

SYBASE®

時系列ガイド

Sybase IQ 15.2

ドキュメント ID：DC01377-01-1520-01

改訂：2010年4月

Copyright © 2010 by Sybase, Inc. All rights reserved.

このマニュアルは Sybase ソフトウェアの付属マニュアルであり、新しいエディションまたはテクニカル・ノートで特に示されない限り、後続のリリースにも付属します。このマニュアルの内容は予告なしに変更されることがあります。このマニュアルに記載されているソフトウェアはライセンス契約に基づいて提供され、使用や複製はこの契約に従って行う場合にのみ許可されます。

追加ドキュメントを注文する場合は、米国、カナダのお客様は、カスタマ・フルフィルメント事業部 (電話 800-685-8225、ファックス 617-229-9845) までご連絡ください。

米国のライセンス契約が適用されるその他の国のお客様は、上記のファックス番号でカスタマ・フルフィルメント事業部までご連絡ください。その他の海外のお客様は、Sybase の関連会社または最寄りの販売代理店にお問い合わせください。アップグレードは定期ソフトウェア リリース日にもみ提供されます。このマニュアルの内容を Sybase, Inc. の書面による事前の許可なく複製、転載、翻訳することは、電子的、機械的、手作業、光学的、その他、形態や手段を問わず禁じられています。

Sybase の商標は Sybase の商標リスト (<http://www.sybase.com/detail?id=1011207>) で確認できます。Sybase および表記されている商標は、Sybase, Inc の商標です。® は、米国で登録されていることを示します。

Java および Java 関連の商標は、米国およびその他の国における Sun Microsystems, Inc. の商標または登録商標です。

Unicode と Unicode のロゴは Unicode, Inc. の登録商標です。

このマニュアルに記載されている上記以外の社名および製品名は、各社の商標または登録商標の場合があります。

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568

目次

対象読者	1
関連マニュアル	3
概要	7
ライセンスの前提条件	7
時系列の予測／分析関数の IMSL ライブラリ	7
IMSL ライブラリのエラー処理とエラー・ロギン グの制御	8
エラー処理値	8
エラー・ロギング値	9
時系列予測／分析関数	11
集合時系列予測／分析関数	11
集合時系列関数のパラメータ	11
スカラ時系列予測／分析関数	12
アルファベット順の関数リスト	14
TS_ARMA_AR 関数 [集合]	14
TS_ARMA_CONST 関数 [集合]	16
TS_ARMA_MA 関数 [集合]	19
TS_AUTOCORRELATION 関数 [集合]	21
TS_AUTO_ARIMA 関数 [集合]	23
TS_AUTO_ARIMA_OUTLIER 関数 [集合]	26
TS_AUTO_ARIMA_RESULT_AIC 関数 [スカラ]	29
TS_AUTO_ARIMA_RESULT_AICC [スカラ]	30
TS_AUTO_ARIMA_RESULT_BIC 関数 [スカラ]	31
TS_AUTO_ARIMA_RESULT_FORECAST_ERROR 関 数 [スカラ]	32
TS_AUTO_ARIMA_RESULT_FORECAST_VALUE 関 数 [スカラ]	33

TS_AUTO_ARIMA_RESULT_MODEL_P 関数 [スカ ラ]	34
TS_AUTO_ARIMA_RESULT_MODEL_Q 関数 [スカ ラ]	35
TS_AUTO_ARIMA_RESULT_MODEL_S 関数 [スカ ラ]	36
TS_AUTO_ARIMA_RESULT_MODEL_D 関数 [スカ ラ]	37
TS_AUTO_ARIMA_RESULT_RESIDUAL_SIGMA 関 数 [スカラ]	38
TS_AUTO_UNI_AR 関数 [集合]	39
TS_BOX_COX_XFORM 関数 [集合]	41
TS_DIFFERENCE 関数 [集合]	44
TS_DOUBLE_ARRAY 関数 [スカラ]	47
TS_ESTIMATE_MISSING 関数 [集合]	48
TS_GARCH 関数 [集合]	51
TS_GARCH_RESULT_A 関数 [スカラ]	54
TS_GARCH_RESULT_AIC 関数 [スカラ]	55
TS_GARCH_RESULT_USER [スカラ]	56
TS_INT_ARRAY 関数 [スカラ]	57
TS_LACK_OF_FIT 関数 [集合]	58
TS_LACK_OF_FIT_P 関数 [集合]	60
TS_MAX_ARMA_AR 関数 [集合]	63
TS_MAX_ARMA_CONST 関数 [集合]	65
TS_MAX_ARMA_LIKELIHOOD 関数 [集合]	67
TS_MAX_ARMA_MA 関数 [集合]	69
TS_OUTLIER_IDENTIFICATION 関数 [集合]	71
TS_PARTIAL_AUTOCORRELATION 関数 [集合]	76
TS_VWAP 関数 [集合]	78
DATASET サンプル入力データ・テーブル	80
索引	83

対象読者

このマニュアルは、時系列の予測と分析に Sybase® IQ を使用する RAP – The Trading Edition™ Enterprise ユーザを対象としています。このマニュアルでは、時系列関数の SQL 構文とパラメータ、およびサード・パーティ製の IMSL™ C Stat/C Math 外部ライブラリへのパラメータのマッピングについて説明します。ユーザ定義のスカラー関数および集合関数の概念情報については、『**ユーザ定義関数ガイド**』を参照してください。

対象読者

関連マニュアル

追加の情報は、Sybase IQ および SQL Anywhere のマニュアルに記載されています。

関連する Sybase IQ マニュアル

Sybase IQ のマニュアル・セットには、以下が含まれています。使用しているプラットフォームの

- 『リリース・ノート』 - マニュアルには記載できなかった最新の情報が記載されています。
リリース・ノートの最新バージョンを利用できる場合があります。製品 CD のリリース後に追加された製品またはマニュアルに関する重要情報がないかどうかを確認するには、Sybase Product Manuals Web サイトを使用してください。
- 使用しているプラットフォームの 『インストールおよび設定ガイド』 - Sybase IQ のインストール、アップグレード、およびいくつかの設定の手順について説明しています。
- 『新機能の概要』 - 最新バージョンの新機能と動作変更の概要がまとめられています。
- 『Sybase IQ による高度なセキュリティ』 - Sybase IQ データ・レポジトリ内でのユーザによるカラムの暗号化の使用について説明しています。この製品のオプションをインストールするには、別途ライセンスが必要です。
- 『エラー・メッセージ』 - Sybase エラー・コード、SQLCode、および SQLState によって参照される Sybase IQ エラー・メッセージ、および SQL プリプロセッサのエラーと警告を示します。
- 『IMSL C 数値関数統計ライブラリ・ユーザー・ガイド Volume 2 of 2: C Stat Library』 - IMSL C Stat ライブラリの時系列 C 関数の簡潔な説明が記載されています。このマニュアルは、RAP - The Trading Edition™ Enterprise のユーザのみが入手できます。
- 『Sybase IQ の概要』 - Sybase IQ や Sybase Central™ データベース管理ツールの操作に慣れていない場合、このマニュアルを参照してください。操作の練習ができます。
- 『パフォーマンス&チューニング・ガイド』 - 巨大なデータベースのクエリ最適化、設計、チューニングについて説明しています。
- 『クイック・スタート』 - Sybase IQ のソフトウェア・インストールを検証するために Sybase IQ に付属のデモ・データベースの構築とクエリを行う手順が説明されています。デモ・データベースのマルチプレックスへの変換についても説明しています。
- 『リファレンス・マニュアル』 - Sybase IQ のリファレンス・ガイドです。

- 『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』 - Sybase IQ でサポートされる SQL、ストアド・プロシージャ、データ型、およびシステム・テーブルについて説明しています。
- 『リファレンス：文とオプション』 - Sybase IQ がサポートしている SQL 文およびオプションについて説明しています。
- 『システム管理ガイド』 - 内容は次のとおりです。
 - 『システム管理ガイド：第 1 巻』 - 起動、接続、データベース作成、自動入力とインデックス作成、バージョン設定、照合、システムのバックアップとリカバリ、トラブルシューティング、およびデータベースの修復方法について説明しています。
 - 『システム管理ガイド：第 2 巻』 - プロシージャおよびバッチの作成と実行、OLAP でのプログラミング、リモート・データへのアクセス、Open Server としての IQ の設定について説明しています。このマニュアルでは、スケジューリングとイベント処理、XML でのプログラミング、デバッグについても説明しています。
- 『時系列ガイド』では、時系列の予測と分析に使用する SQL 関数について説明しています。この製品オプションを使用するには、RAP - The Trading Edition™ Enterprise が必要です。
- 『Sybase IQ の非構造化データ分析の概要』では、Sybase IQ データ・レポジトリ内での非構造化データの格納と取得について説明しています。このオプション製品をインストールするには、別のライセンスが必要です。
- 『ユーザ定義関数ガイド』では、ユーザ定義関数、パラメータ、および使用のシナリオについて説明しています。
- 『Sybase IQ マルチプレックスの使用』では、複数のノードにまたがって発生する大きなクエリの負荷を管理するマルチプレックス機能の使用方法について説明しています。
- 『ユーティリティ・ガイド』では、Sybase IQ ユーティリティ・プログラムのリファレンス項目 (使用可能な構文、パラメータ、オプションなど) について説明しています。

関連する SQL Anywhere マニュアル

注意： Sybase IQ は SQL Anywhere Studio® のコンポーネントである SQL Anywhere® と多くのコンポーネントを共有しているため、Sybase IQ は SQL Anywhere と同じ機能を数多くサポートしています。IQ のマニュアル・セットは、SQL Anywhere Studio のマニュアルの該当する箇所を参照しています。

SQL Anywhere には、次のマニュアルがあります。

- 『SQL Anywhere サーバーデータベース管理』では、SQL Anywhere データベースの実行、管理、設定方法について説明します。データベース接続、データベース・サーバ、データベース・ファイル、バックアップ手順、セキュリティ

ティ、高可用性、Replication Server での複製、管理ユーティリティおよびオプションについて説明します。

- 『SQL Anywhere サーバ - プログラミング』では、C、C++、Java、PHP、Perl、Python、および Visual Basic や Visual C# などの .NET プログラミング言語を使用したデータベース・アプリケーションの構築および展開方法について説明します。このマニュアルでは、ADO.NET や ODBC などの各種プログラミング・インタフェースについても説明します。
- 『SQL Anywhere サーバ - SQL リファレンス』は、システム・プロシージャの参照情報およびカタログ (システム・テーブルとビュー) を提供します。さらに、SQL 言語の SQL Anywhere での実装についても説明しています (検索条件、構文、データ型、関数)。
- 『SQL Anywhere サーバ - SQL の使用法』では、データベースの設計 / 作成方法、データのインポート / エクスポート / 変更方法、データの検索方法、ストアド・プロシージャとトリガの作成方法について説明します。

Product Manuals および DocCommentXchange でも、SQL Anywhere Studio 11.0 コレクションの SQL Anywhere のマニュアルを参照できます。

概要

時系列とは、連続的に測定 (通常は一定間隔をおいて測定) される時間ポイントのシーケンスです。時系列予測では、モデルを使用して、過去のデータ・ポイントに基づいて将来のデータ・ポイントを予測します。時系列分析では、予測モデリング、自己回帰モデリング、短期的変動の補整などの手法を使用して、データ・ポイントの基本となるコンテキストやデータの傾向を特定します。

Sybase® IQ には、時系列の予測と分析に使用する SQL 関数が用意されています。これらの関数は、IMSL™ C Stat および C Math 外部ライブラリ内で実行するユーザ定義関数 (UDF: User Defined Function) を宣言します。

時系列の予測/分析関数には、OLAP スタイルの集合関数とサポート・スカラ関数があります。

ライセンスの前提条件

時系列および予測の機能は、RAP – The Trading Edition Enterprise でのみ使用できません。RAP – The Trading Edition Enterprise には、必要なすべてのライセンスが付属しています。

時系列の予測/分析関数の IMSL ライブラリ

TS-VWAP を除くすべての時系列関数と予測関数は、サード・パーティ製の 2 つの外部ライブラリを呼び出します。Visual Numerics Inc. の IMSL™ C Stat ライブラリおよび C Math ライブラリには、金融時系列の予測/分析計算に使用する C 関数が含まれています。

ラッパ・ライブラリ **libtsudf** には、IMSL C Stat ライブラリと C Math ライブラリに含まれている関数を呼び出すユーザ定義関数 (UDF) が含まれています。

時系列 SQL 関数と予測 SQL 関数は、libtsudf ラッパ・ライブラリを自動的に呼び出します。ユーザが時系列および予測分析用の有効なユーザ定義集合関数を呼び出すと、Sybase IQ は IMSL C Stat ライブラリおよび C Math ライブラリをロードします。

IMSL C Stat ライブラリおよび C Math ライブラリの名前とロケーションは、Sybase IQ がインストールされているプラットフォームによって異なります。

表 1 : IMSL ライブラリのロケーションとファイル名

	Windows 32 ビット版	Windows 64 ビット版	UNIX (AIX を除 く)	AIX
ライブラ リのディ レクトリ	bin32	bin64	lib64	lib64
ライブラ リ・ファ イル名	imslcmath_imsl_dll .dll imslcstat_imsl_dll. dll	imslcmath_imsl_dll .dll imslcstat_imsl_dll. dll	libimslcmath_imsl .so libimslcstat_imsl.s o	libimslcmath_imsl _r.so libimslcstat_imsl_r .so

IMSL ライブラリのエラー処理とエラー・ロギングの制御

IMSL ライブラリを呼び出す時系列関数のエラー処理とエラー・ロギングの動作を制御できます。IMSL ライブラリ関数の呼び出し時に実行時エラーが発生すると、Sybase IQ はユーザが選択したエラー処理オプションに従って応答し、ユーザが選択したエラー・ロギング・オプションに従ってエラーのログを取ります。

1. 次の **SET OPTION** 文を呼び出します。

```
SET OPTION PUBLIC.Time_Series_Error_Level = 'value'
```

エラー・レベルの *value* は 0、1、2、3 のいずれかです。

2. 次の **SET OPTION** 文を呼び出します。

```
SET OPTION PUBLIC.Time_Series_Log_Level = 'value'
```

ログ・レベルの *value* は 0、1、2、3 のいずれかです。

エラー処理値

次の表に、**Time_Series_Error_Level** の値を示します。

表 2 : IMSL ライブラリエラー処理値

値	説明
0 (デフォルト)	IMSL ライブラリ関数の呼び出し中に取得する可能性のあるすべての警告とエラーが無視されます。そのような状況が発生すると、時系列関数は NULL 値を返します。
1	IMSL ライブラリ関数の呼び出し中に時系列関数が警告またはエラー・メッセージを取得すると、Sybase IQ はエラー・メッセージを返し、SQL クエリをアボートします。

値	説明
2	IMSL ライブラリ関数の呼び出し中に時系列関数が致命的なエラー・メッセージを取得すると、Sybase IQ はエラー・メッセージを返し、SQL クエリをアボートします。ただし、警告を取得した場合は、時系列関数が NULL 値を返します。
3	IMSL ライブラリ関数の呼び出し中に時系列関数がターミナル・エラー・メッセージを取得すると、Sybase IQ はエラー・メッセージを返し、SQL クエリをアボートします。しかし、警告または致命的なエラーが取得された場合、時系列関数は NULL 値を返します。

エラー・ロギング値

次の表に、`Time_Series_Log_Level` の値を示します。

表 3 : IMSL ライブラリのエラー・ロギング値

値	説明
0 (デフォルト)	IMSL ライブラリ関数の呼び出し中に返されたすべての警告およびエラーは無視され、ログ・ファイルにログされません。
1	IMSL ライブラリ関数の呼び出し中に時系列関数が警告またはエラー・メッセージを返すと、メッセージがログ・ファイルにログされます。
2	IMSL ライブラリ関数の呼び出し中に時系列関数が致命的なエラー・メッセージを返すと、メッセージがログ・ファイルにログされます。警告はログされません。
3	IMSL ライブラリ関数の呼び出し中に時系列関数がターミナル・エラー・メッセージを返すと、メッセージがログ・ファイルにログされます。警告および致命的なエラーはログされません。

時系列予測／分析関数

時系列 SQL 関数は、OLAP スタイルの集合関数、または OLAP スタイルの集合をサポートするスカラ関数です。時系列関数と予測関数を使用して、自己回帰和分移動平均 (ARIMA: Auto Regressive Integrated Moving Average) モデルまたは一般化自己回帰条件付き分散不均一 (GARCH: Generalized Auto Regressive Conditional Heteroskedasticity) モデルを使用し、Box-Cox 変換などのモデル構築／評価ユーティリティを利用します。

集合時系列予測／分析関数

集合時系列予測／分析関数は、金融時系列統計分析専用設計された OLAP スタイルの集合関数です。集合関数であるこれらの関数は、入力値の多数のセットを受け入れ、1つの結果を返します。

OLAP とウィンドウ集合関数については、『システム管理ガイド：第2巻』の「OLAP の使用」を参照してください。

集合時系列関数のパラメータ

次の表に、各集合時系列関数のパラメータを示します。

表 4：集合時系列関数

時系列関数	パラメータ
TS_ARMA_AR	<i>(timeseries_expression , ar_count, ar_elem, method)</i>
TS_ARMA_CONST	<i>(timeseries_expression , method)</i>
TS_ARMA_MA	<i>(timeseries_expression , ma_count, ma_elem, method)</i>
TS_AUTOCORRELATION	<i>(timeseries_expression , lagmax, lag_elem)</i>
TS_AUTO_ARIMA	<i>(time_value, timeseries_expression [,max_lag [,critical [,epsilon [,criterion [,confidence [,model [,n_predictions]]]]]]])</i>
TS_AUTO_ARIMA_OUTLIER	<i>(time_value, timeseries_expression [,max_lag [,critical [,epsilon [,criterion [,confidence [,model [,delta]]]]]]])</i>

時系列関数	パラメータ
TS_AUTO_UNI_AR	(<i>timeseries_expression</i> , <i>ar_count</i> , <i>ar_elem</i> , <i>method</i>)
TS_BOX_COX_XFORM	(<i>timeseries_expression</i> , <i>power</i> [, <i>shift</i> [, <i>inverse</i>]])
TS_DIFFERENCE	(<i>timeseries_expression</i> , <i>period1</i> [, <i>period2</i> [, ... <i>period 10</i>]])
TS_ESTIMATE_MISSING	(<i>timeseries_expression</i> , <i>method</i>)
TS_GARCH	(<i>timeseries_expression</i> , <i>garch_count</i> , <i>arch_count</i> , <i>xguess_binary_encoding</i> , [<i>max_sigma</i>])
TS_LACK_OF_FIT	(<i>timeseries_expression</i> , <i>p_value</i> , <i>q_value</i> , <i>lagmax</i> , [<i>tolerance</i>])
TS_LACK_OF_FIT_P	(<i>timeseries_expression</i> , <i>p_value</i> , <i>q_value</i> , <i>lagmax</i> , [<i>tolerance</i>])
TS_MAX_ARMA_AR	(<i>timeseries_expression</i> , <i>ar_count</i> , <i>ar_elem</i>)
TS_MAX_ARMA_CONST	(<i>timeseries_expression</i>)
TS_MAX_ARMA_MA	(<i>timeseries_expression</i> , <i>ma_count</i> , <i>ma_elem</i>)
TS_MAX_ARMA_LIKELIHOOD	(<i>timeseries_expression</i>)

スカラ時系列予測／分析関数

スカラ時系列 UDF とスカラ予測 UDF は、**TS_GARCH** および **TS_AUTO_ARIMA** 集合関数をサポートしています。**TS_GARCH** と **TS_AUTO_ARIMA** はそれぞれバイナリ複合を生成しますが、バイナリ入力も受け入れます。**TS_INT_ARRAY** 関数は、**TS_AUTO_ARIMA** と **TS_AUTO_ARIMA_OUTLIER** の入力を提供します。**TS_DOUBLE_ARRAY** 関数は、**TS_GARCH** の入力を提供します。その他のスカラ関数は、集合関数から個々のスカラ結果値を返します。サポート・スカラ関数は、**TS_GARCH** および **TS_AUTO_ARIMA** 関数のパラメータを外部 IMSL ライブラリに含まれる C 関数のパラメータにマップします。

次のスカラ関数は、**TS_GARCH** 集合関数をサポートします。

表 5 : **TS_GARCH** をサポートするスカラ関数

時系列関数	パラメータ
TS_DOUBLE_ARRAY	(<i>xguess1</i> , <i>xguess2</i> , [... [, <i>xguess10</i>] ...])

時系列関数	パラメータ
TS_GARCH_RESULT_A	(<i>ts_garch_result</i>)
TS_GARCH_RESULT_AIC	(<i>ts_garch_result</i>)
TS_GARCH_RESULT_USER	(<i>ts_garch_result, model_element_number</i>)

次のスカラ関数は、TS_AUTO_ARIMA 集合関数をサポートします。

表 6 : TS_AUTO_ARIMA をサポートするスカラ関数

時系列関数	パラメータ
TS_INT_ARRAY	(<i>int1, int2, int3, int4, [... [, int10] ...]</i>)
TS_AUTO_ARIMA_RESULT_AIC	(<i>auto_arima_result</i>)
TS_AUTO_ARIMA_RESULT_AICC	(<i>auto_arima_result</i>)
TS_AUTO_ARIMA_RESULT_BIC	(<i>auto_arima_result</i>)
TS_AUTO_ARIMA_RESULT_FORECAST_VALUE	(<i>auto_arima_result, model_element_number</i>)
TS_AUTO_ARIMA_RESULT_FORECAST_ERROR	(<i>auto_arima_result, forecast_element_number</i>)
TS_AUTO_ARIMA_RESULT_MODEL_P	(<i>auto_arima_result</i>)
TS_AUTO_ARIMA_RESULT_MODEL_Q	(<i>auto_arima_result</i>)
TS_AUTO_ARIMA_RESULT_MODEL_S	(<i>auto_arima_result</i>)
TS_AUTO_ARIMA_RESULT_MODEL_D	(<i>auto_arima_result</i>)
TS_AUTO_ARIMA_RESULT_RESIDUAL_SIGMA	(<i>auto_arima_result</i>)

次のスカラ関数は、TS_AUTO_ARIMA_OUTLIER 集合関数をサポートします。

表 7 : TS_AUTO_ARIMA_OUTLIER をサポートするスカラ関数

時系列関数	パラメータ
TS_INT_ARRAY	(<i>int1, int2, int3, int4, [... [, int10] ...]</i>)

アルファベット順の関数リスト

この項では、すべての時系列関数の詳細（構文、ライセンスの前提条件、パラメータの説明、使用法、IMSL ライブラリ・マッピング、例、標準と互換性の情報）を示します。

TS_ARMA_AR 関数 [集合]

自己回帰移動平均 (ARMA) モデルのパラメータの最小二乗近似を計算し、要求された自己回帰近似を返します。

構文

```
TS_ARMA_AR (timeseries_expression
, ar_count, ar_elem, method)
```

```
OVER (window-spec)
```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **timeseries_expression** – 数値式。通常は時系列の要素を含むカラム名。
- **ar_count** – 計算する自己回帰値の数を含む整数。
- **ar_elem** – 返される計算された AR 配列の要素を示す整数。 *ar_elem* は 0 より大きく、*ar_count* 以下である必要があります。
- **method** – (オプション) 推定の計算に使用するプロシージャのタイプを示す整数。0 (デフォルト値) = 最小 2 乗法、1 = モーメント法
- **window-spec** – **TS_ARMA_AR** は、**OVER ()** 句が必要な OLAP 関数です。

使用法

TS_ARMA_AR 時系列関数は、自己回帰推定を含む倍精度浮動小数点値を返します。**TS_ARMA_AR** は、IMSL ライブラリの関数 **imsls_d_arma** を呼び出します。

IMSL マッピング

TS_ARMA_AR の引数は、次のように IMSL ライブラリの関数 **imsls_d_arma** にマップします。

```
params = imsls_d_arma(n_objs, z, p, q, methodID, 0);
```

- **n_objs** – 現在のウィンドウ・フレームのローの数を含みます。

- **z[]** – 現在のウィンドウ・フレームの *timeseries_expression* の値を含みます。
- **p** – ユーザ定義集合関数の引数 *ar_count* にマップします。
- **q** – =1.
- **methodID** – **TS_ARMA_AR** の method 引数にマップします。
 IMSLS_METHOD_OF_MOMENTS または IMSLS_LEAST_SQUARES に設定できます。

imsls_d_arma が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

例

この例では、**TS_ARMA_AR** 関数を含む SQL 文と、この関数から返されるデータ値を示します。この例では、サンプル入力データ・テーブル (*DATASET*) を入力データとして使用します。DATASET サンプル入力データ (80 ページ) を参照してください。

次の SQL 文では、最小 2 乗法を使用して、*data* カラムの 1 つの値を含む自己回帰推定の最初の要素を返します。

```
SELECT TS_ARMA_AR(data,1,1,0) OVER (ORDER BY rownum ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS res FROM DATASET
```

Sybase IQ は 50 個のローを返します。各ローには同じ値が含まれています。

表 8 : TS_ARMA_AR から返される値

res
0.898793
0.898793
0.898793
0.898793
0.898793
0.898793
0.898793
0.898793
0.898793
0.898793
...
0.898793

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠
- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

システム管理ガイド：第2巻』の「OLAPの使用」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第2/2巻C 統計ライブラリ

TS_ARMA_CONST 関数 [集合]

自己回帰移動平均 (ARMA) モデルのパラメータの最小二乗近似を計算し、推定数を返します。

構文

```
TS_ARMA_CONST (timeseries_expression
, method)
```

```
OVER (window-spec)
```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **timeseries_expression** – 数値式。通常は時系列の要素を含むカラム名。
- **method** – 推定の計算に使用するプロシージャのタイプを示す整数。0 (デフォルト値) = 最小2乗法、1 = モーメント法
- **window-spec** – **TS_ARMA_CONST** は、**OVER ()** 句が必要な OLAP 関数です。

使用法

この時系列関数は、関数によって生成される定数推定を含む倍精度浮動小数点値を返します。**TS_ARMA_CONST** は、IMSL ライブラリの関数 **imsls_d_arma** を呼び出します。

IMSL マッピング

TS_ARMA_CONST の引数は、次のように IMSL ライブラリの関数 **imsls_d_arma** にマップします。

```
params = imsls_d_arma(n_objs, z, p, q, IMSLS_CONSTANT, method_id, 0);
```

- **n_objs** – 現在のウィンドウ・フレームのローの数を含みます。
- **z[]** – 現在のウィンドウ・フレームの *timeseries_expression* の値を含みます。
- **p** – =1.
- **q** – =1.
- **methodID** – **TS_ARMA_CONST** の method 引数にマップします。

関数 **imsls_d_arma** が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第2/2巻C統計ライブラリ』を参照してください。

例 1

この例では、**TS_ARMA_CONST** 関数を含む SQL 文と、この関数から返されるデータ値を示します。この例では、サンプル入力データ・テーブル (*DATASET*) を入力データとして使用します。*DATASET* サンプル入力データ (80 ページ) を参照してください。

次の SQL 文では、最小 2 乗法を使用して、*data* カラムの推定定数を返します。

```
SELECT TS_ARMA_CONST(data,0) OVER (ORDER BY ROWNUM rows BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS res FROM DATASET
```

Sybase IQ は 50 個のローを返します。各ローには同じ値が含まれています。

表 9 : TS_ARMA_CONST の例 1 で返される値

res
0.082077
0.082077
0.082077
0.082077
0.082077
0.082077
0.082077
0.082077
0.082077
...
0.082077

例 2

この例は、AR、MA、および定数パラメータの推定を返すサンプル・クエリを示します。配列内の AR と MA の最初の要素には、1つの要素が含まれます。

```
select ts_arma_ar(data,1,1,0) over (order by rownum rows between
unbounded preceding and unbounded following) as ar_param,
ts_arma_ma(data,1,1,0) over (order by rownum rows between unbounded
preceding and unbounded following) as ma_param, ts_arma_const(data,
0) over (order by rownum rows between UNBOUNDED preceding and
unbounded following) as const_param FROM DATASET
```

Sybase IQ は 50 個のデータ・ローを返します。各ローには同じ 3つの値が含まれています。

表 10 : TS_ARMA_CONST の例 2 で返される値

ar_param	ma_param	const_param
0.898793	0.105075	0.082077
0.898793	0.105075	0.082077
0.898793	0.105075	0.082077
0.898793	0.105075	0.082077
0.898793	0.105075	0.082077
0.898793	0.105075	0.082077
0.898793	0.105075	0.082077
0.898793	0.105075	0.082077
0.898793	0.105075	0.082077
...
0.898793	0.105075	0.082077

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠
- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

システム管理ガイド：第 2 巻』の「OLAP の使用」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ

TS_ARMA_MA 関数 [集合]

自己回帰移動平均 (ARMA) モデルのパラメータの最小二乗近似を計算し、要求された自己回帰移動平均近似を返します。

構文

```
TS_ARMA_MA (timeseries_expression
, ma_count, ma_elem, method)
```

```
OVER (window-spec)
```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **timeseries_expression** – 数値式。通常は時系列の要素を含むカラム名。
- **ma_count** – 計算する自己回帰値の数を含む整数。
- **ma_elem** – 計算された移動平均配列から返す要素を示す整数。この整数は 0 より大きく、ma_count 以下である必要があります。
- **method** – (オプション) 推定の計算に使用するプロシージャを示す整数。0 (デフォルト値) = 最小 2 乗法、1 = モーメント法
- **window-spec** – **TS_ARMA_MA** は、**OVER ()** 句が必要な OLAP 関数です。

使用法

この時系列関数は、移動平均推定を表す倍精度浮動小数点値を返します。

TS_ARMA_MA は、IMSL ライブラリの関数 **imsls_d_arma** を呼び出します。

IMSL マッピング

TS_ARMA_MA の引数は、次のように IMSL ライブラリの関数 **imsls_d_arma** にマップします。

```
params = imsls_d_arma(n_objs, z, p, q, method_id, 0);
```

- **n_objs** – 現在のウィンドウ・フレームのローの数を含みます。
- **z[]** – 現在のウィンドウ・フレームの *timeseries_expression* の値を含みます。
- **p** – =1.
- **q** – ユーザ定義集合関数の引数 ma_count にマップします。
- **method_id** – **TS_ARMA_MA** の method 引数にマップします。

関数 `imsls_d_arma` が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

例

この例では、`TS_ARMA_MA` 関数を含む SQL 文と、この関数から返されるデータ値を示します。この例では、サンプル入力データ・テーブル (`DATASET`) を入力データとして使用します。`DATASET` サンプル入力データ (80 ページ)を参照してください。

次の SQL 文では、最小 2 乗法を使用して、`data` カラムの 1 つの要素を含む配列の最初の要素を返します。

```
SELECT TS_ARMA_MA(data,1,1,0) OVER (ORDER BY rownum ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS res FROM DATASET
```

Sybase IQ は 50 個のローを返します。各ローには同じ値が含まれています。

表 11 : `TS_ARMA_MA` から返される値

res
0.105075
0.105075
0.105075
0.105075
0.105075
0.105075
0.105075
0.105075
0.105075
0.105075
...
0.105075

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠
- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

システム管理ガイド：第2巻』の「OLAPの使用」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第2/2巻 C 統計ライブラリ

TS_AUTOCORRELATION 関数 [集合]

定常時系列のサンプル自己相関関数を計算します。

構文

```
TS_AUTOCORRELATION (timeseries_expression
, lagmax, lag_elem)
```

```
OVER (window-spec)
```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **timeseries_expression** – 数値式。通常は時系列の要素を含むカラム名。
- **lagmax** – 自己共分散の最大遅延、自己相関、および自己相関の標準誤差を示す整数。整数は、1 以上で、時系列の要素数よりも少ない必要があります。
- **lag_elem** – 自己相関配列のどの要素を返すかを指定する整数。整数は、0 よりも大きく、lagmax 以下である必要があります。
- **window-spec** – **TS_AUTOCORRELATION** は、**OVER ()** 句が必要な OLAP 関数です。

使用法

この時系列関数は、自己相関値を表す倍精度浮動小数点値を返します。

TS_AUTOCORRELATION は、IMSL ライブラリの関数 **imsls_d_autocorrelation** を呼び出します。

IMSL マッピング

TS_AUTOCORRELATION の引数は、次のように IMSL ライブラリの関数 **imsls_d_autocorrelation()** にマップします。

```
params = imsls_d_autocorrelation(n_objs, x[], lagmax, 0);
```

- **n_objs** – 現在のウィンドウ・フレームのローの数を含みます。
- **x[]** – 現在のウィンドウ・フレームの *timeseries_expression* の値を含みます。
- **lagmax** – ユーザ定義集合関数の引数 *lag_max* にマップします。

関数 **imsls_d_autocorrelation** が時系列計算を実行する方法の詳細については、『IMSLS 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

例

この例では、**TS_AUTOCORRELATION** 関数を含む SQL 文と、この関数から返されるデータ値を示します。この例では、サンプル入力データ・テーブル (*DATASET*) を入力データとして使用します。DATASET サンプル入力データ (80 ページ) を参照してください。

次の SQL 文では、*data* カラムの時系列データの自己相関を含む配列の 2 番目の要素を返します。

```
SELECT TS_AUTOCORRELATION(data,2,2) OVER (ORDER BY ROWNUM rows
BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS res FROM
DATASET
```

Sybase IQ は 50 個のローを返します。各ローには同じ値が含まれています。

表 12 : **TS_AUTOCORRELATION** から返される値

res
0.803659
0.803659
0.803659
0.803659
0.803659
0.803659
0.803659
0.803659
0.803659
0.803659
...
0.803659

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠

- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

システム管理ガイド：第2巻』の「OLAPの使用」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第2/2巻C 統計ライブラリ

TS_AUTO_ARIMA 関数 [集合]

乗法的季節自己回帰和分移動平均 (ARIMA) モデルのパラメータを決定し、系列の終端を超えて持続する、異常値による影響を組み込んで予測を生成します。

構文

```

TS_AUTO_ARIMA( <
time_value
>, <
timeseries_expression
> [ , <
max_lag
> [ , <
critical
> [ , <
epsilon
> [ , <
criterion
> [ , <
confidence
> [ , <
model
> [ , <
n_predictions
>]]]]]] )
OVER (
window-spec
)

```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **time_value** – 各入力時系列データ・ポイントの時刻値。
- **timeseries_expression** – 数値式。通常は差異を計算する時系列の要素を含むカラム名。

- **max_lag** – (オプション) AR(p) モデルに適合する場合に許容される最大遅延を表します。整数の定数または定数式である必要があります。デフォルトは 10。
- **critical** – (オプション) 異常値を検出するためのスレッシュホールドとして使用する臨界値。この値は 0 より大きく、倍精度浮動小数点の定数または定数式である必要があります。デフォルトは 3.0 です。
- **epsilon** – (オプション) 異常値の検出時にパラメータ推定の精度を制御する正の許容値。この値は、倍精度浮動小数点の定数または定数式である必要があります。デフォルトは 0.001 です。
- **criterion** – (オプション) 最適なモデルを選択するために使用する情報量基準。整数の定数または定数式 (値は 0、1、または 2) である必要があります。
 - 0 – (デフォルト) 赤池情報量基準 (AIC)
 - 1 – 赤池補正情報量基準 (AICC)
 - 2 – ベイズ情報量基準 (BIC)
- **confidence** – (オプション) 排他的間隔 (0, 100) から取得された、予測の信頼限界を計算するための信頼レベル。confidence の一般的な選択肢は、90.0、95.0、99.0 です。倍精度浮動小数点の定数または定数式である必要があります。デフォルトは 95.0 です。
- **n_predictions** – (オプション) 要求された予測の数。正の整数の定数または定数式である必要があります。サポートされる最大値は 2000 です。デフォルトは 0 です。
- **model** – (オプション) p 、 q 、 s 、 d の ARIMA 値をこの順序で含む、4 つの整数の長さのバイナリでコード化された整数配列を表します。TS_INT_ARRAY 関数で model 引数が指定されている場合、TS_AUTO_ARIMA 関数は、IMSL ライブラリにある `imsls_d_auto_arima` 関数の (`IMSL_METHOD`, `int method`) 引数の `method 3 (Specified ARIMA)` パラメータを使用して、 p 、 q 、 s 、 d の各値を決定します。文節拡張 model パラメータが NULL の場合、TS_AUTO_ARIMA 関数は自動的に ARIMA(p,0,0)X(0,d,0) モデルを計算し、指定された誤差基準を最小化します。デフォルトは NULL です。
- **window-spec** – TS_AUTO_ARIMA は、ORDER BY 句を含む OVER() 句が必要な OLAP 関数です。ROWS または RANGE 指定子は、OVER() 句では使用できません。

使用法

OLAP スタイルの集合関数である TS_AUTO_ARIMA は、1 つの SQL 結果 (特別にコード化された可変長のバイナリ結果値) を生成します。サポート・スカラ関数は、バイナリ複合出力値を受け入れ、その出力値から個々のスカラ結果値を返します。

OLAP スタイルの集合関数は入力タプルごとに 1 つの結果値を返すため、パーティション内の各ローに対して同じバイナリ複合結果が返されます。OVER 句に

PARTITION BY 句を指定しない場合は、**SELECT FIRST** を使用して、バイナリ複合結果を含む 1 つのタプルに結果を減らします。文節拡張 **OVER** 句に **PARTITION BY** 句を指定した場合は、**SELECT DISTINCT** を使用してパーティションごとに 1 つのタプルを残し、他のすべてのタプルを削除します。

IMSL マッピング

外部 VNI ライブラリの IMSL C 関数のパラメータへのマッピングは、**TS_AUTO_ARIMA_RESULT** のサポート・スカラ関数によって実行されます。

例

この例では、IQ は 4 つの銘柄記号の **AUTO_ARIMA** モデルをそれぞれ個別に計算します。**DISTINCT** 修飾子は、銘柄記号ごとにタプルのセットを 1 タプルに減らします。最終的に、銘柄記号と、その銘柄用に計算された **AUTO_ARIMA** モデルについて記述するすべての関連情報を含む 1 つの出力ローが返されます。

```
select stock_symbol,
       TS_AUTO_ARIMA_RESULT_RESIDUAL_SIGMA( auto_arima_res ),
       TS_AUTO_ARIMA_RESULT_AIC( auto_arima_res),
       TS_AUTO_ARIMA_RESULT_AICC( auto_arima_res),
       TS_AUTO_ARIMA_RESULT_BIC( auto_arima_res),
       TS_AUTO_ARIMA_RESULT_FORECAST_VALUE( auto_arima_res, 1),
       TS_AUTO_ARIMA_RESULT_FORECAST_ERROR( auto_arima_res, 1),
       TS_AUTO_ARIMA_RESULT_FORECAST_VALUE( auto_arima_res, 2),
       TS_AUTO_ARIMA_RESULT_FORECAST_ERROR( auto_arima_res, 2),

       TS_AUTO_ARIMA_RESULT_FORECAST_VALUE( auto_arima_res, 3),

       TS_AUTO_ARIMA_RESULT_FORECAST_ERROR( auto_arima_res, 3),

       TS_AUTO_ARIMA_RESULT_MODEL_P( auto_arima_res),
       TS_AUTO_ARIMA_RESULT_MODEL_Q( auto_arima_res),

       TS_AUTO_ARIMA_RESULT_MODEL_S( auto_arima_res),

       TS_AUTO_ARIMA_RESULT_MODEL_D( auto_arima_res)
from ( select distinct
       stock_symbol,
       TS_AUTO_ARIMA(stock_trade_time,
                     trade_price,

                     1, 3.0, 4.0, 0, 95.0, TS_INT_ARRAY(4, 0,
                     1, 0, 3))
       over (partition by stock_symbol
            order by stock_trade_time)
       as auto_arima_res
from stock_trades
where stock_symbol in ('XYZ', 'XZZ', 'ZXZ', 'ZZZ')
) as auto_arima_per_stock
```

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠

- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

- システム管理ガイド：第2巻』の「OLAP の使用」
- 『IMSL 数値関数ライブラリ・ユーザ・ガイド：第2/2巻C 統計ライブラリ

TS_AUTO_ARIMA_OUTLIER 関数 [集合]

TS_AUTO_ARIMA_OUTLIER は、入力時系列を受け入れ、乗法的季節自己回帰和分移動平均 (ARIMA) モデルのパラメータを自動的に決定します。 **TS_AUTO_ARIMA** は、ARIMA モデルを使用して入力セットを超える値を予測します。これに対して、 **TS_AUTO_ARIMA_OUTLIER** は、ARIMA モデルを使用して入力時系列内の統計的異常値を特定し、それぞれの異常値タイプを返します。

構文

```

TS_AUTO_ARIMA_OUTLIER( <
time_value
>, <
timeseries_expression
> [ , <
max_lag
> [ , <
critical
> [ , <
epsilon
> [ , <
criterion
> [ , <
confidence
> [ , <
model
> [ , <
delta
> ] ] ] ] ] )
OVER (
window-spec
)

```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **time_value** – 各入力時系列データ・ポイントの時刻値。

- **timeseries_expression** – 数値式。通常は差異を計算する時系列の要素を含むカラム名。
- **max_lag** – (オプション) AR(p) モデルに適合する場合に許容される最大遅延を表します。正の整数の定数または定数式である必要があります。デフォルトは 10。
- **critical** – (オプション) 異常値を検出するためのスレッシュホールドとして使用する臨界値。この値は 0 より大きく、倍精度浮動小数点の定数または定数式である必要があります。デフォルトは 3.0 です。
- **epsilon** – (オプション) 異常値の検出時にパラメータ推定の精度を制御する正の許容値。倍精度浮動小数点の定数または定数式である必要があります。デフォルトは 0.001 です。
- **criterion** – (オプション) 最適なモデルを選択するために使用する情報量基準。整数の定数または定数式 (値は 0、1、または 2) である必要があります。デフォルト値は 0 です。
 - 0 – (デフォルト) 赤池情報量基準 (AIC: Akaike Information Criterion)
 - 1 – 赤池補正情報量基準 (AICC: Akaike Corrected Information Criterion)
 - 2 – ベイズ情報量基準 (BIC: Bayesian Information Criterion)
- **confidence** – (オプション) 排他的間隔 (0, 100) から取得された、予測の信頼限界を計算するための信頼レベル。confidence の一般的な選択肢は、90.0、95.0、99.0 です。倍精度浮動小数点の定数または定数式である必要があります。デフォルトは 95.0 です。
- **delta** – (オプション) 一時的変化異常値の検出に使用する抑制効果パラメータ。倍精度浮動小数点の定数または定数式である必要があります。デフォルトは 0.7 です。
- **model** – (オプション) p 、 q 、 s 、 d の ARIMA 値をこの順序で含む、4 つの整数の長さのバイナリでコード化された整数配列を表します。TS_INT_ARRAY 関数で model 引数が指定されている場合、TS_AUTO_ARIMA_OUTLIER 関数は、IMSL ライブラリにある *imsls_d_auto_arima* 関数の (*IMSL_METHOD*, *int method*) 引数の method 3 (Specified ARIMA) パラメータを使用して、 p 、 q 、 s 、 d の各値を決定します。文節拡張 model パラメータが NULL の場合、TS_AUTO_ARIMA_OUTLIER 関数は自動的に ARIMA(p,0,0)X(0,d,0) モデルを計算し、指定された誤差基準を最小化します。デフォルトは NULL です。
- **window-spec** – TS_AUTO_ARIMA_OUTLIER は、ORDER BY 句を含む OVER() 句が必要な OLAP 関数です。ROWS または RANGE 指定子は、OVER() 句では使用できません。

使用法

TS_AUTO_ARIMA と TS_AUTO_ARIMA_OUTLIER の入力ほぼ同じです。ただし、TS_AUTO_ARIMA_OUTLIER はパーティション内の入力ローごとに異なる値を返

し、**TS_AUTO_ARIMA** はすべてのローに対して同じ値を返します。結果データ型と結果値の範囲が異なるため、**SELECT FIRST** または **SELECT DISTINCT** を使用して、重複する出力値を削除する必要はありません。**TS_AUTO_ARIMA_OUTLIER** には、結果を復号化するためのサポート・スカラー関数はありません。文節拡張

TS_AUTO_ARIMA と同様に、**TS_AUTO_ARIMA_OUTLIER** には **TS_INT_ARRAY** サポート・スカラー関数が必要です。**TS_INT_ARRAY** は、バイナリ複合入力を提供します。

この関数は入力タプルごとに整数値を返します。この整数値は、各タプル内の時刻／データ値の異常値のタイプを示します。時刻／データ値が異常値でない場合は **NULL** が返されます。整数値は次のとおりです。

- **0** – イノベーション異常値 (IO)
- **1** – 加法異常値 (AO)
- **2** – 水準変化 (LS)
- **3** – 一時的変化 (TC)
- **4** – 同定不能 (UI)

5つの異常値タイプの詳細については、『**IMSL 数値関数ライブラリ・ユーザ・ガイド**：第2/2巻 C 統計ライブラリ』を参照してください。

IMSL マッピング

imsls_d_auto_arima の異常値識別ロジックにマップします。

例

この例では、XYZの株価の時系列のARIMAモデルを計算し、そのモデルを使用して統計的異常値である取引を特定します。ローが異常値と見なされなかった場合、**TS_AUTO_ARIMA_OUTLIER** は **NULL** を返します。ローが異常値と見なされた場合、**TS_AUTO_ARIMA_OUTLIER** は現在のローの異常値のタイプを表す0～4の整数値を返します。

```
select stock_trade_time,
       stock_price,
       stock_trade_shares,
       TS_AUTO_ARIMA_OUTLIER(stock_trade_time,
                             stock_price)
       over (order by stock_trade_time) as outlier_type
from stock_trades
where stock_symbol = 'XYZ'
```

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠

- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

- システム管理ガイド：第2巻』の「OLAP の使用」
- 『IMSL 数値関数ライブラリ・ユーザ・ガイド：第2/2巻C 統計ライブラリ

TS_AUTO_ARIMA_RESULT_AIC 関数 [スカラ]

TS_AUTO_ARIMA 関数のサポート関数。 **TS_AUTO_ARIMA** によって生成された赤池情報量基準 (AIC) 出力パラメータを取得します。

構文

```
TS_AUTO_ARIMA_RESULT_AIC(
  auto_arima_result
)
```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **auto_arima_result** – **TS_AUTO_ARIMA** によって生成された varbinary の結果。

使用法

赤池情報量基準出力パラメータを表す倍精度浮動小数点値を返します。

IMSL マッピング

imsls_d_auto_arima の *IMSL_AIC* 引数にマップします。

例

TS_AUTO_ARIMA の例を参照してください。

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠
- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

- システム管理ガイド：第2巻』の「OLAP の使用」

- 『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ

TS_AUTO_ARIMA_RESULT_AICC [スカラ]

TS_AUTO_ARIMA のサポート関数。 **TS_AUTO_ARIMA** によって生成された補正 AIC (AICC) 出力パラメータを取得します。

構文

```
TS_AUTO_ARIMA_RESULT_AICC(  
    auto_arima_result  
)
```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **auto_arima_result** – **TS_AUTO_ARIMA** によって生成された varbinary の結果。

使用法

生成された補正済み赤池情報量基準出力パラメータの倍精度浮動小数点値を返します。

IMSL マッピング

imsls_d_auto_arima の *IMSL_AICC* 引数にマップします。

例

TS_AUTO_ARIMA の例を参照してください。

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠
- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

- システム管理ガイド：第 2 巻』の「OLAP の使用」
- 『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ

TS_AUTO_ARIMA_RESULT_BIC 関数 [スカラ]

TS_AUTO_ARIMA のサポート関数。 **TS_AUTO_ARIMA** によって生成されたベイズ情報量基準 (BIC) 出力パラメータを取得します。

構文

```
TS_AUTO_ARIMA_RESULT_BIC(
  auto_arima_result
)
```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **auto_arima_result** – **TS_AUTO_ARIMA** によって生成された varbinary の結果。

使用法

生成されたベイズ情報量基準出力パラメータの倍精度浮動小数点値を返します。

IMSL マッピング

imsls_d_auto_arima の *IMSL_BIC* 引数にマップします。

例

TS_AUTO_ARIMA の例を参照してください。

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠
- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

- システム管理ガイド：第2巻』の「OLAP の使用」
- 『IMSL 数値関数ライブラリ・ユーザ・ガイド：第2/2巻C 統計ライブラリ

TS_AUTO_ARIMA_RESULT_FORECAST_ERROR 関数 [スカラ]

TS_AUTO_ARIMA のサポート関数。 **TS_AUTO_ARIMA** によって生成された、元の入力系列の予測標準誤差値を取得します。

構文

```
TS_AUTO_ARIMA_RESULT_FORECAST_ERROR(
    auto_arima_result
    ,
    forecast_element_number
)
```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **auto_arima_result** – **TS_AUTO_ARIMA** によって生成された varbinary の結果。
- **forecast_element_number** – 整数定数式の値。許容範囲は 1 から n_predictions までです。

使用法

指定された予測要素の標準誤差の倍精度浮動小数点値を返します。この予測誤差値の暗黙的な時刻値は、**TS_AUTO_ARIMA** に渡される入力 *time_values* と、指定された *forecast_element_number* の最後の時刻値です。

IMSL マッピング

imsls_d_auto_arima の *IMSLS_OUTLIER_FORECAST* 引数にマップします。

例

TS_AUTO_ARIMA の例を参照してください。

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠
- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

- システム管理ガイド：第 2 巻』の「OLAP の使用」

- 『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ

TS_AUTO_ARIMA_RESULT_FORECAST_VALUE 関数 [スカラ]

TS_AUTO_ARIMA のサポート関数。**TS_AUTO_ARIMA** によって生成された、要求された異常値のない系列の予測値を取得します。

構文

```
TS_AUTO_ARIMA_RESULT_FORECAST_VALUE(
    auto_arima_result
    ,
    model_element_number
)
```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- auto_arima_result** – **TS_AUTO_ARIMA** によって生成された varbinary の結果。
- model_element_number** – 整数定数式の値。許容範囲は 1 から n_predictions までです。

使用法

元の入力系列の指定された予測値を表す倍精度浮動小数点値を返します。この予測値の暗黙的な時刻値は、**TS_AUTO_ARIMA** に渡される入力 *time_values* と、指定された *forecast_element_number* の最後の時刻値です。

IMSL マッピング

imsls_d_auto_arima の *IMSL_OUTLIER_FORECAST* 引数にマップします。

例

TS_AUTO_ARIMA 関数の例を参照してください。

標準と互換性

- SQL** – ISO/ANSI SQL 準拠
- Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

- システム管理ガイド：第2巻』の「OLAPの使用」
- 『IMSL 数値関数ライブラリ・ユーザ・ガイド：第2/2巻C統計ライブラリ

TS_AUTO_ARIMA_RESULT_MODEL_P 関数 [スカラ]

TS_AUTO_ARIMA のサポート関数。ARIMA モデル記述の計算時に、**TS_AUTO_ARIMA** によって生成された p 値を取得します。

構文

```
TS_AUTO_ARIMA_RESULT_MODEL_P(  
  auto_arima_result  
)
```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **auto_arima_result** – **TS_AUTO_ARIMA** によって生成された varbinary の結果。

使用法

TS_AUTO_ARIMA によって生成された倍精度浮動小数点 p 出力パラメータを返します。

IMSL マッピング

imsls_d_auto_arima の *IMSL_MODEL* 引数の最初の要素にマップします。

例

TS_AUTO_ARIMA の例を参照してください。

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠
- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

- システム管理ガイド：第2巻』の「OLAPの使用」
- 『IMSL 数値関数ライブラリ・ユーザ・ガイド：第2/2巻C統計ライブラリ

TS_AUTO_ARIMA_RESULT_MODEL_Q 関数 [スカラ]

TS_AUTO_ARIMA のサポート関数。ARIMA モデル記述の計算時に、**TS_AUTO_ARIMA** によって生成された q 値を取得します。

構文

```
TS_AUTO_ARIMA_RESULT_MODEL_Q
(auto_arima_result_
)
```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **auto_arima_result** – **TS_AUTO_ARIMA** によって生成された varbinary の結果。

使用法

TS_AUTO_ARIMA によって生成された倍精度浮動小数点 q 出力パラメータを返します。

IMSL マッピング

imsls_d_auto_arima の *IMSL_MODEL* 引数の 2 番目の要素にマップします。

例

TS_AUTO_ARIMA の例を参照してください。

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠
- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

- システム管理ガイド：第 2 巻』の「OLAP の使用」
- 『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ

TS_AUTO_ARIMA_RESULT_MODEL_S 関数 [スカラ]

TS_AUTO_ARIMA のサポート関数。ARIMA モデル記述の計算時に、**TS_AUTO_ARIMA** によって生成された *s* 値を取得します。

構文

```
TS_AUTO_ARIMA_RESULT_MODEL_S(  
  auto_arima_result  
)
```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **auto_arima_result** – **TS_AUTO_ARIMA** によって生成された varbinary の結果。

使用法

TS_AUTO_ARIMA によって生成された倍精度浮動小数点 *s* 出力パラメータを返します。

IMSL マッピング

imsls_d_auto_arima の *IMSL_MODEL* 引数の 3 番目の要素にマップします。

例

TS_AUTO_ARIMA の例を参照してください。

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠
- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

- システム管理ガイド：第 2 巻』の「OLAP の使用」
- 『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ

TS_AUTO_ARIMA_RESULT_MODEL_D 関数 [スカラ]

TS_AUTO_ARIMA 関数のサポート関数。ARIMA モデル記述の計算時に、**TS_AUTO_ARIMA** によって生成された *d* 値を取得します。

構文

```
TS_AUTO_ARIMA_RESULT_MODEL_D(
  auto_arima_result
)
```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **auto_arima_result** – **TS_AUTO_ARIMA** によって生成された varbinary の結果。

使用法

TS_AUTO_ARIMA によって生成された倍精度浮動小数点 *s* 出力パラメータを返します。

IMSL マッピング

imsls_d_auto_arima の *IMSL_MODEL* 引数の 4 番目の要素にマップします。

例

TS_AUTO_ARIMA の例を参照してください。

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠
- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

- システム管理ガイド：第 2 巻』の「OLAP の使用」
- 『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ

TS_AUTO_ARIMA_RESULT_RESIDUAL_SIGMA 関数 [スカラ]

TS_AUTO_ARIMA のサポート関数。異常値のないデータ・ポイントの残差標準誤差を取得します。

構文

```
TS_AUTO_ARIMA_RESULT_RESIDUAL_SIGMA (  
    auto_arma_result  
)
```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **auto_arma_result** – **TS_AUTO_ARIMA** によって生成された varbinary の結果。

使用法

残差標準誤差出力パラメータを返します。

IMSL マッピング

imsls_d_auto_arma の *IMSL_RESIDUAL_SIGMA* 引数にマップします。

例

TS_AUTO_ARIMA の例を参照してください。

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠
- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

- システム管理ガイド：第2巻』の「OLAP の使用」
- 『IMSL 数値関数ライブラリ・ユーザ・ガイド：第2/2巻 C 統計ライブラリ

TS_AUTO_UNI_AR 関数 [集合]

単変量自己回帰時系列モデルの自動選択と適合を実行します。

構文

```
TS_AUTO_UNI_AR (timeseries_expression  
 , ar_count, ar_elem, method)
```

```
OVER (window-spec)
```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **timeseries_expression** – 数値式。通常は時系列の要素を含むカラム名。
- **ar_count** – 計算する自己回帰値の数を含む整数。
- **ar_elem** – 返す計算済み自己回帰値を示す整数。この整数は 0 より大きく、*ar_count* 以下である必要があります。
- **method** – (オプション) AR 係数の計算時に使用する方法を示す整数。0 = モーメント法、1 = 最小 2 乗法 (デフォルト値)、2 = 最尤法
- **window-spec** – **TS_AUTO_UNI_AR** は、**OVER ()** 句が必要な OLAP 関数です。

使用法

この時系列関数は、自己回帰推定を含む倍精度浮動小数点数を返します。

TS_AUTO_UNI_AR は、IMSL ライブラリの関数 **imsls_d_auto_uni_ar** を呼び出します。

IMSL マッピング

TS_AUTO_UNI_AR の引数は、次のように IMSL ライブラリの関数 **imsls_d_auto_uni_ar** にマップします。

```
params = imsls_d_auto_uni_ar (n_objs, z[], maxlag, p, method, 0);
```

- **n_objs** – 現在のウィンドウ・フレームのローの数を含みます。
- **z[]** – 現在のウィンドウ・フレームの *timeseries_expression* の値を含みます。
- **maxlag** – ユーザ定義集合関数の引数 *ar_count* にマップします。
- **p** – 最小 AIC を使用するモデルの自己回帰パラメータの数を表す出力パラメータ。
- **method** – ユーザ定義集合関数の引数 *method* にマップします。 *ar_elem* が *p* より大きく、IMSL ライブラリの時系列関数のエラー処理値が 0 に設定されている場合、Sybase IQ は null を返します。IMSL ライブラリの時系列関数のエラー

処理値が 0 以外の値に設定されている場合、Sybase IQ は、ar_elem が p より大きいことを示すエラー・メッセージを表示します。

関数 `imsls_d_auto_uni_ar` が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

例

この例では、`TS_AUTO_UNI_AR` 関数を含む SQL 文と、この関数から返されるデータ値を示します。この例では、サンプル入力データ・テーブル (`DATASET`) を入力データとして使用します。`DATASET` サンプル入力データ (80 ページ) を参照してください。

次の SQL 文では、`data` カラムの 2 つの要素を含む配列の最初の要素を返します。

```
SELECT TS_AUTO_UNI_AR(data,2,1,0) OVER (ORDER BY ROWNUM rows BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS res FROM DATASET
```

Sybase IQ は 50 個のローを返します。各ローには同じ値が含まれています。

表 13 : `TS_AUTO_UNI_AR` から返される値

res
0.883453
0.883453
0.883453
0.883453
0.883453
0.883453
0.883453
0.883453
0.883453
0.883453
...
0.883453

標準と互換性

- SQL – ISO/ANSI SQL 準拠

- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

システム管理ガイド：第2巻』の「OLAPの使用」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第2/2巻C 統計ライブラリ

TS_BOX_COX_XFORM 関数 [集合]

前方または逆 Box-Cox ベキ変換を実行します。

構文

```
TS_BOX_COX_XFORM (timeseries_expression
, power
[, shift [, inverse] ]) OVER (window-spec)
```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **timeseries_expression** – 数値式。通常は時系列の要素を含むカラム名。
- **power** – Box-Cox ベキ変換の指数パラメータを表す倍精度浮動小数点値。
- **shift** – (オプション) シフト・パラメータを表す倍精度浮動小数点値。この値は、 $\min(\text{timeseries}) + \text{shift} > 0$ という関係を満たす必要があります。shift のデフォルトは 0.0 です。
- **inverse** – (オプション) tinyint 値。1 に設定すると、Sybase IQ は逆変換を実行します。0 または null の場合、Sybase IQ は前方変換を実行します。デフォルト値は 0 です。
- **window-spec** – **TS_BOX_COX_XFORM** は、無制限ウィンドウを指定した **OVER ()** 句が必要な OLAP 関数です。**TS_BOX_COX_XFORM** は値ベースのウィンドウをサポートしません。たとえば、**OVER ()** 句で範囲指定子を使用できません。

使用法

TS_BOX_COX_XFORM は、時系列の各要素の対応する計算済み変換値を返します。この関数は、IMSL ライブラリの関数 **imsls_d_box_cox_transform** を呼び出します。

IMSL マッピング

TS_BOX_COX_XFORM の引数は、次のように IMSL ライブラリの関数 **imsls_d_box_cox_transform** にマップします。

```
params = imsls_d_box_cox_transform(n_objs, z[], power,
    IMSLS_SHIFT, shift [, IMSLS_INVERSE], 0);
```

- **n_objs** – 現在のウィンドウ・フレームのローの数を含みます。
- **z[]** – 現在のウィンドウ・フレームの *timeseries_expression* の値を含みます。
- **power** – ユーザ定義集合関数の引数 *power* にマップします。
- **shift** – ユーザ定義集合関数の引数 *shift* にマップします。
- **IMSL_INVERSE** – ユーザ定義集合関数の引数 *inverse* が 1 の場合、Sybase IQ は IMSLS_INVERSE で Box-Cox 変換を呼び出します。それ以外の場合は、この引数は関数呼び出しから除外されます。

関数 **imsls_d_box_cox_transform** が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

例

この例では、入力データ・テーブル、**TS_BOX_COX_XFORM** 関数を含む SQL 文、およびこの関数から返されるデータ値を示します。この例では、次のテーブル (**BOX_COX_XFORM_DATASET**) を入力データとして使用します。

BOX_COX_XFORM_DATASET テーブルには、時系列データの 13 のローが含まれています。

表 14：入力データ・テーブル **BOX_COX_XFORM_DATASET**

ロー番号	データ
1	7
2	26
3	6
4	60
5	78.5
6	1
7	29
8	15
9	52
10	74.3
11	11

ロー番号	データ
12	56
13	8

次の SQL 文では、*data* カラムの Box-Cox ベキ変換を返します。

```
SELECT TS_BOX_COX_XFORM(data,1.0,1.0,0) OVER (ORDER BY rownum ROWS
BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS res FROM
BOX_COX_XFORM_DATASET
```

Sybase IQ は次の 13 個のローを返します。

表 15 : TS_BOX_COX_XFORM から返される値

res
8
27
7
61
79.5
2
30
16
53
75.3
12
57
9

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠
- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

システム管理ガイド：第 2 巻』の「OLAP の使用」

TS_DIFFERENCE 関数 [集合]

季節時系列と非季節時系列の差異を計算します。

構文

```

TS_DIFFERENCE (timeseries_expression
, period1
[, period2 [, ...period 10] ]) OVER (window-spec)

```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **timeseries_expression** – 数値式。通常は差異を計算する時系列の要素を含むカラム名。
- **period1 ... period10** – 各間隔は、時系列の差異を計算する間隔を含む整数式です。少なくとも 1 つの間隔を指定する必要があり、最大 10 の間隔を指定できます。
- **window-spec** – **TS_DIFFERENCE** は、無制限ウィンドウを指定した **OVER ()** 句が必要な OLAP 関数です。文節拡張この関数は値ベースのウィンドウをサポートしません。たとえば、**OVER ()** 句では範囲指定子を使用できません。

使用法

時系列の各要素に対して、**TS_DIFFERENCE** は、時系列の対応する計算済み差異値を返します。この関数は、IMSL ライブラリの関数 **imsls_d_difference** を呼び出します。

IMSL マッピング

TS_DIFFERENCE の引数は、次のように IMSL ライブラリの関数 **imsls_d_difference** にマップします。

```

params = imsls_d_difference(n_objs, z[], n_differences,
periods [], 0);

```

- **n_objs** – 現在のウィンドウ・フレームのローの数を含みます。
- **z[]** – 現在のウィンドウ・フレームの *timeseries_expression* の値を含みます。
- **n_differences** – **TS_DIFFERENCE** で定義された *period* 引数にマップします。
- **period** – **TS_DIFFERENCE** で定義された *period* 引数の配列。

関数 `imsls_d_difference` が時系列計算を実行する方法の詳細については、『IMSLS 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

例

この例では、**TS_DIFFERENCE** 関数を含む SQL 文と、この関数から返されるデータ値を示します。この例では、サンプル入力データ・テーブル (*DATASET*) を入力データとして使用します。*DATASET* サンプル入力データ (80 ページ) を参照してください。

次の SQL 文では、*data* カラムのデータの差異を計算します。

```
SELECT TS_DIFFERENCE(data,1) OVER (ORDER BY ROWNUM rows BETWEEN
UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS res FROM DATASET
```

Sybase IQ は 50 個のローを返します。

表 16 : **TS_DIFFERENCE** から返される値

res
NULL
0.170336
0.191027
1.29692
0.801743
-0.038988
-0.09424
1.61886
-1.12477
1.02925
-1.20614
-0.814478
-0.157049
-1.18213
0.027546
1.45734

res
-0.990302
-0.833325
-0.637229
-0.08655
-0.12594
-0.122914
-1.39596
0.919785
-0.449474
0.037273
-0.954345
-0.562983
1.98379
0.88304
-0.345265
0.934656
0.069088
-0.249428
0.795766
-1.8145
1.27016
1.39266
-0.141794
0.934752
0.982506
0.330772
-1.34311

res
1.23124
0.209869
0.791146
-0.259155
0.15124
-0.963484
0.383186

注意：IMSL ライブラリが結果の最初のローに *not a number* (NaN) 値を返したため、最初のローは NULL です。

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠
- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

システム管理ガイド：第2巻』の「OLAPの使用」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第2/2巻 C 統計ライブラリ

TS_DOUBLE_ARRAY 関数 [スカラ]

TS_GARCH のサポート関数。3個から10個の定数の倍精度浮動小数点値を含む論理配列を作成し、単一の varbinary 値を返します。

構文

```
TS_DOUBLE_ARRAY(xguess1, xguess2, xguess3, [ ... [ , xguess10] ... ] )
```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **xguess** – 定数の倍精度浮動小数点値のセット。 p および q の **TS_GARCH** 値に応じて、`varbinary` でコード化された結果の論理配列の値は 3 ~ 10 個になります。

使用法

TS_GARCH をサポートするスカラー関数。 **TS_GARCH** `xguess_binary_encoding` パラメータの `xguess_array` を生成します。

IMSL マッピング

`imsls_d_garch` の `float xguess[]` 引数にマップします。

例

TS_GARCH の例を参照してください。

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠
- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

- システム管理ガイド：第 2 巻』の「OLAP の使用」
- 『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ

TS_ESTIMATE_MISSING 関数 [集合]

時系列内の欠落値を推定し、元の時系列に組み込み、新しい時系列として返します。

構文

```
TS_ESTIMATE_MISSING (timeseries_expression  
 , method)
```

```
OVER (window-spec)
```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **timeseries_expression** – 数値式。通常は差異を計算する時系列の要素を含むコラム名。NULL 以外の値を指定した場合、時系列のギャップ (関数によって計算される値) が反映されます。
- **method** – (オプション) 欠損値を判別するときに使用する方法を指定する整数。
 - 0 (デフォルト) – ギャップの前の最後の 4 個の時系列値とギャップの後の最初の 4 個の値の中央値によって、ギャップ内で失われた時系列観測値を推定します。
 - 1 – 3 次スプライン補間法を使用して欠損値を推定します。この場合も、ギャップの前の最後の 4 個の時系列値とギャップの後の最初の 4 個の値に対して補間が行われます。
 - 2 – ギャップの前の時系列を AR(1) 過程によって適切に説明できることを前提としています。
 - 3 – AR(p) モデルを使用して、短期予測によって欠損値を推定します。
- **window-spec** – **TS_ESTIMATE_MISSING** は、無制限ウィンドウを指定した **OVER ()** 句が必要な OLAP 関数です。この関数は値ベースのウィンドウをサポートしません。たとえば、**OVER ()** 句では範囲指定子を使用できません。

使用法

TS_ESTIMATE_MISSING を使用して、4 つの推定法のいずれかによって欠損している等間隔時間ポイントを推定します。 **TS_ESTIMATE_MISSING** は、IMSL ライブラリの関数 **imsls_d_estimate_missing** を呼び出します。

時間ポイントのセットに 2 つ以上の連続した NULL 値が存在する場合は、**TS_ESTIMATE_MISSING** を使用できません。時間ポイント・セットの最初または最後の 2 つの値が NULL の場合、NULL が返されます。

IMSL マッピング

TS_ESTIMATE_MISSING の引数は、次のように IMSL ライブラリの関数 **imsls_d_estimate_missing** にマップします。

```
params = imsls_d_estimate_missing(n_objs, tpoints[], z[], method, 0);
```

- **n_objs** – 現在のウィンドウ・フレームのローの数を含みます。
- **tpoints** – 時間ポイントのシーケンス内の欠損値を示すインデックスの配列。
- **z[]** – **next_value** の呼び出し時に取得される、累積された **timeseries_expression**。
- **method** – **TS_ESTIMATE_MISSING** で定義された **method** 引数にマップします。

関数 `imsls_d_estimate_missing` が時系列計算を実行する方法の詳細については、『IMSLS 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

例

この例は、入力データ・テーブル、`TS_ESTIMATE_MISSING` 関数を含む SQL 文、および関数から返されるデータ値を示します。この例では、次のテーブル (`EST_MISSING_DATASET`) を入力データとして使用します。`EST_MISSING_DATASET` テーブルには、時系列データの 9 つのローが含まれています。

表 17：入力データ・テーブル `EST_MISSING_DATASET`

ロー番号	データ
1	2.8223
2	-0.5721
3	2.2771
4	NULL
5	1.2648
6	1.0278
7	0.6991
8	-1.7539
9	-2.8875

次の SQL 文では、4 番目のローから欠損しているデータの値を推定します。

```
SELECT ts_estimate_missing(data,0) OVER (order by rownum rows
between unbounded preceding and unbounded following) AS res FROM
EST_MISSING_DATASET
```

Sybase IQ は次の 9 つのローを返し、NULL 値を 1.0278 で置き換えます。

表 18：`TS_ESTIMATE_MISSING` から返される値

res
2.8223
-0.5721
2.2771

res
1.0278
1.2648
1.0278
0.6991
-1.7539
-2.8875

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠
- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

システム管理ガイド：第2巻』の「OLAPの使用」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第2/2巻C統計ライブラリ

TS_GARCH 関数 [集合]

時系列データの変動を分析および予測するときに使用します。**TS_GARCH** は、GARCH(p, q) モデルのパラメータの推定値を計算します。GARCH (一般化自己回帰条件付き分散不均一) は、ARCH の一般化モデルです。ARCH 計算では、誤差分散を前の期間の誤差の2乗に関連付けます。

構文

```

TS_GARCH (
  timeseries
  ,
  garch_count
  ,
  arch_count
  ,
  xguess_binary_encoding
  [ , <
  max_sigma
  > ] )
OVER (
  window-spec
)

```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **timeseries_expression** – 数値式。通常は差異を計算する時系列の要素を含むカラム名。
- **garch_count** – GARCH(p,q) 計算の p GARCH パラメータの定数整数の数。
- **arch_count** – GARCH(p,q) 計算の q ARCH パラメータの定数整数の数。GARCH カウントと ARCH カウントの合計が 9 を超えることはできません。
- **xguess_binary_encoding** – **TS_DOUBLE_ARRAY** スカラ関数の呼び出しによって生成された定数バイナリ式としてコード化された、モデル決定のためのシード値 (*xguess array*) を表します。配列の長さは、 $p + q + 1$ です。この配列には、**TS_GARCH_RESULT_USER** で使用される x パラメータの初期値が含まれます。最初の要素 (シグマ 2 乗値のシード値) は、**max_sigma** の値より小さい 0 以外の正の倍精度浮動小数点値である必要があります。残りの p および q シード値は、0 以上の倍精度浮動小数点値である必要があります。これらの合計が 1.0 未満でなければなりません。
- **max_sigma** – (オプション) シグマ 2 乗値における定数の倍精度浮動小数点上限値。正の値である必要があります。デフォルトは 10.0 です。
- **window-spec** – **TS_GARCH** は、**ORDER BY** 句を含む **OVER()** 句が必要な OLAP 関数です。**ROWS** または **RANGE** 指定子は、**OVER()** 句では使用できません。文節拡張

使用法

OLAP スタイルの集合関数である **TS_GARCH** は、1 つの SQL 結果 (特別にコード化された可変長のバイナリ結果値) を生成します。サポート・スカラ関数は、バイナリ複合出力値を受け入れ、その出力値から個々のスカラ結果値を返します。

OLAP スタイルの集合関数は入力タプルごとに 1 つの結果値を返すため、パーティション内の各ローに対して同じバイナリ複合結果が返されます。**OVER** 句に **PARTITION BY** 句を指定しない場合は、**SELECT FIRST** を使用して、バイナリ複合結果を含む 1 つのタプルに結果を減らします。文節拡張 **OVER** に **PARTITION BY** 句を指定した場合は、**SELECT DISTINCT** を使用してパーティションごとに 1 つのタプルを残し、他のすべてのタプルを削除します。

IMSL マッピング

外部 VNI ライブラリの IMSL C 関数のパラメータへのマッピングは、**TS_GARCH_RESULT** のサポート・スカラ関数によって実行されます。

例

この例では、指定した4社の株価のGARCH(1,2)モデルをそれぞれ個別に計算します。次に、クエリでDISTINCT修飾子を使用して、銘柄記号ごとにタプルのセットを1タプルに減らします。最終的に、銘柄記号と、その銘柄用に計算されたGARCH(1,2)モデルについて記述するすべての関連情報を含む1つの出力ローが返されます。

TS_GARCH 関数への2番目と3番目の引数は、GARCH(p,q)タイプのモデルを使用することを示す p 値と q 値です。これらの値は、正の整数の定数または定数式である必要があります。

4番目の引数は、バイナリ定数式としてコード化される GARCH(p,q) モデルを決定するためのシード値のセットです。これは、サポート・スカラ関数

TS_DOUBLE_ARRAY によって生成される必要があります。

```
select stock_symbol,
       TS_GARCH_RESULT_A( garch_res ) as log_likelihood,
       TS_GARCH_RESULT_AIC( garch_res ) as akaike_info,
       TS_GARCH_RESULT_USER( garch_res, 1) as sigma_squared,
       TS_GARCH_RESULT_USER( garch_res, 2) as q_1,
       TS_GARCH_RESULT_USER( garch_res, 3) as p_1,
       TS_GARCH_RESULT_USER( garch_res, 4) as p_2
from ( select distinct
       stock_symbol,
       TS_GARCH(stock_price, 1, 2,
                TS_DOUBLE_ARRAY(1.2, 0.3, 0.2, 0.3), 4)
       over (partition by stock_symbol
            order by stock_trade_time) as garch_res
       from stock_trades
       where stock_symbol in ('XYZ', 'XZZ', 'ZXX', 'ZZZ')
       as dtl
```

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠
- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

- システム管理ガイド：第2巻』の「OLAPの使用」
- 『IMSL 数値関数ライブラリ・ユーザ・ガイド：第2/2巻 C 統計ライブラリ

TS_GARCH_RESULT_A 関数 [スカラ]

TS_GARCH のサポート関数。 **TS_GARCH** 集合関数によって生成された、対数尤度出力パラメータ *A* を取得します。

構文

```
TS_GARCH_RESULT_A (  
  ts_garch_result  
)
```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **ts_garch_result** – **TS_GARCH** 集合関数の呼び出しによって生成された varbinary の結果引数。

使用法

生成された対数尤度出力パラメータの倍精度浮動小数点値を返します。

IMSL マッピング

imsls_d_garch の *IMSLA_A*, *float *a*, 引数にマップします。

例

TS_GARCH の例を参照してください。

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠
- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

- システム管理ガイド：第2巻』の「OLAP の使用」
- 『IMSL 数値関数ライブラリ・ユーザ・ガイド：第2/2巻C 統計ライブラリ

TS_GARCH_RESULT_AIC 関数 [スカラ]

TS_GARCH のサポート関数。**TS_GARCH** 集合関数によって生成された、赤池情報量基準出力パラメータ *AIC* を取得します。

構文

```
TS_GARCH_RESULT_AIC (
  ts_garch_result
)
```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **ts_garch_result** – **TS_GARCH** 集合関数の呼び出しによって生成された varbinary の結果引数。

使用法

生成された赤池情報量基準出力パラメータの倍精度浮動小数点値を返します。

IMSL マッピング

imsls_d_garch の *IMSL_AIC*, *float *aic*, 引数にマップします。

例

TS_GARCH の例を参照してください。

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠
- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

- システム管理ガイド：第2巻』の「OLAP の使用」
- 『IMSL 数値関数ライブラリ・ユーザ・ガイド：第2/2巻 C 統計ライブラリ

TS_GARCH_RESULT_USER [スカラ]

TS_GARCH のサポート関数。GARCH(p,q) モデルを記述する論理配列内の各要素にアクセスします。

構文

```
TS_GARCH_RESULT_USER (
  ts_garch_result
,
  model_element_number
)
```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **ts_garch_result** – **TS_GARCH** 集合関数の呼び出しによって生成された varbinary の結果引数。
- **model_element_number** – 整数定数式の値。値の範囲は、1 から $(1+p+q)$ までです。

使用法

指定されたモデル記述出力値の倍精度浮動小数点値を返します。出力セットで返される要素の数は、 $p+q+1$ です。出力セットの最初の要素は、結果のシグマ2乗値です。次の q 値は、計算済みの ARCH パラメータです。最後の p 値は、決定された GARCH パラメータです。

出力セットのサイズは可変ですが、 p 値と q 値を定数の入力引数として **TS_GARCH** に渡す必要があるため、特定の **TS_GARCH** 呼び出しに対して、モデル記述値の数は固定されます。

IMSL マッピング

imsls_d_garch の **IMSL_RETURN_USER**, *float x[]*, 引数にマップします。文節拡張

例

TS_GARCH の例を参照してください。

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠

- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

- システム管理ガイド：第2巻』の「OLAPの使用」
- 『IMSL 数値関数ライブラリ・ユーザ・ガイド：第2/2巻C 統計ライブラリ

TS_INT_ARRAY 関数 [スカラ]

TS_AUTO_ARIMA と TS_AUTO_ARIMA_OUTLIER のサポート関数。

構文

```
TS_INT_ARRAY(
  int1
  ,
  int2
  ,
  int3
  ,
  int4
  , [... [,
    int10
  ] ... ])
```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **int1 ... int10** – 定数整数値のセット。TS_AUTO_ARIMA の *model* パラメータが指定されている場合は、4つの整数を指定します。セットの最大値は10個の整数です。

注意： 整数値は必須です。この関数に整数以外の値を渡すと、予期しない結果が生じる可能性があります。

使用法

値の論理配列として指定された整数入力要素をコード化する varbinary 値を返します。

IMSL マッピング

4つの整数を指定し、結果を TS_AUTO_ARIMA または TS_AUTO_ARIMA_OUTLIER に渡すと、TS_INT_ARRAY は IMSL ライブラリにある *imsls_d_auto_arima* の (*IMSL_METHOD*, *int method*) 入力引数の *method 3* にマップします。

例

TS_AUTO_ARIMA の例を参照してください。

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠
- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

- システム管理ガイド：第2巻』の「OLAP の使用」
- 『TMSL 数値関数ライブラリ・ユーザ・ガイド：第2/2巻C 統計ライブラリ

TS_LACK_OF_FIT 関数 [集合]

適切な相関関数を前提として単変量時系列または伝達関数の適合性不足テスト (LOF) を実行します。

構文

```
TS_LACK_OF_FIT (timeseries_expression  
    , p_value, q_value, lagmax,  
    [tolerance])
```

```
OVER (window-spec)
```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **timeseries_expression** – 数値式。通常は時系列の要素を含むカラム名。
- **p_value** – 自己回帰パラメータの数を含む整数。
- **q_value** – 移動平均パラメータの数を含む整数。
- **lagmax** – 相関関数の最大遅延を含む整数。
- **tolerance** – (オプション) 非リニアの最小2乗アルゴリズムの収束を判別するために使用する浮動小数点値レベル。デフォルト値は0です。
- **window-spec** – **TS_LACK_OF_FIT** は、**OVER ()** 句が必要な OLAP 関数です。

使用法

この関数は、時系列の適合性不足統計 (q) を含む倍精度浮動小数点値を返します。**TS_LACK_OF_FIT** は、IMSL ライブラリの関数 **imsls_d_lack_of_fit** を呼び出します。

IMSL マッピング

TS_LACK_OF_FIT の引数は、次のように IMSL ライブラリの関数 **imsls_d_lack_of_fit** にマップします。

```
params = imsls_d_arma(n_objs, z[], p, q,      IMSLS_LEAST_SQUARES,
                    IMSLS_CONVERGENCE_TOLERANCE, tolerance,      IMSL_RESIDUAL, &residual,
                    0); correlations = imsls_d_autocorrelation(n_objs-p+lagmax,
                    residuals, lagmax, 0); result = imsls_d_lack_of_fit(n_objs,
                    correlations, lagmax, npfree, 0);
```

- **n_objs** – 現在のウィンドウ・フレームのローの数を含みます。
- **z[]** – 現在のウィンドウ・フレームの *timeseries_expression* の値を含みます。
- **p** – **TS_LACK_OF_FIT** で定義された *p_value* 引数にマップします。
- **q** – **TS_LACK_OF_FIT** で定義された *q_value* 引数にマップします。
- **lagmax** – **TS_LACK_OF_FIT** で定義された *lagmax* 引数にマップします。
- **npfree** – p+q から派生します。
- **tolerance** – **IMSL_CONVERGENCE_TOLERANCE** を使用するオプション引数。null の場合、IMSL ライブラリはデフォルト値を適用し、**IMSL_CONVERGENCE_TOLERANCE** は使用されません。

IMSL 関数 **imsls_d_lack_of_fit** が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

例

この例では、**TS_LACK_OF_FIT** 関数を含む SQL 文と、この関数から返されるデータ値を示します。この例では、サンプル入力データ・テーブル (*DATASET*) を入力データとして使用します。DATASET サンプル入力データ (80 ページ) を参照してください。

次の SQL 文では、*data* カラムのデータの適合性不足統計を返します。

```
select ts_lack_of_fit(data,1,1,5,0.225) over (order by rownum rows
between unbounded preceding and unbounded following) as res from
DATASET
```

Sybase IQ は 50 個のローを返します。各ローには同じ値が含まれています。

表 19 : TS_LACK_OF_FIT から返される値

res
3.96751
3.96751
3.96751
3.96751
3.96751
3.96751
3.96751
3.96751
3.96751
3.96751
3.96751
...
3.96751

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠
- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

システム管理ガイド：第 2 巻』の「OLAP の使用」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ

TS_LACK_OF_FIT_P 関数 [集合]

単変量時系列に対して適合性不足テストを実行します。TS_LACK_OF_FIT_P 関数は、q を返すのではなく q の p 値を返す点を除き、TS_LACK_OF_FIT と同じです。

構文

```

TS_LACK_OF_FIT_P (timeseries_expression
, p_value, q_value, lagmax, [tolerance])

OVER (window-spec)
    
```


ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **timeseries_expression** – 数値式。通常は時系列の要素を含むカラム名。
- **p_value** – 自己回帰パラメータの数を含む整数。
- **q_value** – 移動平均パラメータの数を含む整数。
- **lagmax** – 相関関数の最大遅延を含む整数。
- **tolerance** – (オプション) 非リニアの最小 2 乗アルゴリズムの収束を判別するために使用する浮動小数点値レベル。デフォルト値は 0 です。
- **window-spec** – **TS_LACK_OF_FIT_P** は、**OVER ()** 句が必要な OLAP 関数です。

使用法

この関数は、時系列の適合性不足統計 (q) の p 値を含む倍精度浮動小数点値を返します。**TS_LACK_OF_FIT_P** は、IMSL ライブラリの関数 **imsls_d_lack_of_fit** を呼び出します。

IMSL マッピング

TS_LACK_OF_FIT_P の引数は、次のように IMSL ライブラリの関数 **imsls_d_lack_of_fit** にマップします。

```
params = imsls_d_arma(n_objs, z[], p, q, IMSLS_LEAST_SQUARES,
IMSL_CONV_TOL, tolerance, IMSL_RESIDUAL, &residual,
0);correlations = imsls_d_autocorrelation(n_objs-p+lagmax,
residuals, lagmax, 0);result = imsls_d_lack_of_fit(n_objs,
correlations, lagmax, npfree, 0);
```

- **n_objs** – 現在のウィンドウ・フレームのローの数を含みます。
- **z[]** – 現在のウィンドウ・フレームの *timeseries_expression* の値を含みます。
- **p** – **TS_LACK_OF_FIT_P** で定義された *p_value* 引数にマップします。
- **q** – **TS_LACK_OF_FIT_P** で定義された *q_value* 引数にマップします。
- **lagmax** – **TS_LACK_OF_FIT_P** で定義された *lagmax* 引数にマップします。
- **npfree** – $p+q$ から派生します。
- **tolerance** – **IMSL_CONV_TOL** を使用するオプション引数。
null の場合、IMSL ライブラリはデフォルト値を適用し、
IMSL_CONV_TOL は使用されません。

IMSL 関数 **imsls_d_lack_of_fit** が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

例

この例では、**TS_LACK_OF_FIT_P** 関数を含む SQL 文と、この関数から返されるデータ値を示します。この例では、サンプル入力データ・テーブル (*DATASET*) を入力データとして使用します。*DATASET* サンプル入力データ (80 ページ) を参照してください。

次の SQL 文では、*data* カラムのデータの適合性不足統計の *p* 値を返します。

```
select ts_lack_of_fit_p(data,1,1,5,0.225) over (order by rownum rows
between unbounded preceding and unbounded following) as res FROM
DATASET
```

Sybase IQ は 50 個のローを返します。各ローには同じ値が含まれています。

表 20 : TS_LACK_OF_FIT_P から返される値

res
0.735006
0.735006
0.735006
0.735006
0.735006
0.735006
0.735006
0.735006
0.735006
0.735006
0.735006
...
0.735006

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠
- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

システム管理ガイド：第 2 巻』の「OLAP の使用」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』

TS_MAX_ARMA_AR 関数 [集合]

単変量 ARMA (自己回帰移動平均) 時系列モデルにある引数の正確な最尤推定を計算し、要求された自己回帰推定を返します。

構文

```
TS_MAX_ARMA_AR (timeseries_expression
, ar_count, ar_elem)
```

```
OVER (window-spec)
```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **timeseries_expression** – 数値式。通常は時系列の要素を含むカラム名。
- **ar_count** – 計算する自己回帰値の数を含む整数。
- **ar_elem** – 計算された自己回帰配列のどの要素を返すかを指定する整数。この整数は 0 より大きく、*ar_count* 以下である必要があります。
- **window-spec** – **TS_MAX_ARMA_AR** は、**OVER ()** 句が必要な OLAP 関数です。

使用法

この時系列関数は、自己回帰近似を含む倍精度浮動小数点値を返します。
TS_MAX_ARMA_AR は、IMSL ライブラリの関数 **imsls_d_max_arma** を呼び出します。

IMSL マッピング

TS_MAX_ARMA_AR の引数は、次のように IMSL ライブラリの関数 **imsls_d_max_arma** にマップします。

```
params = imsls_d_max_arma(n_objs, z[], p, q, 0);
```

- **n_objs** – 現在のウィンドウ・フレームのローの数を含みます。
- **z[]** – 現在のウィンドウ・フレームの *timeseries_expression* の値を含みます。
- **p** – *ar_count* 引数にマップします。
- **q** – =1.

IMSL 関数 **imsls_d_max_arma** が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

例 1

この例では、**TS_MAX_ARMA_AR** 関数を含む SQL 文と、この関数から返されるデータ値を示します。この例では、サンプル入力データ・テーブル (*DATASET*) を入力データとして使用します。*DATASET* サンプル入力データ (80 ページ) を参照してください。

次の SQL 文では、*data* カラムのデータの 2 つの自己回帰推定を含む配列の 2 番目の要素を返します。

```
select ts_max_arma_ar(data,2,2) over (order by rownum rows between
unbounded preceding and unbounded following) as res FROM DATASET
```

Sybase IQ は 50 個のローを返します。各ローには同じ値が含まれています。

表 21 : TS_MAX_ARMA_AR の例 1 で返される値

res
0.179748
0.179748
0.179748
0.179748
0.179748
0.179748
0.179748
0.179748
0.179748
0.179748
0.179748
...
0.179748

例 2

この例は、*DATASET* テーブルの 2 つの結果カラム (自己回帰推定の最初の要素と 2 番目の要素) を返すサンプル・クエリを示しています。

```
select ts_max_arma_ar(data,2,1) over (order by rownum rows between
unbounded preceding and unbounded following) as ar_elem1,
ts_max_arma_ar(data,2,2) over (order by rownum rows between
unbounded preceding and unbounded following) as ar_elem2 FROM DATASET
```

Sybase IQ は、50 個のデータ・ローを返します。各ローには同じ 2 つの値が含まれています。

表 22 : TS_MAX_ARMA_AR の例 2 で返される値

ar_elem1	ar_elem2
0.731164	0.179748
0.731164	0.179748
0.731164	0.179748
0.731164	0.179748
0.731164	0.179748
0.731164	0.179748
0.731164	0.179748
0.731164	0.179748
0.731164	0.179748
0.731164	0.179748
0.731164	0.179748
0.731164	0.179748
...	...
0.731164	0.179748

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠
- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

システム管理ガイド：第 2 巻』の「OLAP の使用」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ

TS_MAX_ARMA_CONST 関数 [集合]

単変量 ARMA (自己回帰移動平均) 時系列モデルにある引数の正確な最尤推定を計算し、定数推定を返します。

構文

```
TS_MAX_ARMA_CONST (timeseries_expression
```

)

OVER (*window-spec*)

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **timeseries_expression** – 数値式。通常は時系列の要素を含むカラム名。
- **window-spec** – **TS_MAX_ARMA_CONST** は、**OVER ()** 句が必要な OLAP 関数です。

使用法

この関数は定数推定を含む倍精度浮動小数点値を返します。

TS_MAX_ARMA_CONST は、IMSL ライブラリの関数 **imsls_d_arma** を呼び出します。

IMSL マッピング

TS_MAX_ARMA_CONST の引数は、次のように IMSL ライブラリの関数 **imsls_d_arma** にマップします。

```
params = imsls_d_max_arma(n_objs, z, p, q, 0);
```

- **n_objs** – 現在のウィンドウ・フレームのローの数を含みます。
- **z[]** – 現在のウィンドウ・フレームの **timeseries_expression** の値を含みます。
- **p** – = 1.
- **q** – = 1.

IMSL 関数 **imsls_d_arma** が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

例

この例では、**TS_MAX_ARMA_CONST** 関数を含む SQL 文と、この関数から返されるデータ値を示します。この例では、サンプル入力データ・テーブル (**DATASET**) を入力データとして使用します。DATASET サンプル入力データ (80 ページ) を参照してください。

次の SQL 文では、**data** カラムのデータの最尤自己回帰計算の定数推定を返します。

```
select ts_max_arma_const(data) over (order by rownum rows between
unbounded preceding and unbounded following) as res FROM DATASET
```

Sybase IQ は 50 個のローを返します。各ローには同じ値が含まれています。

表 23 : TS_MAX_ARMA_CONST から返される値

res
0.107555
0.107555
0.107555
0.107555
0.107555
0.107555
0.107555
0.107555
0.107555
0.107555
0.107555
...
0.107555

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠
- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

システム管理ガイド：第 2 巻』の「OLAP の使用」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ

TS_MAX_ARMA_LIKELIHOOD 関数 [集合]

単変量 ARMA (自己回帰移動平均) 時系列モデルにある引数の正確な最尤推定を計算し、適合モデルの尤度値 (ln) を返します。

構文

```
TS_MAX_ARMA_LIKELIHOOD (timeseries_expression
)
```

```
OVER (window-spec)
```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **timeseries_expression** – 数値式。通常は時系列の要素を含むカラム名。
- **window-spec** – **TS_MAX_ARMA_LIKELIHOOD** は、**OVER ()** 句が必要な OLAP 関数です。

使用法

この関数は、 $-2 * (\ln(\text{likelihood}))$ の値を含む倍精度浮動小数点値を返します。

TS_MAX_ARMA_LIKELIHOOD は、IMSL ライブラリの関数 **imsls_d_max_arma** を呼び出します。

IMSL マッピング

TS_MAX_ARMA_LIKELIHOOD の引数は、次のように IMSL ライブラリの関数 **imsls_d_max_arma** にマップします。

```
params = imsls_d_max_arma(n_objs, z, p, q, IMSLS_LOG_LIKELIHOOD,
&likelihood, 0);
```

- **n_objs** – 現在のウィンドウ・フレームのローの数を含みます。
- **z[]** – 現在のウィンドウ・フレームの **timeseries_expression** の値を含みます。
- **p** – = 1.
- **q** – = 1.
- **likelihood** – 関数呼び出しによって提供されます。ログ推定結果を含みます。

IMSL 関数 **imsls_d_max_arma** が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

例

この例では、**TS_MAX_ARMA_LIKELIHOOD** 関数を含む SQL 文と、この関数から返されるデータ値を示します。この例では、サンプル入力データ・テーブル (**DATASET**) を入力データとして使用します。DATASET サンプル入力データ (80 ページ) を参照してください。

次の SQL 文では、**data** カラムのデータの最尤推定の尤度値を返します。

```
Select ts_max_arma_likelihood(data) over (order by rownum rows
between unbounded preceding and unbounded following) as res FROM
DATASET
```

Sybase IQ は 50 個のローを返します。各ローには同じ値が含まれています。

表 24 : TS_MAX_ARMA_LIKELIHOOD から返される値

res
-11.7818
-11.7818
-11.7818
-11.7818
-11.7818
-11.7818
-11.7818
-11.7818
-11.7818
-11.7818
-11.7818
...
-11.7818

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠
- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

システム管理ガイド：第 2 巻』の「OLAP の使用」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ

TS_MAX_ARMA_MA 関数 [集合]

単変量 ARMA (自己回帰移動平均) 時系列モデルにある引数の正確な最尤推定を計算し、要求された移動平均推定を返します。

構文

```
TS_MAX_ARMA_MA (timeseries_expression
                , ma_count, ma_elem
                )
```

OVER (*window-spec*)

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **timeseries_expression** – 数値式。通常は時系列の要素を含むカラム名。
- **ma_count** – 計算する自己回帰値の数を含む整数。
- **ma_elem** – 計算された移動平均配列から返す要素を示す整数。この整数は 0 より大きく、*ma_count* 以下である必要があります。
- **window-spec** – **TS_MAX_ARMA_MA** は、**OVER ()** 句が必要な OLAP 関数です。

使用法

この関数は、自己回帰推定を含む倍精度浮動小数点値を返します。

TS_MAX_ARMA_MA は、IMSL ライブラリの関数 **imsls_d_max_arma** を呼び出します。

IMSL マッピング

TS_MAX_ARMA_MA の引数は、次のように IMSL ライブラリの関数 **imsls_d_max_arma** にマップします。

```
params = imsls_d_max_arma(n_objs, z[], p, q, 0);
```

- **n_objs** – 現在のウィンドウ・フレームのローの数を含みます。
- **z[]** – 現在のウィンドウ・フレームの **timeseries_expression** の値を含みます。
- **p** – =1.
- **q** – **TS_MAX_ARMA_MA** の引数 *ma_count* にマップします。

IMSL 関数 **imsls_d_max_arma** が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

例

この例では、**TS_MAX_ARMA_MA** 関数を含む SQL 文と、この関数から返されるデータ値を示します。この例では、サンプル入力データ・テーブル (**DATASET**) を入力データとして使用します。「**DATASET** サンプル入力データ・テーブル (80 ページ)」を参照してください。

次の SQL 文では、**data** カラムのデータの最尤推定の移動平均を返します。

```
select ts_max_arma_ma(data,5,4) over (order by rownum rows between unbounded preceding and unbounded following) as res FROM DATASET
```

Sybase IQ は 50 個のローを返します。各ローには同じ値が含まれています。

表 25 : TS_MAX_ARMA_MA から返される値

res
-0.035006
-0.035006
-0.035006
-0.035006
-0.035006
-0.035006
-0.035006
-0.035006
-0.035006
-0.035006
-0.035006
...
-0.035006

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠
- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

システム管理ガイド：第 2 巻』の「OLAP の使用」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ

TS_OUTLIER_IDENTIFICATION 関数 [集合]

異常値の検出および判定を行うと同時に、異常値のない基礎系列が一般的な季節または非季節 ARMA モデルに従う時系列におけるモデル引数を推定します。

構文

```

TS_OUTLIER_IDENTIFICATION (timeseries_expression
, p_value, q_value, s_value
, d_value
, [, delta_value[, critical_value]])
    
```

OVER (*window-spec*)

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **timeseries_expression** – 数値式。通常は時系列の要素を含むカラム名。
- **p_value** – 異常値のない系列が従う自己回帰和分移動平均 (ARIMA) (p, 0, q)x(0, d, 0)s モデルの p 部分を含む整数。
- **q_value** – 異常値のない系列が従う ARIMA (p, 0, q)x(0, d, 0)s モデルの q 部分を含む整数。
- **s_value** – 異常値のない系列が従う ARIMA (p, 0, q)x(0, d, 0)s モデルの s 部分を含む整数。
- **d_value** – 異常値のない系列が従う ARIMA (p, 0, q)x(0, d, 0)s モデルの d 部分を含む整数。
- **delta_value** – (オプション) 一時的変化異常値の検出に使用する抑制効果パラメータを含む倍精度浮動値。整数は、0 よりも大きく、1 よりも小さい必要があります。デフォルト値は 0.7 です。
- **critical_value** – (オプション) 異常値を検出するためのスレッショルドとして使用する倍精度浮動値。デフォルトは 3.0 です。
- **window-spec** – **TS_OUTLIER_IDENTIFICATION** は、無制限ウィンドウを指定した **OVER ()** 句が必要な OLAP 関数です。この関数は値ベースのウィンドウをサポートしません。たとえば、**OVER ()** 句では範囲指定子を使用できません。

使用法

この関数は異常値のない系列を返します。**TS_OUTLIER_IDENTIFICATION** は、IMSL ライブラリの関数 **imsls_d_ts_outlier_identification** を呼び出します。

IMSL マッピング

TS_OUTLIER_IDENTIFICATION の引数は、次のように IMSL ライブラリの関数 **imsls_d_ts_outlier_identification** にマップします。

```
params = imsls_d_ts_outlier_identification(n_objs, model[], z[], 0);
```

- **n_objs** – 現在のウィンドウ・フレームのローの数を含みます。
- **model** – **TS_OUTLIER_IDENTIFICATION** の引数 **p_value**、**s_value**、**q_value**、**d_value** を含む配列。model[0] = p_value、model[1] = s_value、model[2] = q_value、model[3] = d_value
- **z[]** – 現在のウィンドウ・フレームの **timeseries_expression** の値を含みます。

delta_value が null 以外の場合、**TS_OUTLIER_IDENTIFICATION** の引数は、次のように IMSL ライブラリの関数 **imsls_d_ts_outlier_identification** にマップします。

```
params = imsls_d_ts_outlier_identification(n_objs, model[], z[],
IMSL_DELTA, delta_value, 0);
```

critical_value が null 以外の場合、**TS_OUTLIER_IDENTIFICATION** の引数は、次のように IMSL ライブラリの関数 **imsls_d_ts_outlier_identification** にマップします。

```
params = imsls_d_ts_outlier_identification(n_objs, model[],
z[], IMSL_CRITICAL, critical_value, 0);
```

delta_value と *critical_value* の両方が null 以外の場合、**TS_OUTLIER_IDENTIFICATION** の引数は、次のように IMSL ライブラリの関数 **imsls_d_ts_outlier_identification** にマップします。

```
params = imsls_d_ts_outlier_identification(n_objs, model[],
z[], IMSL_DELTA, delta_value, IMSL_CRITICAL, critical_value,
0);
```

IMSL 関数 **imsls_d_ts_outlier_identification** が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

例

この例では、**TS_OUTLIER_IDENTIFICATION** 関数を含む SQL 文と、この関数から返されるデータ値を示します。この例では、サンプル入力データ・テーブル (*DATASET*) を入力データとして使用します。「*DATASET* サンプル入力データ・テーブル (80 ページ)」を参照してください。

次の SQL 文では、*data* カラムのデータの異常値を検出し、判別します。

```
select ts_outlier_identification(data,1,1,1,1,0.7,3.0) over (order
by rownum rows between unbounded preceding and unbounded following)
as res FROM DATASET
```

Sybase IQ は 50 個のローを返します。

表 26 : *ts_outlier_identification* から返される値

res
0.315523
0.485859
0.676886
1.97381
2.77555

res
2.73657
2.64233
4.26118
3.13641
4.16566
2.95952
2.14504
1.98799
0.805859
0.833405
2.29075
1.30045
0.467122
-0.170107
-0.256657
-0.382597
-0.505511
-1.90147
-0.981688
-1.43116
-1.39389
-2.34823
-2.91122
-0.927423
-0.044383
-0.389648
0.545008

res
0.614096
0.364668
1.16043
-0.654063
0.616094
2.00875
1.86696
2.80171
3.78422
4.11499
2.77188
4.00312
4.21298
5.00413
4.74498
4.89621
3.93273
4.31592

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠
- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

システム管理ガイド：第2巻』の「OLAPの使用」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第2/2巻C統計ライブラリ

TS_PARTIAL_AUTOCORRELATION 関数 [集合]

定常時系列のサンプル偏自己相関関数を計算します。

構文

```
TS_PARTIAL_AUTOCORRELATION (timeseries_expression  
, lagmax, lag_elem)
```

```
OVER (window-spec)
```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **timeseries_expression** – 数値式。通常は時系列の要素を含むカラム名。
- **lagmax** – 計算する自己共分散の最大遅延、自己相関、および自己相関の標準誤差を含む整数。整数は、1 以上で、時系列の要素数よりも少ない必要があります。
- **lag_elem** – 返す自己相関配列の要素を示す整数。この整数は 0 より大きく、*lagmax* 以下である必要があります。
- **window-spec** – **TS_PARTIAL_AUTOCORRELATION** は、**OVER ()** 句が必要な OLAP 関数です。

使用法

この関数は異常値のない時系列を返します。**TS_PARTIAL_AUTOCORRELATION** は、IMSL ライブラリの関数 **imsls_d_autocorrelation** と **imsls_d_partial_autocorrelation** を呼び出します。

IMSL マッピング

TS_PARTIAL_AUTOCORRELATION の引数は、次のように IMSL ライブラリの関数 **imsls_d_autocorrelation** と **imsls_d_partial_autocorrelation** にマップします。

```
params = imsls_d_autocorrelation(n_objs, z[], lagmax, 0);
```

```
result = imsls_d_partial_autocorrelation(lagmax, params, 0);
```

- **n_objs** – 現在のウィンドウ・フレームのローの数を含みます。
- **z[]** – 現在のウィンドウ・フレームの **timeseries_expression** の値を含みます。
- **lagmax** – **TS_PARTIAL_AUTOCORRELATION** の引数 *lagmax* にマップします。

IMSL 関数 `imsls_d_autocorrelation` と `imsls_d_partial_autocorrelation` が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

例

この例では、**TS_PARTIAL_AUTOCORRELATION** 関数を含む SQL 文と、この関数から返されるデータ値を示します。この例では、サンプル入力データ・テーブル (*DATASET*) を入力データとして使用します。「DATASET サンプル入力データ・テーブル (80 ページ)」を参照してください。

次の SQL 文では、*data* カラムのデータの偏自己相関を含む配列の最初の要素を返します。

```
select ts_partial_autocorrelation(data,1,1) over (order by rownum
rows between unbounded preceding and unbounded following) as res FROM
DATASET
```

Sybase IQ は 50 個のローを返します。各ローには同じ値が含まれています。

表 27 : TS_PARTIAL_AUTOCORRELATION から返される値

res
0.883453
0.883453
0.883453
0.883453
0.883453
0.883453
0.883453
0.883453
0.883453
0.883453
0.883453
...
0.883453

標準と互換性

- SQL – ISO/ANSI SQL 準拠

- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

システム管理ガイド：第2巻』の「OLAPの使用」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第2/2巻C 統計ライブラリ

TS_VWAP 関数 [集合]

VWAP は Volume-Weighted Average Price (出来高加重平均価格) の略語です。

TS_VWAP は、特定の対象期間の取引値の総出来高に対する比率を計算します。

VWAP は、証券の定義済みの取引期間における平均価格の評価基準です。

TS_VWAP は、単純な集合関数としても OLAP スタイルの集合関数としても使用できます。他の時系列関数とは異なり、**TS_VWAP** は IMSL ライブラリを呼び出しません。

構文 1

```
TS_VWAP (price_expression  
        , volume_expression  
        )
```

構文 2

```
TS_VWAP (price_expression  
        , volume_expression)
```

```
OVER (window-spec)
```

ライセンスの前提条件

RAP – The Trading Edition Enterprise でのみ使用できます。

パラメータ

- **price_expression** – 出来高加重平均に組み込む価格を指定する数値式。
- **volume_expression** – 出来高加重平均の計算で使用する出来高を指定する数値式。
- **window-spec** – 構文 2 で使用した場合、**TS_VWAP** は **OVER ()** 句が必要な OLAP 関数です。

使用法

Sybase IQ は、次の式を使用して **TS_VWAP** を計算します。

図 1 : VWAP の計算

$$P_{vwap} = \frac{\sum_j P_j \cdot Q_j}{\sum_j Q_j}$$

P_{vwap} = 出来高加重平均価格。 $P_j = j$ の取引価格。 $Q_j = j$ の取引高。 j = 対象期間内に発生した 1 つの取引。

例

この例では、入力データ・テーブル、**TS_VWAP** 関数を含む SQL 文、およびこの関数から返されるデータ値を示します。この例では、次のテーブル (**VWAP_DATASET**) を入力データとして使用します。VWAP_DATASET テーブルには、時系列データの 3 つのローが含まれています。

表 28 : 入力データ・テーブル VWAP_DATASET

ロー番号	価格	出来高
1	1	1
2	2	2
3	5	1

次の SQL 文では、出来高加重平均価格を計算します。

```
select ts_vwap(price,volume) over (order by rownum Rows between unbounded preceding and unbounded following) as res FROM VWAP_DATASET
```

Sybase IQ は 3 つのローを返します。

表 29 : TS_VWAP から返される値

res
2.5
2.5
2.5

標準と互換性

- **SQL** – ISO/ANSI SQL 準拠

- **Sybase** – SQL Anywhere および Adaptive Server Enterprise との互換性はありません。

参照

システム管理ガイド：第2巻』の「OLAPの使用」

DATASET サンプル入力データ・テーブル

時系列関数および予測分析関数の例では、次のテーブル (*DATASET*) を入力データとして使用します。DATASET テーブルには、時系列データの50のローが含まれています。

表 30 : 入力データ・テーブル DATASET

ロー番号	データ
1	0.315523
2	0.485859
3	0.676886
4	1.97381
5	2.77555
6	2.73657
7	2.64233
8	4.26118
9	3.13641
10	4.16566
11	2.95952
12	2.14504
13	1.98799
14	0.805859
15	0.833405
16	2.29075
17	1.30045
18	0.467122

ロー番号	データ
19	-0.170107
20	-0.256657
21	-0.382597
22	-0.505511
23	-1.90147
24	-0.981688
25	-1.43116
26	-1.39389
27	-2.34823
28	-2.91122
29	-0.927423
30	-0.044383
31	-0.389648
32	0.545008
33	0.614096
34	0.364668
35	1.16043
36	-0.654063
37	0.616094
38	2.00875
39	1.86696
40	2.80171
41	3.78422
42	4.11499
43	2.77188
44	4.00312
45	4.21298
46	5.00413

時系列予測／分析関数

ロー番号	データ
47	4.74498
48	4.89621
49	3.93273
50	4.31592

索引

I

IMSL ライブラリ
 エラー処理 8
 エラー・ロギング 8
 接続 7

T

TS_ARMA_AR 関数 14
 TS_ARMA_CONST 関数 16
 TS_ARMA_MA 関数 19
 TS_AUTO_ARIMA 12
 TS_AUTO_ARIMA 関数 23
 TS_AUTO_ARIMA_OUTLIER 12
 TS_AUTO_ARIMA_RESULT_AIC 関数 29
 TS_AUTO_ARIMA_RESULT_AICC 関数 30
 TS_AUTO_ARIMA_RESULT_BIC 関数 31
 TS_AUTO_ARIMA_RESULT_FORECAST_ERR
 OR 関数 32
 TS_AUTO_ARIMA_RESULT_FORECAST_VAL
 UE 関数 33
 TS_AUTO_ARIMA_RESULT_MODEL_D 関数
 37
 TS_AUTO_ARIMA_RESULT_MODEL_P 関数
 34
 TS_AUTO_ARIMA_RESULT_MODEL_Q 関数
 35
 TS_AUTO_ARIMA_RESULT_RESIDUAL_SIG
 MA 関数 38
 TS_AUTO_UNI_AR 関数 39
 TS_AUTOCORRELATION 関数 21
 TS_BOX_COX_XFORM 関数 41
 TS_DIFFERENCE 関数 44
 TS_DOUBLE_ARRAY 関数 47
 TS_ESTIMATE_MISSING 関数 48
 TS_GARCH 12
 TS_GARCH 関数 51
 TS_GARCH_RESULT_A 関数 54
 TS_GARCH_RESULT_AIC 関数 55
 TS_GARCH_RESULT_USER 関数 56
 TS_INT_ARRAY 関数 57

TS_LACK OF FIT 関数 58
 TS_LACK_OF_FIT_P 関数 60
 TS_MAX_ARMA_AR 関数 63
 TS_MAX_ARMA_CONST 関数 65
 TS_MAX_ARMA_LIKELIHOOD 関数 67
 TS_MAX_ARMA_MA 関数 69
 TS_OUTLIER_IDENTIFICATION 関数 71
 TS_PARTIAL_AUTOCORRELATION 関数 76
 TS_VWAP 関数 78

え

エラー処理
 IMSL ライブラリ 8
 エラー・ロギング
 IMSL ライブラリ 8

か

関数 11
 TS_ARMA_AR 関数 14
 TS_ARMA_CONST 関数 16
 TS_ARMA_MA 関数 19
 TS_AUTO_ARIMA 23
 TS_AUTO_ARIMA_OUTLIER 26
 TS_AUTO_ARIMA_OUTLIER 関数 26
 TS_AUTO_ARIMA_RESULT_AIC 29
 TS_AUTO_ARIMA_RESULT_AICC 30
 TS_AUTO_ARIMA_RESULT_BIC 31
 TS_AUTO_ARIMA_RESULT_FORECAST_ERROR
 32
 TS_AUTO_ARIMA_RESULT_FORECAST_VALUE
 33
 TS_AUTO_ARIMA_RESULT_MODEL_D 37
 TS_AUTO_ARIMA_RESULT_MODEL_P 34
 TS_AUTO_ARIMA_RESULT_MODEL_Q 35
 TS_AUTO_ARIMA_RESULT_MODEL_S 36
 TS_AUTO_ARIMA_RESULT_MODEL_S 関数 36
 TS_AUTO_ARIMA_RESULT_RESIDUAL_SIGMA
 38
 TS_AUTO_UNI_AR 関数 39
 TS_AUTOCORRELATION 関数 21

TS_BOX_COX_XFORM 関数 41
TS_DIFFERENCE 関数 44
TS_DOUBLE_ARRAY 47
TS_ESTIMATE_MISSING 関数 48
TS_GARCH 関数 51
TS_GARCH_RESULT_A 54
TS_GARCH_RESULT_AIC 55
TS_GARCH_RESULT_USER 56
TS_INT_ARRAY 57
TS_LACK OF FIT 関数 58
TS_LACK_OF_FIT_P 関数 60
TS_MAX_ARMA_AR 関数 63
TS_MAX_ARMA_CONST 関数 65
TS_MAX_ARMA_LIKELIHOOD 関数 67
TS_MAX_ARMA_MA 関数 69
TS_OUTLIER_IDENTIFICATION 関数 71
TS_PARTIAL_AUTOCORRELATION 関数 76
TS_VWAP 関数 78
時系列 11

さ

サンプル入力データ 80

し

時系列関数 7, 11
 IMSL ライブラリ 7
 エラー処理 8
 エラー・ロギング 8

せ

接続

 IMSL ライブラリ 7

て

データセット 80

に

入力データ 80

ま

マニュアル
 SQL Anywhere 4
 Sybase IQ 3

ら

ライブラリ
 IMSL エラー・ロギング 8
 IMSL エラー処理 8