IBM SPSS Modeler 14.2 スクリプト とオートメーション ガイド

注 ： サポートされている情報および製品をご利用いただく前に、「注意事項」(p.) の一般情報をお読みください。

本マニュアルには、SPSS Inc., an IBM Company が所有する情報が含まれています。これらの情報は使用許諾契約書に基づいて提供され、著作権法によって保護されています。本文書に記載されている情報には、製品の保証は含まれていません。また本マニュアルに記載されている文は製品の保証を規定しないものとします。

IBM または SPSS に情報を送信すると、あなたに対する義務を負うことなく、適切とする方法でその情報を使用または配布する非独占的権利と IBM および SPSS 付与するものとします。

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により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

章 1

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 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。ここに挙げた以外にも、ほかの製品やアップデートも利用できる可能性があります。 詳細は、http://www.ibm.com/software/analytics/spss/products/modeler/ を参照してください。

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、ユーザーのデスクトップ 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 Server: SPSS Modeler Server は、1 つまたは複数の IBM® SPSS® Modeler のインストールと同時に分散分析モードで継続的に実行し、大規模なデータセットを使用する際にパフォーマンスが大幅に向上しますが、それは、データをクライアント コンピュータへダウンロードする

ことなく、メモリー主体の操作をサーバー上で実行できるからです。また、SPSS Modeler Server は SQL 最適化のサポート、データベース内モデル作成機能を提供し、パフォーマンスおよび自動化にさらなるメリットをもたらします。分析を実行するには、少なくとも 1 つの SPSS Modeler をインストールしておく必要があります。

IBM SPSS Modeler のオプション

次のコンポーネントおよび機能を個別に購入し、ライセンス供与を受け SPSS Modeler と合わせて使用できます。追加の製品や更新が利用できる可能性があることに注意してください。 詳細は、http://www.ibm.com/software/analytics/spss/products/modeler/ を参照してください。

- SPSS Modeler Server へのアクセスにより、大規模なデータセット上のスケーラビリティおよびパフォーマンスを向上させ、SQL 最適化のサポート、およびインデータベース モデリング能力を提供します。
- SPSS Modeler Solution Publisher は、SPSS Modeler 環境の外側でのリアルタイムまたは自動スコアリングで使用します。 詳細は、 2 章 IBM SPSS Modeler Solution Publisher in IBM SPSS Modeler 14.2 Solution Publisher を参照してください。
- アダプタを使用して IBM SPSS Collaboration and Deployment Services またはシンクライアント アプリケーションの IBM SPSS Modeler Advantage に展開します。 詳細は、 9 章 IBM SPSS Collaboration and Deployment Services Repository オブジェクトの保存と展開 in IBM SPSS Modeler 14.2 ユーザー ガイド を参照してください。

IBM SPSS Text Analytics

IBM® SPSS® Text Analytics は、SPSS Modeler の完全に統合されたアドオンで、高度な言語テクノロジと自然言語処理 (NLP: Natural Language Processing) を使用して、さまざまな無構造テキスト データを高速で処理し、重要な概念を抽出および整理してカテゴリに分類します。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、IBM® 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。

- テキスト マイニング ノードは、テキスト リンクとクラスタの高度な洞察を実行できるインタラクティブ ワークベンチと同様にコンセプトおよびカテゴリ モデル作成を提供、独自のカテゴリを作成し。言語リソースのテンプレートを調整します。

- ブログやその他の Web ベースのソースなど、多くのインポート形式がサポートされています。
- CRM（Customer Relationship Management、顧客関係管理）やゲノム研究のような特定の分野用のカスタム テンプレート、ライブラリ、辞書も付属しています。

注 :このコンポーネントを利用するには、別途、ライセンスが必要です。詳細は、http://www.ibm.com/software/analytics/spss/products/modeler/ を参照してください。

IBM SPSS Modeler ドキュメント

オンライン ヘルプ形式の完全なドキュメントは、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は、各製品 DVD の ¥Documentation フォルダにもあります。

- **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 14.2 Modeling Nodes 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。
- **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］メニューからも利用できます。 詳細は、 アプリケーションの例 in IBM SPSS Modeler 14.2 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 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 手法を使用した SPSS Modeler によるデータ マイニングの段階的なガイドです。

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で、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 マイニング作業者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 14.2 ユーザー ガイド を参照してください。

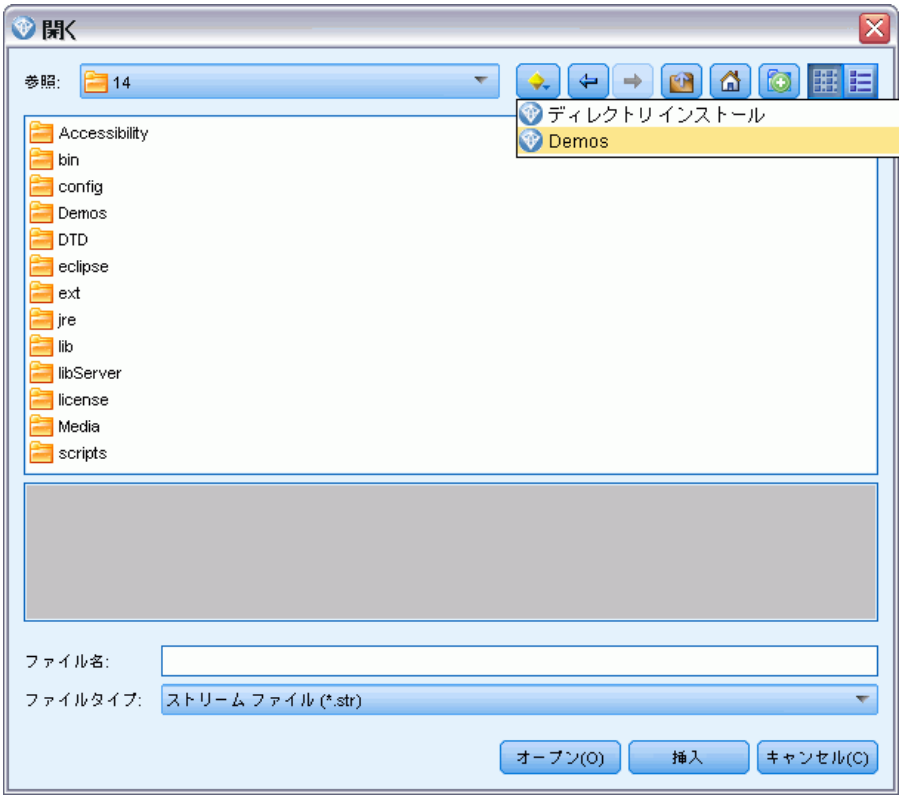
データベース モデル作成の例 : 例は、『IBM SPSS Modeler データベース内マイニング ガイド』を参照してください。

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 14.2 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

章
2

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 インターフェイ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。

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バスの内容ではなく、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

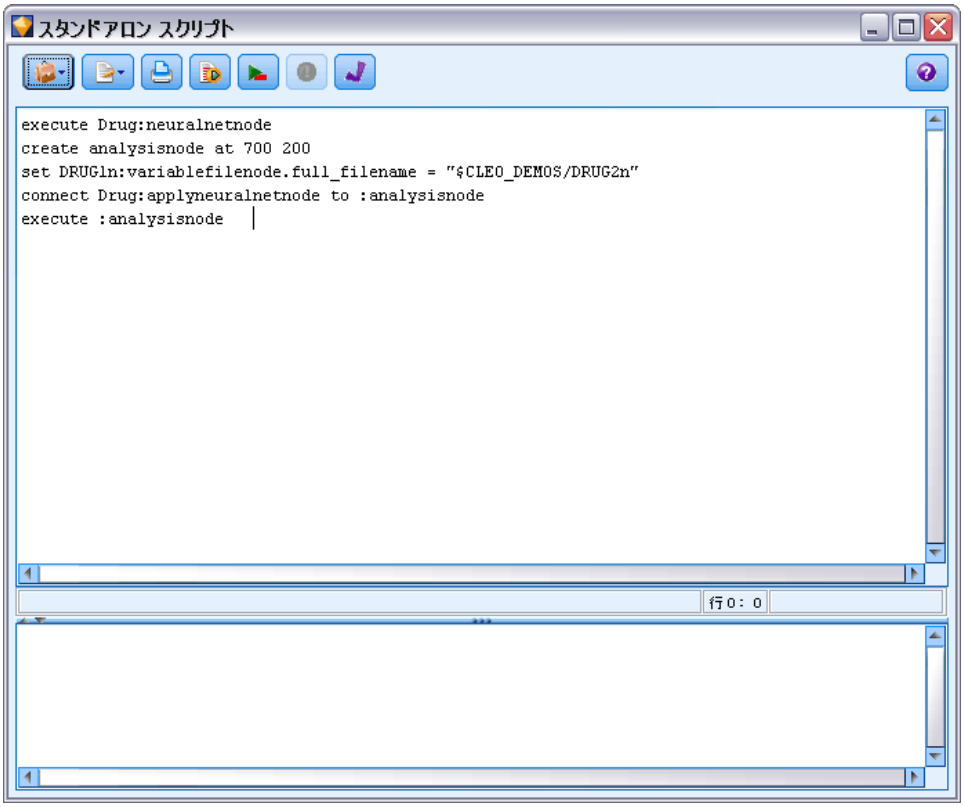
スクリプトを使って特定のストリーム内の操作をカスタマイズできます。また、スクリプトをそのストリームとともに保存することができます。ストリーム スクリプトは、ストリーム内のターミナル ノードの、特定の実行順序を指示するために使用されます。ストリーム スクリプト ダイアログ ボックスを使用して、現在のストリームとともに保存されているスクリプトを編集します。

[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、エラーが見つかれば、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。

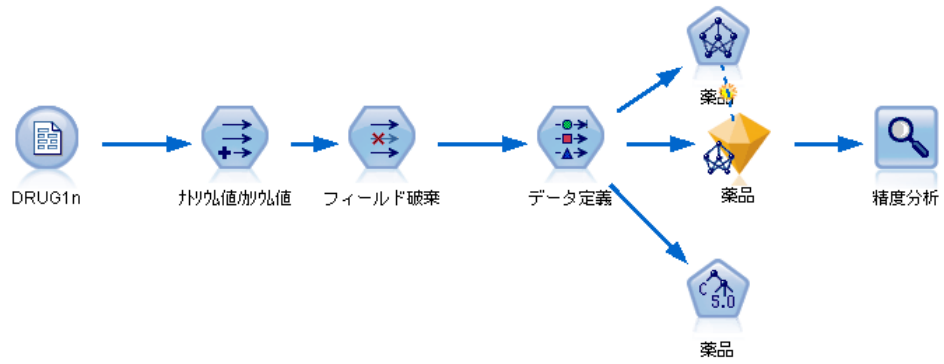
IBM® SPSS® Modeler スクリプトを使用すると、モデル ナゲット作成後のテスト プロセスを自動化できます。たとえば、デモ ストリーム druglearn.str (SPSS Modeler インストールの下の /Demos/streams/ フォルダ内) をテストする次のストリーム スクリプトは、[ストリーム プロパティ] ダイアログ (**[ツール] → [ストリームのプロパティ] → [スクリプト]**) で実行できます。

```
execute Drug:neuralnetworknode
create analysisnode at 700 200
set DRUG1n:variablefilenode.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. 20) 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. 33) を参照してください。

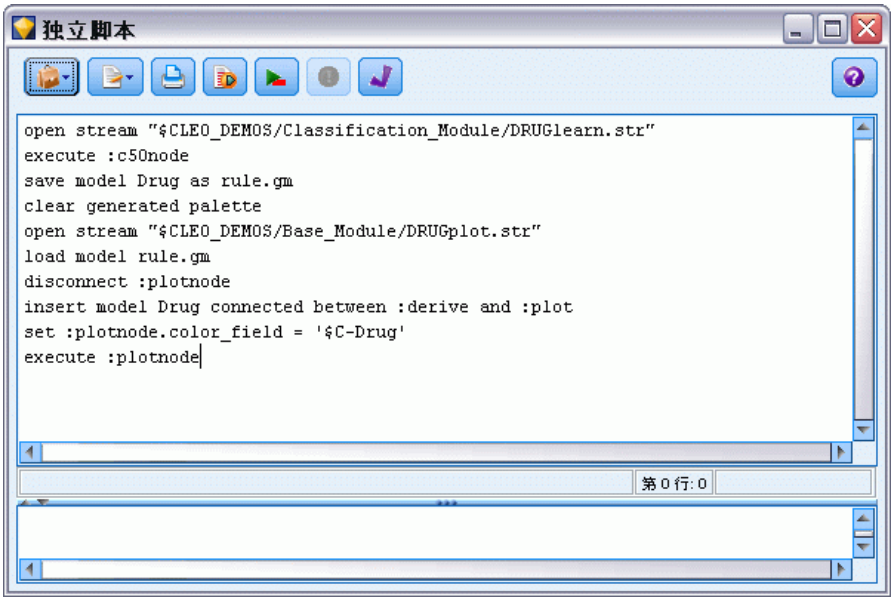
スタンドアロン スクリプト

[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



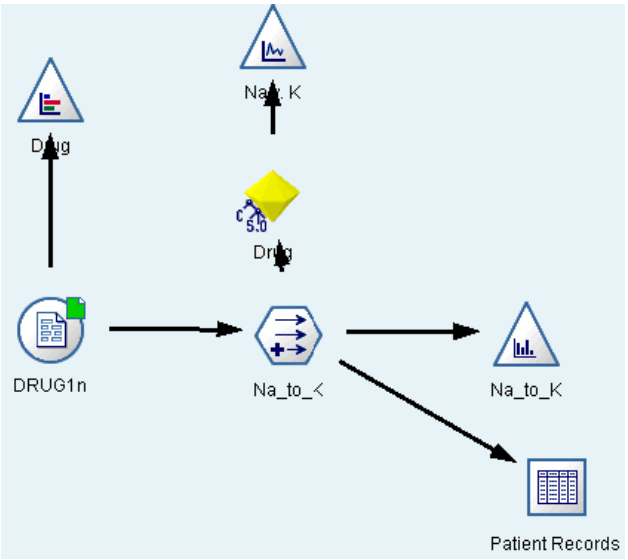
スタンドアロン スクリプトでは、ストリーム スクリプトと同じツール バーやスクリプト シンタックス検査オプションを使用することができます。 詳細は、 p. 8 ストリーム スクリプト を参照してください。

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 "$CLEO_DEMOS/Classification_Module/DRUGlearn.str"
execute :c50node
save model Drug as rule.gm
clear generated palette
open stream "$CLEO_DEMOS/Base_Module/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. 20 ）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. 33 ）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

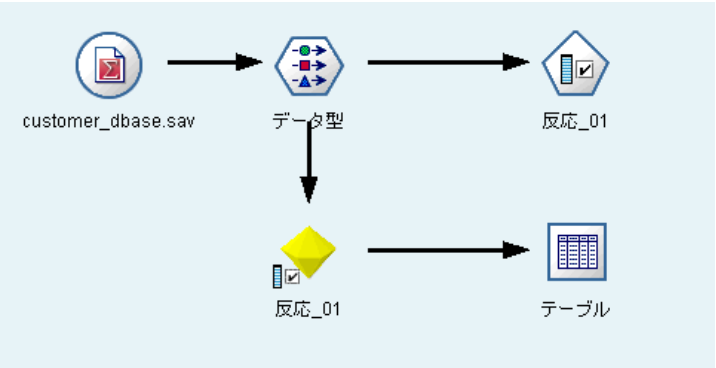
この例では、空の領域からフィールド選択モデルを生成するストリームを構築し、そのモデルに適用して、指定された対象に関連するもっとも重要な上位 15 のフィールドを表示するテーブルを作成します。

```
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. 235 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テーミナル 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. 332 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 14.2 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. 76 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

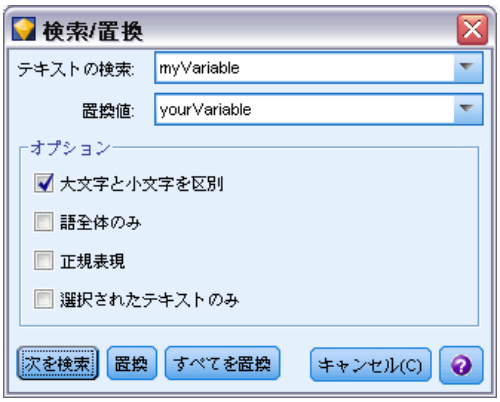
[ストリーム スクリプト] ダイアログ ボックスのツールバーにある赤い中止ボタンは、スクリプト実行時に有効になります。このボタンを使用すると、スクリプトおよび現在のストリームの実行を中止することができます。

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

文字クラスの一致

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、[ad-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

章

3

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、先頭にカレット（^）記号を付けます。ただし、宣言時や変数の値の設定時には、カレット（^）記号を使用しません。 詳細は、p. 21 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。
- databasenode 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もあります。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. 31 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 :derivenode.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 などの名前で指定できます。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 14.2 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（後者は接頭辞の “apply” を使用します）。 詳細は、 4 章 p. 46 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. 73 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すが、両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可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、その結果、使用法に微妙な区別が必要です。 詳細は、 4 章 p. 36 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に設定されます。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にのみ必要です。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存続します。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 14.2 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 14.2 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 14.2 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.33 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. 20 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 :derivenode.formula_expr = "substring(5, 1, Drug)"

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

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

set :derivenode.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、 p. 31 リテラル テキストのブロック 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...
"""
```

章
4

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 : リストの順序を使用して、PARAMETER に割り当てられた LIST 内の各値に対して、STATEMENTS を 1 回ずつ実行します。たとえば次のように、Filter.include プロパティが複数フィールドで true に設定されます。

```
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 I from 1 to 5
 set :selectnode.condition = 'Age > ' >< (I * 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められ、フィールドは返されないので、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
```

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

```
if :samplenode.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、get など）、またはリテラル値であってもかないません。

```
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. 27 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. 123 ）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 :samplenode {
  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 :variablefilenode.read_field_names = true
```

注 ： Off、OFF、off、No、NO、no、n、N、f、F、false、False、FALSE、または 0 を含むバリエーションも、値の設定時には認識されますが、プロパティ値の読み取り時にはエラーを起こすことがあります。その他の値はすべて真と見なされます。小文字の true と false だけを使用すると、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 :derivenode {
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、現在のスクリプトのコンテクストに応じて、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. 26 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 **y** 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.24 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.42 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. 25 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

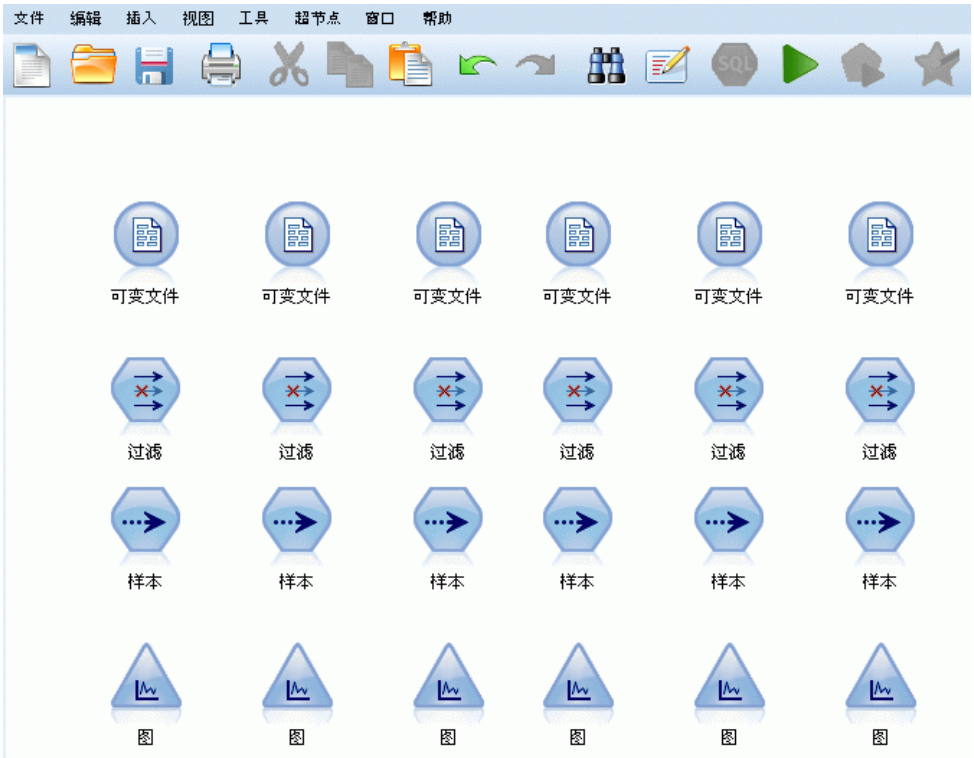
position Drug:net between DRUG2n and analysis

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.64 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.64 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.268 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。通常の状況では、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	タイム シリーズ
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。

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 14.2 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 14.2 Modeling Nodes 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。

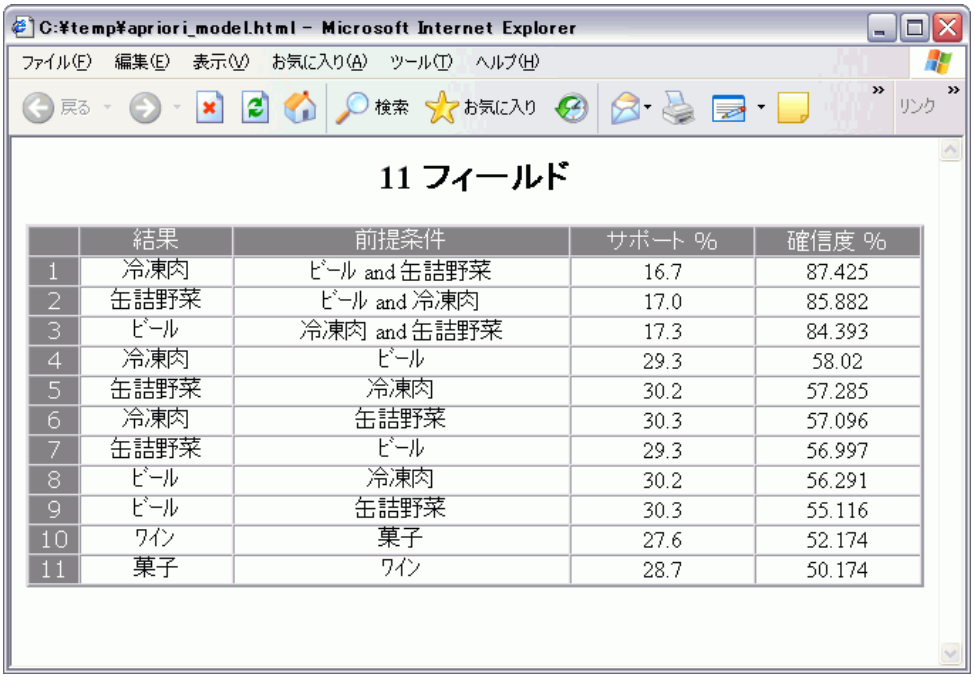
モデルの要約。 モデル ナゲットの参照時に［要約］タブ内に表示されるとおりにモデルの要約を HTML またはテキスト フォーマットでエクスポートするには以下のようにします。

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



11 フィールド

	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.64 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.64 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 “Kohenen” として格納されている場合、そのモデルは同じ名前で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.64 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. 25 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.64 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.64 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.64 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 I from 1 to 3
  write file 'Stream ' >< I
endfor
close file
```

write FILE

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

上記の式は、テキスト式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.64 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.64 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"

章
5

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 14.2 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.79 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 [#l.label]

retrieve object REPOSITORY_PATH {label LABEL | version VERSION}
retrieve object URI [(#m.marker | #l.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、同じ名前で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。

```
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

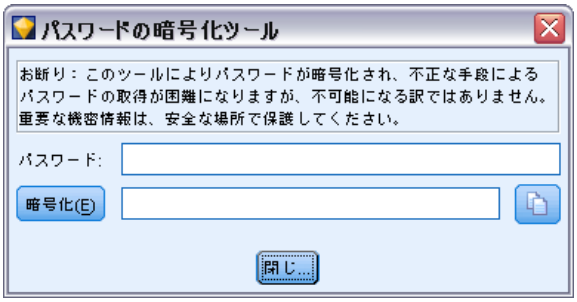
場合によっては、スクリプトにパスワードを記述する必要があるかも知れません。たとえば、パスワードで保護されたデータ ソースにアクセスしたい場合などです。暗号化パスワードは、次の場所で使用することができます。

- 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。databasenode および databaseexportnode 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。

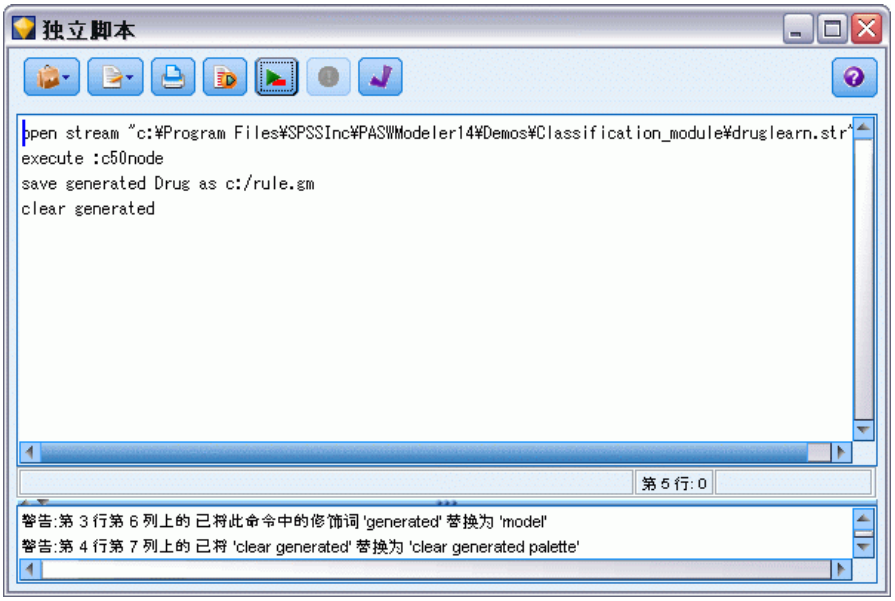
旧リリースとの互換性

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



章

6

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. 187 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. 33 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>IBM SPSS Modeler ã€® ãƒ‡ãƒ¼ã‚¿åž‹ãƒŽãƒ¼ãƒ‰ã€® æƒ…å ±</header>"
writeln myreport "<body><br/><br/>"

#create the table and write out the headers
writeln myreport "<table border=\"1\">"
writeln myreport "<tr bgcolor=\"C0C0C0\">"
writeln myreport "<td>ãƒ•ã‚£ãƒ¼ãƒ«ãƒ‰</td><td>ç¨® é¡ž</td><td>å€¤</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>IBM SPSS Modeler ã€® ãƒ‡ãƒ¼ã‚¿ãƒžãƒŽã€® æƒ…å ±</header>"
writeln myreport "<body><br/><br/>"

#create the table and write out the headers
writeln myreport "<table border=\"1\">"
writeln myreport "<tr bgcolor=\"C0C0C0\">"
writeln myreport "<td>ãƒ•ã‚£ãƒ¼ãƒ«ãƒ‰</td><td>ç¨® é¡ž</td><td>å€¤</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（
 タグ）で区切られます。

```
# 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
```

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 >< " "
 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（ ）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 >< " "
 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
```

章
7

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。

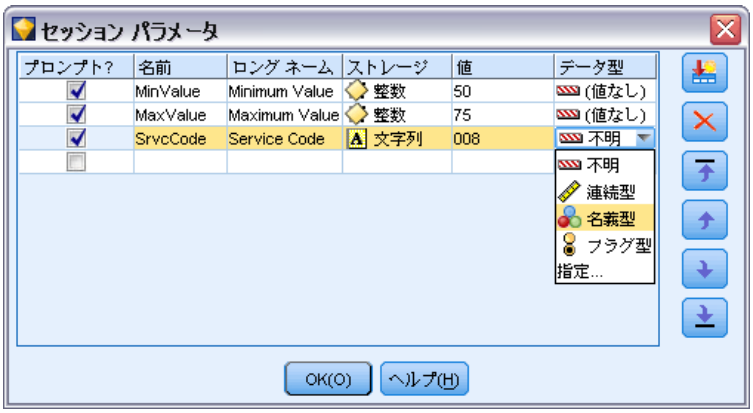
```
modelerclient -stream mystream.str -Pusername="Joe User" -execute
```

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\nn\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、フラグ -hostname、-use_ssl、-port、-username、-password、および -domain は SPSS Modeler にパブリック サーバーへの接続方法を指示します。-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.79 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
-cluster <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 hostname、port および use_ssl 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、 p.79 IBM SPSS Collaboration and Deployment Services Repository 接続の引数 を参照してください。
-username <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。
-password <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 : -password 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。
-epassword <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。
-domain <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。
-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。

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。詳細は、 p.77 複数の引数の組み合わせ を参照してください。
-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。-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.64 ）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.126 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.username=george
-P:databasenode.password=jetson
```

章

8

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 14.2 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.30 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 14.2 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 14.2 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、その後に続く小数点、その後に続く 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 14.2 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 ストリームのオプションの設定 in IBM SPSS Modeler 14.2 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、4 Q 2004 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 ��� 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 INT1 && INT2 = 0 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 INT1 && INT2 = 0 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。 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。- 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 is_string 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 to_timestamp 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、<= 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	if、then、else などの論理演算を行うために用いられます。
Numeric	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 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 stripchar 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、@MEAN 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、@BLANK(FIELD) 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、@FIELD 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 extract_concept のような任意の実数
DATE	start_date 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	power_flux 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 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 14.2 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.30 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.30 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 >< 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、"xx" >< Site は "xxBRAMLEY" を返します。>< 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、V1 >< 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 ストリームのオプションの設定 in IBM SPSS Modeler 14.2 ユーザー ガイド を参照してください。

関数	結果	説明
ITEM1 >< 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。

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になります。

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。 詳細は、 7 章 複数フィールドの要約 in IBM SPSS Modeler 14.2 ユーザー ガイド を参照してください。
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。

関数	結果	説明
date_before(DATE1, DATE2)	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。
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 14.2 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 14.2 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。

関数	結果	説明
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 14.2 ユーザー ガイド を参照してください。
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 14.2 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 14.2 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 14.2 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、length(Label) >=6 や Label(6) = 'x' 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。
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 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。

関数	結果	説明
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(POWER * log(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。
intof(NUM)	Integer	引数を切り捨てて整数にします。NUM と同じ符号で、abs(INT) <= 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(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 intof(NUM+0.5) または NUM が負の数の場合は intof(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 14.2 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ク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ク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。
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 < ANGLE <= pi (radians) – 180 < ANGLE <= 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.30 スクリプト内の CLEM 式 を参照してください。

関数	結果	説明
~~ INT1	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。~~ INT = –(INT + 1) 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.30 スクリプト内の CLEM 式 を参照してください。
INT1 || INT2	整数	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。
INT1 ||/& INT2	整数	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。
INT1 && INT2	整数	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。
INT1 &&~~ INT2	整数	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 INT1&& (~~INT2) 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。
INT << N	整数	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。
INT >> N	整数	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。
INT1 &&=_0 INT2	ブール	ブール式 INT1 && INT2 /== 0 と同じですが、より効率的です。
INT1 &&/=_0 INT2	ブール	ブール式 INT1 && INT2 == 0 と同じですが、より効率的です。

関数	結果	説明
integer_bitcount(INT)	整数	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。integer_bitcount(INT) = integer_bitcount(-(INT+1)) は常に true となります。
integer_leastbit(INT)	整数	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。
integer_length(INT)	整数	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 << N) if INT >= 0 INT >= (–1 << 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 複数フィールドの要約 in IBM SPSS Modeler 14.2 ユーザー ガイド 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、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
allbutfirst(N, STRING)	String	STRING の先頭 N 文字,ð削除した文字列を返します。
allbutlast(N, STRING)	String	STRING の最後の文字を削除した文字列を返します。
alphabefore(STRING1, STRING2)	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。
endstring(LENGTH, STRING)	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か、またはそれに等しい場合は、変更されません。
hasendstring(STRING, SUBSTRING)	整数	この関数は、isendstring(SUBSTRING, STRING). と同じです。
hasmidstring(STRING, SUBSTRING)	整数	この関数は、ismidstring(SUBSTRING, STRING)（埋め込みサブ文字列）と同じです。
hasstartstring(STRING, SUBSTRING)	整数	この関数は、isstartstring(SUBSTRING, STRING) と同じです。
hassubstring(STRING, N, SUBSTRING)	整数	この関数は、issubstring(SUBSTRING, N, STRING) と同じです。N のデフォルトは 1 です。
count_substring(STRING, SUBSTRING)	整数	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。count_substring("foooo.txt", "oo") は 3 を返します。
hassubstring(STRING, SUBSTRING)	整数	この関数は、issubstring(SUBSTRING, 1, STRING) と同じです。N のデフォルトは 1 です。

関数	結果	説明
isalphacode(CHAR)	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、isalphacode(produce_num(1)) のようになります。
isendstring(SUBSTRING, STRING)	整数	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。
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)	整数	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)	整数	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)	整数	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。

関数	結果	説明
issubstring(SUBSTRING, STRING)	整数	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):	整数	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。
issubstring_lim(SUBSTRING, N, STARTLIM, ENDLIM, STRING)	整数	この関数は issubstring と同じですが、添字 STARTLIM から、またはその前から始まり、添字 ENDLIM 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。STARTLIM 制約または ENDLIM 制約は、どちらかの引数に偽（false）の値を指定することによって無効にできます。たとえば、issubstring_lim(SUBSTRING, N, false, false, STRING) は issubstring と同じです。
isuppercode(CHAR)	ブール	この関数は CHAR が大文字の場合に、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、isuppercode(``) や isuppercode(country_name(2)) などが有効な式になります。
last(CHAR)	String	この関数は、STRING の最後の文字 CHAR 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)。
length(STRING)	整数	文字列 STRING の長さ（つまり文字列内の半角文字数）を返します。
locchar(CHAR, N, STRING)	整数	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 STRING 中の N 番目の文字から、文字 CHAR の検索を開始します。この関数は、文字が見つかった(N 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 (N) 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 $null$を返します。 たとえば、locchar(`n`, 2, web_page) と指定すると、フィールド web_page 中の 2番目の文字から、文字 `n` を検索します。 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。

関数	結果	説明
locchar_back(CHAR, N, STRING)	整数	locchar に似ていますが、N 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、locchar_back(`n`, 9, web_page) と指定すると、フィールド web_page 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 $null$ を返します。できる限り、locchar_back とともに関数 length(<field>) 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。たとえば、locchar_back(`n`, (length(web_page)), web_page) となります。
lowertoupper(CHAR) lowertoupper (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、lowertoupper(`a`)、lowertoupper("My string")、および lowertoupper(field_name(2)) となります。
matches	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。
replace(SUBSTRING, NEWSUBSTRING, STRING)	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。
replicate(COUNT, STRING)	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。
stripchar(CHAR,STRING)	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 stripchar(`$`, 'Cost') 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。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。

関数	結果	説明
skipchar(CHAR, N, STRING)	整数	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 $null$ を返します。 locchar は、よく関数 skipchar 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、skipchar(`s`, (locchar(`s`, 1, "MyString")), "MyString") となります。
skipchar_back(CHAR, N, STRING)	整数	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)	整数	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。

関数	結果	説明
unicode_value(CHAR)	NUM	CHAR の Unicode 値を返します。
uppertolower(CHAR) 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。 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ても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)	整数	指定した STRING の 4 文字の SoundEx コードを返します。
soundex_difference(STRING1, STRING2)	整数	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. 30 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. 30 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)	整数	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)	整数	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)	整数	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)	整数	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を返します。
datetime_in_seconds(DATE), 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)	整数	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)	整数	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)	整数	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)	整数	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)	整数	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プ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	時間またはタイプスタンプ 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 time_in_hours('0130') 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)	整数	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	時間またはタイプスタンプ 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)	整数	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 ストリームのオプションの設定 in IBM SPSS Modeler 14.2 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、

15 章 時系列ノードによる予測 in IBM SPSS Modeler 14.2 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 @SINCE0 を使用します。現在のレコードで条件が真（true）の場合は、@SINCE0 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.30 スクリプト内の CLEM 式 を参照してください。

関数	結果	説明
MEAN(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。
@MEAN(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

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.30 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	整数	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、$null$ 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。
@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

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。 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が、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。 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。さもなければ、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みます–、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.30 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.30 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.30 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.30 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.30 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.30 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。 @PARTITION_FIELD = @TRAINING_PARTITION 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,ð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 14.2 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。 詳細は、 7 章 複数フィールドの要約 in IBM SPSS Modeler 14.2 ユーザー ガイド を参照してください。
@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。 詳細は、 7 章 複数レスポンス データの処理 in IBM SPSS Modeler 14.2 ユーザー ガイド を参照してください。

パート II:
プロパティ参照

章

9

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 14.2 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.36 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に、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。

s.max_size	ノード s のプロパティ max_size を表します。
s:samplenode.max_size	ノード s のプロパティ max_size 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。
:samplenode.max_size	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 max_size 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）。
s:sample.max_size	ノード s のプロパティ max_size 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。
t.direction.Age	データ型ノードt の Age フィールドの役割を表します。
:.max_size	*** 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もあります。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 process_supernode terminal_supernode 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 real_income のような名前だけで参照する代わりに、userinputnode または filternode 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.332 スーパーノードのプロパティ を参照してください。

章
10

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	データの型	プロパティの説明
execute_method	標準 Script	
date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY	
date_baseline	数値型	
date_2digit_baseline	数値型	
time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S"	
time_rollover	フラグ型	
import_datetime_as_string	フラグ型	

プロパティ名	データの型	プロパティの説明
decimal_places	数値型	
decimal_symbol	Default Period Comma	
angles_in_radians	フラグ型	
use_max_set_size	フラグ型	
max_set_size	数値型	
ruleset_evaluation	Voting 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	文字列	
annotation	文字列	例 ： set ^stream.annotation = "something interesting"
name	文字列	例 ： set x = ^stream.name 注: このプロパティは読み取り専用です。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。 例 ： set ^stream.parameters.height = 23
nodes		詳細は以下を参照してください。
encoding	SystemDefault "UTF-8"	

章
11

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 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 ExcludedItems 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 RecentItems 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	select_mode が RecentItems の場合に使用。
old_item_limit	integer	select_mode が OldItems の場合に使用。
ordering	TypeNameAdded TypeAddedName NameAddedType 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。

章
12

入力ノードのプロパティ

入力ノードの共通プロパティ

すべての入力ノードに共通するプロパティを次に一覧にします。その後に、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 Target Both None Partition Split Frequency RecordID	フィールドの役割のキープロパティ。 使用フォーマット： NODE.direction.FIELDNAME 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 Flag Set Typeless Discrete 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。 使用フォーマット： NODE.type.FIELDNAME
storage	Unknown String Integer Real Time Date 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。 使用フォーマット： NODE.storage.FIELDNAME

プロパティ名	データ型	プロパティの説明
check	None Nullify Coerce Discard Warn Abort	フィールド タイプと範囲の検査用のキー プロパティ。 使用フォーマット : 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、value_mode プロパティの値が自動的に Specify に設定されます。 使用フォーマット : NODE.values.FIELDNAME
value_mode	Read Pass 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。 使用フォーマット : NODE.value_mode.FIELDNAME 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、values プロパティを設定します。
default_value_mode	Read 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。 使用フォーマット : NODE.default_value_mode 例: set mynode.default_value_mode = Pass この設定による特定のフィールドの設定は、value_mode 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	value_mode 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。 使用フォーマット : NODE.extend_values.FIELDNAME
value_labels	string	値ラベルの指定に使用します。例: set :varfilenode.value_labels.Age = [{3 three}{5 five}] 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。 使用フォーマット : 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。 使用フォーマット : NODE.missing_values.FIELDNAME

プロパティ名	データ型	プロパティの説明
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。 使用フォーマット ： 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）だけを含む値は欠損値とみなされます。 使用フォーマット ： 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。 NODE.default_include 例 : 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 : NODE.include.FIELDNAME. 例 : set mynode:filternode.include.Age = true
new_name	string	例 : set mynode:filternode.new_name.'Age' = "years"

cognosimportnode のプロパティ

IBM Cognos BI 入力ノードは、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 IBM Cognos BI 入力ノード in IBM SPSS Modeler 14.2 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 cognosimportnode
set :cognosimportnode.cognos_connection = {'http://mycogsrv1:9300/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
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。 {" Cognos_server_URL", login_mode, "namespace", "username", "password "} ここでの意味は次の通りです。 Cognos_server_URL は、ソースを含むする Cognos サーバーの URL です。 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。 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。 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。 /Public Folders/GOSALES
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 つまたは複数のオブジェクトの Cognos パスおよび名前。フィールドの形式は [namespace].[query_subject].[query_item] となります。

databasenode 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 14.2 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 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。詳細は、 5 章 p.66 暗号化パスワードの生成 を参照してください。 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 Left Right 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 Always 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）二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。

databasenodeプロパティ	データ型	プロパティの説明
datasource	string	書式 ： {database_name,username,password[,true | false]} 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 databasenode
set :databasenode.mode = Table
```

```
set :databasenode.query = "SELECT * FROM drug4n"
set :databasenode.datasource = {"ORA 10gR2", user1, mypsw, true}
var test_e
set test_e = :databasenode.epassword
set :databasenode.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 14.2 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 DimensionsMDD 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。 mrADODsc mrI2dDsc mrLogDsc mrQdiDrsDsc mrQvDsc mrSampleReportingMDSC mrSavDsc mrSCDsc mrScriptMDSC

datacollectionimportnodeプロパティ	データ型	プロパティの説明
		特殊な値の none 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。 mrADODsc mrI2dDsc mrLogDsc mrPunchDSC mrQdiDrsDsc mrQvDsc mrRdbDsc2 mrSavDsc mrScDSC mrXmlDsc 特殊な値の none は、CDSC がないことを示します。
casedata_source_type	Unknown File Folder UDL DSN	CDSC のソース タイプを示します。
casedata_file	string	casedata_source_type が File 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	casedata_source_type が Folder 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	casedata_source_type が UDL 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。
casedata_dsn_string	string	casedata_source_type 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 Latest Specify	各バージョンの取り扱い方法を定義します。
specific_version	string	version_import_mode 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。

datacollectionimportnodeプロパティ	データ型	プロパティの説明
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	use_language 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	use_context 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	use_label_type 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 None 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	
import_sourcefile_variables	フラグ型	
import_multi_response	MultipleFlags 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 14.2 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 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 named_range 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 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。
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 ExplicitRange	範囲の決定方法を指定します。
blank_rows	StopReading ReturnBlankRows	data_range_mode が FirstNonBlank のときに、空白行の処理方法を指定します。
explicit_range_start	string	data_range_mode が ExplicitRange のときに、読み込む範囲の開始点を指定します。

excelimportnodeプロパティ	データ型	プロパティの説明
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 14.2 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。 使用フォーマット： evimportnode.connection = [description,app_view_path, app_view_version_label,environment,DPD_path]
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 14.2 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' 1 3}{'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 Comma Period	データ ソースで使われている小数点記号。例: 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: set :fixedfilenode.lines_to_scan = 50.
fields	リスト	構造化プロパティ。 使用フォーマット : fixedfilenode.fields = [{field start length} {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 Left Right 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 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 : set :varfilenode.use_custom_values.Age = true

fixedfilenodeプロパティ	データ型	プロパティの説明
custom_storage	Unknown String Integer Real Time Date Timestamp	次のフォーマットのキー スロット： set :varfilenode.custom_storage.'Age' = "Real"
custom_date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY 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。 例： set:varfilenode.custom 次のフォーマットのキー スロット： set :varfilenode.custom_date_format.'LaunchDate' = "DDMMYY"
custom_time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(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。 次のフォーマットのキー スロット： set :varfilenode.custom_time_format.'Initialize' = "HHMM"

fixedfilenodeプロパティ	データ型	プロパティの説明
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。次のフォーマットのキー スロット：set :varfilenode.custom_decimal_symbol.'Revenue' = "Comma"
encoding	StreamDefault SystemDefault "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 14.2 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 UNIX Transport SAS7 SAS8 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 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 14.2 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.328) 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 14.2 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： set :userinputnode.data.Age = "1 2 3 4" 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: 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。 例: ['Field1' 'Field2']

userinputnodeプロパティ	データ型	プロパティの説明
custom_storage	Unknown String Integer Real Time Date 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。 例: set :userinputnode.custom_storage.'Age' = "Real"
data_mode	Combined 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 14.2 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。 使用フォーマット： variablefilenode:skip_header = 3
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。 使用フォーマット： variablefilenode:num_fields_auto
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、delimit_other を true に設定し、other 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。
delimit_other	フラグ型	other 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	delimit_other 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 Comma 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	複数の隣接するブランク区切り文字を 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 Left Right 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 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)。

variablefilenodeプロパティ	データ型	プロパティの説明
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 PairAndDiscard 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 PairAndDiscard 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： set :varfilenode.use_custom_values.Age = true
custom_storage	Unknown String Integer Real Time Date Timestamp	次のフォーマットのキー スロット：set :varfilenode.custom_storage.'Age' = "Real"
custom_date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY 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。 例： set:varfilenode.custom 次のフォーマットのキー スロット： set :varfilenode.custom_date_format. 'LaunchDate' = "DDMMYY"

variablefilenodeプロパティ	データ型	プロパティの説明
custom_time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(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。 次のフォーマットのキー スロット： set :varfilenode.custom_time_format. '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。 次のフォーマットのキー スロット： set :varfilenode.custom_decimal_ symbol.'Revenue' = "Comma"
encoding	StreamDefault SystemDefault "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 14.2 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 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。

xmlimportnodeプロパティ	データ型	プロパティの説明
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 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。

章

13

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 14.2 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 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 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 14.2 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、Sex や Region がキー フィールドの場合、M と F および地域 N と地域 S の一意の組み合せ（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: set :aggregatenode. aggregates.Age = [Sum Mean Min Max SDev]。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 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 14.2 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 14.2 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 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 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 14.2 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
```

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 Keys	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 (Keys) を指定します。
key_fields	[フィールド フィールド フィールド]	
common_keys	フラグ型	
join	Inner FullOuter PartialOuter Anti	一例を以下に挙げます。 set :merge.join = FullOuter
outer_join_tag.n	フラグ型	このプロ◆◆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ので、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 、outer_join_tag 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} ¥ {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 14.2 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 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 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。
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	

rfmaggregatenodeプロパティ	データ型	プロパティの説明
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 Weeks Months 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 14.2 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 Complex	
mode	Include 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 OneInN 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。 set :samplenode.sample_type = First set :samplenode.first_n = 100
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	maximum_size 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、First と Include 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。
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 Systematic	
sample_units	Proportions Counts	
sample_size_proportions	Fixed Custom Variable	

samplenodeプロパティ	データ型	プロパティの説明
sample_size_counts	Fixed Custom 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、Sample units=Proportions 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、Sample units=Proportions 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 ... 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 value}{stringstring value}…]	sample_units=proportions および sample_size_proportions=Custom 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 value}{stringstring 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 sizes_proportions 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 14.2 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 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 14.2 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} ¥ {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。

章

14

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 14.2 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 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。

autodataprepnode 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 14.2 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 autodataprepnode
set:autodataprepnode.objective = Balanced
set:autodataprepnode.excluded_fields = Filter
set:autodataprepnode.prepare_dates_and_times = true
set:autodataprepnode.compute_time_until_date = true
set:autodataprepnode.reference_date = Today
set:autodataprepnode.units_for_date_durations = Automatic
```

autodataprepnodeプロパティ	データ型	プロパティの説明
objective	Balanced Speed Accuracy 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 ... 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	

autodataprepnode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 None	
if_fields_do_not_match	StopExecution 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 Fixed	
fixed_date	日付	
units_for_date_durations	Automatic Fixed	
fixed_date_units	Years Months Days	
compute_time_until_time	フラグ型	
reference_time	CurrentTime Fixed	
fixed_time	時間	
units_for_time_durations	Automatic Fixed	
fixed_date_units	Hours Minutes 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	

autodataprepnode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 Delete	
rescale_continuous_inputs	フラグ型	
rescaling_method	MinMax 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	

autodataprepnode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 14.2 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 フィールド2 ... 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 EqualCount Rank SDev 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 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 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 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 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。

binningnodeプロパティ	データ型	プロパティの説明
custom_tile_add_as	Suffix Prefix	
custom_tile	integer	
equal_count_method	RecordCount 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 Current 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 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 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 Prefix	
sdev_count	One Two Three	
optimal_name_extension	string	デフォルトの拡張子は _OPTIMAL です。
optimal_add_as	Suffix 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 Exclusive	
optimal_first_bin	Unbounded Bounded	
optimal_last_bin	Unbounded 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 14.2 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 derivenode
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 derivenode
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"
```

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 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 フィールド 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 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 Flag Set State Count 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。 使用フォーマット : set :derivenode. set_value_cond. 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 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て 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 14.2 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 ConfidenceWeighted-Voting RawPropensity-WeightedVoting AdjustedPropensity-WeightedVoting HighestConfidence AverageRawPropen-sity 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にのみ適用されます。
set_ensemble_method	Voting ConfidenceWeighted-Voting 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にのみ適用されます。

ensemblenodeプロパティ	データ型	プロパティの説明
flag_voting_tie_selection	Random HighestConfidence RawPropensity 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 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 14.2 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 フィールド 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 Conditional Blank Null 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 14.2 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 inBP 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、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。 NODE.include.FIELDNAME 一例を以下に挙げます。 set mynode:filternode.default_include = false 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。 使用フォーマット ： NODE.include.FIELDNAME 一例を以下に挙げます。 set mynode: filternode.include.Age = false
new_name	string	一例を以下に挙げます。 set mynode: filternode.new_name. Age = "age"

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 14.2 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 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 Leave 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 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 14.2 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)。

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 SystemAndLabel 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 14.2 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 Multiple	1 つのフィールドのカテゴリを再分類する場合、Single を選択します。Multiple（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 フィールド2 ... 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 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。使用フォーマット ： NODE.reclassify. OLD_VALUE 一例を以下に挙げます。 set :reclassifynode.reclassify.'drugB' = 'Yes'
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 … 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。 一例を以下に挙げます。 set :reclassify.pick_list = [fruit dairy cereals]

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 14.2 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 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 Type Storage	
ascending	フラグ型	
start_fields	[フィールド1 フィールド2 … 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 フィールド2 … 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、名義型またはグラグ型フィールドを、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 14.2 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 カテゴリ] all	例をあげると、次のようになります。 set :restructurenode.fields_from.Drug = [drugA drugB] は Drug_drugA および Drug_drugB というフィールドを作成します。 特定フィールドのすべてのカテゴリを使用するには set :restructurenode.fields_from.Drug = all
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 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。OtherFields 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）。Flags を指定する場合、値は数値のフラグです。
value_fields	[フィールド フィールド フィールド]	value_mode が OtherFields 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 14.2 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
```

```
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 Current	可否同数の値のデータに配置されるビンを指定。
recalculate_bins	Always 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 Frequency 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。

rfmanalysisnode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 14.2 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 カテゴリ] all	例をあげると、次のようになります。 set :settoflagnode.fields_from.Drug = [drugA drugB] 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。 特定フィールドのすべてのカテゴリを使用するには set :settoflagnode.fields_from.Drug = all
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 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 フィールド 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 14.2 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.328 ）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 14.2 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 Periods CyclicPeriods Years Quarters Months DaysPerWeek DaysNonPeriodic HoursPerDay HoursNonPeriodic MinutesPerDay MinutesNonPeriodic SecondsPerDay SecondsNonPeriodic	
mode	Label 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 February March April May June July August September October November December	
day_start	integer	
hour_start	integer	
minute_start	integer	
second_start	integer	

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 February March April May June July August September October November 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 Monday Tuesday Wednesday Thursday Friday Saturday 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 2 3 4 5 6 10 15 20 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" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY	

timeintervalsnodeプロパティ	データ型	プロパティの説明
time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S"	
aggregate	Mean Sum Mode Min Max First Last TrueIfAnyTrue	フィールドに対して集計方法を指定します（たとえば、aggregate.AGE=Mean）.
pad	Blank MeanOfRecentPoints True False	フィールドを埋める方法を指定します（たとえば、pad.AGE=MeanOfRecentPoints）.
agg_mode	All 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 Sum Mode Min 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 First 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 Mode First Last	
pad_range_default	Blank 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 MostRecentValue	

timeintervalsnodeプロパティ	データ型	プロパティの説明
pad_flag_default	Blank True 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 14.2 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 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ことが必要です。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 String 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 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 14.2 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 Target Both None Partition Split Frequency RecordID	フィールドの役割のキープロパティ。 使用フォーマット： NODE.direction.FIELDNAME 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 Flag Set Typeless Discrete 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。 使用フォーマット： NODE.type.FIELDNAME
storage	Unknown String Integer Real Time Date 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。 使用フォーマット： NODE.storage.FIELDNAME
check	None Nullify Coerce Discard Warn Abort	フィールド タイプと範囲の検査用のキー プロパティ。 使用フォーマット： NODE.check.FIELDNAME

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、value_mode プロパティの値が自動的に Specify に設定されます。 使用フォーマット： NODE.values.FIELDNAME
value_mode	Read Pass Specify	値の設定方法を決定します。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、values プロパティを設定します。 使用フォーマット： NODE.value_mode.FIELDNAME
extend_values	フラグ型	value_mode に Read 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、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。 使用フォーマット： NODE.extend_values.FIELDNAME
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。 使用フォーマット： 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。 使用フォーマット： 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	range_missing が真（true）の場合、欠損値範囲の下限値を指定します。
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。
null_missing	フラグ型	T 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。 使用フォーマット： 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。 使用フォーマット： NODE.whitespace _missing. FIELDNAME
description	string	フィールドの説明を指定します。
value_labels	[{Value LabelString} { Value LabelString} ...]	値のペアのためのラベルを指定します。 一例を以下に挙げます。 set :typenode.value_labels.'Drug'=[{drugA label1} {drugB label2}]
display_places	integer	フィールドが表示されるとき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。 使用フォーマット： NODE.display_places.FIELDNAME
export_places	integer	フィールドが表示されるとき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。 使用フォーマット： NODE.export_places.FIELDNAME
decimal_separator	DEFAULT PERIOD 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）。 使用フォーマット： NODE.decimal_separator.FIELDNAME
date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "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）。 使用フォーマット： NODE.date_format.FIELDNAME 一例を以下に挙げます。 set :tablenode.date_format.'LaunchDate' = "DDMMYY"

typenodeプロパティ	データ型	プロパティの説明
	"DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY	
time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(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）。 使用フォーマット： NODE.time_format.FIELDNAME 一例を以下に挙げます。 set :tablenode.time_format.'BOF_enter' = "HHMMSS"
number_format	DEFAULT STANDARD SCIENTIFIC CURRENCY	フィールドに数値の表示フォーマットを設定します。 使用フォーマット： 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の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。 使用フォーマット： 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。 使用フォーマット： 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の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。 使用フォーマット： 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 NONE LOCALE PERIOD COMMA SPACE	フィールドにグループ化シンボルを設定します。 使用フォーマット： NODE.grouping_symbol.FIELDNAME
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 Auto に設定されます。 使用フォーマット： NODE.column_width.FIELDNAME
justify	AUTO CENTER LEFT RIGHT	フィールドに列調整を設定します。 使用フォーマット： NODE.justify.FIELDNAME

章

15

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	文字列	タイトルを指定します。例:”This is a title.”
caption	文字列	解説を指定します。例:”This is a caption.”
output_mode	Screen 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 JPEG PNG HTML 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	文字列	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	数値型	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	数値型	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 14.2 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 のプロパティ	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	文字列	
collect_field	フィールド	
collect_label_auto	フラグ型	
collect_label	文字列	
three_D	フラグ型	
by_field	フィールド	
by_label_auto	フラグ型	
by_label	文字列	
operation	Sum Mean Min Max SDev	
color_field	文字列	
panel_field	文字列	
animation_field	文字列	
range_mode	Automatic UserDefined	
range_min	数値型	
range_max	数値型	
ビン	ByNumber ByWidth	
num_bins	数値型	
bin_width	数値型	
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 14.2 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 のプロパティ	データ型	プロパティの説明
plot	SelectedFields 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 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 14.2 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'
```

```
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 のプロパティ	データ型	プロパティの説明
chart_type	Gains Response Lift Profit ROI	
inc_baseline	フラグ型	
field_detection_method	Metadata 名前	
use_fixed_cost	フラグ型	
cost_value	数値型	
cost_field	文字列	
use_fixed_revenue	フラグ型	
revenue_value	数値型	
revenue_field	文字列	
use_fixed_weight	フラグ型	
weight_value	数値型	
weight_field	フィールド	
n_tile	Quartiles Quintles Deciles Vingtiles Percentiles 1000-tiles	
cumulative	フラグ型	
style	Line Point	
point_type	Rectangle Dot Triangle Hexagon Plus Pentagon Star BowTie HorizontalDash VerticalDash IronCross Factory House Cathedral OnionDome	

evaluationnode のプロパティ	データ型	プロパティの説明
	ConcaveTriangle OblateGlobe CatEye FourSidedPillow RoundRectangle Fan	
export_data	フラグ型	
data_filename	文字列	
delimiter	文字列	
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	文字列	
plot_score_fields	フラグ型	
score_fields	[フィールド1 ... フィールドN]	
target_field	フィールド	
use_hit_condition	フラグ型	
hit_condition	文字列	
use_score_expression	フラグ型	
score_expression	文字列	
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 14.2 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 のプロパティ	データ型	プロパティの説明
graph_type	Histogram Dotplot Scatterplot BinnedScatter HexBinScatter Line Path Area 3DHistogram 3DDensity Bubble PieCounts Pie 3DPie BarCounts Bar Surface Ribbon 3DArea Boxplot Heatmap SPLOM Parallel LinkAnalysis	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。
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	
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	

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 14.2 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 のプロパティ	データ型	プロパティの説明
フィールド	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 UserDefined	
range_min	数値型	
range_max	数値型	
ビン	ByNumber ByWidth	
num_bins	数値型	
bin_width	数値型	
normalize	フラグ型	
separate_bands	フラグ型	
x_label_auto	フラグ型	
x_label	文字列	

histogramnode のプロパティ	データ型	プロパティの説明
y_label_auto	フラグ型	
y_label	文字列	
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 14.2 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 のプロパティ	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 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	

multiplotnode のプロパティ	データ型	プロパティの説明
use_overlay_expr	フラグ型	
overlay_expression	文字列	
records_limit	数値型	
if_over_limit	PlotBins PlotSample PlotAll	
x_label_auto	フラグ型	
x_label	文字列	
y_label_auto	フラグ型	
y_label	文字列	
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。

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 14.2 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 のプロパティ	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。
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 Smoother 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	文字列	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 Line	
point_type	Rectangle Dot Triangle Hexagon Plus Pentagon Star BowTie HorizontalDash VerticalDash IronCross Factory House Cathedral	

plotnode のプロパティ	データ型	プロパティの説明
	OnionDome ConcaveTriangle OblateGlobe CatEye FourSidedPillow RoundRectangle Fan	
x_mode	Sort Overlay AsRead	
x_range_mode	Automatic UserDefined	
x_range_min	数値型	
x_range_max	数値型	
y_range_mode	Automatic UserDefined	
y_range_min	数値型	
y_range_max	数値型	
z_range_mode	Automatic UserDefined	
z_range_min	数値型	
z_range_max	数値型	
ジッター	フラグ型	
records_limit	数値型	
if_over_limit	PlotBins PlotSample PlotAll	
x_label_auto	フラグ型	
x_label	文字列	
y_label_auto	フラグ型	
y_label	文字列	
z_label_auto	フラグ型	
z_label	文字列	
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 14.2 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 のプロパティ	データ型	プロパティの説明
plot_series	Series Models	
use_custom_x_field	フラグ型	
x_field	フィールド	
y_fields	[フィールド 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	

timeplotnode のプロパティ	データ型	プロパティの説明
point_type	Rectangle Dot Triangle Hexagon Plus Pentagon Star BowTie HorizontalDash VerticalDash IronCross Factory House Cathedral OnionDome ConcaveTriangle OblateGlobe CatEye FourSidedPillow RoundRectangle 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	整数	
symbol_size	数値型	マーカー サイズを指定します。 例をあげると、次のようになります。 set :webnode.symbol_size = 5.5 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 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 14.2 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 のプロパティ	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 フィールド フィールド]	
to_field	フィールド	
from_fields	[フィールド フィールド フィールド]	
true_flags_only	フラグ型	
line_values	Absolute OverallPct PctLarger PctSmaller	
strong_links_heavier	フラグ型	
num_links	ShowMaximum ShowLinksAbove ShowAll	
max_num_links	数値型	
links_above	数値型	
discard_links_min	フラグ型	
links_min_records	数値型	
discard_links_max	フラグ型	
links_max_records	数値型	
weak_below	数値型	
strong_above	数値型	
link_size_continuous	フラグ型	

webnode のプロパティ	データ型	プロパティの説明
web_display	Circular Network Directed 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	数値型	マーカー サイズを指定します。 例をあげると、次のようになります。 set :webnode.symbol_size = 5.5 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。

章
16
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 or targets	フィールド or [フィールド 1 ... 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 ... 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 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っ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 14.2 Modeling Nodes 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 Properties	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. 209 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 Simple	
anomaly_method	IndexLevel PerRecords 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	

anomalydetectionnode Properties	値	プロパティの説明
adjustment_coeff	number	距離の計算時,É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、メモリーの一·” 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 14.2 Modeling Nodes 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
```

apriorinodeProperties	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、 p. 209 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。
antecedents	[フィールド 1 ... フィールド N]	
min_supp	number	
min_conf	number	
max_antecedents	number	
true_flags	フラグ型	
optimize	Speed Memory	
use_transactional_data	フラグ型	
contiguous	フラグ型	
id_field	string	
content_field	string	
mode	Simple Expert	
evaluation	RuleConfidence DifferenceToPrior ConfidenceRatio InformationDifference NormalizedChiSquare	

apriorinodeProperties	値	プロパティの説明
lower_bound	number	
optimize	Speed 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 14.2 Modeling Nodes 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
```

autoclassifiernodeProperties	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、 p. 209 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。
ranking_measure	Accuracy Area_under_curve Profit Lift Num_variables	
ranking_dataset	Training 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。
calculate_variable_importance	フラグ型	

autoclassifiernodeProperties	値	プロパティの説明
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。 set :autoclassifiernode.chaid=true
<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、p. 215 アルゴリズム プロパティの設定 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 14.2 Modeling Nodes 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
```

autoclusternodeProperties	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 Num_clusters Size_smallest_cluster Size_largest_cluster Smallest_to_largest Importance	
ranking_dataset	Training 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 Counts	

autoclusternodeProperties	値	プロパティの説明
smallest_cluster_limit_percent-age	number	
smallest_cluster_limit_count	integer	1 以上の整数。
enable_largest_cluster_limit	フラグ型	
largest_cluster_units	Percentage 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 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。 set :autoclusternode.kohonen=true
<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、p.215 アルゴリズム プロパティの設定 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 14.2 Modeling Nodes 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

autonumericnodeProperties	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、 p. 209 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。
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 NumberOfFields	
ranking_dataset	Test 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	

autonumericnodeProperties	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。 set :autonumericnode.chaid=true
<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、p. 215 アルゴリズム プロパティの設定 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 14.2 Modeling Nodes 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 Properties	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。 詳細は、p. 209 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	
structure_type	TAN 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 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 Properties	値	プロパティの説明
mode	Expert Simple	
missing_values	フラグ型	
all_probabilities	フラグ型	
independence	Likelihood 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 ... フィールドN]	ベイズ 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。 注 : 対象フィールドは常に選択されます。
maximum_number_inputs	number	ベイズ 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 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 14.2 Modeling Nodes 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
```

```
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 Properties	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、p. 209 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。
output_type	DecisionTree 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 Expert	
favor	Accuracy 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_winnowing	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 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、データからルールの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 14.2 Modeling Nodes 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
```

carmanodeProperties	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、 p.209 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。
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	

carmanodeProperties	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 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「純粋」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 14.2 Modeling Nodes 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"
```

cartnodeProperties	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。 詳細は、 p. 209 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	
objective	Standard Boosting Bagging 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 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。詳細は、 6 章 ツリー成長ディレクティブ in IBM SPSS Modeler 14.2 Modeling Nodes を参照してください。
use_max_depth	Default Custom	

cartnodeProperties	値	プロパティの説明
max_depth	integer	最大ツリー深さ（0 ～ 1000）。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。[{drugA drugB 1.5} {drugA drugC 2.1}]。{} 内の引数は実際の予測コストです。
priors	Data Equal 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。 set :cartnode. custom_priors = [{ 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 HighestProbability HighestMeanProbability	カテゴリ型対象のデフォルト結合ルール。
range_ensemble_method	Mean 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 Twoing 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	

cartnodeProperties	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 Validation	

chaidnode のプロパティ

CHAID ノードはディシジョン 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 個以上の枝葉を持てることを意味します。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 14.2 Modeling Nodes 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
```

set :chaidnode.bonferroni_adjustment = true

chaidnodeProperties	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。 詳細は、p. 209 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	
objective	Standard Boosting Bagging 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 InteractiveBuilder	
use_tree_directives	フラグ型	
tree_directives	string	
method	Chaid ExhaustiveChaid	
use_max_depth	Default Custom	
max_depth	integer	最大ツリー深さ（0 ～ 1000）。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。[{drugA drugB 1.5} {drugA drugC 2.1}]。{} 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 HighestProbability HighestMeanProbability	カテゴリ型対象のデフォルト結合ルール。
range_ensemble_method	Mean 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：

chaidnodeProperties	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 LR	カイ 2 乗統計の計算に使用する方法、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 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 14.2 Modeling Nodes 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
```

set :coxregnode.survival = True

coxregnode Properties	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。 詳細は、 p. 209 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 Stepwise BackwardsStepwise	
groups	フィールド	
model_type	MainEffects Custom	
custom_terms	[“BP*Sex” “BP*Age”]	例 : set :coxregnode. custom_terms = ["BP*Sex" "BP" "Age"]
mode	Expert Simple	
max_iterations	number	
p_converge	1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 0	
p_converge	1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 0	
l_converge	1.0E-1 1.0E-2 1.0E-3 1.0E-4 1.0E-5 0	
removal_criterion	LR Wald Conditional	
probability_entry	number	
probability_removal	number	
output_display	EachStep LastStep	
ci_enable	フラグ型	

coxregnode Properties	値	プロパティの説明
ci_value	90 95 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。 数値型フィールドの使用 : coxnode.value = [{"age" "35.8"}] カテゴリ フィールドの使用 : 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 14.2 Modeling Nodes 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
```

set :decisionlistnode.min_group_size_pct = 15

decisionlistnodeProperties	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す。 詳細は、 p. 209 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 InteractiveBuilder	
search_direction	Up 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 Expert	
bin_method	EqualWidth 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 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 14.2 Modeling Nodes 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
```

discriminantnodeProperties	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. 209 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 Stepwise	
mode	Simple Expert	
prior_probabilities	AllEqual ComputeFromSizes	
covariance_matrix	WithinGroups 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	

discriminantnodeProperties	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 UnexplainedVariance MahalanobisDistance SmallestF RaosV	
V_to_enter	number	
criteria	UseValue 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 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 14.2 Modeling Nodes 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
```

factornodeProperties	値	プロパティの説明
inputs	[フィールド 1 ... 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、p. 209 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	PC ULS GLS ML PAF Alpha Image	
mode	Simple Expert	
max_iterations	number	
complete_records	フラグ型	
matrix	Correlation Covariance	
extract_factors	ByEigenvalues ByFactors	
min_eigenvalue	number	
max_factor	number	

factornodeProperties	値	プロパティの説明
rotation	None Varimax DirectOblimin Equamax Quartimax 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？ 詳細は、 4 章 フィールド選択ノード in IBM SPSS Modeler 14.2 Modeling Nodes 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. 13 ）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 Properties	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●●●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. 209 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 Likelihood CramersV 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。

featureselectionnode Properties	値	プロパティの説明
important_above	number	0.0 ～ 1.0 の値を指定します。
unimportant_label	string	非重要ランクのラベルを指定します。
marginal_label	string	
important_label	string	
selection_mode	ImportanceLevel ImportanceValue 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、そして区間打切り生存モデルなど、統計モデルの機能性が数多く含まれています。 詳細は、 10 章 GenLin ノード in IBM SPSS Modeler 14.2 Modeling Nodes 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
```

genlinnodeProperties	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、 p. 209 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。
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 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 MainAndAllTwoWayEffects	
offset_type	Variable 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	実数であることが必要です。
base_category	Last First	
include_intercept	フラグ型	
mode	Simple Expert	
distribution	BINOMIAL GAMMA IGAUSS NEGBIN NORMAL POISSON TWEEDIE MULTINOMIAL	IGAUSS: 逆ガウス。 NEGBIN: 負の 2 項分布。
negbin_para_type	Specify 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	

genlinnodeProperties	値	プロパティの説明
link_function	IDENTITY CLOGLOG LOG LOGC LOGIT NEGBIN NLOGLOG ODDSPOWER PROBIT POWER CUMCAUCHIT CUMCLOGLOG CUMLOGIT CUMNLOGLOG CUMPROBIT	CLOGLOG: 補ログ・マイナス・ログ。 LOGC:対数の補数。 NEGBIN: 負の 2 項分布。 NLOGLOG: 負ログ・マイナス・ログ。 CUMCAUCHIT:累積コーチット。 CUMCLOGLOG: 累積補ログ マイナス ログ。 CUMLOGIT: 累積ロジット。 CUMNLOGLOG: 累積負ログ マイナス ログ。 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 Fisher 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 Deviance PearsonChiSquare 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 RobustEstimator	
max_iterations	number	デフォルト値は 100 です。0 以上の整数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 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 Relative	
hessian_convergence	フラグ型	

genlinnodeProperties	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 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 TypeIII TypeIAndTypeIII	
statistics	Wald LR	
citype	Wald 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 Kernel	
singularity_tolerance	1E-007 1E-008 1E-009 1E-010 1E-011 1E-012	
value_order	Ascending Descending 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 Validation	

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）へ、クラスタリングされます。この方法で、固定数のクラスタを定義し、クラスタ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、入力フィールドのセット内のパターンを明らかにするために、「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 14.2 Modeling Nodes 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
```

kmeansnodeProperties	値	プロパティの説明
inputs	[フィールド 1 ... 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が、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、 p.209 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。
num_clusters	number	
gen_distance	フラグ型	
cluster_label	String Number	
label_prefix	string	
mode	Simple Expert	

kmeansnodeProperties	値	プロパティの説明
stop_on	Default Custom	
max_iterations	number	
tolerance	number	
encoding_value	number	
optimize	Speed 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 14.2 Modeling Nodes 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
```

knnnodeProperties	値	プロパティの説明
analysis	PredictTarget IdentifyNeighbors	
objective	Balance Speed Accuracy 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チェック ボックス。
focal_cases_field	フィールド	
automatic_k_selection	フラグ型	

knnnodeProperties	値	プロパティの説明
fixed_k	integer	automatic_k_selectio が False の場合にのみ有効です。
minimum_k	integer	automatic_k_selectio が True の場合にのみ有効です。
maximum_k	integer	
distance_computation	Euclidean CityBlock	
weight_by_importance	フラグ型	
range_predictions	Mean Median	
perform_feature_selection	フラグ型	
forced_entry_inputs	[フィールド 1 ... 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 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 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	
calculate_adjusted_propensities	フラグ型	
adjusted_propensity_partition	Test 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 14.2 Modeling Nodes 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
```

kohonennodeProperties	値	プロパティの説明
inputs	[フィールド 1 ... 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. 209 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 Time	
time	number	
optimize	Speed 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 Expert	
width	number	
length	number	
decay_style	Linear Exponential	
phase1_neighborhood	number	
phase1_eta	number	
phase1_cycles	number	
phase2_neighborhood	number	

kohonennodeProperties	値	プロパティの説明
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 14.2 Modeling Nodes 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
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 ... 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 Bagging Boosting 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 BestSubsets None	
criteria_forward_stepwise	AICC Fstatistics AdjustedRSquare ASE	
probability_entry	数値型	
probability_removal	数値型	
use_max_effects	フラグ型	

linearnode プロパティ	値	プロパティの説明
max_effects	数値型	
use_max_steps	フラグ型	
max_steps	数値型	
criteria_best_subsets	AICC AdjustedRSquare ASE	
combining_rule_continuous	Mean 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	
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 14.2 Modeling Nodes 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
```

logregnodeProperties	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、 p. 209 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。
logistic_procedure	Binomial Multinomial	
include_constant	フラグ型	
mode	Simple Expert	
method	Enter Stepwise Forwards Backwards BackwardsStepwise	
binomial_method	Enter Forwards Backwards	

logregnodeProperties	値	プロパティの説明
model_type	MainEffects FullFactorial 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： set :logregnode. custom_terms = [{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 Simple Difference Helmert Repeated Polynomial 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。 使用フォーマット ： NODE.binomial_input_contrast.FIELD-NAME
binomial_input_category	First 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。 使用フォーマット ： NODE.binomial_input_category.FIELD-NAME
scale	None UserDefined Pearson Deviance	
scale_value	number	
all_probabilities	フラグ型	
tolerance	1.0E-5 1.0E-6 1.0E-7 1.0E-8 1.0E-9 1.0E-10	
min_terms	number	
use_max_terms	フラグ型	
max_terms	number	
entry_criterion	Score LR	
removal_criterion	LR Wald	
probability_entry	number	

logregnodeProperties	値	プロパティの説明
probability_removal	number	
binomial_probability_entry	number	
binomial_probability_removal	number	
requirements	HierarchyDiscrete HierarchyAll Containment None	
max_iterations	number	
max_steps	number	
p_converge	1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 0	
l_converge	1.0E-1 1.0E-2 1.0E-3 1.0E-4 1.0E-5 0	
delta	number	
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 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	

logregnodeProperties	値	プロパティの説明
binomial_residual	outliers all	
binomial_residual_enable	フラグ型	
binomial_outlier_threshold	number	
binomial_classification_cutoff	number	
binomial_removal_criterion	LR Wald 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
```

neuralnetnodeProperties	値	プロパティの説明
targets	[フィールド 1 ... 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、 p. 209 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	Quick Dynamic Multiple Prune ExhaustivePrune RBFN	
prevent_overtrain	フラグ型	
train_pct	number	
set_random_seed	フラグ型	
random_seed	number	
mode	Simple Expert	
stop_on	Default Accuracy Cycles Time	停止モード。
accuracy	number	停止精度。
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 Two Three	
hl_units_one	number	
hl_units_two	number	

neuralnetnodeProperties	値	プロパティの説明
hl_units_three	number	
persistence	number	
m_topologies	string	
m_non_pyramids	フラグ型	
m_persistence	number	
p_hid_layers	One Two 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	
optimize	Speed 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 sensitivity_analysis 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、calculate_variable_importance 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 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 - [アンサンブル] 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 ... 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 ... 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 ... 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_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します。
continue	フラグ型	既存モデルの学習を継続 ：
objective	Standard Bagging Boosting 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 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	

neuralnetworknode プロパティ	値	プロパティの説明
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 HighestProbability HighestMeanProbability	
combining_rule_continuous	Mean 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 missingValueImputation	
use_custom_model_name	フラグ型	
custom_model_name	文字列	
confidence	onProbability 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	
keywords	文字列	
annotation	文字列	

questnode のプロパティ

QUEST ノードには、ディシジョン 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）にできますが、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 14.2 Modeling Nodes 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
```

questnodeProperties	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. 209 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	
objective	Standard Boosting Bagging 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 InteractiveBuilder	
use_tree_directives	フラグ型	
tree_directives	string	
use_max_depth	Default Custom	
max_depth	integer	最大ツリー深さ（0 ～ 1000）。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。

questnodeProperties	値	プロパティの説明
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。 [{drugA drugB 1.5} {drugA drugC 2.1}]。 {} 内の引数は実際の予測コストです。
priors	Data Equal 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。 set :cartnode. custom_priors = [{ 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 HighestProbability HighestMeanProbability	カテゴリ型対象のデフォルト結合ルール。
range_ensemble_method	Mean 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：
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 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

regressionnodeProperties	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、p. 209 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 Stepwise Backwards Forwards	
include_constant	フラグ型	
use_weight	フラグ型	
weight_field	フィールド	
mode	Simple Expert	
complete_records	フラグ型	
tolerance	1.0E-1 1.0E-2 1.0E-3 1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 1.0E-9 1.0E-10 1.0E-11 1.0E-12	引数には二重引用符を使用します。
stepping_method	useP useF	useP： F 値確率を使用 useF: F 値を使用
probability_entry	number	
probability_removal	number	
F_value_entry	number	
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	

regressionnodeProperties	値	プロパティの説明
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 14.2 Modeling Nodes 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

sequencenodeProperties	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、p. 209 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。
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 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 14.2 Modeling Nodes 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']
```

slrmnodeProperties	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. 209 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。 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 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 14.2 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. 329 ）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 14.2 Modeling Nodes 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
```

svmnodeProperties	値	プロパティの説明
all_probabilities	フラグ型	
stopping_criteria	1.0E-1 1.0E-2 1.0E-3（デフォルト） 1.0E-4 1.0E-5 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） Polynomial Sigmoid Linear	変換に使用されるカーネル関数のタイプ。
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	

svmnodeProperties	値	プロパティの説明
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 14.2 Modeling Nodes を参照してください。

例

```
create timeseriesnode
set :timeseriesnode.method = Exsmooth
set :timeseriesnode.exsmooth_model_type = HoltsLinearTrend
set :timeseriesnode.exsmooth_transformation_type = None
```

timeseriesnodeProperties	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、 p.209 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	
method	ExpertModeler Exsmooth Arima 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	

timeseriesnodeProperties	値	プロパティの説明
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 HoltsLinearTrend BrownsLinearTrend DampedTrend SimpleSeasonal WintersAdditive WintersMultiplicative	
exsmooth_transformation_type	None SquareRoot NaturalLog	
arima_p	integer	
arima_d	integer	
arima_q	integer	
arima_sp	integer	
arima_sd	integer	
arima_sq	integer	
arima_transformation_type	None SquareRoot 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。
tf_arima_transformation_type.fieldname	None SquareRoot NaturalLog	転送関数用。
arima_detect_outlier_mode	None Automatic	
arima_outlier_additive	フラグ型	
arima_outlier_level_shift	フラグ型	
arima_outlier_innovational	フラグ型	
arima_outlier_transient	フラグ型	
arima_outlier_seasonal_additive	フラグ型	

timeseriesnodeProperties	値	プロパティの説明
arima_outlier_local_trend	フラグ型	
arima_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 14.2 Modeling Nodes 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
```

twostepnodeProperties	値	プロパティの説明
inputs	[フィールド 1 ... 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、 p.209 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。
standardize	フラグ型	

twostepnodeProperties	値	プロパティの説明
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 Number	
label_prefix	string	
distance_measure	Euclidean Loglikelihood	
clustering_criterion	AIC BIC	

章
17

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.127 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.210 ）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
anomaly_score_method	FlagAndScore FlagOnly ScoreOnly	スコアリング用に、作成される出力を決めます。
num_fields	整数	報告するフィールド数。
discard_records	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。
discard_anomalous_records	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 off です。それに対し、on 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、discard_records 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.211) 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
max_predictions	数値（整数）	
ignore_unmatached	フラグ型	
allow_repeats	フラグ型	
check_basket	NoPredictions Predictions NoCheck	
criterion	Confidence Support RuleSupport Lift 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.213) 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
flag_ensemble_method	Voting ConfidenceWeightedVoting RawPropensityWeightedVoting HighestConfidence 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。
flag_voting_tie_selection	Random HighestConfidence 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 ConfidenceWeightedVoting 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 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.216) 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.217) 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.219) 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.220) 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 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.222) 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.223) 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 MissingValues 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.226) 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 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.228) 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 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.230) 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 へ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.232 ）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.233 ）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.235 ）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.237) 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.241) 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.242) 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	

applykohonennode 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、applykohonennode 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「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.245) 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
use_custom_name	フラグ型	
custom_name	文字列	
enable_sql_generation	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.246) 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
calculate_raw_propensities	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.251) 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。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 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.254) 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 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.255) 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 MissingValues 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.258) 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.261) 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 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クリプトの詳細は、16 章 (p.260) 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 ナゲットのスクリプト名は、applysvmnode 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「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 ナゲットのスクリプト名は、applytimeseriesnode 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.264) 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 ナゲットのスクリプト名は、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クリプトの詳細は、16 章 (p.266) の「twostepnode のプロパティ」を参照してください。

章
18

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 14.2 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 フィールド 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。 詳細は、p. 283 アルゴリズム パラメータ を参照してください。
with_drillthrough	フラグ型	[ドリルスルーあり] オプション。

MS ディシジョン 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。

msassocnodeProperties	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 フィールド 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。

mssequenceclusternodeProperties	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 フィールド 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。

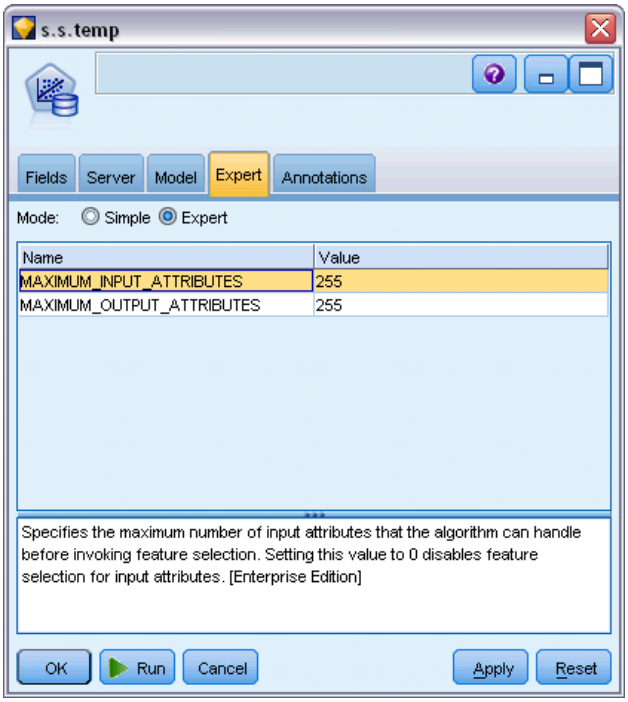
```
set :msregressionnode.msas_parameters =
[{"MAXIMUM_INPUT_ATTRIBUTES" 255}{"MAXIMUM_OUTPUT_ATTRIBUTES" 255}]
```

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 ディシジョン ツリー

applymstreenodeProperties	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。 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

applymsregressionnodeProperties	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。 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

applymsneuralnetworknodeProperties	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。 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

applymslogisticnodeProperties	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。 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

applymstimeseriesnodeProperties	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。 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 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 クラスタリング

applymssequenceclustern-odeProperties	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。 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	コスト行列	構造化プロパティ、使用フォーマット：[{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 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。

oranbnodeProperties	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 Equal Custom	
custom_priors	構造化	構造化プロパティ、使用フォーマット : 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。

oraabnnodeProperties	値	プロパティの説明
model_type	SingleFeature MultiFeature 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 Equal Custom	
custom_priors	構造化	構造化プロパティ、使用フォーマット : 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。

orasvmnodeProperties	値	プロパティの説明
active_learning	Enable Disable	
kernel_function	Linear Gaussian System	
normalization_method	zscore minmax none	
kernel_cache_size	integer	Gaussian カーネル専用。値は 1 以上でなければなりません。*
convergence_tolerance	number	値は 1 以上でなければなりません。*
use_standard_deviation	フラグ型	Gaussian カーネル専用。*
standard_deviation	number	値は 1 以上でなければなりません。*
use_epsilon	フラグ型	回帰モデルのみです。*
epsilon	number	値は 1 以上でなければなりません。*
use_complexity_factor	フラグ型	*
complexity_factor	number	*
use_outlier_rate	フラグ型	単一バリアントのみです。*
outlier_rate	number	単一バリアントのみです。 0.0-1.0.*
weights	Data Equal Custom	
custom_weights	構造化	構造化プロパティ、使用フォーマット : set :orasvmnode.custom_weights = [{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 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。

oraglmnodeProperties	値	プロパティの説明
normalization_method	zscore minmax none	
missing_value_handling	ReplaceWith-Mean UseCompleteRe-cords	
use_row_weights	フラグ型	*

oraglmnodeProperties	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 Off 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。

oradecisiontreenodeProperties	値	プロパティの説明
use_costs	フラグ型	
impurity_metric	Entropy 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、oraoclusternode タイプのノードで使用できます。

oraoclusternodeProperties	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。

orakmeansnodeProperties	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 minmax none	
distance_function	Euclidean Cosine	
iterations	integer	0-20.*
conv_tolerance	number	0.0-0.5.*
split_criterion	Variance 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。

oranmfnodeProperties	値	プロパティの説明
normalization_method	minmax 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。

oraapriorinodeProperties	値	プロパティの説明
content_field	フィールド	
id_field	フィールド	

oraapriorinodeProperties	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)

oramdlnode 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。

oraainodeProperties	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 ImportanceValue 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、applyoradecisiontreenode タイプのノードで使用できます。

applyoradecisiontreenodeProperties	値	プロパティの説明
use_costs	フラグ型	
display_rule_ids	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。

applyoranmfnodeProperties	値	プロパティの説明
display_all_features	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 ディシジョン 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。

db2imtreenodeProperties	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。

db2imtreenodeProperties	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：[{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。

db2imassocnodeProperties	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	
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 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。

db2imsequencenodeProperties	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。

db2imsequencenodeProperties	値	プロパティの説明
min_rule_support	number	0-100%
min_rule_confidence	number	0-100%
use_item_constraints	フラグ型	
item_constraints_type	Include 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。

db2imregnodeProperties	値	プロパティの説明
target	フィールド	
regression_method	transform linear polynomial rbf	
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	
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_selec-tion_method	フラグ型	
auto_feature_selection_method	normal adjusted	
use_min_significance_level	フラグ型	

db2imregnodeProperties	値	プロパティの説明
min_significance_level	number	
use_min_significance_level	フラグ型	
次のプロパティは、regression_method = rbf の場合にのみ適用されます。		
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。 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。 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。 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。 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。 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。 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。

db2imclusternodeProperties	値	プロパティの説明
cluster_method	demographic kohonen	
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。
use_execution_time_limit	フラグ型	
execution_time_limit_mins	integer	値は 1 以上です。

db2imclusternodeProperties	値	プロパティの説明
min_data_percentage	number	0-100%
maximum_CF_leaf_nodes	integer	デフォルト値は 1000 です。
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。

db2imnbnodeProperties	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。 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： [{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。

db2imlognodeProperties	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： [{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。

db2imtime-seriesnodeProperties	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	

db2imtime-seriesnodeProperties	値	プロパティの説明
forecasting_algorithm	arima exponen-tial_smoothing sea-sonal_trend_de-composition	
forecasting_end_time	auto integer date 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 linear exponen-tial_splines 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 ... 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_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_to_connection	フラグ型	true の場合、データは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。
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。
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。

共通の Netezza ノード プロパティ	値	プロパティの説明
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。
include_input_fields	フラグ型	

Netezza ディシジョン 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。

netezzadectreenodeProperties	値	プロパティの説明
target	フィールド	対象フィールド（連続型またはカテゴリ型）。
impurity_measure	Entropy Gini	
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： set :netezza_dectree.weights = [{drugA 0.3}{drugB 0.6}] デフォルトの重みはすべてのクラスで 1 です。
pruning_measure	Acc 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 partitionTraining-Data 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。
compute_probabilities	フラグ型	

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。

netezzakmeansnodeProperties	値	プロパティの説明
distance_measure	Euclidean Manhattan Canberra 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。

netezzakmeansnodeProperties	値	プロパティの説明
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 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。
model_name	string	モデル名。
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。

Netezza 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。

Netezza ディシジョン ツリー

applynetezzadectreenode 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。

Netezza K-Means

applynetezzakmeansnode 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。

章
19

出力ノードのプロパティ

出力ノードのプロパティは、ほかの種類の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 14.2 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） HTML（.html） Output（.cou）	出力のタイプを指定します。
by_fields	[フィールド 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 14.2 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 … 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 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 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。

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 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） Delimited（.csv） HTML（.html） 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 14.2 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 Flags 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 Function	
function_field	string	
function	Sum Mean Min Max SDev	
sort_mode	Unsorted Ascending Descending	
highlight_top	number	ゼロでない場合に真（true）。
highlight_bottom	number	ゼロでない場合に真（true）。
display	[Counts Expected Residuals RowPct ColumnPct 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	use_output_name が真（true）のときに、使用する名前を指定します。

matrixnode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。
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) 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 transposed を伴うことができます。例を挙げます。NODE.output_format=transposed Delimited
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 14.2 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 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} {field3 field4} ...]	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 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 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) Delimited (.csv) HTML (.html) Output (.cou)	出力のタイプを指定します。
full_filename	string	
output_view	Simple 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 14.2 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 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) Text (.txt) 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 式で使用でき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 14.2 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。 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 14.2 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 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 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） HTML（.html） Output （.cou）	出力のタイプを指定します。
full_filename	string	
examine	[フィールド フィールド フィールド]	
correlate	[フィールド 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 14.2 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.330 ）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 14.2 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 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	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 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) Delimited (.csv) HTML (.html) 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	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。
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} { Value LabelString}]	値のペアのためのラベルを指定します。 例をあげると、次のようになります。 set :typenode.value_labels. 'Drug'=[{drugA label1} {drugB label2}]
display_places	integer	フィールドが表示されるとき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。 使用フォーマット : NODE.display_places. FIELDNAME
export_places	integer	フィールドが出力されるとき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。 使用フォーマット : NODE.export_places.FIELDNAME
decimal_separator	DEFAULT PERIOD 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)。 使用フォーマット : NODE.decimal_separator. FIELDNAME
date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "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) 。 使用フォーマット : NODE.date_format.FIELDNAME 例をあげると、次のようになります。 set :tablenode.date_format. 'LaunchDate' = "DDMMYY"

tablenodeプロパティ	データ型	プロパティの説明
	"DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY	
time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(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）。 使用フォーマット： NODE.time_format.FIELDNAME 例をあげると、次のようになります。 set :tablenode.time_format. set 'BOF_enter' = "HHMMSS"
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 Auto に設定されます。 使用フォーマット： NODE.column_width.FIELDNAME
justify	AUTO CENTER LEFT RIGHT	フィールドに列調整を設定します。 使用フォーマット： NODE.justify.FIELDNAME

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 14.2 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 … フィールドN]	変換で使用するフィールド。
formula	All 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 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) 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。

章

20

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 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 14.2 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。 {" Cognos_server_URL", login_mode, "namespace", "username", "password "} ここでの意味は次の通りです。 Cognos_server_URL は、エクスポートする Cognos サーバーの URL です。 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。 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。 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。 /Public Folders/MyPackage
cognos_datasource	string	
cognos_export_mode	Publish 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している必要があります。 詳細は、 7 章 データベース エクスポート ノード in IBM SPSS Modeler 14.2 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ê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.66 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 Append Merge	

databaseexportnodeプロパティ	データ型	プロパティの説明
map	string	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（write_mode が Merge の場合にのみ有効）。 例： set :databaseexportnode.map.streamBP = 'databaseBP' 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。 set :databaseexportnode.map.[streamfield1 streamfield2 streamfield3] = [field1 field2 field3] 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。
key_fields	[フィールド フィールド ... フィールド]	キーに使用されるストリーム フィールドを指定します。map 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。
join	Database Add	例： set : databaseexportnode..join = Database
drop_existing_table	フラグ型	
delete_existing_rows	フラグ型	
default_string_size	integer	
type		スキーマ タイプの設定に用いられる構造化プロパティ。 使用フォーマット ： set :databaseexportnode.type.BP = 'VARCHAR(10)'
generate_import	フラグ型	
use_custom_create_table_command	フラグ型	custom_create_table スロットを使用して、標準の CREATE TABLE SQL コマンドを変更します。
custom_create_table_command	string	標準の CREATE TABLE 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。

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。Use_batch 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 ODBC 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 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 Space 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。Other は、loader_other_delimiter 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 data_file 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 loader_program 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。

databaseexportnodeプロパティ	データ型	プロパティの説明
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。
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。

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 14.2 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 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 14.2 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 Excel2007	
export_mode	Create 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 14.2 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 Append	
inc_field_names	フラグ型	
use_newline_after_records	フラグ型	
delimit_mode	Comma Tab Space Other	
other_delimiter	char	
quote_mode	None Single Double Other	
other_quote	フラグ型	
generate_import	フラグ型	
encoding	StreamDefault SystemDefault "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 14.2 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 UNIX SAS7 SAS8	バリアント プロパティ ラベル フィールド。
full_filename	string	
export_names	NamesAndLabels 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 14.2 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.331) 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 14.2 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。 例: set :xmlexportnode.map = [{"/top/node1" field1}{"/top/node2" field2}] 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。

章

21

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 14.2 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 LabelsAsNames	変数名と変数ラベルを処理する方法。
import_data	DataAndLabels 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 14.2 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.172 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.172 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.172 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 14.2 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.172 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.172 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.172 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 14.2 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ンタックス 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 SPV 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 14.2 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 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	

章
22
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 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 14.2 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。

付録

A

注意事項

This information was developed for products and services offered worldwide.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785, U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing, Legal and Intellectual Property Law, IBM Japan Ltd., 1623-14, Shimotsuruma, Yamato-shi, Kanagawa 242-8502 Japan.

次の文は、条項が法律と一致しないイギリスなどの国には適用されません。本出版物は、SPSS INC., AN IBM COMPANY によって提供され、明示的および暗黙的なあらゆる保証、制限されていない場合を除く商品性や特定の目的への適合性、および無違反に関する暗黙的な保証を含む意思表示と保証を放棄します。特定の取引では明示的または暗黙的な保証の免責が許可されないため、この文が適用されない場合があります。

この情報には、技術的な誤りや誤植を含まれる場合があります。本文では変更が定期的に行われます。これらの変更は本書の次の版に組み込まれます。SPSS は、本文書に記載された製品やプログラムは予告なしに改善または変更される場合があります。

この情報内にある SPSS 以外または IBM 以外の Web サイトに対する参照は、便宜上提供されたものであり、これらのWebサイトを推奨するものではありませ提供されます。これらの Web サイトの資料は、この SPSS 社製品の使用の一部ではなく、これらの Web サイトの使用は個人の責任によるものです。

IBM または SPSS に情報を送信すると、あなたに対する義務を負うことなく、適切とする方法でその情報を使用または配布する非独占的権利と IBM および SPSS 付与するものとします。

SPSS 以外の製品に関する情報は、これらの製品、公開された通知、公表されているソースの供給者から得たものです。SPSS は、それらの製品をテストしていません。また、SPSS 以外の製品に関連するパフォーマンスの正確性、互換性、またはsポの他の要求を確認することはできません。SPSS 以外の製品の機能に関する質問は、これらの製品の供給者にお問い合わせください。

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Software Group, Attention: Licensing, 233 S. Wacker Dr., Chicago, IL 60606, USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the

capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

この情報には、日常の業務で使用されているデータおよびレポートの例が含まれています。それらを可能な限り詳細に説明するために、例には個人、企業、ブランド、製品の名前が含まれます。これらの名前はすべて架空のものであり、実際の名前や住所に似ているものでも、まったくの偶然によるものです。

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

商標

IBM、IBM ロゴ、ibm.com は世界各国の四方に基づく IBM 社の登録商標です。IBM の商標の現在のリストは Web サイト http://www.ibm.com/legal/copytrade.shmtl を参照してください。

SPSS Inc., an IBM Company の SPSS の商標 は、世界各国の司法に基づく登録商標です。

Adobe、Adobe のロゴ、PostScript、および PostScript ロゴはアメリカ合衆国およびその他各国のアドビシステムズ社の登録商標または商標です。

IT Infrastructure Library は、イギリス商務局の一部である中央電子計算機局の登録商標です。

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、 Pentium はアメリカ合衆国およびその他各国のインテル社およびその子会社の商標または登録商標です。

Linux は、アメリカ合衆国およびその他各国の Linus Torvalds の登録商標です。

Microsoft、Windows、Windows NT、および Windows ロゴは、アメリカ合衆国およびその他合衆国のマイクロソフト社の商標です。

ITIL は、米国特許商標局の登録商標および登録共同体商標です。

UNIX は、アメリカ合衆国およびその他各国の The Open Group の登録商標です。

Cell Broadband Engine は、アメリカ合衆国およびその他各国のソニーコンピュータエンタテインメント株式会社の使用許諾に基づいて使用されています。

Java および Java ベースの商標およびロゴは、アメリカ合衆国およびその他各国のサン・マイクロシステムズ株式会社の商標です。

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

その他の製品およびサービス名は、IBM、SPSS、またはその他の企業の商標である場合があります。

索引