頼性を考慮します。                    |

## applysequencenode のプロパティ

シーケンス モデル作成ノードを使用して、シーケンス モデル ナゲットを生成することができます。このモデル ナゲットのスクリプト名は、applysequencenode です。このモデル ナゲットの他のプロパティはありま

せん。モデル作成ノード自体のスキプットの詳細は、16 章（ p.267 ）の「sequencenode のプロパティ」を参照してください。

## applysvmnode のプロパティ

SVM モデル作成ノードを使用して、SVM モデル ナゲットを生成することができます。このモデル ナゲットのスキプット名は、applysvmnode です。モデル作成ノード自体のスキプットの詳細は、16 章（ p.270 ）の「svmnode プロパティ」を参照してください。

| applysvmnode のプロパティ             | 値    | プロパティの説明 |
|---------------------------------|------|----------|
| all_probabilities               | フラグ型 |          |
| calculate_raw_propensities      | フラグ型 |          |
| calculate_adjusted_propensities | フラグ型 |          |

## applytimeseriesnode のプロパティ

時系列モデル作成ノードを使用して、時系列モデル ナゲットを生成することができます。このモデル ナゲットのスキプット名は、applytimeseriesnode です。モデル作成ノード自体のスキプットの詳細は、16 章（ p.271 ）の「timeseriesnode のプロパティ」を参照してください。

| applytimeseriesnode のプロパティ | 値    | プロパティの説明 |
|----------------------------|------|----------|
| calculate_conf             | フラグ型 |          |
| calculate_residuals        | フラグ型 |          |

## applytwostepnode のプロパティ

TwoStep モデル作成ノードを使用して、TwoStep モデル ナゲットを生成することができます。このモデル ナゲットのスキプット名は、applytwostepnode です。このモデル ナゲットの他のプロパティはありません。モデル作成ノード自体のスキプットの詳細は、16 章（ p.273 ）の「twostepnode のプロパティ」を参照してください。

# データベース モデル作成ノードのプロパティ

IBM® SPSS® Modeler は、Microsoft SQL Server Analysis Services、Oracle Data Mining、IBM® DB2® InfoSphere Warehouse、IBM® Netezza® Analytics を含む、データベース ベンダーから入手可能なデータ マイニングとモデル作成ツールとの統合をサポートしています。詳細は、[2 章 データベースモデル作成の概要 in IBM SPSS Modeler 15 データベース内 マイニング ガイド](#) を参照してください。SPSS Modeler ネイティブ データベース アルゴリズムを使用して、アプリケーション内からのモデルの構築およびスコアリングがすべて可能です。データベース モデルは、このセクションで説明するプロパティを使用してスクリプトで作成および処理することも可能です。

たとえば、次のスクリプトの引用は、SPSS Modeler スクリプト インターフェイスを使用した Microsoft デシジョン ツリー モデルの作成を示します。

```
create mstreenode
rename :mstreenode as msbuilder
set msbuilder.analysis_server_name = 'localhost'
set msbuilder.analysis_database_name = 'TESTDB'
set msbuilder.mode = 'Expert'
set msbuilder.datasource = 'LocalServer'
set msbuilder.target = 'Drug'
set msbuilder.inputs = ['Age' 'Sex']
set msbuilder.unique_field = 'IDX'
set msbuilder.custom_fields = true
set msbuilder.model_name = 'MSDRUG'

connect :typenode to msbuilder
execute msbuilder

insert model MSDRUG connected between :typenode and :tablenode
set MSDRUG.sql_generate = true
execute :tablenode
```

## Microsoft モデル作成ノードのプロパティ

### Microsoft モデル作成ノードのプロパティ

#### 共通のプロパティ

次のプロパティは、Microsoft データベース モデル作成ノードに共通です。

| 共通の Microsoft ノード プロパティ | 値                         | プロパティの説明   |
|-------------------------|---------------------------|--|
| analysis_database_name  | string                    | Analysis Services データベースの名前。                                     |
| analysis_server_name    | string                    | Analysis Services ホストの名前。  |
| use_transactional_data  | フラグ型                      | 入力データがテーブル形式またはトランザクション形式かを指定します。                                |
| inputs                  | [フィールド<br>フィールド<br>フィールド] | テーブル形式の入力フィールド。  |
| target                  | フィールド                     | 予測フィールド (MS クラスタリング ノードまたはシーケンス クラスタリング ノードには該当しない)。             |
| unique_field            | フィールド                     | キー フィールド。  |
| msas_parameters         | 構造化                       | アルゴリズム パラメータ。 <a href="#">詳細は、p. 290 アルゴリズム パラメータ を参照してください。</a> |
| with_drillthrough       | フラグ型                      | [ドリルスルーあり] オプション。  |

#### MS ディジジョン ツリー

mstreenode タイプのノードには、特定のプロパティが定義されていません。このセクションの冒頭にある共通 Microsoft プロパティを参照してください。

#### MS クラスタリング

msclusternode タイプのノードには、特定のプロパティが定義されていません。このセクションの冒頭にある共通 Microsoft プロパティを参照してください。



## MS アソシエーション ルール

次のプロパティは、`msassocnode` タイプのノードで使用できます。

| msassocnode のプロパティ   | 値                         | プロパティの説明              |
|----------------------|---------------------------|-----------------------|
| id_field             | フィールド                     | データの各トランザクションを特定します。  |
| trans_inputs         | [フィールド<br>フィールド<br>フィールド] | トランザクションデータの入力フィールド。  |
| transactional_target | フィールド                     | 予測データ (トランザクション データ)。 |

### MS Naive Bayes

`msbayesnode` タイプのノードには、特定のプロパティが定義されていません。このセクションの冒頭にある共通 Microsoft プロパティを参照してください。

### MS Linear Regression

`msregressionnode` タイプのノードには、特定のプロパティが定義されていません。このセクションの冒頭にある共通 Microsoft プロパティを参照してください。

### MS Neural Network

`msneuralnetworknode` タイプのノードには、特定のプロパティが定義されていません。このセクションの冒頭にある共通 Microsoft プロパティを参照してください。

### MS Logistic Regression

`mslogisticnode` タイプのノードには、特定のプロパティが定義されていません。このセクションの冒頭にある共通 Microsoft プロパティを参照してください。

### MS タイム シリーズ

`mstimeseriesnode` タイプのノードには、特定のプロパティが定義されていません。このセクションの冒頭にある共通 Microsoft プロパティを参照してください。

## MS シーケンス クラスタリング

次のプロパティは、`mssequenceclusternode` タイプのノードで使用できます。

| mssequenceclusternode のプロパティ | 値                         | プロパティの説明             |
|------------------------------|---------------------------|----------------------|
| id_field                     | フィールド                     | データの各トランザクションを特定します。 |
| input_fields                 | [フィールド<br>フィールド<br>フィールド] | トランザクションデータの入力フィールド。 |
| sequence_field               | フィールド                     | シーケンス ID。            |
| target_field                 | フィールド                     | 予測フィールド (テーブル形式データ)。 |

## アルゴリズム パラメータ

各 Microsoft データベース モデル タイプには、`msas_parameters` プロパティを使用して設定できる、次のような特定のパラメータがあります。

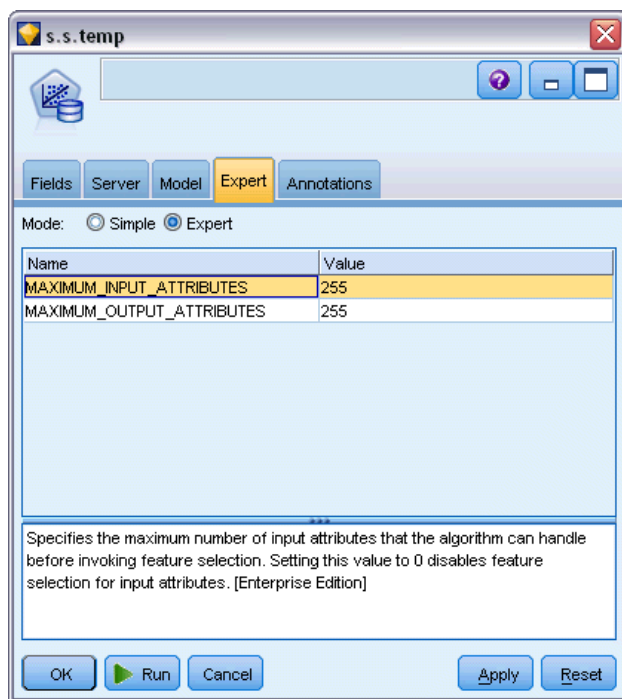
```
set :msregressionnode.msas_parameters =
[{"MAXIMUM_INPUT_ATTRIBUTES" 255},{"MAXIMUM_OUTPUT_ATTRIBUTES" 255}]
```

これらのパラメータは SQL から取得されます。各ノードに関連するパラメータを見るには

- ▶ キャンバスにデータベース入力ノードを配置します。
- ▶ データベース入力ノードを開きます。
- ▶ [データソース] ドロップダウン リストから有効なソースを選択します。
- ▶ [テーブル名] リストから有効なテーブルを選択します。
- ▶ [OK] をクリックして、データベース入力ノードを閉じます。
- ▶ プロパティを一覧表示したい Microsoft データベース モデル作成ノードを追加します。
- ▶ データベース モデル作成ノードを開きます。
- ▶ [エキスパート] タブを選択します。

このノードの使用できる `msas_parameters` プロパティが表示されます。

図 18-1  
アルゴリズム パラメータ表示の例



## Microsoft モデル ナゲットのプロパティ

Microsoft データベース モデル作成ノードを使用して作成されるモデル ナゲットのプロパティを、次に示します。

### MS ディジジョン ツリー

| appliednode のプロパティ     | 値      | 説明  |
|------------------------|--------|---|
| analysis_database_name | string | このノードは、ストリームの中で直接スコアされます。<br>このプロパティは Analysis Services データベース名の識別に使用します。 |
| analysis_server_name   | string | Analysis サーバー ホストの名前  |
| datasource             | string | SQL Server の ODBC データ ソース 名 (DSN) の名前                                     |
| sql_generate           | フラグ型   | SQL 生成を有効にします。  |

### MS Linear Regression

| appliesregressionnode のプロパティ | 値      | 説明  |
|------------------------------|--------|---|
| analysis_database_name       | string | このノードは、ストリームの中で直接スコアされます。<br>このプロパティは Analysis Services データベース名の識別に使用します。 |
| analysis_server_name         | string | Analysis サーバー ホストの名前  |

### MS Neural Network

| appliesneuralnetworknode のプロパティ | 値      | 説明  |
|---------------------------------|--------|---|
| analysis_database_name          | string | このノードは、ストリームの中で直接スコアされます。<br>このプロパティは Analysis Services データベース名の識別に使用します。 |
| analysis_server_name            | string | Analysis サーバー ホストの名前  |

### MS Logistic Regression

| applieslogisticnode のプロパティ | 値      | 説明  |
|----------------------------|--------|---|
| analysis_database_name     | string | このノードは、ストリームの中で直接スコアされます。<br>このプロパティは Analysis Services データベース名の識別に使用します。 |
| analysis_server_name       | string | Analysis サーバー ホストの名前  |

### MS タイム シリーズ

| appliestimeseriesnode のプロパティ | 値                                       | 説明  |
|------------------------------|---|---|
| analysis_database_name       | string                                  | このノードは、ストリームの中で直接スコアされます。<br>このプロパティは Analysis Services データベース名の識別に使用します。 |
| analysis_server_name         | string                                  | Analysis サーバー ホストの名前  |
| start_from                   | new_prediction<br>historical_prediction | 将来の予測を行うか過去の予測を行うかを指定します。   |
| new_step                     | number                                  | 将来の予測の開始時間を定義します。   |
| historical_step              | number                                  | 過去の予測の開始時間を定義します。   |
| end_step                     | number                                  | 予測の終了時間を定義します。  |

## MS シーケンス クラスターリング

| appliessequenceclusternode のプロパティ | 値      | 説明  |
|-----------------------------------|--------|---|
| analysis_database_name            | string | このノードは、ストリームの中で直接スコアされます。<br>このプロパティは Analysis Services データベース名の識別に使用します。 |
| analysis_server_name              | string | Analysis サーバー ホストの名前  |

## Oracle モデル作成ノードのプロパティ

## Oracle モデル作成ノードのプロパティ

次のプロパティは、各 Oracle データベース モデリング ノードに共通です。

| 一般的な Oracle ノードのプロパティ | 値                | プロパティの説明   |
|-----------------------|------------------|--|
| target                | フィールド            |  |
| inputs                | フィールドのリスト        |  |
| partition             | フィールド            | モデル構築の学習、テスト、および検証の各ステージ用に、データを独立したサブセット（サンプル）に分割するフィールド。                      |
| datasource            |                  |  |
| username              |                  |  |
| password              |                  |  |
| epassword             |                  |  |
| use_model_name        | フラグ型             |  |
| model_name            | string           | ユーザーが指定する新規モデル名。   |
| use_partitioned_data  | フラグ型             | 区分フィールドが定義される場合、このオプションは学習データ区分からのデータのみがモデル構築に使用されるようにします。                     |
| unique_field          | フィールド            |  |
| auto_data_prep        | フラグ型             | Oracle 自動データ準備機能を有効化または無効化します (11g データベースのみ)。                                  |
| costs                 | 構造化              | 構造化プロパティ、使用フォーマット：<br>[[drugA drugB 1.5]{drugA drugC 2.1}]。{} 内の引数は実際の予測コストです。 |
| mode                  | Simple<br>Expert | Simple に設定されている場合、個々のノード プロパティに記述されているように、特定のプロパティは無視されます。                     |

| 一般的な Oracle ノードのプロパティ      | 値      | プロパティの説明 |
|----------------------------|--------|----------|
| use_prediction_probability | フラグ型   |          |
| prediction_probability     | string |          |
| use_prediction_set         | フラグ型   |          |

### Oracle Naive Bayes

次のプロパティは、`oranbnode` タイプのノードで使用できます。

| <code>oranbnode</code> のプロパティ | 値                       | プロパティの説明  |
|-------------------------------|-------------------------|---|
| singleton_threshold           | number                  | 0.0-1.0.*   |
| pairwise_threshold            | number                  | 0.0-1.0.*   |
| priors                        | Data<br>Equal<br>Custom |   |
| custom_priors                 | 構造化                     | 構造化プロパティ、使用フォーマット：<br>set:oranbnode.custom_priors = [{drugA 1}{drugB 2}{drugC 3}{drugX 4}{drugY 5}] |

\* `mode` が `Simple` に設定されている場合、プロパティは無視されます。

### Oracle Adaptive Bayes

次のプロパティは、`oraabnnode` タイプのノードで使用できます。

| <code>oraabnnode</code> のプロパティ | 値   | プロパティの説明   |
|--------------------------------|---|--|
| model_type                     | SingleFeature<br>MultiFeature<br>NaiveBayes |  |
| use_execution_time_limit       | フラグ型  | *  |
| execution_time_limit           | integer                                     | 値は 1 以上でなければなりません。*  |
| max_naive_bayes_predictors     | integer                                     | 値は 1 以上でなければなりません。*  |
| max_predictors                 | integer                                     | 値は 1 以上でなければなりません。*  |
| priors                         | Data<br>Equal<br>Custom                     |  |
| custom_priors                  | 構造化   | 構造化プロパティ、使用フォーマット：<br>set:oraabnnode.custom_priors = [{drugA 1}{drugB 2}{drugC 3}{drugX 4}{drugY 5}] |

\* `mode` が `Simple` に設定されている場合、プロパティは無視されます。

### Oracle Support Vector Machines

次のプロパティは、`orasvmnode` タイプのノードで使用できます。

| orasvmnode のプロパティ                   | 値                            | プロパティの説明  |
|-------------------------------------|------------------------------|---|
| <code>active_learning</code>        | Enable<br>Disable            |   |
| <code>kernel_function</code>        | Linear<br>Gaussian<br>System |   |
| <code>normalization_method</code>   | zscore<br>minmax<br>none     |   |
| <code>kernel_cache_size</code>      | integer                      | Gaussian カーネル専用。値は 1 以上でなければなりません。*   |
| <code>convergence_tolerance</code>  | number                       | 値は 1 以上でなければなりません。*   |
| <code>use_standard_deviation</code> | フラグ型                         | Gaussian カーネル専用。*   |
| <code>standard_deviation</code>     | number                       | 値は 1 以上でなければなりません。*   |
| <code>use_epsilon</code>            | フラグ型                         | 回帰モデルのみです。*   |
| <code>epsilon</code>                | number                       | 値は 1 以上でなければなりません。*   |
| <code>use_complexity_factor</code>  | フラグ型                         | *   |
| <code>complexity_factor</code>      | number                       | *   |
| <code>use_outlier_rate</code>       | フラグ型                         | 単一バリエーションのみです。*   |
| <code>outlier_rate</code>           | number                       | 単一バリエーションのみです。0.0-1.0.*   |
| <code>weights</code>                | Data<br>Equal<br>Custom      |   |
| <code>custom_weights</code>         | 構造化                          | 構造化プロパティ、使用フォーマット：<br>set :orasvmnode.custom_weights = [{drugA<br>1}{drugB 2}{drugC 3}{drugX 4}{drugY 5}] |

\* mode が Simple に設定されている場合、プロパティは無視されます。

### Oracle 一般化線型モデル

次のプロパティは、`oraglmnode` タイプのノードで使用できます。

| oraglmnode のプロパティ                   | 値   | プロパティの説明 |
|-------------------------------------|---|----------|
| <code>normalization_method</code>   | zscore<br>minmax<br>none                        |          |
| <code>missing_value_handling</code> | ReplaceWith-<br>Mean<br>UseCompleteRe-<br>cords |          |
| <code>use_row_weights</code>        | フラグ型  | *        |

| oraglmnode のプロパティ      | 値                 | プロパティの説明 |
|------------------------|-------------------|----------|
| row_weights_field      | フィールド             | *        |
| save_row_diagnostics   | フラグ型              | *        |
| row_diagnostics_table  | string            | *        |
| coefficient_confidence | number            | *        |
| use_reference_category | フラグ型              | *        |
| reference_category     | string            | *        |
| ridge_regression       | Auto<br>Off<br>On | *        |
| parameter_value        | number            | *        |
| vif_for_ridge          | フラグ型              | *        |

\* mode が Simple に設定されている場合、プロパティは無視されます。

### Oracle Decision Tree

次のプロパティは、oradecisiontreenode タイプのノードで使用できます。

| oradecisiontreenode のプロパティ | 値               | プロパティの説明            |
|----------------------------|-----------------|---------------------|
| use_costs                  | フラグ型            |                     |
| impurity_metric            | Entropy<br>Gini |                     |
| term_max_depth             | integer         | 2-20.*              |
| term_minpct_node           | number          | 0.0-10.0.*          |
| term_minpct_split          | number          | 0.0-20.0.*          |
| term_minrec_node           | integer         | 値は 1 以上でなければなりません。* |
| term_minrec_split          | integer         | 値は 1 以上でなければなりません。* |
| display_rule_ids           | フラグ型            | *                   |

\* mode が Simple に設定されている場合、プロパティは無視されます。

### Oracle O-Cluster

次のプロパティは、oraclusternode タイプのノードで使用できます。

| oraclusternode のプロパティ | 値       | プロパティの説明            |
|-----------------------|---------|---------------------|
| max_num_clusters      | integer | 値は 1 以上でなければなりません。  |
| max_buffer            | integer | 値は 1 以上でなければなりません。* |
| sensitivity           | number  | 0.0-1.0.*           |

\* mode が Simple に設定されている場合、プロパティは無視されます。



**Oracle KMeans**

次のプロパティは、**orakmeansnode** タイプのノードで使用できます。

| orakmeansnode のプロパティ | 値                        | プロパティの説明             |
|----------------------|--------------------------|----------------------|
| num_clusters         | integer                  | 値は 1 以上でなければなりません。   |
| normalization_method | zscore<br>minmax<br>none |                      |
| distance_function    | Euclidean<br>Cosine      |                      |
| iterations           | integer                  | 0-20.*               |
| conv_tolerance       | number                   | 0.0-0.5.*            |
| split_criterion      | Variance<br>Size         | デフォルトは Variance です。* |
| num_bins             | integer                  | 値は 1 以上でなければなりません。*  |
| block_growth         | integer                  | 1-5.*                |
| min_pct_attr_support | number                   | 0.0-1.0.*            |

\* mode が Simple に設定されている場合、プロパティは無視されます。

**Oracle NMF**

次のプロパティは、**oranmfnode** タイプのノードで使用できます。

| oranmfnode のプロパティ    | 値              | プロパティの説明                           |
|----------------------|----------------|------------------------------------|
| normalization_method | minmax<br>none |                                    |
| use_num_features     | フラグ型           | *                                  |
| num_features         | integer        | 0-1。デフォルト値はアルゴリズムによってデータから推定されます。* |
| random_seed          | number         | *                                  |
| num_iterations       | integer        | 0-500.*                            |
| conv_tolerance       | number         | 0.0-0.5.*                          |
| display_all_features | フラグ型           | *                                  |

\* mode が Simple に設定されている場合、プロパティは無視されます。

**Oracle Apriori**

次のプロパティは、**oraapriorinode** タイプのノードで使用できます。

| oraapriorinode のプロパティ | 値     | プロパティの説明 |
|-----------------------|-------|----------|
| content_field         | フィールド |          |
| id_field              | フィールド |          |

| oraapriorinode のプロパティ  | 値       | プロパティの説明 |
|------------------------|---------|----------|
| max_rule_length        | integer | 2-20.    |
| min_confidence         | number  | 0.0-1.0. |
| min_support            | number  | 0.0-1.0. |
| use_transactional_data | フラグ型    |          |

### Oracle 最小記述長 (MDL)

oramlnode タイプのノードには、特定のプロパティが定義されていません。このセクションの冒頭にある共通 Oracle プロパティを参照してください。

### Oracle Attribute Importance (AI)

次のプロパティは、oraainode タイプのノードで使用できます。

| oraainode のプロパティ   | 値  | プロパティの説明  |
|--------------------|--|---|
| custom_fields      | フラグ型                                       | 真 (true) の場合は、現在のノードのターゲット、入力、その他フィールドなどを指定することができます。偽 (false) の場合は、上流のデータ型ノードから現在の設定が使用されます。 |
| selection_mode     | ImportanceLevel<br>ImportanceValue<br>TopN |   |
| select_important   | フラグ型                                       | selection_mode が ImportanceLevel に設定されているときに、重要なフィールドを選択するかどうかを指定します。                         |
| important_label    | string                                     | 「重要」ランクのラベルを指定します。  |
| select_marginal    | フラグ型                                       | selection_mode が ImportanceLevel に設定されているときに、境界フィールドを選択するかどうかを指定します。                          |
| marginal_label     | string                                     | 「境界」ランクのラベルを指定します。  |
| important_above    | number                                     | 0.0-1.0.  |
| select_unimportant | フラグ型                                       | selection_mode が ImportanceLevel に設定されているときに、重要でないフィールドを選択するかどうかを指定します。                       |
| unimportant_label  | string                                     | 「非重要」ランクのラベルを指定します。   |
| unimportant_below  | number                                     | 0.0-1.0.  |
| importance_value   | number                                     | selection_mode が ImportanceValue に設定されているときに、使用する分割値を指定します。0 ~ 100 の値を指定します。                  |
| top_n              | number                                     | selection_mode が TopN に設定されているときに、使用する分割値を指定します。0 ~ 1000 の値を指定します。                            |

## Oracle モデル ナゲットのプロパティ

Oracle ノードを使用して作成されるモデル ナゲットのプロパティを、次に示します。

### Oracle Naive Bayes

`applyoranbnode` タイプのノードには、特定のプロパティが定義されていません。

### Oracle Adaptive Bayes

`applyoraabnnode` タイプのノードには、特定のプロパティが定義されていません。

### Oracle Support Vector Machines

`applyorasvmnode` タイプのノードには、特定のプロパティが定義されていません。

### Oracle Decision Tree

次のプロパティは、`applyradecisiontreenode` タイプのノードで使用できます。

| <code>applyradecisiontreenode</code> のプロパティ | 値    | プロパティの説明 |
|---|------|----------|
| <code>use_costs</code>                      | フラグ型 |          |
| <code>display_rule_ids</code>               | フラグ型 |          |

### Oracle O-Cluster

`applyoraoclusternode` タイプのノードには、特定のプロパティが定義されていません。

### Oracle KMeans

`applyorakmeansnode` タイプのノードには、特定のプロパティが定義されていません。

### Oracle NMF

次のプロパティは、`applyoranmfnode` タイプのノードで使用できます。

| <code>applyoranmfnode</code> のプロパティ | 値    | プロパティの説明 |
|-------------------------------------|------|----------|
| <code>display_all_features</code>   | フラグ型 |          |

**Oracle Apriori**

このモデル ナゲットはスクリプトに適用できません。

**Oracle MDL**

このモデル ナゲットはスクリプトに適用できません。

**IBM DB2 モデル作成ノードのプロパティ****IBM DB2 モデル作成ノードのプロパティ**

次のプロパティは、各 IBM InfoSphere Warehouse (ISW) データベースモデリング ノードに共通です。

| ISW ノードの共通プロパティ            | 値         | プロパティの説明            |
|----------------------------|-----------|---------------------|
| inputs                     | フィールドのリスト |                     |
| datasource                 |           |                     |
| username                   |           |                     |
| password                   |           |                     |
| epassword                  |           |                     |
| enable_power_options       | フラグ型      |                     |
| power_options_max_memory   | integer   | 値は 33 以上でなければなりません。 |
| power_options_cmdline      | string    |                     |
| mining_data_custom_sql     | string    |                     |
| logical_data_custom_sql    | string    |                     |
| mining_settings_custom_sql |           |                     |

**ISW デシジョン ツリー**

次のプロパティは、db2imtreenode タイプのノードで使用できます。

| db2imtreenode のプロパティ       | 値       | プロパティの説明         |
|----------------------------|---------|------------------|
| target                     | フィールド   |                  |
| perform_test_run           | フラグ型    |                  |
| use_max_tree_depth         | フラグ型    |                  |
| max_tree_depth             | integer | 値は 1 以上です。       |
| use_maximum_purity         | フラグ型    |                  |
| maximum_purity             | number  | 0 と 100 の間の数値です。 |
| use_minimum_internal_cases | フラグ型    |                  |
| minimum_internal_cases     | integer | 値は 2 以上です。       |

## データベース モデル作成ノードのプロパティ

| db2imtreenode のプロパティ | 値    | プロパティの説明  |
|----------------------|------|---|
| use_costs            | フラグ型 |   |
| costs                | 構造化  | 構造化プロパティ、使用フォーマット：<br>[[drugA drugB 1.5]{drugA drugC 2.1}]. {} 内の引数は実際の予測コストです。 |

## ISW アソシエーション

次のプロパティは、db2imassocnode タイプのノードで使用できます。

| db2imassocnode のプロパティ       | 値                       | プロパティの説明  |
|-----------------------------|-------------------------|---|
| use_transactional_data      | フラグ型                    |   |
| id_field                    | フィールド                   |   |
| content_field               | フィールド                   |   |
| data_table_layout           | basic<br>limited_length |   |
| max_rule_size               | integer                 | 値は 3 以上でなければなりません。  |
| min_rule_support            | number                  | 0-100%  |
| min_rule_confidence         | number                  | 0-100%  |
| use_item_constraints        | フラグ型                    |   |
| item_constraints_type       | Include<br>Exclude      |   |
| use_taxonomy                | フラグ型                    |   |
| taxonomy_table_name         | string                  | DB2 テーブルの名前は、分類の詳細に格納されます。  |
| taxonomy_child_column_name  | string                  | 分類テーブルの子カラムの名前。子カラムには、項目名またはカテゴリ名が含まれます。  |
| taxonomy_parent_column_name | string                  | 分類テーブルの親カラムの名前。親カラムには、カテゴリ名が含まれます。  |
| load_taxonomy_to_table      | フラグ型                    | IBM® SPSS® Modeler に保存されている分類情報をモデルの構築時に、分類テーブルにアップロードするかどうかをコントロールします。すでに分類テーブルが存在する場合、そのテーブルは削除されます。分類情報は、モデル構築ノードと共に保存され、[カテゴリの編集] ボタンと [分類法の編集] ボタンを使用して編集できます。 |

## ISW シーケンス

次のプロパティは、db2imsequencenode タイプのノードで使用できます。

| db2imsequencenode のプロパティ    | 値                  | プロパティの説明  |
|-----------------------------|--------------------|---|
| id_field                    | フィールド              |   |
| group_field                 | フィールド              |   |
| content_field               | フィールド              |   |
| max_rule_size               | integer            | 値は 3 以上でなければなりません。  |
| min_rule_support            | number             | 0-100%  |
| min_rule_confidence         | number             | 0-100%  |
| use_item_constraints        | フラグ型               |   |
| item_constraints_type       | Include<br>Exclude |   |
| use_taxonomy                | フラグ型               |   |
| taxonomy_table_name         | string             | DB2 テーブルの名前は、分類の詳細に格納されます。  |
| taxonomy_child_column_name  | string             | 分類テーブルの子カラムの名前。子カラムには、項目名またはカテゴリ名が含まれます。  |
| taxonomy_parent_column_name | string             | 分類テーブルの親カラムの名前。親カラムには、カテゴリ名が含まれます。  |
| load_taxonomy_to_table      | フラグ型               | SPSS Modeler に保存されている分類情報をモデルの構築時に、分類テーブルにアップロードするかどうかをコントロールします。すでに分類テーブルが存在する場合、そのテーブルは削除されます。分類情報は、モデル構築ノードと共に保存され、[カテゴリの編集] ボタンと [分類法の編集] ボタンを使用して編集できます。 |

## ISW 回帰

次のプロパティは、db2imregnode タイプのノードで使用できます。

| db2imregnode のプロパティ      | 値  | プロパティの説明   |
|--------------------------|--|--|
| target                   | フィールド                                    |  |
| regression_method        | transform<br>linear<br>polynomial<br>rbf | regression_method が rbf に設定されている場合にのみ適用されるプロパティについては、次の表を参照してください。 |
| perform_test_run         | フィールド                                    |  |
| limit_rsquared_value     | フラグ型                                     |  |
| max_rsquared_value       | number                                   | 値の範囲は 0.0～1.0 です。  |
| use_execution_time_limit | フラグ型                                     |  |

## データベース モデル作成ノードのプロパティ

| db2imregnode のプロパティ               | 値                  | プロパティの説明   |
|-----------------------------------|--------------------|------------|
| execution_time_limit_mins         | integer            | 値は 1 以上です。 |
| use_max_degree_polynomial         | フラグ型               |            |
| max_degree_polynomial             | integer            |            |
| use_intercept                     | フラグ型               |            |
| use_auto_feature_selection_method | フラグ型               |            |
| auto_feature_selection_method     | normal<br>adjusted |            |
| use_min_significance_level        | フラグ型               |            |
| min_significance_level            | number             |            |
| use_min_significance_level        | フラグ型               |            |

次のプロパティは、`regression_method` が `rbf` に設定されている場合にのみ適用されます。

| db2imregnode のプロパティ    | 値       | プロパティの説明                     |
|------------------------|---------|------------------------------|
| use_output_sample_size | フラグ型    | true の場合、値はデフォルトに自動的に設定されます。 |
| output_sample_size     | integer | デフォルトは 2 です。<br>最小値は 1 です。   |
| use_input_sample_size  | フラグ型    | true の場合、値はデフォルトに自動的に設定されます。 |
| input_sample_size      | integer | デフォルトは 2 です。<br>最小値は 1 です。   |
| use_max_num_centers    | フラグ型    | true の場合、値はデフォルトに自動的に設定されます。 |
| max_num_centers        | integer | デフォルトは 20 です。<br>最小値は 1 です。  |
| use_min_region_size    | フラグ型    | true の場合、値はデフォルトに自動的に設定されます。 |
| min_region_size        | integer | デフォルトは 15 です。<br>最小値は 1 です。  |
| use_max_data_passes    | フラグ型    | true の場合、値はデフォルトに自動的に設定されます。 |
| max_data_passes        | integer | デフォルトは 5 です。<br>最小値は 2 です。   |
| use_min_data_passes    | フラグ型    | true の場合、値はデフォルトに自動的に設定されます。 |
| min_data_passes        | integer | デフォルトは 5 です。<br>最小値は 2 です。   |

## ISW クラスタリング

次のプロパティは、db2imclusternode タイプのノードで使用できます。

| db2imclusternode のプロパティ   | 値                               | プロパティの説明                  |
|---------------------------|---------------------------------|---------------------------|
| cluster_method            | demographic<br>kohonen<br>birch |                           |
| kohonen_num_rows          | integer                         |                           |
| kohonen_num_columns       | integer                         |                           |
| kohonen_passes            | integer                         |                           |
| use_num_passes_limit      | フラグ型                            |                           |
| use_num_clusters_limit    | フラグ型                            |                           |
| max_num_clusters          | integer                         | 値は 2 以上です。                |
| birch_dist_measure        | log_likelihood<br>euclidean     | デフォルトは log_likelihood です。 |
| birch_num_cfleaves        | integer                         | デフォルトは 1000 です。           |
| birch_num_refine_passes   | integer                         | デフォルトは 3、最小値は 1 です。       |
| use_execution_time_limit  | フラグ型                            |                           |
| execution_time_limit_mins | integer                         | 値は 1 以上です。                |
| min_data_percentage       | number                          | 0-100%                    |
| use_similarity_threshold  | フラグ型                            |                           |
| similarity_threshold      | number                          | 値の範囲は 0.0~1.0 です。         |

## ISW Naive Bayes

次のプロパティは、db2imnbsnode タイプのノードで使用できます。

| db2imnbsnode のプロパティ   | 値      | プロパティの説明  |
|-----------------------|--------|---|
| perform_test_run      | フラグ型   |   |
| probability_threshold | number | デフォルトは 0.001 です。<br>最小値は 0、最大値は 1.000 です。                                       |
| use_costs             | フラグ型   |   |
| costs                 | 構造化    | 構造化プロパティ、使用フォーマット：<br>{{drugA drugB 1.5} {drugA drugC 2.1}}。{} 内の引数は実際の予測コストです。 |



## ISW ロジスティック回帰

次のプロパティは、db2imlognode タイプのノードで使用できます。

| db2imlognode のプロパティ | 値    | プロパティの説明   |
|---------------------|------|--|
| perform_test_run    | フラグ型 |  |
| use_costs           | フラグ型 |  |
| costs               | 構造化  | 構造化プロパティ、使用フォーマット：<br>[[drugA drugB 1.5] {drugA drugC 2.1}]. {} 内の引数は実際の予測コストです。 |

## ISW 時系列

注： 入力フィールド パラメータはこのノードには使用されません。入力フィールド パラメータがスクリプトにない場合、ノードに入力フィールドではなく、受信フィールドとして時間および対象があることを示す警告が表示されます。

次のプロパティは、db2imtimeseriesnode タイプのノードで使用できます。

| db2imtimeseriesnode のプロパティ | 値  | プロパティの説明  |
|----------------------------|--|---|
| time                       | フィールド  | 整数、時間、日付が使用できます。  |
| targets                    | フィールドのリスト  |   |
| forecasting_algorithm      | arima<br>exponential_smoothing<br>seasonal_trend_decomposition |   |
| forecasting_end_time       | auto<br>integer<br>date<br>time                                |   |
| use_records_all            | boolean  | false の場合、use_records_start および use_records_end を設定する必要があります。 |
| use_records_start          | 整数 / 時間 / 日付   | 時間フィールドの種類によって異なります   |
| use_records_end            | 整数 / 時間 / 日付   | 時間フィールドの種類によって異なります   |
| interpolation_method       | none<br>linear<br>exponential_splines<br>cubic_splines         |   |

## IBM DB2 モデル ナゲットのプロパティ

IBM DB2 ISW ノードを使用して作成されるモデル ナゲットのプロパティを、次に示します。

### ISW ディシジョン ツリー

`applydb2imtreenode` タイプのノードには、特定のプロパティが定義されていません。

### ISW アソシエーション

このモデル ナゲットはスクリプトに適用できません。

### ISW シーケンス

このモデル ナゲットはスクリプトに適用できません。

### ISW 回帰

`applydb2imregnode` タイプのノードには、特定のプロパティが定義されていません。

### ISW クラスターリング

`applydb2imclusternode` タイプのノードには、特定のプロパティが定義されていません。

### ISW Naive Bayes

`applydb2imnbnode` タイプのノードには、特定のプロパティが定義されていません。

### ISW ロジスティック回帰

`applydb2imlognode` タイプのノードには、特定のプロパティが定義されていません。

### ISW 時系列

このモデル ナゲットはスクリプトに適用できません。

## IBM Netezza Analytics モデル作成ノードのプロパティ

### Netezza モデル作成ノードのプロパティ

次のプロパティは、各 IBM Netezza データベース モデリング ノードに共通です。

| 共通の Netezza ノード プロパティ   | 値                      | プロパティの説明  |
|-------------------------|------------------------|---|
| custom_fields           | フラグ型                   | 真 (true) の場合は、現在のノードのターゲット、入力、その他フィールドなどを指定することができます。偽 (false) の場合は、上流のデータ型ノードから現在の設定が使用されます。   |
| inputs                  | [フィールド1<br>... フィールドN] | モデルで使用される入力または予測変数フィールド。  |
| target                  | フィールド                  | 対象フィールド (連続型またはカテゴリ型)。  |
| record_id               | フィールド                  | 一意のレコード ID として使用されるフィールド。   |
| use_upstream_connection | フラグ型                   | true (デフォルト) の場合、上流のノードで指定された接続の詳細。move_data_to_connection が指定されている場合は使用されません。  |
| move_data_connection    | フラグ型                   | true の場合、データは connection に指定されたデータベースに移動します。use_upstream_connection が指定されている場合は使用されません。   |
| connection              | 構造化                    | モデルが保存される Netezza データベースの接続文字列。構造化プロパティ、使用フォーマット：<br>['odbc' '<dsn>' '<username>' '<psw>' '<catname>' '<conn_attribs>' {true false}]<br>ここでの意味は次の通りです。<br><dsn> は データ ソース名です。<br><username> と <psw> は、データベースのユーザー名とパスワードです。<br><catname> はカタログ名です。<br><conn_attribs> は接続の属性です。<br>true   false は、パスワードが必要かどうかを示します。 |
| table_name              | string                 | モデルが保存されるデータベース テーブルの名前。  |
| use_model_name          | フラグ型                   | true の場合、model_name によって指定された名前をモデルの名前として使用します。そうでない場合、モデル名はシステムによって作成されます。   |

| 共通の Netezza ノード プロパティ | 値      | プロパティの説明   |
|-----------------------|--------|--|
| model_name            | string | ユーザーが指定する新規モデル名。   |
| include_input_fields  | フラグ型   | true の場合、すべての入力フィールドを下流に渡します。そうでない場合、record_id とモデルによって生成されたフィールドのみが渡されます。 |

### Netezza ディジジョン ツリー

次のプロパティは、netezzadectreenode タイプのノードで使用できます。

| netezzadectreenode のプロパティ | 値   | プロパティの説明  |
|---------------------------|---|---|
| impurity_measure          | Entropy<br>Gini   | ツリーの分割に最も良い場所を評価するのに使用される、不純度の測定。   |
| max_tree_depth            | integer   | ツリーが成長可能な最大レベル数。デフォルトは 62 です (可能な最大値)。  |
| min_improvement_splits    | number  | 分割が発生する不純度の改善の最小値。デフォルトは 0.01 です。   |
| min_instances_split       | integer   | 分割が発生する前に残る分割されていないレコードの最小数。デフォルトは 2 です (可能な最小値)。   |
| weights                   | 構造化   | クラスの相対的重み構造化プロパティ、使用フォーマット：<br><code>set :netezza_dectree.weights = [{drugA 0.3}{drugB 0.6}]</code><br>デフォルトの重みはすべてのクラスで 1 です。          |
| pruning_measure           | Acc<br>wAcc   | デフォルトは Acc (精度) です。wAcc (重み付き精度) は、剪定を適用する際にクラスの重みを考慮します。   |
| prune_tree_options        | allTrainingData<br>partitionTrainingData<br>useOtherTable | デフォルトでは、allTrainingData を使用してモデルの精度を推定します。partitionTrainingData を使用して、使用する学習データの割合を、useOtherTable を使用して指定したデータベース テーブルの学習データ セットを使用します。 |
| perc_training_data        | number  | prune_tree_options が partitionTrainingData に設定されている場合、学習に使用するデータの割合を指定します。  |
| prune_seed                | integer   | prune_tree_options が partitionTrainingData に設定されている場合、分析結果の複製に使用するランダム シード。デフォルトは 1 です。   |
| pruning_table             | string  | モデルの精度を推定するために個別の剪定データセットのテーブル名。  |
| compute_probabilities     | フラグ型  | true の場合、予測フィールドのほか、確信度 (確率) フィールドを生成します。   |

### Netezza K-Means

次のプロパティは、netezzakmeansnode タイプのノードで使用できます。

| netezzakmeansnode のプロパティ | 値   | プロパティの説明                           |
|--------------------------|---|------------------------------------|
| distance_measure         | Euclidean<br>Manhattan<br>Canberra<br>maximum | データ ポイント間の教理を測定する方法。               |
| num_clusters             | integer                                       | 作成するクラスタ数。デフォルトは 3。                |
| max_iterations           | integer                                       | モデルの学習を停止する前のアルゴリズムの反復数。デフォルトは 5。  |
| rand_seed                | integer                                       | 分析結果の反復に使用するランダム シード。デフォルトは 12345。 |

### Netezza ベイズ ネットワーク

次のプロパティは、netezzabayesnode タイプのノードで使用できます。

| netezzabayesnode のプロパティ        | 値                                 | プロパティの説明  |
|--------------------------------|-----------------------------------|---|
| base_index                     | integer                           | 内部管理の最初の入力フィールドに割り当てられる数値の識別子。デフォルトは 777。                           |
| sample_size                    | integer                           | 属性の値が非常に大きい場合に最小するサンプルのサイズ。デフォルトは 10,000。                           |
| display_additional_information | フラグ型                              | true の場合、メッセージのダイアログ ボックスに追加の進捗状況の情報を表示します。                         |
| type_of_prediction             | best<br>neighbors<br>nn-neighbors | 使用する予測アルゴリズムの種類: 最適 (相関度が最も高い近隣)、近隣 (近隣の重み付き予測)、NN 近隣 (null 以外の近隣)。 |

### Netezza Naive Bayes

次のプロパティは、netezzanaivebayesnode タイプのノードで使用できます。

| netezzanaivebayesnode のプロパティ | 値    | プロパティの説明                                  |
|------------------------------|------|---|
| compute_probabilities        | フラグ型 | true の場合、予測フィールドのほか、確信度 (確率) フィールドを生成します。 |
| use_m_estimation             | フラグ型 | true の場合、推定時に 0 の確立を回避する m 推定方法を使用します。    |

### Netezza KNN

次のプロパティは、`netezzaknnnode` タイプのノードで使用できます。

| netezzaknnnode のプロパティ    | 値   | プロパティの説明  |
|--------------------------|---|---|
| weights                  | 構造化   | 重みを各クラスに割り当てる構造化プロパティ。例：<br><code>set :netezzaknnnode.weights = [{drugA 0.3}{drugB 0.6}]</code> |
| distance_measure         | Euclidean<br>Manhattan<br>Canberra<br>Maximum | データ ポイント間の教理を測定する方法。  |
| num_nearest_neighbors    | integer                                       | 特定のケースの最近隣数。デフォルトは 3。   |
| standardize_measurements | フラグ型  | true の場合、距離の値を計算する前に連続型入力フィールドの測定を標準化します。   |
| use_coresets             | フラグ型  | true の場合、大規模なデータセットに対して計算を高速化するコアセット サンプリングを使用しています   |

### Netezza 分裂クラスタリング

次のプロパティは、`netezzadivclusternode` タイプのノードで使用できます。

| netezzadivclusternode のプロパティ | 値   | プロパティの説明                                     |
|------------------------------|---|--|
| distance_measure             | Euclidean<br>Manhattan<br>Canberra<br>Maximum | データ ポイント間の教理を測定する方法。                         |
| max_iterations               | integer                                       | モデルの学習が停止する前に、実行するアルゴリズム反復の最大回数。デフォルトは 5 です。 |
| max_tree_depth               | integer                                       | データセットを分割することができるレベルの最大数。デフォルトは 3 です。        |
| rand_seed                    | integer                                       | 分析を複製するために使用されるランダムシード。デフォルトは 12345。         |
| min_instances_split          | integer                                       | 分割可能な最小レコード数。デフォルトは 5。                       |
| level                        | integer                                       | レコードをスコアリングする階層レベル。デフォルトは -1。                |

### Netezza PCA

次のプロパティは、netezzapcanode タイプのノードで使用できます。

| netezzapcanode のプロパティ | 値       | プロパティの説明  |
|-----------------------|---------|---|
| center_data           | フラグ型    | true (デフォルト) の場合、このオプションをチェックした場合、分析前にデータのセンタリングを (または「平均値減算」) を実行します。    |
| perform_data_scaling  | フラグ型    | true の場合、分析前にデータのスケールリングを行います。そうすることで、別の変数が異なる単位で測定されるとき、分析が恣意的でないようにします。 |
| force_eigensolve      | フラグ型    | true の場合、を計算する精度が低くなくてもより高速な方法を使用します。                                     |
| pc_number             | integer | データセットを減少する主要成分の数。デフォルトは 1。   |

### Netezza 回帰ツリー

次のプロパティは、netezzaregreenode タイプのノードで使用できます。

| netezzaregreenode のプロパティ | 値  | プロパティの説明  |
|--------------------------|--|---|
| max_tree_depth           | integer  | ルート ノードの前にツリーが成長できるレベルの最大数。デフォルトは 10 です。  |
| split_evaluation_measure | Variance   | ツリーを分割するのに最適な場所を評価するために使用される、クラスの不純度の測定。デフォルト (現在唯一のオプション) は Variance。  |
| min_improvement_splits   | number   | ツリー内に新しい分割が作成される前に純度を減少させる最小数。  |
| min_instances_split      | integer  | 分割可能な最小レコード数。   |
| pruning_measure          | mse<br>r2<br>pearson<br>spearman                           | 剪定に使用する方法   |
| prune_tree_options       | allTrainingData<br>partitionTraining-Data<br>useOtherTable | デフォルトでは、allTrainingData を使用してモデルの精度を推定します。partitionTrainingData を使用して、使用する学習データの割合を、useOtherTable を使用して指定したデータベース テーブルの学習データ セットを使用します。 |
| perc_training_data       | number   | prune_tree_options が PercTrainingData に設定されている場合、学習に使用するデータの割合を指定します。   |
| prune_seed               | integer  | prune_tree_options が PercTrainingData に設定されている場合、分析結果の複製に使用するランダム シード。デフォルトは 1 です。  |

| netezzaregreetreenode のプロパティ | 値      | プロパティの説明                                    |
|------------------------------|--------|---|
| pruning_table                | string | モデルの精度を推定するために個別の剪定データセットのテーブル名。            |
| compute_probabilities        | フラグ型   | true の場合、割り当てられたクラスの分散が出力に含まれるべきかどうかを指定します。 |

### Netezza 線型回帰

次のプロパティは、netezzalineressionionnode タイプのノードで使用できます。

| netezzalineressionionnode のプロパティ | 値    | プロパティの説明   |
|----------------------------------|------|--|
| use_svd                          | フラグ型 | true の場合、元のマトリックスの代わりに特異値分解マトリックスを使用して速度と数値の精度を向上させます。 |
| include_intercept                | フラグ型 | true (デフォルト) の場合、ソリューションの全体の精度が向上します。                  |
| calculate_model_diagnostics      | フラグ型 | true の場合、モデルの診断を計算します。                                 |

### Netezza 時系列

次のプロパティは、netezzatimeseriesnode タイプのノードで使用できます。

| netezzatimeseriesnode のプロパティ | 値   | プロパティの説明  |
|------------------------------|---|---|
| time_points                  | フィールド   | 時系列の日付または時刻の値を含む入力フィールド。                          |
| time_series_ids              | フィールド   | 時系列 ID を含むフィールド。入力に複数の時系列が含まれる場合に使用します。           |
| model_table                  | フィールド   | Netezza 時系列モデルが保存されるデータベース テーブルの名前。               |
| description_table            | フィールド   | 時系列名および説明を含む入力テーブルの名前。                            |
| seasonal_adjustment_table    | フィールド   | 指数平滑化または季節的傾向分解アルゴリズムによって計算された季節性調整値を保存する出力テーブル名。 |
| algorithm_name               | SpectralAnalysis<br>or spectral<br>ExponentialSmoothing<br>or smoothing<br>ARIMA<br>SeasonalTrend-<br>Decomposition<br>or std | 時系列モデリングに使用するアルゴリズム                               |



## データベース モデル作成ノードのプロパティ

| netezzatimeseriesnode のプロパティ | 値   | プロパティの説明   |
|------------------------------|---|--|
| trend_name                   | string  | 指数平滑化の傾向タイプ。   |
| seasonality_type             | string  | 指数平滑化の季節性タイプ。  |
| interpolation_method         | linear<br>cubicspline<br>exponential-spline                                       | 使用する補間方法。  |
| earliest_time                | フィールド   | 時系列の部分を使用する場合、開始時間。  |
| latest_time                  | フィールド   | 時系列の部分を使用する場合、終了時間。  |
| p                            | integer   | ARIMA - 自己相関の非季節性の度合い。   |
| q                            | integer   | ARIMA - 自己相関の非季節性導出値。  |
| d                            | integer   | ARIMA - モデル内の移動平均の非季節性数値。  |
| sp                           | integer   | ARIMA - 自己相関の季節性の度合い。  |
| sq                           | integer   | ARIMA - 自己相関の季節性導出値。   |
| sd                           | integer   | ARIMA - モデル内の移動平均の季節性数値。   |
| period                       | integer   | units_period と組み合わせて指定した季節性サイクルの長さ。スペクトル解析には適用できません。               |
| units_period                 | Milliseconds<br>Seconds<br>Minutes<br>Hours<br>Days<br>Weeks<br>Quarters<br>Years | period が表現される単位。たとえば、1 週間は period に 1、units_period に Weeks を指定します。 |
| include_history              | フラグ型  | 過去の値を出力に含めるかどうかを示します。  |
| include_interpolated_values  | フラグ型  | 補間されたの値を出力に含めるかどうかを示します。include_history が false の場合は使用されません。       |

## Netezza 一般化線型モデル

次のプロパティは、netezzaglmnode タイプのノードで使用できます。

| netezzaglmnode のプロパティ          | 値   | プロパティの説明                    |
|--------------------------------|---|-----------------------------|
| response_variable_distribution | bernoulli<br>gaussian<br>poisson<br>binomial<br>negativebinomial<br>wald<br>gamma | 分布のタイプ。デフォルトは bernoulli です。 |

| netezzaglmnode のプロパティ   | 値  | プロパティの説明   |
|-------------------------|--|--|
| distribution_parameter  | number   | 使用する分布パラメータ値。 <b>distribution</b> が <b>Negativebinomial</b> の場合のみ適用されます。   |
| trials                  | integer  | <b>distribution</b> が <b>Binomial</b> の場合のみ適用されます。ターゲット応答が一連の試行が発生するさまざまなイベントの場合、 <b>target</b> フィールドにはイベント数、 <b>trials</b> フィールドには試行回数が含まれます。 |
| model_table             | フィールド  | Netezza 一般化線型モデルが保存されるデータベース テーブルの名前。  |
| max_iterations          | integer  | アルゴリズムが実行できる反復の最大回数。デフォルトは 20 です。  |
| max_error               | number   | アルゴリズムが適合度モデルの検索を停止する最大誤差の値 (科学的表記)。デフォルトは -3、つまり $1E-3$ または 0.001 です。   |
| error_threshold         | number   | 誤差が 0 として扱われる値 (科学的表記)。デフォルトは -7、つまり $1E-7$ (または 0.0000001) を下回る誤差の値が有意でないとカウントされます。   |
| link_function           | identity<br>inverse<br>invnegative<br>invsquare<br>sqrt<br>power<br>oddspower<br>log<br>clog<br>loglog<br>cloglog<br>logit<br>probit<br>gaussit<br>cauchit<br>canbinom<br>cangeom<br>cannegbinom | 使用するリンク関数。デフォルトは <b>logit</b> です。  |
| link_function_parameter | number   | 使用するリンク関数パラメータ値。 <b>link_function</b> が <b>power</b> または <b>oddspower</b> の場合のみ適用されます。   |
| intercept               | フラグ型   | <b>true</b> の場合、モデルに定数項を含みます。  |

## Netezza モデル ナゲットのプロパティ

次のプロパティは、Netezza データベース モデリング ナゲットに共通です。

| Netezza モデル ナゲットの共通プロパティ | 値      | プロパティの説明                        |
|--------------------------|--------|---------------------------------|
| connection               | string | モデルが保存される Netezza データベースの接続文字列。 |
| table_name               | string | モデルが保存されるデータベース テーブルの名前。        |

他のモデルナゲットのプロパティは、対応するモデリングのノードの場合と同じです。

モデル ナゲットのスクリプト名は以下の通りです。

| モデル ナゲット    | スクリプト名                        |
|-------------|-------------------------------|
| ディジション ツリー  | applynetezzadectreenode       |
| K-Means     | applynetezzakmeansnode        |
| ベイズ ネットワーク  | applynetezزابayesnode         |
| Naive Bayes | applynetezzanaivebayesnode    |
| KNN         | applynetezzaknnnode           |
| 分裂クラスタリング   | applynetezзадivclusternode    |
| 主成分分析       | applynetezzapcanode           |
| 回帰ツリー       | applynetezzaregtreenode       |
| 線型回帰        | applynetezalineregressionnode |
| 時系列         | applynetezatimeseriesnode     |
| 一般化線型モデル    | applynetezzaglmnode           |

# 出力ノードのプロパティ

出力ノードのプロパティは、ほかの種類ノードのプロパティと少し異なっています。出力ノードのプロパティは、特定のノード オプションを参照するというよりは、参照を出力オブジェクトに格納します。このことはテーブルから値を取得して、それをストリーム パラメータとして設定するような場合などに役立ちます。

このセクションで、出力ノードで使用できるスクリプト用のプロパティを説明します。

## analysisnode のプロパティ



精度分析ノードで、予測モデルの能力を評価して正確な予測を生成します。分析ノードでは、1 つ以上のモデル ナゲットについて、予測値と実際値をさまざまな方法で比較します。また、分析ノードでは予測モデル同士を比較できます。詳細は、[6 章 精度分析ノード in IBM SPSS Modeler 15 入力ノード、プロセスノード、出力ノード](#) を参照してください。

### 例

```
create analysisnode
# "Analysis" tab
set :analysisnode.coincidence = True
set :analysisnode.performance = True
set :analysisnode.confidence = True
set :analysisnode.threshold = 75
set :analysisnode.improve_accuracy = 3
set :analysisnode.inc_user_measure = True
# "Define User Measure..."
set :analysisnode.user_if = "@TARGET = @PREDICTED"
set :analysisnode.user_then = "101"
set :analysisnode.user_else = "1"
set :analysisnode.user_compute = [Mean Sum]
set :analysisnode.by_fields = ['Drug']
# "Output" tab
set :analysisnode.output_format = HTML
```

```
set :analysisnode.full_filename = "C:/output/analysis_out.html"
```

| analysisnode プロパティ | データ型   | プロパティの説明                                     |
|--------------------|--|--|
| output_mode        | Screen File                                  | 出力ノードから生成される出力の、出力先を指定します。                   |
| use_output_name    | フラグ型   | ユーザー設定の出力名が使用されるかどうかを指定します。                  |
| output_name        | string                                       | use_output_name が真 (true) のときに、使用する名前を指定します。 |
| output_format      | Text (.txt)<br>HTML (.html)<br>Output (.cou) | 出力のタイプを指定します。                                |
| by_fields          | [フィールド<br>フィールド フィールド]                       |  |
| full_filename      | string                                       | ディスク、データ、または HTML の出力を選択した場合の、出力ファイルの名前。     |
| coincidence        | フラグ型   |  |
| performance        | フラグ型   |  |
| confidence         | フラグ型   |  |
| threshold          | number                                       |  |
| improve_accuracy   | number                                       |  |
| inc_user_measure   | フラグ型   |  |
| user_if            | 廃止   |  |
| user_then          | 廃止   |  |
| user_else          | 廃止   |  |
| user_compute       | [Mean Sum Min Max SDev]                      |  |

## dataauditnode のプロパティ



データ検査ノードでは、欠損値、外れ値、および極値に関する情報の他、各フィールドの要約統計量、ヒストグラムや棒グラフを含む、データを広範に検査するための手段を提供しています。結果は把握しやすい行列形式で表示され、ソートしたり、フルサイズのグラフやデータ準備ノードを生成することができます。詳細は、[6 章 データ検査ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

## 例

```

create dataauditnode
connect :variablefilenode to :dataauditnode
set :dataauditnode.custom_fields = True
set :dataauditnode.fields = [Age Na K]
set :dataauditnode.display_graphs = True
set :dataauditnode.basic_stats = True
set :dataauditnode.advanced_stats = True
set :dataauditnode.median_stats = False
set :dataauditnode.calculate = [Count Breakdown]
set :dataauditnode.outlier_detection_method = std
set :dataauditnode.outlier_detection_std_outlier = 1.0
set :dataauditnode.outlier_detection_std_extreme = 3.0
set :dataauditnode.output_mode = Screen

```

| dataauditnodeプロパティ            | データ型                   | プロパティの説明  |
|-------------------------------|------------------------|---|
| custom_fields                 | フラグ型                   |   |
| fields                        | [フィールド1 ...<br>フィールドN] |   |
| overlay                       | フィールド                  |   |
| display_graphs                | フラグ型                   | 出力行列中のグラフ表示をオンまたはオフにするために使用されます。                        |
| basic_stats                   | フラグ型                   |   |
| advanced_stats                | フラグ型                   |   |
| median_stats                  | フラグ型                   |   |
| calculate                     | Count<br>Breakdown     | 欠損値の計算に使用します。計算方法のいずれか、または両方を選択するか、またはどちらも選択しません。       |
| outlier_detection_method      | std<br>iqr             | 外れ値および極値の検出方法を指定します。                                    |
| outlier_detection_std_outlier | number                 | outlier_detection_method が std の場合、外れ値の定義に使用する数値を指定します。 |
| outlier_detection_std_extreme | number                 | outlier_detection_method が std の場合、外れ値の定義に使用する数値を指定します。 |
| outlier_detection_iqr_outlier | number                 | outlier_detection_method が iqr の場合、外れ値の定義に使用する数値を指定します。 |

## 出力ノードのプロパティ

| dataauditnodeプロパティ            | データ型  | プロパティの説明  |
|-------------------------------|---|---|
| outlier_detection_iqr_extreme | number  | outlier_detection_method が iqr の場合、外れ値の定義に使用する数値を指定します。 |
| use_output_name               | フラグ型  | ユーザー設定の出力名が使用されるかどうかを指定します。                             |
| output_name                   | string  | use_output_name が真 (true) のときに、使用する名前を指定します。            |
| output_mode                   | Screen<br>File  | 出力ノードから生成される出力の、出力先を指定します。                              |
| output_format                 | Formatted (. tab)<br>Delimited (. csv)<br>HTML (. html)<br>Output (. cou) | 出力のタイプを指定します。   |
| paginate_output               | フラグ型  | output_format が HTML の場合、出力がページに分割されるようにします。            |
| lines_per_page                | number  | paginate_output と共に使用する場合は、出力ページあたりの行数を指定します。           |
| full_filename                 | string  |   |

## matrixnode のプロパティ



クロス集計ノードで、フィールド間の関係を示すテーブルを作成します。一般的にこのノードは、2つのシンボル値フィールドの関係を示す場合によく使用されますが、フラグ型フィールド間または数値型フィールド間の関係を示すこともできます。詳細は、[6章 クロス集計ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

## 例

```
create matrixnode
# "Settings" tab
set :matrixnode.fields = Numerics
set :matrixnode.row = 'K'
set :matrixnode.column = 'Na'
set :matrixnode.cell_contents = Function
set :matrixnode.function_field = 'Age'
set :matrixnode.function = Sum
# "Appearance" tab
set :matrixnode.sort_mode = Ascending
set :matrixnode.highlight_top = 1
```

```

set :matrixnode.highlight_bottom = 5
set :matrixnode.display = [Counts Expected Residuals]
set :matrixnode.include_totals = True
# "Output" tab
set :matrixnode.full_filename = "C:/output/matrix_output.html"
set :matrixnode.output_format = HTML
set :matrixnode.paginate_output = true
set :matrixnode.lines_per_page = 50

```

| matrixnodeプロパティ        | データ型   | プロパティの説明  |
|------------------------|--|---|
| fields                 | Selected<br>Flags<br>Numerics  |   |
| row                    | フィールド  |   |
| column                 | フィールド  |   |
| include_missing_values | フラグ型   | ユーザーによる欠損値（空白）とシステムによる欠損値（ヌル）が、行と列の出力に含まれるかどうかを指定します。 |
| cell_contents          | CrossTabs<br>Function  |   |
| function_field         | string   |   |
| function               | Sum<br>Mean<br>Min<br>Max<br>SDev                                    |   |
| sort_mode              | Unsorted<br>Ascending<br>Descending                                  |   |
| highlight_top          | number   | ゼロでない場合に真 (true)。<br>。                                |
| highlight_bottom       | number   | ゼロでない場合に真 (true)。<br>。                                |
| display                | [Counts<br>Expected<br>Residuals<br>RowPct<br>ColumnPct<br>TotalPct] |   |
| include_totals         | フラグ型   |   |
| use_output_name        | フラグ型   | ユーザー設定の出力名が使用されるかどうかを指定します。                           |
| output_name            | string   | <b>use_output_name</b> が真 (true) のときに、使用する名前を指定します。   |



## 出力ノードのプロパティ

| matrixnode プロパティ | データ型  | プロパティの説明  |
|------------------|---|---|
| output_mode      | Screen<br>File  | 出力ノードから生成される出力の、出力先を指定します。  |
| output_format    | Formatted (. tab)<br>Delimited (. csv)<br>HTML (. html)<br>Output (. cou) | 出力のタイプを指定します。Formatted と Delimited の両方が、テーブル内で行と列を入れ替える修飾子 <b>transposed</b> を伴うことができます。例を挙げます。<br><b>NODE.output_format=transposed Delimited</b> |
| paginate_output  | フラグ型  | output_format が HTML の場合、出力がページに分割されるようにします。  |
| lines_per_page   | number  | paginate_output と共に使用する場合は、出力ページあたりの行数を指定します。   |
| full_filename    | string  |   |

## meansnode のプロパティ



平均比較ノードでは、独立したグループ間で、または関連するフィールドのペア間で著しい違いがあるかどうかを調べるために、平均を比較します。たとえば、販売促進活動の前後で平均収益を比較したり、販売促進活動を受けなかった顧客と受けた顧客からの収益を比較することができます。詳細は、[6 章 平均比較ノード in IBM SPSS Modeler 15 入力ノード、プロセスノード、出力ノード](#) を参照してください。

## 例

```
create meansnode
set :meansnode.means_mode = BetweenFields
set :meansnode.paired_fields = [{'OPEN_BAL' 'CURR_BAL'}]
set :meansnode.label_correlations = true
set :meansnode.output_view = Advanced
set :meansnode.output_mode = File
set :meansnode.output_format = HTML
set :meansnode.full_filename = "C:/output/means_output.html"
```

| meansnode プロパティ | データ型                           | プロパティの説明  |
|-----------------|--------------------------------|---|
| means_mode      | BetweenGroups<br>BetweenFields | データに実行する平均統計処理の種類を指定します。                              |
| test_fields     | [field1 ... fieldn]            | means_mode が BetweenGroups に設定されているときのテストフィールドを指定します。 |

| meansnodeプロパティ           | データ型  | プロパティの説明  |
|--------------------------|---|---|
| grouping_field           | フィールド                                       | グループにまとめるフィールドを指定します。   |
| paired_fields            | [[field1 field2]<br>{field3 field4}<br>...] | means_mode が BetweenFields に設定されているときに使用するフィールドのペアを指定します。   |
| label_correlations       | フラグ型  | 相関ラベルが出力に表示されるかどうかを指定します。means_mode が BetweenFields に設定されているときにのみ、この設定が適用されます。                    |
| correlation_mode         | Probability<br>Absolute                     | 確率 (Probability) または絶対値 (Absolute) のどちらかで相関にラベルを付けることを指定します。                                      |
| weak_label               | string                                      |   |
| medium_label             | string                                      |   |
| strong_label             | string                                      |   |
| weak_below_probability   | number                                      | correlation_mode が Probability に設定されているときに、弱い相関の分割値を指定します。この値は、たとえば 0.90 のように、0 と 1 の間にする必要があります。 |
| strong_above_probability | number                                      | 強い相関の分割値。   |
| weak_below_absolute      | number                                      | correlation_mode が Absolute に設定されているときに、弱い相関の分割値を指定します。この値は、たとえば 0.90 のように、0 と 1 の間にする必要があります。    |
| strong_above_absolute    | number                                      | 強い相関の分割値。   |
| unimportant_label        | string                                      |   |
| marginal_label           | string                                      |   |
| important_label          | string                                      |   |
| unimportant_below        | number                                      | 低いフィールド重要度の分割値。この値は、たとえば 0.90 のように、0 と 1 の間にする必要があります。  |
| important_above          | number                                      |   |
| use_output_name          | フラグ型  | ユーザー設定の出力名が使用されるかどうかを指定します。   |
| output_name              | string                                      | 使用する名前。   |

| meansnodeプロパティ | データ型  | プロパティの説明   |
|----------------|---|--|
| output_mode    | Screen<br>File  | 出力ノードから生成された出力の出力先を指定します。                                      |
| output_format  | Formatted (. tab)<br>Delimited (. csv)<br>HTML (. html)<br>Output (. cou) | 出力のタイプを指定します。  |
| full_filename  | string  |  |
| output_view    | Simple<br>Advanced  | 出力に単純な (Simple) ビューが表示されるか、または詳細な (Advanced) ビューが表示されるかを指定します。 |

## reportnode のプロパティ



レポート ノードで、固定テキスト、およびデータやデータから導かれた他の式を含む、フォーマット済みレポートを作成します。レポートの書式は、固定テキストとデータの出力構成を定義するテキスト テンプレートを使用して指定します。テンプレート内の HTML タグを使用し、また [出力] タブでオプションを設定することで、カスタムのテキスト書式設定を提供できます。テンプレート内の CLEM 式を使用して、データ値やその他の条件出力を含めることができます。 [詳細は、6 章 レポート ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード を参照してください。](#)

### 例

```
create reportnode
set :reportnode.output_format = HTML
set :reportnode.full_filename = "C:/report_output.html"
set :reportnode.lines_per_page = 50
set :reportnode.title = "Report node created by a script"
set :reportnode.highlights = False
```

| reportnodeプロパティ | データ型  | プロパティの説明                    |
|-----------------|---|-----------------------------|
| output_mode     | Screen<br>File                                  | 出力ノードから生成される出力の、出力先を指定します。  |
| output_format   | HTML (. html)<br>Text (. txt)<br>Output (. cou) | 出力のタイプを指定します。               |
| use_output_name | フラグ型  | ユーザー設定の出力名が使用されるかどうかを指定します。 |

| reportnodeプロパティ | データ型   | プロパティの説明                                     |
|-----------------|--------|--|
| output_name     | string | use_output_name が真 (true) のときに、使用する名前を指定します。 |
| text            | string |  |
| full_filename   | string |  |
| highlights      | フラグ型   |  |
| title           | string |  |
| lines_per_page  | number |  |

## setglobalsnode のプロパティ



グローバル ノードで、データを走査し、CLEM 式で使用する要約値を算出します。たとえば、グローバル ノードを使用して、[年齢] という名前のフィールドの統計量を算出し、次に CLEM 式に @GLOBAL\_MEAN(年齢) 関数を挿入して年齢の全体的な平均を算出することができます。詳細は、6 章 [グローバル ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

### 例

```
create setglobalsnode
connect :typenode to :setglobalsnode
set :setglobalsnode.globals.Na = [Max Sum Mean]
set :setglobalsnode.globals.K = [Max Sum Mean]
set :setglobalsnode.globals.Age = [Max Sum Mean SDev]
set :setglobalsnode.clear_first = False
set :setglobalsnode.show_preview = True
```

| setglobalsnodeプロパティ | データ型                    | プロパティの説明   |
|---------------------|-------------------------|--|
| globals             | [Sum Mean Min Max SDev] | フィールドを設定する構造化プロパティは、次のフォーマットで参照する必要があります。<br>set :setglobalsnode.globals.Age = [Sum Mean Min Max SDev] |
| clear_first         | フラグ型                    |  |
| show_preview        | フラグ型                    |  |

## statisticsnode のプロパティ



記述統計ノードでは、数値型フィールドに関する基本的な集計情報が提供されます。このノードで、個々のフィールドの要約統計量とフィールド間の相関が計算されます。詳細は、6章 記述統計ノード in IBM SPSS Modeler 15 入力ノード、プロセスノード、出カノード を参照してください。

### 例

```
create statisticsnode
# "Settings" tab
set :statisticsnode.examine = ['Age' 'BP' 'Drug']
set :statisticsnode.statistics = [Mean Sum SDev]
set :statisticsnode.correlate = ['BP' 'Drug']
# "Correlation Labels..." section
set :statisticsnode.label_correlations = True
set :statisticsnode.weak_below_absolute = 0.25
set :statisticsnode.weak_label = "lower quartile"
set :statisticsnode.strong_above_absolute = 0.75
set :statisticsnode.medium_label = "middle quartiles"
set :statisticsnode.strong_label = "upper quartile"
# "Output" tab
set :statisticsnode.full_filename = "c:/output/statistics_output.html"
set :statisticsnode.output_format = HTML
```

| statisticsnode プロパティ | データ型   | プロパティの説明                                     |
|----------------------|--|--|
| use_output_name      | フラグ型   | ユーザー設定の出力名が使用されるかどうかを指定します。                  |
| output_name          | string                                       | use_output_name が真 (true) のときに、使用する名前を指定します。 |
| output_mode          | Screen<br>File                               | 出力ノードから生成される出力の、出力先を指定します。                   |
| output_format        | Text (.txt)<br>HTML (.html)<br>Output (.cou) | 出力のタイプを指定します。                                |
| full_filename        | string                                       |  |
| examine              | [フィールド<br>フィールド フィールド]                       |  |
| correlate            | [フィールド<br>フィールド フィールド]                       |  |

| statisticsnode プロパティ     | データ型  | プロパティの説明  |
|--------------------------|---|---|
| statistics               | [Count Mean Sum Min Max Range Variance SDev SErr Median Mode] |   |
| correlation_mode         | Probability Absolute  | 確率 (Probability) または絶対値 (Absolute) のどちらかで相関にラベルを付けることを指定します。                                      |
| label_correlations       | フラグ型  |   |
| weak_label               | string  |   |
| medium_label             | string  |   |
| strong_label             | string  |   |
| weak_below_probability   | number  | correlation_mode が Probability に設定されているときに、弱い相関の分割値を指定します。この値は、たとえば 0.90 のように、0 と 1 の間にする必要があります。 |
| strong_above_probability | number  | 強い相関の分割値。   |
| weak_below_absolute      | number  | correlation_mode が Absolute に設定されているときに、弱い相関の分割値を指定します。この値は、たとえば 0.90 のように、0 と 1 の間にする必要があります。    |
| strong_above_absolute    | number  | 強い相関の分割値。   |

## statisticsoutputnode のプロパティ



Statistics 出力ノードを使用すると、IBM® SPSS® Statistics プロシージャを呼び出し、IBM® SPSS® Modeler データを分析することができます。さまざまな SPSS Statistics 分析プロシージャにアクセスできます。このノードは、ライセンスが与えられた SPSS Statistics のコピーが必要です。詳細は、[8 章 Statistics 出力ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

このノードのプロパティについては、「[statisticsoutputnode のプロパティ](#)」( p. 345 ) に記載されています。

## tablenode のプロパティ



テーブル ノードで、データがテーブル形式で表示されます。このデータは、ファイルにも書き込めます。この機能は、データの値を調査したり、データを読みやすいフォーマットでエクスポートする必要がある場合に役立ちます。詳細は、[6 章 テーブル ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出カノード](#) を参照してください。

### 例

```
create tablenode
set :tablenode.highlight_expr = "Age > 30"
set :tablenode.output_format = HTML
set :tablenode.transpose_data = true
set :tablenode.full_filename = "C:/output/table_output.htm"
set :tablenode.paginate_output = true
set :tablenode.lines_per_page = 50
```

| tablenode プロパティ | データ型  | プロパティの説明   |
|-----------------|---|--|
| full_filename   | string  | ディスク、データ、または HTML の出力を選択した場合の、出力ファイルの名前。             |
| use_output_name | フラグ型  | ユーザー設定の出力名が使用されるかどうかを指定します。                          |
| output_name     | string  | <b>use_output_name</b> が真 (true) のときに、使用する名前を指定します。  |
| output_mode     | Screen<br>File  | 出力ノードから生成される出力の、出力先を指定します。                           |
| output_format   | Formatted (. tab)<br>Delimited (. csv)<br>HTML (. html)<br>Output (. cou) | 出力のタイプを指定します。  |
| transpose_data  | フラグ型  | エクスポート前にデータの行列を入れ替えて、行がフィールドを、列がレコードを表すようにします。       |
| paginate_output | フラグ型  | <b>output_format</b> が HTML の場合、出力がページに分割されるようにします。  |
| lines_per_page  | number  | <b>paginate_output</b> と共に使用する場合は、出力ページあたりの行数を指定します。 |
| highlight_expr  | string  |  |
| output          | string  | ノードで直前に構築されたテーブルへの参照を保持する、読み取り専用プロパティ。               |

| tablenodeプロパティ    | データ型  | プロパティの説明   |
|-------------------|---|--|
| value_labels      | [ {Value LabelString}<br>{ Value LabelString} ]   | 値のペアのためのラベルを指定します。<br>例をあげると、次のようになります。<br><b>set :typenode.value_labels.</b><br><b>'Drug'={drugA label1} {drugB label2}</b>   |
| display_places    | integer   | フィールドが表示される時の小数部の桁数を設定します (REAL ストレージのフィールドにのみ適用)。-1 を設定すると、ストリームのデフォルトが使用されます。<br>使用フォーマット：<br><b>NODE.display_places.</b><br><b>FIELDNAME</b>  |
| export_places     | integer   | フィールドが出力される時の小数部の桁数を設定します (REAL ストレージのフィールドにのみ適用)。-1 を設定すると、ストリームのデフォルトが使用されます。<br>使用フォーマット：<br><b>NODE.export_places.FIELDNAME</b>  |
| decimal_separator | DEFAULT<br>PERIOD<br>COMMA  | フィールドの小数点記号を指定します (REAL ストレージのフィールドにのみ適用)。<br>使用フォーマット：<br><b>NODE.decimal_separator.</b><br><b>FIELDNAME</b>  |
| date_format       | "DDMMYY"<br>"MMDDYY"<br>"YYMMDD"<br>"YYYYMMDD"<br>"YYYYDDD"<br>DAY<br>MONTH<br>"DD-MM-YY"<br>"DD-MM-YYYY"<br>"MM-DD-YY"<br>"MM-DD-YYYY"<br>"DD-MON-YY"<br>"DD-MON-YYYY"<br>"YYYY-MM-DD"<br>"DD.MM.YY"<br>"DD.MM.YYYY"<br>"MM.DD.YY"<br>"MM.DD.YYYY"<br>"DD.MON.YY"<br>"DD.MON.YYYY"<br>"DD/MM/YY"<br>"DD/MM/YYYY"<br>"MM/DD/YY"<br>"MM/DD/YYYY" | フィールドの日付フォーマットを設定します (DATE または TIMESTAMP ストレージのフィールドにのみ適用されます)。<br>使用フォーマット：<br><b>NODE.date_format.FIELDNAME</b><br>例をあげると、次のようになります。<br><b>set</b><br><b>:tablenode.date_format.</b><br><b>'LaunchDate' = "DDMMYY"</b> |



| tablenode プロパティ | データ型   | プロパティの説明   |
|-----------------|--|--|
|                 | "DD/MON/YY"<br>"DD/MON/YYYY"<br>MON YYYY<br>q Q YYYY<br>ww WK YYYY   |  |
| time_format     | "HHMMSS"<br>"HHMM"<br>"MMSS"<br>"HH:MM:SS"<br>"HH:MM"<br>"MM:SS"<br>"(H)H:(M)M:(S)S"<br>"(H)H:(M)M"<br>"(M)M:(S)S"<br>"HH.MM.SS"<br>"HH.MM"<br>"MM.SS"<br>"(H)H.(M)M.(S)S"<br>"(H)H.(M)M"<br>"(M)M.(S)S" | フィールドの日付フォーマットを設定します (TIME または <b>TIMESTAMP</b> ストレージのフィールドにのみ適用されます)。<br>使用フォーマット：<br><b>NODE.time_format.FIELDNAME</b><br>例をあげると、次のようになります。<br><b>set</b><br><b>:tablenode.time_format.</b><br><b>set 'BOF_enter' = "HHMMSS"</b> |
| column_width    | integer  | フィールドに列幅を設定します。-1 という値を指定すると、列幅は <b>Auto</b> に設定されます。<br>使用フォーマット：<br><b>NODE.column_width.FIELDNAME</b>   |
| justify         | AUTO<br>CENTER<br>LEFT<br>RIGHT  | フィールドに列調整を設定します。<br>使用フォーマット：<br><b>NODE.justify.FIELDNAME</b>   |

## transformnode のプロパティ



変換ノードによって、選択フィールドに適用する前に変換の結果を選択し、視覚的に確認することができます。詳細は、6章 [変換ノード in IBM SPSS Modeler 15](#) 入力ノード、プロセスノード、出力ノード を参照してください。

### 例

```
create transformnode
set :transformnode.fields = [AGE INCOME]
set :transformnode.formula = Select
set :transformnode.formula_log_n = true
```

```
set :transformnode.formula_log_n_offset = 1
```

| transformnodeプロパティ     | データ型                          | プロパティの説明   |
|------------------------|-------------------------------|--|
| fields                 | [[フィールド1 ...<br>フィールドN]       | 変換で使用するフィールド。  |
| formula                | All<br>Select                 | すべての変換を計算するか、選択した変換を計算するかを指定します。                     |
| formula_inverse        | フラグ型                          | 逆変換を使用するかどうかを指定します。                                  |
| formula_inverse_offset | number                        | 式で使用するデータ オフセットを指定します。ユーザーが指定しない限り、デフォルトで 0 に設定されます。 |
| formula_log_n          | フラグ型                          | $\log_n$ 変換を使用するかどうかを指定します。                          |
| formula_log_n_offset   | number                        |  |
| formula_log_10         | フラグ型                          | $\log_{10}$ 変換を使用するかどうかを指定します。                       |
| formula_log_10_offset  | number                        |  |
| formula_exponential    | フラグ型                          | 指数変換 ( $e^x$ ) を使用するかどうかを指定します。                      |
| formula_square_root    | フラグ型                          | 平方根変換を使用するかどうかを指定します。                                |
| use_output_name        | フラグ型                          | ユーザー設定の出力名が使用されるかどうかを指定します。                          |
| output_name            | string                        | use_output_name が真 (true) のときに、使用する名前を指定します。         |
| output_mode            | Screen<br>File                | 出力ノードから生成される出力の、出力先を指定します。                           |
| output_format          | HTML (.html)<br>Output (.cou) | 出力のタイプを指定します。  |
| paginate_output        | フラグ型                          | output_format が HTML の場合、出力がページに分割されるようにします。         |
| lines_per_page         | number                        | paginate_output と共に使用する場合は、出力ページあたりの行数を指定します。        |
| full_filename          | string                        | ファイル出力に使用するファイル名を指定します。                              |

# エクスポート ノードのプロパティ

## 共通のエクスポート ノード プロパティ

次のプロパティは、すべてのエクスポート ノードに共通しています。

| プロパティ                  | 値                      | プロパティの説明  |
|------------------------|------------------------|---|
| publish_path           | string                 | 公開されたイメージおよびパラメータ ファイルに使用するルート名を指定します。                    |
| publish_metadata       | フラグ型                   | イメージの入力および出力、それらのデータ モデルを説明するメタデータ ファイルを作成するかどうかを指定します。   |
| publish_use_parameters | フラグ型                   | ストリーム パラメータが *.par ファイルに含まれるかどうかを指定します。                   |
| publish_parameters     | 文字列のリスト                | 使用するパラメータを指定します。  |
| execute_mode           | export_data<br>publish | ストリームを公開せずにノードを実行するかどうか、ノードの実行時にストリームを自動的に公開するかどうかを指定します。 |

## cognosexportnode のプロパティ



IBM Cognos BI エクスポート ノードは、Cognos BI データベースで読み取ることができる形式でデータをエクスポートできます。詳細は、7 章 [IBM Cognos BI エクスポート ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

注：このノードの場合 Cognos 接続と ODBC 接続を定義する必要があります。

## Cognos の接続

Cognos 接続のプロパティは次のとおりです。

| cognosexportnode プロパティ | データ型                                | プロパティの説明   |
|------------------------|-------------------------------------|--|
| cognos_connection      | { "フィールド", "フィールド", ... , "フィールド" } | Cognos サーバーの接続の詳細を含むリストのプロパティ。形式は次のとおりです。<br>{ "Cognos_server_URL", login_mode, "namespace", "username", "password" }<br>ここでの意味は次の通りです。<br>Cognos_server_URL は、エクスポートする Cognos サーバーの URL です。<br>login_mode は匿名ログインが使用されるかどうかを示し、true または false となります。true の場合、次のフィールドは "" に設定されます。<br>namespace はサーバーへのログインに使用するセキュリティ認証プロバイダを指定します。<br>username および password は Cognos サーバーにログインする際に使用するユーザー名とパスワードです。 |
| cognos_package_name    | string                              | データをエクスポートしている Cognos データ ソース（通常はデータベース）のパスおよび名前。次に例を示します。<br>/Public Folders/MyPackage  |
| cognos_datasource      | string                              |  |
| cognos_export_mode     | Publish<br>ExportFile               |  |
| cognos_filename        | string                              |  |

## ODBC 接続

ODBC 接続のプロパティは次のセクションの **databaseexportnode** に示されているものと同じです。ただし、**datasource** プロパティは有効ではありません。

## databaseexportnode のプロパティ



データベース エクスポート ノードで、データを ODBC 対応のリレーショナル データ ソースに書き込みます。ODBC データ ソースに書き込むには、データ ソースが存在し、そのデータ ソースに対する書き込み権限を取得する必要があります。詳細は、7 章 データベース エクスポート ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード を参照してください。

## 例

```

/*
Use this sample with fraud.str from demo folder
Assumes a datasource named "MyDatasource" has been configured
*/
create databaseexport
connect claimvalue:applyneuralnetwork to :databaseexport
# Export tab
set :databaseexport.username = "user"
set :databaseexport.datasource = "MyDatasource"
set :databaseexport.password = "password"
set :databaseexport.table_name = "predictions"
set :databaseexport.write_mode = Create
set :databaseexport.generate_import = true
set :databaseexport.drop_existing_table = true
set :databaseexport.delete_existing_rows = true
set :databaseexport.default_string_size = 32

# Schema dialog
set :databaseexport.type.region = "VARCHAR(10)"
set :databaseexport.export_db_primarykey.id = true
set :databaseexportnode.use_custom_create_table_command = true
set :databaseexportnode.custom_create_table_command = "My SQL Code"

# Indexes dialog
set :databaseexport.use_custom_create_index_command = true
set :databaseexport.custom_create_index_command = \
  "CREATE BITMAP INDEX <index-name> ON <table-name> <(index-columns)>"
set :databaseexport.indexes.MYINDEX.fields = [id region]

```

| databaseexportnode プロパティ | データ型                      | プロパティの説明   |
|--------------------------|---------------------------|--|
| datasource               | string                    |  |
| username                 | string                    |  |
| password                 | string                    |  |
| epassword                | string                    | このスロットは、実行時に読み込み専用になります。暗号化パスワードを生成するには、[ツール]メニューの [パスワード暗号化ツール] を使用してください。詳細は、5 章 p.69 暗号化パスワードの生成 を参照してください。 |
| table_name               | string                    |  |
| write_mode               | Create<br>Append<br>Merge |  |

| databaseexportnodeプロパティ         | データ型                          | プロパティの説明   |
|---------------------------------|-------------------------------|--|
| map                             | string                        | <p>ストリーム フィールド名をデータベース列名にマッピングします (<code>write_mode</code> が <code>Merge</code> の場合にのみ有効)。</p> <p>例:</p> <pre>set :databaseexportnode.map.streamBP = 'databaseBP'</pre> <p>複数のマッピングが、次のようにフィールドの位置に従ってサポートされています。</p> <pre>set :databaseexportnode.map=[{streamfield1 field1}{streamfield2 field2}{streamfield3 field3}]</pre> <p>結合の場合、すべてのフィールドをマッピングしてエクスポートする必要があります。データベース内に存在しないフィールド名が、新しい列として追加されます。</p> |
| key_fields                      | [フィールド<br>フィールド ...<br>フィールド] | <p>キーに使用されるストリーム フィールドを指定します。 <code>map</code> プロパティは、データベースでストリーム フィールド内で対応する内容を表示します。</p>  |
| join                            | Database<br>Add               | <p>例:</p> <pre>set :databaseexportnode.join = Database</pre>   |
| drop_existing_table             | フラグ型                          |  |
| delete_existing_rows            | フラグ型                          |  |
| default_string_size             | integer                       |  |
| type                            |                               | <p>スキーマ タイプの設定に用いられる構造化プロパティ。</p> <p>使用フォーマット :</p> <pre>set :databaseexportnode.type.BP = 'VARCHAR(10)'</pre>   |
| generate_import                 | フラグ型                          |  |
| use_custom_create_table_command | フラグ型                          | <p><code>custom_create_table</code> スロットを使用して、標準の <code>CREATE TABLE</code> SQL コマンドを変更します。</p>  |
| custom_create_table_command     | string                        | <p>標準の <code>CREATE TABLE</code> SQL コマンドの代わりに使用する文字列コマンドを指定します。</p>   |

## エクスポート ノードのプロパティ

| databaseexportnodeプロパティ | データ型                    | プロパティの説明   |
|-------------------------|-------------------------|--|
| use_batch               | フラグ型                    | 次のプロパティは、データベースのバルク ロード用の詳細オプションです。<br><b>Use_batch</b> に真 (true) の値を指定すると、行単位のデータベースへのコミットが無効になります。                               |
| batch_size              | number                  | メモリーにコミットする前にデータベースに送信するレコード数を指定します。   |
| bulk_loading            | Off<br>ODBC<br>External | バルク ロードの種類を指定します。ODBC および External 用の付加オプションを次に示します。   |
| odbc_binding            | Row<br>Column           | ODBC 経由のバルク ロードにおける、行方向または列方向のバインドを指定します。  |
| loader_delimit_mode     | Tab<br>Space<br>Other   | 外部プログラム経由のバルク ロードの場合に、区切り文字の種類を指定します。<br><b>Other</b> は、<br><b>loader_other_delimiter</b> プロパティと組み合わせて選択し、カンマ (,) のような区切り文字を指定します。 |
| loader_other_delimiter  | string                  |  |
| specify_data_file       | フラグ型                    | 真 (true) を設定すると、以下の <b>data_file</b> プロパティが有効になります。このプロパティには、データベースにバルク ロードする際<br>の書き込み先のファイル名とパスを指定することができます。                     |
| data_file               | string                  |  |
| specify_loader_program  | フラグ型                    | 真 (true) を設定すると、以下の <b>loader_program</b> プロパティが有効になります。このプロパティには、外部ローダー スクリプトまたはプログラムの名前と場所を指定することができます。                          |
| loader_program          | string                  |  |

| databaseexportnodeプロパティ                           | データ型   | プロパティの説明  |
|---|--------|---|
| gen_logfile                                       | フラグ型   | 真 (true) を設定すると、以下の logfile_name が有効になります。このプロパティには、エラー ログを生成するための、サーバー上のファイル名を指定することができます。       |
| logfile_name                                      | string |   |
| check_table_size                                  | フラグ型   | 真 (true) を設定すると、IBM® SPSS® Modeler からエクスポートされる行数に対応してデータベースのテーブル サイズを確実に増加させるために、テーブル検査が実施されます。   |
| loader_options                                    | string | ローダー プログラムに対して、-comment および -specialdir のような、他の引数を指定します。  |
| export_db_primarykey                              | フラグ型   | 指定されたフィールドがプライマリ キーかどうかを指定します。  |
| use_custom_create_index_command                   | フラグ型   | true の場合、すべてのインデックスに対してカスタム SQL (ユーザー指定のSQL) を有効にします。   |
| custom_create_index_command                       | string | カスタム SQL (ユーザー指定のSQL) が有効にされている場合、インデックスの作成に使用される SQL コマンドを指定します。(この値は、下に示す特定のインデックスに対して上書きできます。) |
| indexes.INDEXNAME.fields                          |        | 必要な場合は指定されたインデックスを作成し、そのインデックスに含まれるフィールド名を一覧表示します。  |
| indexes.INDEXNAME.use_custom_create_index_command | フラグ型   | 特定のインデックスに対してカスタム SQL (ユーザー指定のSQL) を有効または無効にするのに使用されます。   |
| indexes.INDEXNAME.custom_create_command           |        | 指定されたインデックスに使用されるカスタム SQL (ユーザー指定のSQL) を使用します。  |



## エクスポート ノードのプロパティ

| databaseexportnode プロパティ | データ型   | プロパティの説明                                |
|--------------------------|--------|---|
| indexes.INDEXNAME.remove | フラグ型   | true の場合、指定されたインデックスをインデックスのセットから削除します。 |
| table_space              | string | 作成されるテーブル スペースを指定します。                   |
| use_partition            | フラグ型   | 分布ハッシュ フィールドが使用されるよう指定します。              |
| partition_field          | string | 分布ハッシュ フィールドの内容を消去します。                  |

注：一部のデータベースで、データベース テーブルが圧縮してエクスポートのために作成されるよう指定できます (SQL の **CREATE TABLE MYTABLE (...)** **COMPRESS YES;** と同等)。次のようにプロパティ **use\_compression** および **compression\_mode** を指定して、この機能をサポートします。

| databaseexportnode プロパティ | データ型   | プロパティの説明   |
|--------------------------|--|--|
| use_compression          | ブール  | true に設定した場合は、圧縮によるエクスポート用のテーブルを作成します。   |
| compression_mode         | Row<br>Page  | SQL Server データベースの圧縮レベルを設定します。   |
|                          | Default<br>Direct_Load_Operations<br>All_Operations<br>Basic<br>OLTP<br>Query_High<br>Query_Low<br>Archive_High<br>Archive_Low | Oracle データベースの圧縮レベルを設定します。値 <b>OLTP</b> 、 <b>Query_High</b> 、 <b>Query_Low</b> 、 <b>Archive_High</b> 、および <b>Archive_Low</b> には最低限 Oracle 11gR2 が必要です。 |

## 例 - SQL Server

```
var DBSource
set DBSource = get node TestCompressionSQL
set ^DBSource.use_compression = true
set ^DBSource.compression_mode = Page
```

```
execute DBSource
```

## 例 - Oracle 11gR1

```
var DBSource
set DBSource = get node TestCompressionOracle11gR1
set ^DBSource.use_compression = true
set ^DBSource.compression_mode = Direct_Load_Operations
```

```
execute DBSource
```

### 例 - Oracle 11gR2

```
var DBSource
set DBSource = get node TestCompressionOracle11gR2
set ^DBSource.use_compression = true
set ^DBSource.compression_mode = Basic
```

```
execute DBSource
```

## datacollectionexportnode のプロパティ



IBM® SPSS® Data Collection エクスポート ノードは、Data Collection の市場調査ソフトウェアで使用する形式でデータを出力します。このノードを使用するには、Data Collection Data Library がインストールされている必要があります。詳細は、7 章 IBM SPSS Data Collection エクスポート ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード を参照してください。

### 例

```
create datacollectionexportnode
set :datacollectionexportnode.metadata_file = "c:\museums.mdd"
set :datacollectionexportnode.merge_metadata = Overwrite
set :datacollectionexportnode.casedata_file = "c:\museumdata.sav"
set :datacollectionexportnode.generate_import = true
set :datacollectionexportnode.enable_system_variables = true
```

| datacollectionexportnode プロパティ | データ型                      | プロパティの説明  |
|--------------------------------|---------------------------|---|
| metadata_file                  | string                    | 出力するメタデータ ファイルの名前。  |
| merge_metadata                 | Overwrite<br>MergeCurrent |   |
| enable_system_variables        | フラグ型                      | エクスポートされた .mdd ファイルに Data Collection システム変数を含むかどうかを指定します。 |
| casedata_file                  | string                    | ケース データがエクスポートされる .sav ファイルの名前。                           |
| generate_import                | フラグ型                      |   |

## excelexportnode のプロパティ



Excel エクスポート ノードは、Microsoft Excel 形式 (.xls) でデータを出力します。オプションで、ノードが実行されるときに自動的に Excel が起動し、エクスポートするファイルを開けるように選択できます。詳細は、7 章 Excel エクスポート ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード を参照してください。

### 例

```
create excelexportnode
set :excelexportnode.full_filename = "C:/output/myexport.xls"
set :excelexportnode.excel_file_type = Excel2007
set :excelexportnode.inc_field_names = True
set :excelexportnode.inc_labels_as_cell_notes = False
set :excelexportnode.launch_application = True
set :excelexportnode.generate_import = True
```

| excelexportnode プロパティ | データ型                   | プロパティの説明  |
|-----------------------|------------------------|---|
| full_filename         | string                 |   |
| excel_file_type       | Excel2003<br>Excel2007 |   |
| export_mode           | Create<br>Append       |   |
| inc_field_names       | フラグ型                   | フィールド名がワークシートの最初の行に表示されるかどうかを指定します。   |
| start_cell            | string                 | エクスポートの開始セルを指定します。  |
| worksheet_name        | string                 | 書き込むワークシートの名前。  |
| launch_application    | フラグ型                   | Excel が結果のファイルで呼び出されるかどうかを指定します。Excel を起動するパスは、[ヘルパー アプリケーション] ダイアログ ボックス ([ツール] メニューから [ヘルパー アプリケーション]) 内で指定する必要があります。 |
| generate_import       | フラグ型                   | 出力されたデータ ファイルを読み込む Excel 入力ノードが生成されるかどうかを指定します。   |

## outputfilenode プロパティ



ファイル ノードでは、データが区切り文字で区切られたテキスト ファイルへ出力されます。このことは、他の分析ソフトウェアや表計算ソフトウェアに読み込めるフォーマットでデータをエクスポートする場合に、役立ちます。詳細は、7 章 [ファイルエクスポート ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

### 例

```
create outputfile
set :outputfile.full_filename = "c:/output/flatfile_output.txt"
set :outputfile.write_mode = Append
set :outputfile.inc_field_names = False
set :outputfile.use_newline_after_records = False
set :outputfile.delimit_mode = Tab
set :outputfile.other_delimiter = ";"
set :outputfile.quote_mode = Double
set :outputfile.other_quote = "*"
set :outputfile.decimal_symbol = Period
set :outputfile.generate_import = True
```

| outputfilenode プロパティ      | データ型                                      | プロパティの説明   |
|---------------------------|---|------------|
| full_filename             | string                                    | 出力ファイルの名前。 |
| write_mode                | Overwrite<br>Append                       |            |
| inc_field_names           | フラグ型                                      |            |
| use_newline_after_records | フラグ型                                      |            |
| delimit_mode              | Comma<br>Tab<br>Space<br>Other            |            |
| other_delimiter           | char                                      |            |
| quote_mode                | None<br>Single<br>Double<br>Other         |            |
| other_quote               | フラグ型                                      |            |
| generate_import           | フラグ型                                      |            |
| encoding                  | StreamDefault<br>SystemDefault<br>"UTF-8" |            |

## sasexportnode のプロパティ



SAS エクスポート ノードで、SAS または SAS 互換ソフトウェア パッケージで読み込むデータを、SAS フォーマットで出力できます。3 つの SAS ファイル形式が利用可能です。SAS for Windows/OS2、SAS for UNIX、または SAS バージョン 7/8 [詳細](#)は、[7 章 SAS エクスポート ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

### 例

```
create sasexportnode
set :sasexportnode.full_filename = "c:/output/SAS_output.sas7bdat"
set :sasexportnode.format = SAS8
set :sasexportnode.export_names = NamesAndLabels
set :sasexportnode.generate_import = True
```

| sasexportnode プロパティ | データ型                            | プロパティの説明  |
|---------------------|---------------------------------|---|
| format              | Windows<br>UNIX<br>SAS7<br>SAS8 | バリエーション プロパティ ラベル フィールド。  |
| full_filename       | string                          |   |
| export_names        | NamesAndLabels<br>NamesAsLabels | エクスポート時にフィールド名を IBM® SPSS® Modeler から IBM® SPSS® Statistics または SAS 変数名に関連付けます。 |
| generate_import     | フラグ型                            |   |

## statisticsexportnode のプロパティ



Statistics エクスポート ノードでは、IBM® SPSS® Statistics.sav フォーマットでデータを出力します。.sav ファイルは、SPSS Statistics Base およびその他の製品で読み込むことができます。このフォーマットは、IBM® SPSS® Modeler のキャッシュ ファイルでも使用されます。 [詳細は、8 章 Statistics エクスポート ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

このノードのプロパティについては、「[statisticsexportnode のプロパティ](#)」 ( p. 346 ) に記載されています。

## xmlexportnode のプロパティ



XML エクスポート ノードでは、XML 形式のファイルにデータを出力します。オプションで、エクスポートしたデータをストリームに読み込む XML 入力ノードを作成できます。詳細は、7 章 XML エクスポート ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード を参照してください。

### 例

```
create xmlexportnode
set :xmlexportnode.full_filename = "c:\export\data.xml"
set :xmlexportnode.map = [{"/catalog/book/genre" genre}{"catalog/book/title" title}]
```

| xmlexportnode プロパティ  | データ型   | プロパティの説明  |
|----------------------|--------|---|
| full_filename        | string | (必須) XML エクスポート ファイルの完全パスおよびファイル名。  |
| use_xml_schema       | フラグ型   | XML スキーマ (XSD ファイルまたは DTD ファイル) を使用して、エクスポートされたデータの構造を制御するかどうかを指定します。   |
| full_schema_filename | string | 使用する XSD ファイルまたは DTD ファイルの完全パスおよびファイル名。use_xml_schema が true に設定されている場合にのみ必須です。   |
| generate_import      | フラグ型   | エクスポートされたデータ ファイルをストリームに読み込む XML 入力ノードを、自動的に生成します。  |
| records              | string | レコードの境界を示す XPath 式。   |
| map                  | string | XML 構造にフィールド名をマッピングします。<br>例:<br>set :xmlexportnode.map = [{"/top/node1" field1}{"top/node2" field2}]<br>ここでは、ストリーム フィールド field1 を XML 要素 /top/node1 にマッピングするなどします。 |

# IBM SPSS Statistics ノードのプロパティ

## statisticsimportnode のプロパティ



Statistics ファイル ノードは、同じフォーマットを使用する IBM® SPSS® Statistics で使用される .sav ファイル形式のデータおよび IBM® SPSS® Modeler に保存されたキャッシュ ファイルを読み込みます。詳細は、[8 章 Statistics ファイル ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

### 例

```
create statisticsimportnode
set :statisticsimportnode.full_filename = "C:/data/drug1n.sav"
set :statisticsimportnode.import_names = true
set :statisticsimportnode.import_data = true
```

| statisticsimportnode のプロパティ  | データ型                            | プロパティの説明  |
|------------------------------|---------------------------------|---|
| full_filename                | 文字列                             | パスを含む、完全なファイル名。                                   |
| import_names                 | NamesAndLabels<br>LabelsAsNames | 変数名と変数ラベルを処理する方法。                                 |
| import_data                  | DataAndLabels<br>LabelsAsData   | 値とラベルを処理する方法。                                     |
| use_field_format_for_storage | ブール                             | インポート時に SPSS Statistics フィールド形式情報を使用するかどうかを指定します。 |

## statisticstransformnode プロパティ



Statistics 変換ノードは、IBM® SPSS® Modeler のデータ ソースに対する IBM® SPSS® Statistics シンタックス コマンドの選択を行います。このノードは、ライセンスが与えられた SPSS Statistics のコピーが必要です。詳細は、[8 章 Statistics 変換ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

## 例

```
create statisticstransformnode
set :statisticstransformnode.syntax = "COMPUTE NewVar = Na + K."
set :statisticstransformnode.new_name.NewVar = "Mixed Drugs"
set :statisticstransformnode.check_before_saving = true
```

| statisticstransformnode のプロパティ | データ型 | プロパティの説明  |
|--------------------------------|------|---|
| syntax                         | 文字列  |   |
| check_before_saving            | フラグ型 | 項目を保存する前に、入力されたシンタックスを検証します。シンタックスが無効な場合は、エラーメッセージを表示します。 |
| default_include                | フラグ型 | 詳細は、14 章 p.176 filternode のプロパティを参照してください。                |
| include                        | フラグ型 | 詳細は、14 章 p.176 filternode のプロパティを参照してください。                |
| new_name                       | 文字列  | 詳細は、14 章 p.176 filternode のプロパティを参照してください。                |

## statisticsmodelnode のプロパティ



Statistics モデル ノードを使用すると、PMML を作成する IBM® SPSS® Statistics プロシージャを実行してデータを分析および使用することができます。このノードは、ライセンスが与えられた SPSS Statistics のコピーが必要です。詳細は、8 章 [Statistics モデル ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

## 例

```
create statisticsmodelnode
set :statisticsmodelnode.syntax = "COMPUTE NewVar = Na + K."
set :statisticsmodelnode.new_name.NewVar = "Mixed Drugs"
```

| statisticsmodelnode のプロパティ | データ型 | プロパティの説明                                   |
|----------------------------|------|--|
| syntax                     | 文字列  |  |
| default_include            | フラグ型 | 詳細は、14 章 p.176 filternode のプロパティを参照してください。 |



| statisticsmodelnode のプロパティ | データ型 | プロパティの説明                                   |
|----------------------------|------|--|
| include                    | フラグ型 | 詳細は、14 章 p.176 filternode のプロパティを参照してください。 |
| new_name                   | 文字列  | 詳細は、14 章 p.176 filternode のプロパティを参照してください。 |

## statisticsoutputnode のプロパティ



Statistics 出力ノードを使用すると、IBM® SPSS® Statistics プロシージャを呼び出し、IBM® SPSS® Modeler データを分析することができます。さまざまな SPSS Statistics 分析プロシージャにアクセスできます。このノードは、ライセンスが与えられた SPSS Statistics のコピーが必要です。詳細は、8 章 Statistics 出力ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード を参照してください。

### 例

```
create statisticsoutputnode
set :statisticsoutputnode.syntax = "SORT CASES BY Age(A) Sex(A) BP(A) Cholesterol(A)"
set :statisticsoutputnode.use_output_name = False
set :statisticsoutputnode.output_mode = File
set :statisticsoutputnode.full_filename = "Cases by Age, Sex and Medical History"
set :statisticsoutputnode.file_type = HTML
```

| statisticsoutputnode のプロパティ | データ型               | プロパティの説明   |
|-----------------------------|--------------------|--|
| mode                        | Dialog Syntax      | 「SPSS Statistics ダイアログ」オプションまたはシNTAX エディタを選択します。 |
| syntax                      | 文字列                |  |
| use_output_name             | フラグ型               |  |
| output_name                 | 文字列                |  |
| output_mode                 | Screen File        |  |
| full_filename               | 文字列                |  |
| file_type                   | HTML<br>SPV<br>SPW |  |

## statisticsexportnode のプロパティ



Statistics エクスポート ノードでは、IBM® SPSS® Statistics.sav フォーマットでデータを出力します。.sav ファイルは、SPSS Statistics Base およびその他の製品で読み込むことができます。このフォーマットは、IBM® SPSS® Modeler のキャッシュファイルでも使用されます。詳細は、8 章 Statistics エクスポート ノード in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード を参照してください。

### 例

```
create statisticsexportnode
set :statisticsexportnode.full_filename = "c:/output/SPSS_Statistics_out.sav"
set :statisticsexportnode.field_names = Names
set :statisticsexportnode.launch_application = True
set :statisticsexportnode.generate_import = True
```

| statisticsexportnode のプロパティ | データ型                            | プロパティの説明   |
|-----------------------------|---------------------------------|--|
| full_filename               | 文字列                             |  |
| launch_application          | フラグ型                            |  |
| export_names                | NamesAndLabels<br>NamesAsLabels | エクスポート時にフィールド名を SPSS Modeler から SPSS Statistics または SAS変数名に関連付けます。 |
| generate_import             | フラグ型                            |  |

# スーパーノードのプロパティ

スーパーノード固有のプロパティを次の表に示します。共通のノード プロパティもスーパーノードに適用されることに注意してください。

テーブル 22-1  
source\_supernode

| プロパティ名     | プロパティの種類/値のリスト | プロパティの説明   |
|------------|----------------|--|
| parameters | いずれでも可         | このプロパティを使って、スーパーノードのパラメータ テーブルに指定されるパラメータを作成、アクセスします。詳細は以下を参照してください。 |

テーブル 22-2  
process\_supernode

| プロパティ名     | プロパティの種類/値のリスト | プロパティの説明   |
|------------|----------------|--|
| parameters | いずれでも可         | このプロパティを使って、スーパーノードのパラメータ テーブルに指定されるパラメータを作成、アクセスします。詳細は以下を参照してください。 |

テーブル 22-3  
terminal\_supernode

| プロパティ名         | プロパティの種類/値のリスト   | プロパティの説明   |
|----------------|------------------|--|
| parameters     | いずれでも可           | このプロパティを使って、スーパーノードのパラメータ テーブルに指定されるパラメータを作成、アクセスします。詳細は以下を参照してください。 |
| execute_method | Script<br>Normal |  |
| script         | 文字列              |  |

## スーパーノードのパラメータ

次の一般フォーマットを使用して、スーパーノードのパラメータを作成または設定するためにスクリプトを使用できます。

```
set mySuperNode.parameters.minvalue = 30
```

また、次のように、名前に加えて（または名前の代わりに）スーパーノードのデータ型を指定できます。

```
set :process_supernode.parameters.minvalue = 30
```

```
set mySuperNode:process_supernode.parameters.minvalue = 30
```

次のように、CLEM 式を使用してパラメータ値も設定できます。

```
set :process_supernode.parameters.minvalue = "<expression>"
```

### カプセル化ノードのプロパティ設定

設定するノードおよびプロパティのリテラル名前に一致するスーパーノード パラメータを作成することで、スーパーノードの中にカプセル化された特定のノードにプロパティを設定できます。たとえば、データを読み込むためにカプセル化された可変長ファイルのある入力スーパーノードがあるとします。読み込みファイルの名前（`full_filename` プロパティを使用して指定します）を次のように渡すことができます。

```
set :source_supernode.parameters.':variablefilenode.full_filename' = "c:/mydata.txt"
```

こうすることで、`c:/mydata.txt` の値がある `:variablefilenode.full_filename` という名前のスーパーノード パラメータが作成されます。指定したデータ型のノードがスーパーノードの中に存在することを仮定して、その名前つきプロパティの値がふさわしく設定されます。ただし、これは、スーパーノード スクリプトではなく、ストリーム スクリプト（スーパーノードを「含む」ストリームのためのスクリプト）の中で行われます。パラメータ名を指定するには必ず一重引用符を使用します。

このアプローチは、有効なノードおよびプロパティ参照結果がある限り、任意のカプセル化ノードで使用できます。たとえば、カプセル化されたサンプル ノードのために `rand_pct` プロパティを設定するには、次のどれでも使用できます。

```
set mySuperNode.parameters.':samplenode.rand_pct' = 50
```

または

```
set mySuperNode.parameters.'Sample.rand_pct'= 50
```

または

```
set mySuperNode.parameters.'Sample:samplenode.rand_pct'= 50
```

上記の最初の例は、ストリームの中のサンプル ノードが 1 つだけであることを想定しています。2 番目はデータ型に関わらず “Sample” と名づけられた唯一のノードが存在することを想定しています。3 番目の例は、ノードの名前とデータ型の両方を指定しているために最も明示的です。

詳細は、[9 章 スーパーノードのパラメータ in IBM SPSS Modeler 15 入力ノード、プロセス ノード、出力ノード](#) を参照してください。

**スーパーノード スクリプトの制約** スーパーノードは、他のストリームを操作したり、現在のストリームを変更することはできません。このために、`open stream`、`get stream`、`execute_script` などのストリームに適用されるコマンドは、スーパーノードスクリプトでは使用できません。

# 注意事項

この情報は、世界各国で提供される製品およびサービス向けに作成されています。

IBMはこのドキュメントで説明する製品、サービス、機能は他の国では提供していない場合があります。現在お住まいの地域で利用可能な製品、サービス、および、情報については、お近くの IBM の担当者にお問い合わせください。IBM 製品、プログラム、またはサービスに対する参照は、IBM 製品、プログラム、またはサービスのみが使用することができることを説明したり意味するものではありません。IBM の知的所有権を侵害しない機能的に同等の製品、プログラム、またはサービスを代わりに使用することができます。ただし、IBM 以外の製品、プログラム、またはサービスの動作を評価および確認するのはユーザーの責任によるものです。

IBMは、本ドキュメントに記載されている内容に関し、特許または特許出願中の可能性があります。本ドキュメントの提供によって、これらの特許に関するいかなる権利も使用者に付与するものではありません。ライセンスのお問い合わせは、書面にて、下記住所に送ることができます。

IBM Director of Licensing, IBM Corporation, North Castle Drive,  
Armonk, NY 10504-1785, U. S. A.

2 バイト文字セット (DBCS) 情報についてのライセンスに関するお問い合わせは、お住まいの国の IBM Intellectual Property Department に連絡するか、書面にて下記宛先にお送りください。

神奈川県大和市下鶴間1623番14号 日本アイ・ビー・エム株式会社 法務・知的財産 知的財産権ライセンス渉外

**以下の条項は、イギリスまたはこのような条項が法律に反する他の国では適用されません。** International Business Machines は、明示的または黙示的に関わらず、第三者の権利の侵害しない、商品性または特定の目的に対する適合性の暗黙の保証を含むがこれに限定されない、いかなる保証なく、本出版物を「そのまま」提供します一部の州では、特定の取引の明示的または暗示的な保証の免責を許可していないため、この文が適用されない場合があります。

この情報には、技術的に不適切な記述や誤植を含む場合があります。情報については変更が定期的に行われます。これらの変更は本書の新版に追加されます。IBM は、本書に記載されている製品およびプログラムについて、事前の告知なくいつでも改善および変更を行う場合があります。

IBM 以外の Web サイトに対するこの情報内のすべての参照は、便宜上提供されているものであり、決してそれらの Web サイトを推奨するものではありません。これらの Web サイトの資料はこの IBM 製品の資料に含まれるものではなく、これらの Web サイトの使用はお客様の責任によるものとします。

IBM はお客様に対する一切の義務を負うことなく、自ら適切と考える方法で、情報を使用または配布することができるものとします。

本プログラムのライセンス取得者が (i) 別途作成されたプログラムと他のプログラム（本プログラムを含む）との間の情報交換および (ii) 交換された情報の相互利用を目的とした本プログラムに関する情報の所有を希望する場合、下記住所にお問い合わせください。

IBM Software Group, Attention:Licensing, 233 S. Wacker Dr., Chicago, IL 60606, USA.

上記のような情報は、該当する条項および条件に従い、有料で利用できるものとします。

本ドキュメントに記載されている許可されたプログラムおよびそのプログラムに使用できるすべてのライセンス認証された資料は、IBM Customer Agreement、IBM International Program License Agreement、および当社とかわした同等の契約の条件に基づき、IBM によって提供されます。

ここに記載されているパフォーマンスデータは、すべて管理環境下で確認されたものです。そのため、他の操作環境で得られた結果は大きく異なる可能性があります。開発レベルのシステムで測定が行われている場合があり、これらの測定値は一般に利用可能なシステムと同じであることを保証するものではありません。また、測定値が推定値である可能性があり、実際の結果は異なる場合があります。本ドキュメントのユーザーは、特定の環境に適したデータを検証する必要があります。

IBM 以外の製品に関する情報は、それらの製品の供給業者、公開済みの発表、または公開で使用できるソースから取得しています。IBM は、それらの製品のテストは行っておらず、IBM 以外の製品に関連する性能、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給業者に通知する必要があります。

IBM の将来の方向性または意向に関する記述については、予告なく変更または取り消すことがあり、目的や目標のみを示すものです。

この情報には、日常の業務処理で用いられるデータや報告書の例が含まれています。できる限り詳細に説明するため、例には、個人、企業、ブランド、製品などの名前が使用されています。これらの名称はすべて架空のものであり、実際の企業で使用される名称および住所とは一切関係ありません。

この情報をソフトコピーでご覧になっている場合は、写真やカラーのイラストが表示されない場合があります。

## 商標

IBM、IBM ロゴ、および [ibm.com](http://ibm.com)、SPSS は、世界の多くの国で登録された IBM Corporation の商標です。IBM の商標の現在のリストは、<http://www.ibm.com/legal/copytrade.shtml> を参照してください。

Intel、Intel のロゴ、Intel Inside、Intel Inside のロゴ、Intel Centrino、Intel Centrino のロゴ、Celeron、Intel Xeon、Intel SpeedStep、Itanium、および Pentium は、米国およびその他の国の Intel Corporation または関連会社の商標または登録商標です。

Linux は、米国およびその他の国における Linus Torvalds の登録商標です。

Microsoft、Windows、Windows NT、および Windows のロゴは、米国およびその他の国における Microsoft 社の商標です。

UNIX は、米国およびその他の国における The Open Group の登録商標です。

Java およびすべての Java ベースの商標およびロゴは、米国およびその他の国の Sun Microsystems, Inc. の商標です。

その他の製品名およびサービス名等は、IBM または他の会社の商標です。





# 索引

- 非等号演算子, 97
- 文字列関数, 66, 104
- 等号演算子, 97
- 三角関数, 101
- 優先順位, 91
- 分布関数, 102
- 変換関数, 96
- 実行順序
  - スクリプトによる変更, 66
- 情報関数, 95
- 指数関数, 100
- 数値関数, 100
- 日付関数, 89-90
  - date\_before, 97, 110
  - date\_days\_difference, 110
  - date\_in\_days, 110
  - date\_in\_months, 110
  - date\_in\_weeks, 110
  - date\_in\_years, 110
  - date\_months\_difference, 110
  - date\_weeks\_difference, 110
  - date\_years\_difference, 110
  - @TODAY 関数, 110
- 時間関数, 89-90
  - time\_before, 97, 110
  - time\_hours\_difference, 110
  - time\_in\_hours, 110
  - time\_in\_mins, 110
  - time\_in\_secs, 110
  - time\_mins\_difference, 110
  - time\_secs\_difference, 110
- 正規分布
  - 確率関数, 102
- 比較関数, 97
- 特殊変数, 27
- 特殊関数, 122
- 確率関数, 102
- 空白文字
  - 文字列から削除する, 104
- 表記方法, 94
- 論理関数, 99
- 報告書
  - スクリプトを使用して作成, 73, 76
- 文字列, 86, 88
  - ケースの変更, 66
  - スクリプト, 24
- 演算子
  - スクリプト, 32
  - 文字列の結合, 77, 96
- 商標, 352
- 変数, 29, 39
  - スクリプト, 23, 27
  - ノードの参照, 24
- 実数, 86-87
- 引数
  - IBM SPSS Collaboration and Deployment Services Repository 接続, 82
  - コマンド ライン, 80
  - サーバー接続, 81
  - システム, 83
- 数値, 87
- 整数, 86-87
- 文字, 86, 88
- 日付
  - 変換, 114
  - 操作, 114
- 概要, 86
- 注釈
  - スクリプトでのアクセス, 78
- 継続
  - スクリプト, 33
- 関数, 89-90, 94-95, 115
  - @FIELD, 122
  - @GLOBAL\_MAX, 120
  - @GLOBAL\_MEAN, 120
  - @GLOBAL\_MIN, 120
  - @GLOBAL\_SDEV, 120
  - @GLOBAL\_SUM, 120
  - @PARTITION, 122
  - @PREDICTED, 122
  - @TARGET, 122
- 例
  - 概要, 7
  - アプリケーション ガイド, 5
- 式, 86
  
- abs 関数, 100
- aggregatenode のプロパティ, 155
- allbutfirst 関数, 104
- allbutlast 関数, 104
- alphabefore 関数, 104
- analysisnode のプロパティ, 316
- AND 演算子, 99
- anomalydetectionnode のプロパティ, 214
- anonymizenode のプロパティ, 165
- appendnode のプロパティ, 155
- applyanomalydetectionnode のプロパティ, 275
- applyapriorinode のプロパティ, 276
- applyautoclassifiernode のプロパティ, 276
- applyautoclusternode のプロパティ, 277
- applyautonumericnode プロパティ, 277
- applybayesnetnode のプロパティ, 277
- applyc50node のプロパティ, 278

- applycarmanode のプロパティ, 278
- applycartnode のプロパティ, 279
- applychaidnode のプロパティ, 279
- applycoxregnode のプロパティ, 280
- applydb2imclusternode のプロパティ, 306
- applydb2imlognode のプロパティ, 306
- applydb2imbnnode のプロパティ, 306
- applydb2imregnode のプロパティ, 306
- applydb2imtreenode のプロパティ, 306
- applydecisionlistnode のプロパティ, 280
- applydiscriminantnode のプロパティ, 280
- applyfactornode のプロパティ, 281
- applyfeatureselectionnode のプロパティ, 281
- applygeneralizedlinearnode のプロパティ, 281
- applykmeansnode のプロパティ, 282
- applyknnnode のプロパティ, 282
- applykohonenode のプロパティ, 282
- applylinearnode プロパティ, 283
- applylogregnode のプロパティ, 283
- applymlogisticnode のプロパティ, 292
- applymneuralnetworknode のプロパティ, 292
- applymregressionnode のプロパティ, 292
- applymsequenceclusternode properties, 293
- applymstimeseriesnode properties, 292
- applymstreenode のプロパティ, 291
- applynetez zabayesnode のプロパティ, 315
- applynetez zadectreenode のプロパティ, 315
- applynetez zativclusternode のプロパティ, 315
- applynetez zakmeansnode のプロパティ, 315
- applynetez zaknnnode のプロパティ, 315
- applynetez zalineresgressionnode のプロパティ, 315
- applynetez zanaivebayesnode のプロパティ, 315
- applynetez zapcanode のプロパティ, 315
- applynetez zaregtreenode のプロパティ, 315
- applyneuralnetnode のプロパティ, 283
- applyneuralnetworknode プロパティ, 284
- applyoraabnnode のプロパティ, 299
- applyoradecisiontreenode のプロパティ, 299
- applyorakmeansnode のプロパティ, 299
- applyoranbnode のプロパティ, 299
- applyoranmfnode のプロパティ, 299
- applyoraoclusternode のプロパティ, 299
- applyorasvmnode のプロパティ, 299
- applyquestnode のプロパティ, 284
- applyregressionnode のプロパティ, 285
- applyselflearningnode のプロパティ, 285
- applysequencenode のプロパティ, 285
- applysvmnode のプロパティ, 286
- applytimeseriesnode のプロパティ, 286
- applytwostepnode のプロパティ, 286
- apriori モデル
  - ノードのスクリプト プロパティ, 215, 276
- apriorinode のプロパティ, 215
- arccos 関数, 101
- arccosh 関数, 101
- arcsin 関数, 101
- arcsinh 関数, 101
- arctan 関数, 101
- arctan2 関数, 101
- arctanh 関数, 101
- autoclassifiernode のプロパティ, 217
- autoclusternode のプロパティ, 220
- autodatapreprenode のプロパティ, 166
- autonumericnode のプロパティ, 221
- balancenode のプロパティ, 156
- binningnode のプロパティ, 169
- @BLANK 関数, 95, 121
- C&R Tree モデル
  - ノードのスクリプト プロパティ, 227, 279
- C5.0 モデル
  - ノードのスクリプト プロパティ, 224, 278
- c50node のプロパティ, 224
- CARMA モデル
  - ノードのスクリプト プロパティ, 226, 278
- carmanode のプロパティ, 226
- cartnode のプロパティ, 227
- cdf\_chisq 関数, 102
- cdf\_f 関数, 102
- cdf\_normal 関数, 102
- cdf\_t 関数, 102
- CHAID モデル
  - ノードのスクリプト プロパティ, 230, 279
- chaidnode のプロパティ, 230
- clear generated palette コマンド, 52, 71
- clear stream コマンド, 56
- CLEM
  - 式, 86
  - language, 86
  - スクリプト, 10, 23
  - データ型, 87-89
- CLEM 関数
  - 三角関数, 101
  - 特殊関数, 122
  - 数値型, 100
  - 無作為, 104
  - 変換, 96
  - 情報, 95
  - 比較, 97
  - 論理, 99
  - probability, 102
  - sequence, 115-116
  - string, 104
  - グローバル, 120

- 日付と時間, 110
- 空白とヌル, 121
- ビット単位, 102
- 利用可能リスト, 93
- CLEM 式
  - parameters, 30
  - スクリプト, 33, 39
  - テキストの検索および置換, 19
- CLEM 式内の円記号, 88
- Clem 式ビルダー
  - テキストの検索および置換, 19
- close FILE コマンド, 62
- close STREAM コマンド, 56
- cognosimportnode プロパティ, 137
- collectionnode のプロパティ, 198
- column\_count プロパティ, 62
- connect NODE コマンド, 44
- cos 関数, 101
- cosh 関数, 101
- count\_equal 関数, 97
- count\_greater\_than 関数, 97
- count\_less\_than 関数, 97
- count\_non\_nulls 関数, 97
- count\_not\_equal 関数, 97
- count\_nulls 関数, 97
- count\_substring 関数, 104
- Cox 回帰モデル
  - ノードのスクリプト プロパティ, 232, 280
- coxregnode のプロパティ, 232
- create NODE コマンド, 43
- create stream コマンド, 56
- dataauditnode のプロパティ, 317
- databaseexportnode のプロパティ, 332
- databasenode のプロパティ, 139
- datacollectionexportnode のプロパティ, 338
- datacollectionimportnode のプロパティ, 141
- date\_before 関数, 97
- datetime\_date 関数, 96
- db2imassocnode のプロパティ, 301
- db2imclusternode のプロパティ, 304
- db2imlognode のプロパティ, 305
- db2imbnnode のプロパティ, 304
- db2imregnode のプロパティ, 302
- db2imsequencenode のプロパティ, 302
- db2imtseriesnode のプロパティ, 305
- db2imtreenode のプロパティ, 300
- decisionlist のプロパティ, 234
- delete model コマンド, 52
- delete NODE コマンド, 44
- delete output コマンド, 64
- derivnode のプロパティ, 172
- DIFF 関数, 116
- @DIFF 関数, 115-116
- directedwebnode のプロパティ, 211
- disable NODE コマンド, 44
- disconnect NODE コマンド, 45
- discriminantnode のプロパティ, 236
- distinctnode のプロパティ, 157
- distributionnode のプロパティ, 199
- div 関数, 100
- duplicate NODE コマンド, 45
- enable NODE コマンド, 45
- endstring 関数, 104
- ensemblenode のプロパティ, 174
- Enterprise View ノード
  - プロパティ, 145
- evaluationnode のプロパティ, 200
- evimportnode のプロパティ, 145
- Excel エクスポート ノード
  - プロパティ, 339
- Excel 入力ノード
  - プロパティ, 144
- excelexportnode のプロパティ, 339
- excelimportnod のプロパティ, 144
- execute NODE コマンド, 45
- execute\_all コマンド, 36
- execute\_project コマンド, 60
- execute\_script コマンド, 36
- exit コマンド, 32, 36
- export model コマンド, 53
- export NODE コマンド, 46
- export output コマンド, 64
- f 分布
  - 確率関数, 102
- factornode のプロパティ, 237
- featureselectionnode のプロパティ, 16, 239
- @FIELD 関数, 122
- fields, 86, 89
  - スクリプトの無効化, 197
- @FIELDS\_BETWEEN 関数, 122
- @FIELDS\_MATCHING 関数, 122
- fillernode のプロパティ, 175
- filternode のプロパティ, 176
- first\_index 関数, 97
- first\_non\_null 関数, 97
- first\_non\_null\_index 関数, 97
- fixedfilenode のプロパティ, 146
- flags
  - 複数のフラグの組み合わせ, 80
- flush NODE コマンド, 47
- for コマンド, 28, 32, 66, 73, 76
- for...endfor コマンド, 37
- fracof 関数, 100
- generated キーワード, 71
- genlinnode のプロパティ, 241
- get node コマンド, 47
- get output コマンド, 64
- get stream コマンド, 57

## 索引

- get コマンド, 26
- GLMM モデル
  - ノードのスクリプト プロパティ, 245
- glmnode のプロパティ, 245
- graphboardnode のプロパティ, 202
- hasendstring 関数, 104
- hasmidstring 関数, 104
- hasstartstring 関数, 104
- hassubstring 関数, 104
- histogramnode のプロパティ, 204
- historynode のプロパティ, 177
- HTML 出力
  - スクリプトを使用して作成, 73, 76
- HTML 形式
  - ノードのエクスポート, 46
  - モデルのエクスポート, 53
- IBM Cognos BI 入力ノード
  - プロパティ, 137
- IBM DB2 モデル
  - ノードのスクリプト プロパティ, 300
- IBM ISW Naive Bayes モデル
  - ノードのスクリプト プロパティ, 304, 306
- IBM ISW アソシエーション モデル
  - ノードのスクリプト プロパティ, 301, 306
- IBM ISW クラスタリング モデル
  - ノードのスクリプト プロパティ, 304, 306
- IBM ISW シーケンス モデル
  - ノードのスクリプト プロパティ, 302, 306
- IBM ISW ディシジョン ツリー モデル
  - ノードのスクリプト プロパティ, 300, 306
- IBM ISW 時系列モデル
  - ノードのスクリプト プロパティ, 305
- IBM ISW 回帰モデル
  - ノードのスクリプト プロパティ, 302, 306
- IBM ISW ロジスティック回帰モデル
  - ノードのスクリプト プロパティ, 305-306
- IBM SPSS Collaboration and Deployment Services Repository
  - コマンド ラインの引数, 82
  - スクリプト, 67
- IBM SPSS Data Collection エクスポート ノード
  - プロパティ, 338
- IBM SPSS Data Collection 入力ノード
  - プロパティ, 141
- IBM SPSS Modeler, 1
  - コマンド ラインからの実行, 79
  - ドキュメンテーション, 5
- IBM SPSS Statistics エクスポート ノード
  - プロパティ, 346
- IBM SPSS Statistics 入力ノード
  - プロパティ, 343
- IBM SPSS Statistics 出力ノード
  - プロパティ, 345
- IBM SPSS Statistics 変換ノード
  - プロパティ, 343
- IBM SPSS Statistics モデル
  - ノードのスクリプト プロパティ, 344
- if コマンド, 32, 73
- if...then...else コマンド, 38
- if、then、else 関数, 99
- INDEX 関数, 116
- @INDEX 関数, 115-116
- insert model コマンド, 54
- integer\_bitcount 関数, 102
- integer\_leastbit 関数, 102
- integer\_length 関数, 102
- intof 関数, 100
- is\_date 関数, 95
- is\_datetime 関数, 95
- is\_integer 関数, 95
- is\_number 関数, 95
- is\_real 関数, 95
- is\_string 関数, 95
- is\_time 関数, 95
- is\_timestamp 関数, 95
- isalphacode 関数, 104
- isendstring 関数, 104
- islowercode 関数, 104
- ismidstring 関数, 104
- isnumbercode 関数, 104
- isstartstring 関数, 104
- issubstring 関数, 104
- issubstring\_count 関数, 104
- issubstring\_lim 関数, 104
- isuppercode 関数, 104
- K-Means モデル
  - ノードのスクリプト プロパティ, 248, 282
- kmeansnode のプロパティ, 248
- KNN モデル
  - ノードのスクリプト プロパティ, 282
- knnnode のプロパティ, 249
- Kohonen モデル
  - ノードのスクリプト プロパティ, 251, 282
- kohonnenode のプロパティ, 251
- last\_index 関数, 97
- LAST\_NON\_BLANK 関数, 116
- @LAST\_NON\_BLANK 関数, 115-116, 121
- last\_non\_null 関数, 97
- last\_non\_null\_index 関数, 97
- length 関数, 104
- linear プロパティ, 252
- load model コマンド, 54
- load node コマンド, 47
- load output コマンド, 65
- load project コマンド, 60
- load state コマンド, 61
- load stream コマンド, 57

- locchar 関数, 104
- locchar\_back 関数, 104
- log 関数, 100
- log10 関数, 100
- logregnode のプロパティ, 254
- lowertoupper 関数, 66, 104
- matches 関数, 104
- matrixnode のプロパティ, 319
- max 関数, 97
- MAX 関数, 116
- @MAX 関数, 115-116
- max\_index 関数, 97
- max\_n 関数, 97
- MEAN 関数, 115-116
- @MEAN 関数, 115-116
- mean\_n 関数, 100
- meansnode のプロパティ, 321
- member 関数, 97
- mergenode のプロパティ, 158
- Microsoft モデル
  - ノードのスクリプト プロパティ, 288, 291
- min 関数, 97
- MIN 関数, 116
- @MIN 関数, 115-116
- min\_index 関数, 97
- min\_n 関数, 97
- mod 関数, 100
- MS Linear Regression
  - ノードのスクリプト プロパティ, 288, 292
- MS Logistic Regression
  - ノードのスクリプト プロパティ, 288, 292
- MS Neural Network
  - ノードのスクリプト プロパティ, 288, 292
- MS シーケンス クラスタリング
  - ノードのスクリプト プロパティ, 293
- MS タイム シリーズ
  - ノードのスクリプト プロパティ, 292
- MS ディシジョン ツリー
  - ノードのスクリプト プロパティ, 288, 291
- msassocnode のプロパティ, 288
- msbayesnode のプロパティ, 288
- msclusternode のプロパティ, 288
- mslogisticnode のプロパティ, 288
- msneuralnetworknode のプロパティ, 288
- msregressionnode のプロパティ, 288
- mssequenceclusternode properties, 288
- mstimeseriesnode properties, 288
- mstreenode のプロパティ, 288
- @MULTI\_RESPONSE\_SET 関数, 122
- multiplotnode のプロパティ, 206
- multiset コマンド, 127
- negate 関数, 100
- Netezza K-Means モデル
  - ノードのスクリプト プロパティ, 309, 315
- Netezza KNN モデル
  - ノードのスクリプト プロパティ, 310, 315
- Netezza Naive Bayes モデル
  - ノードのスクリプト プロパティ, 309
- Netezza Naive Bayesmodels
  - ノードのスクリプト プロパティ, 315
- Netezza 分裂クラスタリング モデル
  - ノードのスクリプト プロパティ, 310, 315
- Netezza 回帰ツリー モデル
  - ノードのスクリプト プロパティ, 311, 315
- Netezza ディシジョン ツリー モデル
  - ノードのスクリプト プロパティ, 308, 315
- Netezza ベイズ ネットワーク モデル
  - ノードのスクリプト プロパティ, 309, 315
- Netezza モデル
  - ノードのスクリプト プロパティ, 307
- Netezza 一般化線型モデル
  - ノードのスクリプト プロパティ, 313
- Netezza 主成分分析モデル
  - ノードのスクリプト プロパティ, 311, 315
- Netezza 線型回帰モデル
  - ノードのスクリプト プロパティ, 312, 315
- Netezza 時系列モデル
  - ノードのスクリプト プロパティ, 312
- netezabayesnode のプロパティ, 309
- netezadectreenode のプロパティ, 308
- netezadivclusternode のプロパティ, 310
- netezzaglmnode のプロパティ, 313
- netezzakmeansnode のプロパティ, 309
- netezzaknnnode のプロパティ, 310
- netezzalineressionnode のプロパティ, 312
- netezzanaivebayesnode のプロパティ, 309
- netezzapcanode のプロパティ, 311
- netezzaregtreenode のプロパティ, 311
- netezzatimeseriesnode のプロパティ, 312
- neuralnetnode のプロパティ, 258
- neuralnetworknode プロパティ, 261
- nodes
  - スクリプトでのループ, 66
- NOT 演算子, 99
- @NULL 関数, 95, 121
- OFFSET 関数, 116
- @OFFSET 関数, 115-116
- oneof 関数, 104
- open FILE コマンド, 62
- open stream コマンド, 28, 57
- OR 演算子, 99
- oraabnnode のプロパティ, 294
- oraainode properties, 298
- oraapriorinode のプロパティ, 297
- Oracle Adaptive Bayes モデル
  - ノードのスクリプト プロパティ, 294, 299

## 索引

- Oracle AI モデル
  - ノードのスクリプト プロパティ, 298
- Oracle Apriori モデル
  - ノードのスクリプト プロパティ, 297, 300
- Oracle Decision Tree モデル
  - ノードのスクリプト プロパティ, 296, 299
- Oracle KMeans モデル
  - ノードのスクリプト プロパティ, 297, 299
- Oracle MDL モデル
  - ノードのスクリプト プロパティ, 298, 300
- Oracle Naive Bayes モデル
  - ノードのスクリプト プロパティ, 294, 299
- Oracle NMF モデル
  - ノードのスクリプト プロパティ, 297, 299
- Oracle O-Cluster
  - ノードのスクリプト プロパティ, 296, 299
- Oracle Support Vector Machines モデル
  - ノードのスクリプト プロパティ, 295, 299
- Oracle モデル
  - ノードのスクリプト プロパティ, 293
- Oracle 一般化線型モデル
  - ノードのスクリプト プロパティ, 295
- oradecisiontreenode のプロパティ, 296
- oraglmnode のプロパティ, 295
- orakmeansnode のプロパティ, 297
- oramdlnode のプロパティ, 298
- oranbnode のプロパティ, 294
- oranmfnode のプロパティ, 297
- oraoclusternode のプロパティ, 296
- orasvmnode のプロパティ, 295
- outputfilenode のプロパティ, 340
- parameters, 17, 39, 126-128, 131
  - スクリプト, 23, 33
  - ストリーム, 30
  - セッション, 30
- @PARTITION\_FIELD 関数, 122
- partitionnode のプロパティ, 178
- pi 関数, 101
- plotnode のプロパティ, 207
- PMML フォーマット
  - ノードのエクスポート, 46
  - モデルのエクスポート, 53
- position NODE コマンド, 47
- power (指数) 関数, 100
- @PREDICTED 関数, 122
- QUEST モデル
  - ノードのスクリプト プロパティ, 263, 284
- questnode のプロパティ, 263
- random 関数, 104
- random0 関数, 104
- reclassifynode のプロパティ, 179
- regressionnode のプロパティ, 265
- rem 関数, 100
- rename NODE コマンド, 29, 48
- reordernode のプロパティ, 181
- replace 関数, 104
- replicate 関数, 104
- reportnode のプロパティ, 323
- restructurenode のプロパティ, 181
- retrieve model コマンド, 55
- retrieve node コマンド, 48
- retrieve output コマンド, 65
- retrieve project コマンド, 60
- retrieve stream コマンド, 57
- retrieve コマンド, 67
- RFM 分析ノード
  - プロパティ, 182
- RFM レコード集計ノード
  - プロパティ, 160
- rfmaggregatenode のプロパティ, 160
- rfmanalysisnode のプロパティ, 182
- round 関数, 100
- row\_count プロパティ, 62
- samplenode のプロパティ, 161
- SAS エクスポート ノード
  - プロパティ, 341
- SAS 入力ノード
  - プロパティ, 148
- sasexportnode のプロパティ, 341
- sasimportnode のプロパティ, 148
- save model コマンド, 55
- save node コマンド, 49
- save output コマンド, 65
- save project コマンド, 60
- save STREAM コマンド, 58
- save コマンド, 27
- SDEV 関数, 116
- @SDEV 関数, 115-116
- sdev\_n 関数, 100
- selectnode のプロパティ, 164
- sequencenode のプロパティ, 267
- set コマンド, 24, 28-30, 39
- setglobalsnode のプロパティ, 324
- settoflagnode のプロパティ, 184
- sign 関数, 100
- sin 関数, 101
- SINCE 関数, 116
- @SINCE 関数, 115-116
- sinh 関数, 101
- skipchar 関数, 104
- skipchar\_back 関数, 104
- SLRM モデル
  - ノードのスクリプト プロパティ, 269, 285
- slrmnode のプロパティ, 269
- sortnode のプロパティ, 164
- soundex 関数, 110
- soundex\_difference 関数, 110
- SPSS Modeler Server, 2

- SQL フォーマット
    - ノードのエクスポート, 46, 53
  - sqrt 関数, 100
  - startstring 関数, 104
  - statisticsexportnode のプロパティ, 346
  - statisticsimportnode のプロパティ, 16, 343
  - statisticsmodelnode のプロパティ, 344
  - statisticsnode のプロパティ, 325
  - statisticsoutputnode のプロパティ, 345
  - statisticstransformnode プロパティ, 343
  - store model コマンド, 55
  - store node コマンド, 49
  - store output コマンド, 65
  - store project コマンド, 60
  - store stream コマンド, 59
  - store コマンド, 67
  - stream.nodes のプロパティ, 66
  - stripchar 関数, 104
  - strmember 関数, 104
  - subscrs 関数, 104
  - substring 関数, 104
  - substring\_between 関数, 104
  - SUM 関数, 116
  - @SUM 関数, 115-116
  - sum\_n 関数, 100
  - SVM モデル
    - ノードのスクリプト プロパティ, 270
  - svmlnode プロパティ, 270
  - t 分布
    - 確率関数, 102
  - tablenode のプロパティ, 327
  - tan 関数, 101
  - tanh 関数, 101
  - @TARGET 関数, 122
  - testbit 関数, 102
  - @TESTING\_PARTITION 関数, 122
  - THIS 関数, 116
  - @THIS 関数, 115-116
  - time\_before 関数, 97
  - timeintervalnode のプロパティ, 185
  - timeplotnode のプロパティ, 209
  - timeseriesnode のプロパティ, 271
  - to\_date 関数, 96, 110
  - to\_dateline 関数, 110
  - to\_datetime 関数, 96
  - to\_integer 関数, 96
  - to\_number 関数, 96
  - to\_real 関数, 96
  - to\_string 関数, 96
  - to\_time 関数, 96, 110
  - to\_timestamp 関数, 96, 110
  - @TODAY 関数, 110
  - @TRAINING\_PARTITION 関数, 122
  - transformnode のプロパティ, 329
  - transposenode のプロパティ, 190
  - trim 関数, 104
  - trim\_start 関数, 104
  - trimend 関数, 104
  - TwoStep モデル
    - ノードのスクリプト プロパティ, 273, 286
  - twostepnode のプロパティ, 273
  - typenode のプロパティ, 16, 74, 191
  - undef 関数, 121
  - unicode\_char 関数, 104
  - unicode\_value 関数, 104
  - uppertolower 関数, 104
  - userinputnode のプロパティ, 149
  - @VALIDATION\_PARTITION 関数, 122
  - value コマンド, 61
  - value\_at 関数, 97
  - var コマンド, 24, 29, 42
  - variablefilenode のプロパティ, 151
  - Web グラフ ノード
    - プロパティ, 211
  - webnode のプロパティ, 211
  - with stream コマンド, 28, 59
  - write FILE コマンド, 62
  - writeln FILE コマンド, 62, 73, 76
  - XML エクスポート ノード
    - プロパティ, 342
  - XML 入力ノード
    - プロパティ, 154
  - xmlexportnode のプロパティ, 342
  - xmlexportnode のプロパティ, 154
- 
- アプリケーションの例, 5
  - アンサンブル ノード
    - プロパティ, 174
- 
- エクスポート
    - nodes, 46
    - PMML, 46, 53
    - SQL, 46, 53
    - モデル, 53
  - エクスポート ノード
    - ノードのスクリプト プロパティ, 331
  - エラーのチェック
    - スクリプト, 70
- 
- 出力オブジェクト
    - スクリプト名, 63
    - スクリプト コマンド, 63
  - 結果オブジェクト
    - スクリプト コマンド, 61

## 索引

- カイ 2 乗分布
  - 確率関数, 102
- カレットのシンタックス
  - 変数の参照, 24, 29
- 自動クラスター ノード
  - ノードのスクリプト プロパティ, 220
- 自動クラスター モデル
  - ノードのスクリプト プロパティ, 277
- 集計棒グラフ ノード
  - プロパティ, 198
- 棒グラフ ノード
  - プロパティ, 199
- 線グラフ ノード
  - プロパティ, 206
- グラフ作成ノード
  - スクリプトのプロパティ, 197
- グラフボード ノード
  - プロパティ, 202
- クロス集計ノード
  - プロパティ, 319
- グローバル関数, 120
- グローバル ノード
  - プロパティ, 324
- コマンド ライン
  - IBM SPSS Modeler の実行, 79
  - IBM SPSS Modeler の起動, 79
  - parameters, 85
  - スクリプト, 71
  - 複数の引数, 80
  - 引数のリスト, 81-83
- コメント
  - スクリプト, 33
- サポート ベクター マシン モデル
  - ノードのスクリプト プロパティ, 286
- サポート ベクトル マシン モデル
  - ノードのスクリプト プロパティ, 270
- 生成されたモデル
  - スクリプト名, 49, 52
- サンプル ノード
  - プロパティ, 161
- サーバー
  - コマンド ラインの引数, 81
- システム
  - コマンド ラインの引数, 83
- シーケンス関数, 115-116
- シーケンス モデル
  - ノードのスクリプト プロパティ, 267, 285
- スクリプト
  - 演算子, 32
  - 中断, 19
  - 保存, 11
  - 実行, 19
  - 概要, 10, 23
  - 継続, 33
  - 例, 73, 76
  - CLEM 式, 33
  - nodes, 24
  - syntax, 23
  - エラーのチェック, 70
  - グラフ作成ノード, 197
  - コマンド ラインから, 71
  - コメント, 33
  - 使用されている省略形, 128
  - スクリプトの実行, 32
  - スタンドアロン スクリプト, 10
  - ストリーム, 10
  - ストリームの実行順序, 66
  - スーパーノード内, 17
  - スーパーノード スクリプト, 10
  - テキスト ファイルからのインポート, 11
  - テキストの検索および置換, 19
  - 現在のオブジェクト, 27
  - 以前のバージョンとの互換性, 71
  - 共通のプロパティ, 130
  - 出力ノード, 316
  - フィールド選択モデル, 16
  - ユーザー インターフェイス, 11, 14, 17
- スクリプトの中断, 19
- スクリプトの実行, 19
- スタンドアロン スクリプト, 10, 14
- ステート型オブジェクト
  - スクリプト コマンド, 61
- ストリーム
  - multiset コマンド, 126
  - スクリプト, 10-11
  - プロパティ, 131
- ストリーム名
  - スクリプトでのアクセス, 77
- ストリーム オブジェクト
  - 参照, 28
  - 開く, 28
  - スクリプト コマンド, 55
- ストリーム パラメータ, 30, 39
- ストリームの実行順序
  - スクリプトによる変更, 66
- ストリームのプロパティ, 77
- スペース
  - 文字列から削除する, 104
- スロット パラメータ, 17, 39, 126, 129
- スーパーノード, 126
- parameters, 30, 39



- スクリプト, 10, 17-18, 347
- パラメータ, 347
- プロパティ, 347
- プロパティの設定, 347
- セキュリティ
  - 暗号化パスワード, 69, 81
  - セッション パラメータ, 30, 39
- ソート ノード
  - プロパティ, 164
- ツリー成長ディレクティブ
  - スクリプト内への埋め込み, 34
- ディシジョン リスト モデル
  - ノードのスクリプト プロパティ, 234, 280
- テキスト文字列
  - スクリプト内への埋め込み, 34
- テキスト形式
  - ノードのエクスポート, 46
  - モデルのエクスポート, 53
- テキストの検索, 19
- テキストの置換, 19
- 自動データ準備
  - プロパティ, 166
- データ分割ノード
  - プロパティ, 169
- データ分類ノード
  - プロパティ, 179
- データ区分ノード
  - プロパティ, 178
- データ検査ノード
  - プロパティ, 317
- データ型ノード
  - プロパティ, 191
- データベース エクスポート ノード
  - プロパティ, 332
- データベース ノード
  - プロパティ, 139
- データベース モデル作成, 287
- テーブル ノード
  - プロパティ, 327
- 日付と時間関数
  - datetime\_date, 110
  - datetime\_day, 110
  - datetime\_day\_name, 110
  - datetime\_day\_short\_name, 110
  - datetime\_hour, 110
  - datetime\_in\_seconds, 110
  - datetime\_minute, 110
  - datetime\_month, 110
  - datetime\_month\_name, 110
  - datetime\_month\_short\_name, 110
  - datetime\_now datetime\_second, 110
  - datetime\_time, 110
  - datetime\_timestamp, 110
  - datetime\_weekday, 110
  - datetime\_year, 110
- ドキュメンテーション, 5
- 時間と日付の関数, 89-90
- ナゲット
  - ノードのスクリプト プロパティ, 275
- 法律に関する注意事項, 350
- ニューラル ネットワーク
  - ノードのスクリプト プロパティ, 261, 284
- ニューラル ネットワーク モデル
  - ノードのスクリプト プロパティ, 258, 283
- 演算子の優先順位, 91
- 文字列の連結, 96
- 日付の書式, 89-90
- 空白の処理
  - CLEM 関数, 121
- 現在のオブジェクト
  - スクリプト内での参照, 27
- 時間のフォーマット, 89-90
- 値のプロパティ, 75
- 平均比較ノード
  - プロパティ, 321
- 時間区分ノード
  - プロパティ, 185
- 条件抽出ノード
  - プロパティ, 164
- 精度分析ノード
  - プロパティ, 316
- 自動分類ノード
  - ノードのスクリプト プロパティ, 217
- 行列入替ノード
  - プロパティ, 190
- 記述統計ノード
  - プロパティ, 325
- 再構成ノード
  - プロパティ, 181
- 匿名化ノード
  - プロパティ, 165
- 固定長ノード
  - プロパティ, 146
- 散布図ノード
  - プロパティ, 207
- 時系列ノード
  - プロパティ, 177, 209

## 索引

- 入力ノード
  - プロパティ, 135
- 出力ノード
  - スクリプトのプロパティ, 316
- 変換ノード
  - プロパティ, 329
- 置換ノード
  - プロパティ, 175
- 評価ノード
  - プロパティ, 200
- 順序ノード
  - プロパティ, 181
- ノード ID
  - スクリプト内での参照, 24
- ノード オブジェクト
  - スクリプト, 24
  - スクリプト コマンド, 42
- 数値予測ノード プロパティ, 221
- ノードのスクリプト プロパティ, 287
  - エクスポート ノード, 331
  - モデル ナゲット, 275
  - モデル作成ノード, 213
- ノードのプロパティ
  - スクリプトでのアクセス, 78
  
- パスワード
  - 暗号化, 81
  - スクリプトへの追加, 69
- 暗号化パスワード
  - スクリプトへの追加, 69
- パラメータ
  - スーパーノード, 347
- バランス ノード
  - プロパティ, 156
  
- ヒストグラム ノード
  - プロパティ, 204
- ビット単位関数, 102
  
- ファイル オブジェクト
  - スクリプト コマンド, 62
- ファイル ノード
  - プロパティ, 340
- 可変長ファイル ノード
  - プロパティ, 151
- フィルタ ノード
  - プロパティ, 176
- フィールド名
  - ケースの変更, 66
- 名義型フィールド
  - 値のプロパティ, 75
- 連続型フィールド
  - 値のプロパティ, 75
- 時間フィールド
  - 変換, 114
- フィールド生成ノード
  - プロパティ, 172
- フィールド順序ノード
  - プロパティ, 181
- フィールド選択モデル
  - 適用, 16
  - スクリプト, 16
  - ノードのスクリプト プロパティ, 239, 281
- フラグ
  - コマンド ラインの引数, 79
- フラグ設定ノード
  - プロパティ, 184
- フラグ型フィールド
  - 値のプロパティ, 75
- プロジェクト
  - プロパティ, 134
- プロパティ, 39
  - スクリプト, 126-129, 213, 275, 331
  - 共通スクリプト, 130
  - ストリーム, 131
  - スーパーノード, 347
  - データベース モデル作成ノード, 287
  - フィルタ ノード, 127
  - プロジェクト, 134
- 構造化プロパティ, 127
  
- ベイズ ネットワーク モデル
  - ノードのスクリプト プロパティ, 223, 277
- ベイズネット プロパティ, 223
  
- モデル
  - エクスポート, 53
  - スクリプト, 53
  - スクリプト名, 49, 52
- 自己学習応答モデル
  - ノードのスクリプト プロパティ, 269, 285
- 一般化線型モデル
  - ノードのスクリプト プロパティ, 241, 281
- 主成分分析モデル
  - ノードのスクリプト プロパティ, 237, 281
- 異常値検出モデル
  - ノードのスクリプト プロパティ, 214, 275
- 判別分析モデル
  - ノードのスクリプト プロパティ, 236, 280
- 因子分析モデル
  - ノードのスクリプト プロパティ, 237, 281
- 線型回帰モデル
  - ノードのスクリプト プロパティ, 265, 285
- 自動分類モデル
  - ノードのスクリプト プロパティ, 276

自動数値モデル  
ノードのスクリプト プロパティ, 221, 277  
時系列モデル  
ノードのスクリプト プロパティ, 271, 286  
最近隣モデル  
ノードのスクリプト プロパティ, 249  
線型モデル  
ノードのスクリプト プロパティ, 252, 283  
モデル オブジェクト  
スクリプト名, 49, 52  
スクリプト コマンド, 49  
モデル ナゲット  
スクリプト名, 49, 52  
ノードのスクリプト プロパティ, 275  
モデル作成ノード  
ノードのスクリプト プロパティ, 213

ユーザー入力ノード  
プロパティ, 149

より大演算子, 97  
より小演算子, 97

リスト, 86, 88  
リスト パラメータ  
スクリプト内での変更, 32  
リテラル  
スクリプト, 23, 34  
リテラル文字列  
スクリプト内への埋め込み, 34

ループ  
スクリプトでの使用, 66, 75-76

重複レコード ノード  
プロパティ, 157  
レコード結合ノード  
プロパティ, 158  
レコード追加ノード  
プロパティ, 155  
レコード集計ノード  
プロパティ, 155  
レポート ノード, 73, 76  
プロパティ, 323

ロジスティック回帰モデル  
ノードのスクリプト プロパティ, 254, 283  
ローカル変数, 29, 39