

SYBASE®

时序指南

Sybase IQ 15.2

文档 ID: DC01378-01-1520-01

最后修订日期: 2010 年 4 月

版权所有 © 2010 Sybase, Inc. 保留所有权利。

本出版物适用于 Sybase 软件及任何后续版本, 除非在新版本或技术声明中另有说明。本文档中的信息如有更改, 恕不另行通知。此处说明的软件按许可协议提供, 其使用和复制必须符合该协议的条款。

若要订购附加文档, 美国和加拿大的客户请拨打客户服务部门电话 (800) 685-8225 或发传真至 (617) 229-9845。

持有美国许可协议的其他国家/地区的客户可通过上述传真号码与客户服务部门联系。所有其他国际客户请与 Sybase 子公司或当地分销商联系。仅在定期安排的软件发布日期提供升级。未经 Sybase, Inc. 事先书面许可, 不得以任何形式或任何手段 (电子的、机械的、手工的、光学的或其它手段) 复制、传播或翻译本书的任何部分。

Sybase 商标可在 Sybase 商标页面 (<http://www.sybase.com/detail?id=1011207>) 中进行查看。Sybase 和列出的标记均是 Sybase, Inc. 的商标。® 表示已在美国注册。

Java 和所有基于 Java 的标记均为 Sun Microsystems, Inc. 在美国和其它国家/地区的商标或注册商标。

Unicode 和 Unicode 徽标是 Unicode, Inc. 的注册商标。

本书中使用的所有其它公司名和产品名均可能是相应公司的商标或注册商标。

以下文本仅适用于作为美国政府的机构或承包人的客户: 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

目录

读者	1
相关文档	3
概述	5
许可前提条件	5
时序预测和分析函数的 IMSL 库	5
控制 IMSL 库的错误处理和日志记录	6
错误处理值	6
错误日志记录值	7
时序预测和分析函数	9
集合时序预测和分析函数	9
集合时序函数参数	9
标量时序预测和分析函数	10
按字母顺序排列的函数列表	11
TS_ARMA_AR 函数 [集合]	11
TS_ARMA_CONST 函数 [集合]	13
TS_ARMA_MA 函数 [集合]	16
TS_AUTOCORRELATION 函数 [集合]	18
TS_AUTO_ARIMA 函数 [集合]	20
TS_AUTO_ARIMA_OUTLIER 函数 [集合]	23
TS_AUTO_ARIMA_RESULT_AIC 函数 [标量]	25
TS_AUTO_ARIMA_RESULT_AICC [标量]	26
TS_AUTO_ARIMA_RESULT_BIC 函数 [标量]	27
TS_AUTO_ARIMA_RESULT_FORECAST_ERROR 函 数 [标量]	28
TS_AUTO_ARIMA_RESULT_FORECAST_VALUE 函 数 [标量]	29
TS_AUTO_ARIMA_RESULT_MODEL_P 函数 [标量]	30
TS_AUTO_ARIMA_RESULT_MODEL_Q 函数 [标量]	31

TS_AUTO_ARIMA_RESULT_MODEL_S 函数 [标量]	32
TS_AUTO_ARIMA_RESULT_MODEL_D 函数 [标量]	32
TS_AUTO_ARIMA_RESULT_RESIDUAL_SIGMA 函数 [标量]	33
TS_AUTO_UNI_AR 函数 [集合]	34
TS_BOX_COX_XFORM 函数 [集合]	36
TS_DIFFERENCE 函数 [集合]	39
TS_DOUBLE_ARRAY 函数 [标量]	42
TS_ESTIMATE_MISSING 函数 [集合]	43
TS_GARCH 函数 [集合]	46
TS_GARCH_RESULT_A 函数 [标量]	48
TS_GARCH_RESULT_AIC 函数 [标量]	49
TS_GARCH_RESULT_USER [标量]	49
TS_INT_ARRAY 函数 [标量]	50
TS_LACK_OF_FIT 函数 [集合]	51
TS_LACK_OF_FIT_P 函数 [集合]	54
TS_MAX_ARMA_AR 函数 [集合]	56
TS_MAX_ARMA_CONST 函数 [集合]	58
TS_MAX_ARMA_LIKELIHOOD 函数 [集合]	60
TS_MAX_ARMA_MA 函数 [集合]	62
TS_OUTLIER_IDENTIFICATION 函数 [集合]	64
TS_PARTIAL_AUTOCORRELATION 函数 [集合]	68
TS_VWAP 函数 [集合]	70
DATASET 示例输入数据表	72
索引	75

读者

本书面向使用 Sybase® IQ 进行时序预测和分析的 RAP - Trading Edition™ Enterprise 用户。使用本书可获得有关时序函数 SQL 语法和参数的信息，以及有关参数如何映射到 IMSL™ C Stat 和 C Math 第三方外部库的信息。有关标量和集合用户定义的函数的概念信息，请参见《用户定义的函数指南》。

相关文档

Sybase IQ 和 SQL Anywhere 文档中还囊括其它信息。

相关 Sybase IQ 文档

Sybase IQ 文档集包括：

- 针对所用平台的《发行公告》 - 包含未能及时写入书中的最新信息。最新版本的发行公告可能已提供。若要了解本产品 CD 发行以后增加的重要产品或文档信息，请访问 [Sybase Product Manuals Web](#) 站点。
- 针对所用平台的《安装和配置指南》 - 介绍 Sybase IQ 的安装、升级和某些配置过程。
- **New Features Summary Sybase IQ** (《Sybase IQ 新增功能摘要》) - 汇总了当前版本的新增功能和行为更改。
- 《Sybase IQ 的高级安全性》 - 涵盖了 Sybase IQ 数据存储库中用户加密列的用法。需要有单独的许可证才能安装此产品选件。
- 《错误消息》 - 列出了由 Sybase 错误代码、SQLCode 和 SQLState 引用的 Sybase IQ 错误消息以及 SQL 预处理器错误和警告。
- 《IMSL 数字库用户指南第二卷 (共两卷) : C Stat 库》 - 包含 IMSL C Stat 库时序 C 函数的简要说明。本书仅适用于 RAP - Trading Edition™ Enterprise 用户。
- 《Sybase IQ 简介》 - 包括针对不熟悉 Sybase IQ 或 Sybase Central™ 数据库管理工具的用户的练习。
- 《性能和调优指南》 - 介绍了针对较大数据库的查询优化、设计和调优问题。
- 《快速入门》 - 讨论了生成并查询 Sybase IQ 提供的演示数据库以验证 Sybase IQ 软件安装的步骤。其中包括有关将演示数据库转换为 Multiplex 数据库的信息。
- 《参考手册》 - Sybase IQ 的参考指南：
 - 《参考：构件块、表和过程》 - 介绍了 Sybase IQ 支持的 SQL、存储过程、数据类型和系统表。
 - 《参考：语句和选项》 - 介绍了 Sybase IQ 支持的 SQL 语句和选项。
- 《系统管理指南》 - 包括：
 - 《系统管理指南：第一卷》 - 介绍了启动、连接、数据库创建、填充和编制索引、版本控制、归类、系统备份和恢复、故障排除和数据库修复。
 - 《系统管理指南：第二卷》 - 介绍如何编写和运行过程和批处理、使用 OLAP 编程、访问远程数据以及将 IQ 设置为 Open Server。本书还讨论调度和事件处理、XML 编程和调试。
- 《时序指南》介绍用于时序预测和分析的 SQL 函数。需要 RAP - Trading Edition™ Enterprise 才能使用此产品选件。
- **Unstructured Data Analytics in Sybase IQ** (《Sybase IQ 中的非结构化数据分析》) 说明如何在 Sybase IQ 数据存储库中存储和检索非结构化数据。需要有单独的许可证才能安装此产品选件。

- 《用户定义的函数指南》提供有关用户定义的函数、这些函数的参数以及可能的使用情形的信息。
- Using Sybase IQ Multiplex (《使用 Sybase IQ Multiplex》) 说明如何使用可管理跨多个节点的大型查询负载的 Multiplex 功能。
- 《实用程序指南》提供了 Sybase IQ 实用程序参考资料, 如可用的语法、参数和选项。

相关 SQL Anywhere 文档

注意: 因为 Sybase IQ 与 SQL Anywhere Studio® 的组件 SQL Anywhere® 共享许多组件, 所以 Sybase IQ 支持许多与 SQL Anywhere 相同的功能。IQ 文档集会在适当的时候提示您参考 SQL Anywhere Studio 文档。

SQL Anywhere 的文档包括:

- 《SQL Anywhere Server — 数据库管理》介绍如何运行、管理和配置 SQL Anywhere 数据库。它介绍数据库连接、数据库服务器、数据库文件、备份过程、安全性、高可用性和使用 Replication Server 复制以及管理实用程序和选项。
- 《SQL Anywhere Server — 编程》介绍如何使用 C、C++、Java、PHP、Perl、Python 和 .NET 编程语言 (例如 Visual Basic 和 Visual C#) 生成和配置数据库应用程序。本书还介绍诸如 ADO.NET 和 ODBC 之类的各种编程接口。
- 《SQL Anywhere Server — SQL 参考》提供系统过程的参考信息和目录 (系统表和视图)。它还提供 SQL 语言的 SQL Anywhere 实现的说明 (搜索条件、语法、数据类型和函数)。
- SQL Anywhere Server — SQL Usage (《SQL Anywhere Server — SQL 用法》) 介绍如何设计和创建数据库, 如何导入、导出和修改数据, 如何检索数据以及如何生成存储过程和触发器。

此外, 还可以参见 “Product Manuals” (产品手册) 的 “SQL Anywhere Studio 11.0 集合” 中以及 “DocCommentXchange” 中的 SQL Anywhere 文档。

概述

时序是指按连续时间测量，并且通常具有相同间隔的一系列时间点。时序预测使用一种模型来基于过去的的数据点预测未来的数据点。时序分析使用预测建模、自回归建模和平滑的短期波动等技术来识别数据点的基础上下文和数据中的趋势。

Sybase® IQ 包括用于时序预测和分析的 SQL 函数。这些函数轮流声明在 IMSL™ C Stat 和 C Math 外部库中执行的用户定义的函数 (UDF)。

时序预测和分析函数包括 OLAP 样式的集合和支持的标量函数。

许可前提条件

时序和预测功能仅可用于 RAP - Trading Edition Enterprise。RAP - Trading Edition Enterprise 中包括所有需要的许可证。

时序预测和分析函数的 IMSL 库

除 **TS-VWAP** 以外的所有时序和预测函数都调用两个第三方外部库。Visual Numerics Inc. 提供的 IMSL™ C Stat 和 C Math 库包含用于财务时序预测和分析计算的 C 函数。

包装库 **libsudf** 包含调用 IMSL C Stat 和 C Math 库中所含函数的用户定义的函数 (UDF)。

时序和预测 SQL 函数会自动调用 **libsudf** 包装库。在您调用用于时序和预测分析的有效用户定义的集合函数时，Sybase IQ 会装载 IMSL C Stat 和 C Math 库。

IMSL C Stat 和 C Math 库的名称和位置会因安装 Sybase IQ 的平台不同而不同：

表 1. IMSL 库位置和文件名

	Windows 32 位	Windows 64 位	UNIX (AIX 除外)	AIX
库所在的目录	bin32	bin64	lib64	lib64
库文件名	imslcmath_imsl_dll.dll imslcstat_imsl_dll.dll	imslcmath_imsl_dll.dll imslcstat_imsl_dll.dll	libimslcmath_imsl.so libimslcstat_imsl.so	libimslcmath_imsl_r.so libimslcstat_imsl_r.so

控制 IMSL 库的错误处理和日志记录

您可以控制调用 IMSL 库的时序函数的错误处理和错误日志记录行为。如果在调用 IMSL 库函数时出现运行时错误，则 Sybase IQ 会根据您的错误处理选择进行响应并根据您的错误日志记录选择来记录错误。

1. 调用以下 SET OPTION 语句：

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

错误级别 *value* 为 0、1、2 或 3。

2. 调用以下 SET OPTION 语句：

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

日志级别 *value* 为 0、1、2 或 3。

错误处理值

下表列出了 Time_Series_Error_Level 值。

表 2. IMSL 库错误处理值

值	说明
0 (缺省值)	将忽略在调用 IMSL 库函数时可获取的所有警告和错误。遇到这种情况时，时序函数将返回空值。
1	如果在调用 IMSL 库函数时，时序函数获得了警告或错误消息，则 Sybase IQ 会返回错误消息并中止 SQL 查询。
2	如果在调用 IMSL 库函数时，时序函数获得了致命错误消息，则 Sybase IQ 会返回错误消息并中止 SQL 查询。但是，如果获得了警告，则时序函数将返回一个空值。
3	如果在调用 IMSL 库函数时，时序函数获得了终端错误消息，则 Sybase IQ 会返回错误消息并中止 SQL 查询。但是，如果获得了警告或致命错误，则时序函数将返回一个空值。

错误日志记录值

下表列出了 `Time_Series_Log_Level` 值。

表 3. IMSL 库错误日志记录值

值	说明
0 (缺省值)	将忽略在调用 IMSL 库函数时返回的所有警告和错误，并且不会将其记录到日志文件中。
1	如果在调用 IMSL 库函数时，时序函数返回警告或错误消息，则会在日志文件中记录一条消息。
2	如果在调用 IMSL 库函数时，时序函数返回致命错误消息，则会在日志文件中记录一条消息。但不会记录警告。
3	如果在调用 IMSL 库函数时，时序函数返回终端错误消息，则会在日志文件中记录一条消息。但不会记录警告和致命错误。

时序预测和分析函数

时序 SQL 函数是 OLAP 样式的集合函数，或支持 OLAP 样式的集合的标量函数。通过时序和预测函数，可以使用自回归集成移动平均值 (ARIMA) 或广义自回归条件异方差 (GARCH) 建模并应用模型构建和评估实用程序，例如 Box-Cox 转换。

集合时序预测和分析函数

集合时序预测和分析函数是专门为财务时序统计分析而设计的 OLAP 样式的集合。作为集合，这些函数接受许多组的输入值并返回单个结果。

有关 OLAP 和窗口化集合函数的信息，请参见《系统管理指南：第二卷》>“使用 OLAP”。

集合时序函数参数

下表列出了每个集合时序函数的参数。

表 4. 集合时序函数

时序函数	参数
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>
TS_AUTO_UNI_AR	<i>(timeseries_expression , ar_count, ar_elem, method)</i>
TS_BOX_COX_XFORM	<i>(timeseries_expression , power [, shift [, inverse]])</i>
TS_DIFFERENCE	<i>(timeseries_expression , period1 [, period2 [, ...period 10]])</i>

时序函数	参数
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>)

标量时序预测和分析函数

标量时序和预测 UDF 函数支持 **TS_GARCH** 和 **TS_AUTO_ARIMA** 集合函数。**TS_GARCH** 和 **TS_AUTO_ARIMA** 各生成一个二进制组合，但也接受二进制输入。**TS_INT_ARRAY** 函数为 **TS_AUTO_ARIMA** 和 **TS_AUTO_ARIMA_OUTLIER** 提供输入；**TS_DOUBLE_ARRAY** 函数为 **TS_GARCH** 提供输入。其它标量函数从集合函数返回各个标量结果值。支持的标量函数将 **TS_GARCH** 和 **TS_AUTO_ARIMA** 函数的参数映射到外部 IMSL 库中包含的 C 函数的参数。

以下标量函数支持 **TS_GARCH** 集合函数。

表 5. 支持 **TS_GARCH** 的标量函数

时序函数	参数
TS_DOUBLE_ARRAY	(<i>xguess1</i> , <i>xguess2</i> , [<i>⋯</i> [, <i>xguess10</i>] <i>⋯</i>])
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</i> , <i>model_element_number</i>)

以下标量函数支持 **TS_AUTO_ARIMA** 集合函数：

表 6. 支持 TS_AUTO_ARIMA 的标量函数

时序函数	参数
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>)

以下标量函数支持 TS_AUTO_ARIMA_OUTLIER 集合函数：

表 7. 支持 TS_AUTO_ARIMA_OUTLIER 的标量函数

时序函数	参数
TS_INT_ARRAY	(<i>int1, int2, int3, int4, [... [, int10] ...]</i>)

按字母顺序排列的函数列表

本节提供了有关所有时序函数的详细信息，包括语法、许可前提条件、参数说明、用法、IMSL 库映射、示例和标准/兼容性信息。

TS_ARMA_AR 函数 [集合]

计算用于自回归移动平均值 (ARMA) 模型的参数的最小二乘估计值，并返回请求的自回归估计值。

语法

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

OVER (*window-spec*)

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **timeseries_expression** - 包含时序中元素的数值表达式（通常为列名）。
- **ar_count** - 包含要计算的自回归值数量的整数。
- **ar_elem** - 一个整数，标识所计算的 AR 数组中应返回的元素。*ar_elem* 必须大于 0 且小于或等于 *ar_count*。
- **method** - （可选）一个整数，标识用于计算估计值的过程的类型。0（缺省值）= 最小二乘法；1 = 矩量法。
- **window-spec** - **TS_ARMA_AR** 是一个需要 **OVER ()** 子句的 OLAP 函数。

用法

TS_ARMA_AR 时序函数返回一个双精度浮点值，其中包含自回归估计值。

TS_ARMA_AR 调用 IMSL 库中的函数 **imsls_d_arma**。

IMSL 映射

TS_ARMA_AR 的参数映射到 IMSL 库函数 **imsls_d_arma** 的方式如下：

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

- **n_objs** - 包含当前窗口架构中的行数。
- **z[]** - 包含当前窗口架构的 *timeseries_expression* 值。
- **p** - 映射到用户定义的集合函数参数 *ar_count*。
- **q** - =1.
- **methodID** - 映射到 **TS_ARMA_AR** 的 *method* 参数。可以设置为 **IMLS_METHOD_OF_MOMENTS** 或 **IMLS_LEAST_SQUARES**。

有关 **imsls_d_arma** 如何执行时序计算的详细信息，请参见 *IMSL C Numerical Library User's Guide: Volume 2 of 2 C Stat Library*（《IMSL C 数字库用户指南：第二卷（共两卷）C Stat 库》）。

示例

此示例显示了一个包含 **TS_ARMA_AR** 函数的 SQL 语句，以及该函数返回的数据值。此示例使用示例输入数据表（名为 *DATASET*）作为输入数据。请参见 *DATASET* 示例输入数据（第 72 页）。

下面的 SQL 语句使用最小二乘法返回包含 *data* 列中一个值的自回归估计值的第一个元素：

```
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 返回 50 行，每行都包含相同的值：

表 8. TS_ARMA_AR 返回的值

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

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

《系统管理指南：第二卷》 > “使用 OLAP”

IMSL C Numerical Library User’s Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_ARMA_CONST 函数 [集合]

计算用于自回归移动平均值 (ARMA) 模型的参数的最小二乘估计值，并返回估计的常量。

语法

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

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

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **timeseries_expression** - 包含时序中元素的数值表达式（通常为列名）。
- **method** - 一个整数，标识用于计算估计值的过程的类型。0（缺省值）= 最小二乘法；1 = 矩量法。
- **window-spec** - **TS_ARMA_CONST** 是一个需要 **OVER ()** 子句的 OLAP 函数。

用法

此时序函数返回一个双精度浮点值，其中包含函数生成的常量估计值。

TS_ARMA_CONST 调用 IMSL 库中的函数 **imsls_d_arma**。

IMSL 映射

TS_ARMA_CONST 的参数映射到 IMSL 库函数 **imsls_d_arma** 的方式如下：

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

- **n_objs** - 包含当前窗口架构中的行数。
- **z[]** - 包含当前窗口架构的 *timeseries_expression* 值。
- **p** - =1.
- **q** - =1.
- **methodID** - 映射到 **TS_ARMA_CONST** 的 **method** 参数。

有关函数 **imsls_d_arma** 如何执行时序计算的详细信息，请参见 *IMSL C Numerical Library User's Guide: Volume 2 of 2 C Stat Library*（《IMSL C 数字库用户指南：第二卷（共两卷）C Stat 库》）。

示例 1

此示例显示了一个包含 **TS_ARMA_CONST** 函数的 SQL 语句，以及该函数返回的数据值。此示例使用示例输入数据表（名为 **DATASET**）作为输入数据。请参见 **DATASET** 示例输入数据（第 72 页）。

下面的 SQL 语句使用最小二乘法返回 **data** 列中的一个估计常量：

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

Sybase IQ 返回 50 行，每行都包含相同的值：

表 9. TS_ARMA_CONST 示例 1 返回的值

res
0.082077

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

示例 2

此示例提供一个返回 AR、MA 和常量参数的估计值的查询示例。数组中 AR 和 MA 的第一个元素包含一个元素。

```
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 返回 50 行数据，每行包含相同的三个值：

表 10. TS_ARMA_CONST 示例 2 返回的值

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

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

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

《系统管理指南：第二卷》 > “使用 OLAP”

IMSL C Numerical Library User’s Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_ARMA_MA 函数 [集合]

计算用于自回归移动平均值 (ARMA) 模型的参数的最小二乘估计值，并返回请求的移动平均估计值。

语法

```

TS_ARMA_MA (timeseries_expression
            , ma_count, ma_elem, method)

OVER (window-spec)
    
```

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **timeseries_expression** - 包含时序中元素的数值表达式 (通常为列名)。
- **ma_count** - 包含要计算的自回归值数量的整数。
- **ma_elem** - 一个整数，标识要从计算的移动平均值数组中返回的元素。该整数必须大于 0 且小于或等于 **ma_count**。
- **method** - (可选) 一个整数，标识要用于计算估计值的过程。0 (缺省值) = 最小二乘法；1 = 矩量法。
- **window-spec** - **TS_ARMA_MA** 是一个需要 **OVER ()** 子句的 OLAP 函数。

用法

此时序函数返回一个表示移动平均估计值的双精度浮点值。**TS_ARMA_MA** 调用 IMSL 库中的函数 **imsls_d_arma**。

IMSL 映射

TS_ARMA_MA 的参数映射到 IMSL 库函数 **imsls_d_arma** 的方式如下：

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

- **n_objs** - 包含当前窗口架构中的行数。
- **z[]** - 包含当前窗口架构的 *timeseries_expression* 值。
- **p** - =1.
- **q** - 映射到用户定义的集合函数参数 **ma_count**。
- **method_id** - 映射到 **TS_ARMA_MA** 的 **method** 参数。

有关函数 **imsls_d_arma** 如何执行时序计算的详细信息，请参见 IMSL C Numerical Library User's Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)。

示例

此示例显示了一个包含 **TS_ARMA_MA** 函数的 SQL 语句，以及该函数返回的数据值。此示例使用示例输入数据表 (名为 *DATASET*) 作为输入数据。请参见 *DATASET* 示例输入数据 (第 72 页)。

下面的 SQL 语句使用最小二乘法返回包含 *data* 列中一个元素的数组的第一个元素：

```
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 返回 50 行，每行都包含相同的值：

表 11. TS_ARMA_MA 返回的值

res
0.105075
0.105075
0.105075
0.105075
0.105075
0.105075
0.105075

res
0.105075
0.105075
...
0.105075

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

《系统管理指南：第二卷》 > “使用 OLAP”

IMSL C Numerical Library User's Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_AUTOCORRELATION 函数 [集合]

计算固定时序的示例自动相关函数。

语法

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

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

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **timeseries_expression** - 包含时序中元素的数值表达式 (通常为列名)。
- **lagmax** - 一个整数, 用于指定自协方差、自动相关以及自动相关的标准误差的最大滞后。该整数必须大于或等于 1, 并小于时序中元素的数量。
- **lag_elem** - 用于指定将返回自动相关数组中哪个元素的整数。该整数必须大于零, 并小于或等于 lagmax。
- **window-spec** - **TS_AUTOCORRELATION** 是一个需要 **OVER ()** 子句的 OLAP 函数。

用法

此时序函数返回表示自动相关值的双精度浮点值。**TS_AUTOCORRELATION** 调用 IMSL 库中的函数 **imsls_d_autocorrelation**。

IMSL 映射

TS_AUTOCORRELATION 的参数映射到 IMSL 库函数 **imsls_d_autocorrelation()** 的方式如下：

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

- **n_objs** - 包含当前窗口架构中的行数。
- **x[]** - 包含当前窗口架构的 *timeseries_expression* 值。
- **lagmax** - 映射到用户定义的集合函数参数 *lag_max*。

有关函数 **imsls_d_autocorrelation** 如何执行时序计算的详细信息，请参见 **IMSL C Numerical Library User's Guide: Volume 2 of 2 C Stat Library** (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)。

示例

此示例显示了一个包含 **TS_AUTOCORRELATION** 函数的 SQL 语句，以及该函数返回的数据值。此示例使用示例输入数据表 (名为 **DATASET**) 作为输入数据。请参见 **DATASET** 示例输入数据 (第 72 页)。

下面的 SQL 语句从包含 *data* 列中时序数据的自动相关的数组中返回第二个元素：

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

Sybase IQ 返回 50 行，每行都包含相同的值：

表 12. TS_AUTOCORRELATION 返回的值

res
0.803659
0.803659
0.803659
0.803659
0.803659
0.803659
0.803659
0.803659
0.803659

res
0.803659
...
0.803659

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

《系统管理指南：第二卷》> “使用 OLAP”

IMSL C Numerical Library User’s Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_AUTO_ARIMA 函数 [集合]

确定乘积季节性自回归集成移动平均值 (ARIMA) 模型的参数，并生成合并了其效果持续至序列末尾之后的离群值效果的预测。

语法

```

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


许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **time_value** - 每个输入时序数据点的时间值。
- **timeseries_expression** - 包含要区分的时序中元素的数值表达式（通常为列名）。
- **max_lag** - （可选）表示符合 AR(p) 模型时所允许的最大滞后。必须为整数常量或常量表达式。缺省值为 10。
- **critical** - （可选）用作离群检测阈值的临界值。此值必须大于 0，并且必须是双精度浮点常量或常量表达式。缺省值为 3.0。
- **epsilon** - （可选）在离群检测期间控制参数估计值的准确性的正容错值。此值必须是双精度浮点常量或常量表达式。缺省值为 0.001。
- **criterion** - （可选）用于选择最优模型的信息准则。必须是值为 0、1 或 2 的整数常量或常量表达式：
 - 0 - （缺省值）阿凯克信息准则 (AIC)
 - 1 - 修正的阿凯克信息准则 (AICC)
 - 2 - 贝叶斯信息准则 (BIC)
- **confidence** - （可选）用于计算预测置信界限的置信度，从开区间 (0, 100) 中选择。置信度的典型选择是 90.0、95.0 和 99.0。必须是双精度浮点常量或常量表达式。缺省值为 95.0。
- **n_predictions** - （可选）请求的预测值的数量。必须为正整数常量或常量表达式。支持的最大值为 2000。缺省值为 0。
- **model** - （可选）表示二进制编码的整数数组，长度为四个整数，其中依次包含 p 、 q 、 s 和 d 的 ARIMA 值。如果在 **TS_INT_ARRAY** 函数中指定了 *model* 实参，则 **TS_AUTO_ARIMA** 函数确定 p 、 q 、 s 和 d 值时会使用 *method 3 (Specified ARIMA)* 形参，它是 (*IMSLS_METHOD*, *int method*) 实参（属于 IMSL 库中的 *imsls_d_auto_arima* 函数）的形参。如果 *model* 参数为 NULL，则 **TS_AUTO_ARIMA** 函数会自动计算 ARIMA(p,0,0)X(0,d,0) 模型，这会最大程度地降低指定的误差准则。缺省值为 NULL。
- **window-spec** - **TS_AUTO_ARIMA** 是需要包含 **ORDER BY** 子句的 **OVER()** 子句的 OLAP 函数。**OVER()** 子句中不允许 **ROWS** 或 **RANGE** 分类符。

用法

作为 OLAP 样式的集合函数，**TS_AUTO_ARIMA** 生成单个 SQL 结果 — 一个专门编码的可变长度的二进制结果值。支持的标量函数接受二进制组合输出值并返回其中的各个标量结果值。

因为 OLAP 样式的集合函数为每个输入元组返回一个结果值，所以为分区中的每行返回相同的二进制组合结果。如果未指定 **PARTITION BY** 子句（在 **OVER** 子句中），请使用 **SELECT FIRST** 将结果减少到包含二进制组合结果的单个元组。如果在 **OVER**

子句中指定了 **PARTITION BY** 子句，请使用 **SELECT DISTINCT** 为每个分区只留下其中一个元组，而删除所有其它元组。

IMSL 映射

TS_AUTO_ARIMA_RESULT 支持的标量函数将执行到外部 VNI 库中的 IMSL C 函数参数的映射。

示例

在此示例中，**IQ** 将分别为四种股票符号中的每一种计算 **AUTO_ARIMA** 模型。**DISTINCT** 限定符将元组集减少为每个股票符号一个元组。最后，返回一个输出行，其中包含股票符号和介绍为该股票计算的 **AUTO_ARIMA** 模型的所有相关信息。

```
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', 'ZXX', 'ZZZ')
) as auto_arima_per_stock
```

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

- 《系统管理指南：第二卷》 > “使用 OLAP”
- IMSL C Numerical Library User’s Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_AUTO_ARIMA_OUTLIER 函数 [集合]

TS_AUTO_ARIMA_OUTLIER 接受输入时序并自动确定乘积季节性自回归集成移动平均值 (ARIMA) 模型的参数。而 **TS_AUTO_ARIMA** 使用 ARIMA 模型来预测超出输入集的值，**TS_AUTO_ARIMA_OUTLIER** 使用 ARIMA 模型来识别输入时序中的统计离群值，并返回每个的离群值类型。

语法

```

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

```

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **time_value** - 每个输入时序数据点的时间值。
- **timeseries_expression** - 包含要区分的时序中元素的数值表达式 (通常为列名)。
- **max_lag** - (可选) 表示符合 AR(p) 模型时所允许的最大滞后。必须为正整数常量或常量表达式。缺省值为 10。

- **critical** - (可选) 用作离群检测阈值的临界值。此值必须大于 0，并且必须是双精度浮点常量或常量表达式。缺省值为 3.0。
- **epsilon** - (可选) 在离群检测期间控制参数估计值的准确性的正容错值。必须是双精度浮点常量或常量表达式。缺省值为 0.001。
- **criterion** - (可选) 用于选择最优模型的信息准则。必须是值为 0、1 或 2 的整数常量或常量表达式。缺省值为 0。
 - 0 - (缺省值) 阿凯克信息准则 (AIC)
 - 1 - 修正的阿凯克信息准则 (AICC)
 - 2 - 贝叶斯信息准则 (BIC)
- **confidence** - (可选) 用于计算预测置信界限的置信度，从开区间 (0, 100) 中选择。置信度的典型选择是 90.0、95.0 和 99.0。必须是双精度浮点常量或常量表达式。缺省值为 95.0。
- **delta** - (可选) 检测临时变化离群时使用的阻尼影响参数。必须是双精度浮点常量或常量表达式。缺省值为 0.7。
- **model** - (可选) 表示二进制编码的整数数组，长度为四个整数，其中依次包含 p 、 q 、 s 和 d 的 ARIMA 值。如果在 **TS_INT_ARRAY** 函数中指定了 *model* 实参，则 **TS_AUTO_ARIMA_OUTLIER** 函数确定 p 、 q 、 s 和 d 值时会使用 *method 3 (Specified ARIMA)* 形参，它是 (*IMSLS_METHOD, int method*) 实参 (属于 IMSL 库中的 *imsls_d_auto_arima* 函数) 的形参。如果 *model* 参数为 **NULL**，则 **TS_AUTO_ARIMA_OUTLIER** 函数会自动计算 ARIMA($p,0,0$)X($0,d,0$) 模型，这会最大程度地降低指定的误差准则。缺省值为 **NULL**。
- **window-spec** - **TS_AUTO_ARIMA_OUTLIER** 是需要 **OVER()** 子句 (其中包含 **ORDER BY** 子句) 的 OLAP 函数。**OVER()** 子句中不允许 **ROWS** 或 **RANGE** 分类符

用法

TS_AUTO_ARIMA 和 **TS_AUTO_ARIMA_OUTLIER** 的输入几乎相同。不过，**TS_AUTO_ARIMA_OUTLIER** 为分区中的每个输入行返回不同的值，而 **TS_AUTO_ARIMA** 为每行返回相同的值。由于结果数据类型和结果值范围中存在这些区别，因此您无需使用 **SELECT FIRST** 或 **SELECT DISTINCT** 来消除重复的输出值。**TS_AUTO_ARIMA_OUTLIER** 没有任何支持的可解码结果的标量函数。

与 **TS_AUTO_ARIMA** 一样，**TS_AUTO_ARIMA_OUTLIER** 需要支持标量函数的 **TS_INT_ARRAY**。**TS_INT_ARRAY** 提供二进制组合输入。

此函数为每个输入元组返回一个整数值，从而为每个元组内的时间和数据值指定离群值类型。如果时间和数据值不是离群值，则此函数返回 **NULL**。整数值有：

- **0** - 创新离群值 (IO)。
- **1** - 附加离群值 (AO)。
- **2** - 级别移动 (LS)。
- **3** - 临时变化 (TC)。
- **4** - 无法识别 (UI)。

有关五种离散值类型的详细信息，请参见 *IMSL C Numerical Library User's Guide: Volume 2 of 2 C Stat Library*（《IMSL C 数字库用户指南：第二卷（共两卷）C Stat 库》）。

IMSL 映射

映射到 *imsls_d_auto_arima* 的离群值识别逻辑。

示例

此示例为 XYZ 的股票价格的时序计算 ARIMA 模型，然后使用该模型来识别哪些交易在统计上属于离群值。对于未识别为离群值的行，**TS_AUTO_ARIMA_OUTLIER** 返回 NULL。对于识别为离群值的行，**TS_AUTO_ARIMA_OUTLIER** 返回一个介于 0 和 4 之间的整数，表示当前行的特定离群值类型。

```
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'
```

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

- 《系统管理指南：第二卷》> “使用 OLAP”
- *IMSL C Numerical Library User's Guide: Volume 2 of 2 C Stat Library*（《IMSL C 数字库用户指南：第二卷（共两卷）C Stat 库》）

TS_AUTO_ARIMA_RESULT_AIC 函数 [标量]

TS_AUTO_ARIMA 函数的支持函数。检索 **TS_AUTO_ARIMA** 生成的阿凯克信息准则 (AIC) 输出参数。

语法

```
TS_AUTO_ARIMA_RESULT_AIC(
  auto_arima_result
)
```

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **auto_arma_result** - **TS_AUTO_ARIMA** 生成的 varbinary 结果。

用法

返回一个表示阿凯克信息准则输出参数的双精度浮点值。

IMSL 映射

映射到 *imsls_d_auto_arma* 的 *IMSL_AIC* 参数。

示例

请参见 **TS_AUTO_ARIMA** 的示例。

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

- 《系统管理指南：第二卷》> “使用 OLAP”
- IMSL C Numerical Library User's Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_AUTO_ARIMA_RESULT_AICC [标量]

TS_AUTO_ARIMA 的支持函数。检索 **TS_AUTO_ARIMA** 生成的修正的 AIC (AICC) 输出参数。

语法

```
TS_AUTO_ARIMA_RESULT_AICC(  
    auto_arma_result  
)
```

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **auto_arma_result** - **TS_AUTO_ARIMA** 生成的 varbinary 结果。

用法

返回生成的修正的阿凯克信息准则输出参数的双精度浮点值。

IMSL 映射

映射到 *imsls_d_auto_arima* 的 *IMSL_S_AICC* 参数。

示例

请参见 **TS_AUTO_ARIMA** 的示例。

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

- 《系统管理指南：第二卷》 > “使用 OLAP”
- IMSL C Numerical Library User’s Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_AUTO_ARIMA_RESULT_BIC 函数 [标量]

TS_AUTO_ARIMA 的支持函数。检索 **TS_AUTO_ARIMA** 生成的贝叶斯信息准则 (BIC) 输出参数。

语法

```
TS_AUTO_ARIMA_RESULT_BIC(
    auto_arima_result
)
```

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **auto_arima_result** - **TS_AUTO_ARIMA** 生成的 varbinary 结果。

用法

返回生成的贝叶斯信息准则输出参数的双精度浮点值。

IMSL 映射

映射到 *imsls_d_auto_arima* 的 *IMSL_S_BIC* 参数。

示例

请参见 **TS_AUTO_ARIMA** 的示例。

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

- 《系统管理指南：第二卷》> “使用 OLAP”
- IMSL C Numerical Library User’s Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_AUTO_ARIMA_RESULT_FORECAST_ERROR 函数 [标量]

TS_AUTO_ARIMA 的支持函数。检索 **TS_AUTO_ARIMA** 生成的原始输入序列的预测标准差值。

语法

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

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **auto_arima_result** - **TS_AUTO_ARIMA** 生成的 varbinary 结果。
- **forecast_element_number** - 整数常量表达式值。允许的范围是 1 至 `n_predictions`。

用法

返回指定预测元素的标准差的双精度浮点值。此预测差值的隐式时间值是传递给 **TS_AUTO_ARIMA** 的输入 *time_values* 中的最后一个时间值加上指定的 *forecast_element_number*。

IMSL 映射

映射到 *imsls_d_auto_arima* 的 *IMSLs_OUTLIER_FORECAST* 参数。

示例

请参见 **TS_AUTO_ARIMA** 的示例。

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

- 《系统管理指南：第二卷》 > “使用 OLAP”
- IMSL C Numerical Library User’s Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_AUTO_ARIMA_RESULT_FORECAST_VALUE 函数 [标量]

TS_AUTO_ARIMA 的支持函数。检索 **TS_AUTO_ARIMA** 生成的没有离群值的请求序列的预测值。

语法

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

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **auto_arima_result** - **TS_AUTO_ARIMA** 生成的 varbinary 结果。
- **model_element_number** - 整数常量表达式值。允许的范围是 1 至 n_predictions。

用法

返回一个表示原始输入序列的指定预测值的双精度浮点值。此预测值的隐式时间值是传递给 **TS_AUTO_ARIMA** 的输入 *time_values* 中的最后一个时间值加上指定的 *forecast_element_number*。

IMSL 映射

映射到 *imsls_d_auto_arima* 的 *IMSL_OUTLIER_FORECAST* 参数

示例

请参见 **TS_AUTO_ARIMA** 函数的示例。

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

- 《系统管理指南：第二卷》 > “使用 OLAP”
- IMSL C Numerical Library User's Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_AUTO_ARIMA_RESULT_MODEL_P 函数 [标量]

TS_AUTO_ARIMA 的支持函数。检索 **TS_AUTO_ARIMA** 在计算 ARIMA 模型说明时生成的 p 值。

语法

```
TS_AUTO_ARIMA_RESULT_MODEL_P(  
    auto_arima_result  
)
```

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **auto_arima_result** - **TS_AUTO_ARIMA** 生成的 varbinary 结果。

用法

返回 **TS_AUTO_ARIMA** 生成的双精度浮点 p 输出参数。

IMSL 映射

映射到 *imsls_d_auto_arima* 的 *IMSL_MODEL* 参数的第一个元素。

示例

请参见 **TS_AUTO_ARIMA** 的示例。

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

- 《系统管理指南：第二卷》> “使用 OLAP”
- IMSL C Numerical Library User’s Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_AUTO_ARIMA_RESULT_MODEL_Q 函数 [标量]

TS_AUTO_ARIMA 的支持函数。检索 **TS_AUTO_ARIMA** 在计算 ARIMA 模型说明时生成的 q 值。

语法

```
TS_AUTO_ARIMA_RESULT_MODEL_Q
(auto_arima_result
)
```

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **auto_arima_result** - **TS_AUTO_ARIMA** 生成的 varbinary 结果。

用法

返回 **TS_AUTO_ARIMA** 生成的双精度浮点 q 输出参数。

IMSL 映射

映射到 *imsls_d_auto_arima* 的 *IMSLS_MODEL* 参数的第二个元素。

示例

请参见 **TS_AUTO_ARIMA** 的示例。

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

- 《系统管理指南：第二卷》> “使用 OLAP”
- IMSL C Numerical Library User’s Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_AUTO_ARIMA_RESULT_MODEL_S 函数 [标量]

TS_AUTO_ARIMA 的支持函数。检索 **TS_AUTO_ARIMA** 在计算 ARIMA 模型说明时生成的 *s* 值。

语法

```
TS_AUTO_ARIMA_RESULT_MODEL_S(  
  auto_arima_result  
)
```

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **auto_arima_result** - **TS_AUTO_ARIMA** 生成的 varbinary 结果。

用法

返回 **TS_AUTO_ARIMA** 生成的双精度浮点 *s* 输出参数。

IMSL 映射

映射到 *imsls_d_auto_arima* 的 *IMSLS_MODEL* 参数的第三个元素。

示例

请参见 **TS_AUTO_ARIMA** 的示例。

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

- 《系统管理指南：第二卷》> “使用 OLAP”
- IMSL C Numerical Library User’s Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_AUTO_ARIMA_RESULT_MODEL_D 函数 [标量]

TS_AUTO_ARIMA 函数的支持函数。检索 **TS_AUTO_ARIMA** 在计算 ARIMA 模型说明时生成的 *d* 值。

语法

```
TS_AUTO_ARIMA_RESULT_MODEL_D(  
  auto_arima_result  
)
```

```
auto_arma_result
)
```

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **auto_arma_result** - **TS_AUTO_ARIMA** 生成的 varbinary 结果。

用法

返回 **TS_AUTO_ARIMA** 生成的双精度浮点 *s* 输出参数。

IMSL 映射

映射到 *imsls_d_auto_arma* 的 *IMSL_MODEL* 参数的第四个元素。

示例

请参见 **TS_AUTO_ARIMA** 的示例。

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

- 《系统管理指南：第二卷》> “使用 OLAP”
- IMSL C Numerical Library User’s Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_AUTO_ARIMA_RESULT_RESIDUAL_SIGMA 函数 [标量]

TS_AUTO_ARIMA 的支持函数。检索没有离群值的数据点的残差标准差。

语法

```
TS_AUTO_ARIMA_RESULT_RESIDUAL_SIGMA (
auto_arma_result
)
```

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **auto_arima_result** - **TS_AUTO_ARIMA** 生成的 varbinary 结果。

用法

返回残差标准差输出参数。

IMSL 映射

映射到 *imsls_d_auto_arima* 的 *IMSLS_RESIDUAL_SIGMA* 参数。

示例

请参见 **TS_AUTO_ARIMA** 的示例。

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

- 《系统管理指南：第二卷》> “使用 OLAP”
- IMSL C Numerical Library User’s Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_AUTO_UNI_AR 函数 [集合]

执行单变量自回归时序模型的自动选择和拟合。

语法

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

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

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **timeseries_expression** - 包含时序中元素的数值表达式 (通常为列名)。
- **ar_count** - 包含要计算的自回归值数量的整数。
- **ar_elem** - 用于标识要返回哪个已计算自回归值的整数。该整数必须大于 0, 并小于或等于 *ar_count*。

- **method** - (可选) 用于标识计算 AR 系数时应使用哪个方法的整数, 其中: 0 = 矩量法; 1 = 最小二乘法 (缺省值); 2 = 最大似然。
- **window-spec** - **TS_AUTO_UNI_AR** 是一个需要 **OVER ()** 子句的 OLAP 函数。

用法

此时序函数返回一个双精度浮点数, 其中包含自回归估计值。**TS_AUTO_UNI_AR** 调用 IMSL 库中的函数 **imsls_d_auto_uni_ar**。

IMSL 映射

TS_AUTO_UNI_AR 的参数映射到 IMSL 库函数 **imsls_d_auto_uni_ar** 的方式如下:

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

- **n_objs** - 包含当前窗口架构中的行数。
- **z[]** - 包含当前窗口架构的 *timeseries_expression* 值。
- **maxlag** - 映射到用户定义的集合函数参数 *ar_count*。
- **p** - 输出参数, 表示模型中具有最小 AIC 的自回归参数的数量。
- **method** - 映射到用户定义的集合函数参数 *method*。如果 *ar_elem* 大于 *p*, 并且 IMSL 库时序函数错误处理值设置为 0, 则 Sybase IQ 会返回 null。如果 IMSL 库时序函数错误处理值设置为不为 0 的值, 则 Sybase IQ 会显示一条错误消息, 指示 *ar_elem* 大于 *p*。

有关函数 **imsls_d_auto_uni_ar** 如何执行时序计算的详细信息, 请参见 IMSL C Numerical Library User's Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南: 第二卷 (共两卷) C Stat 库》)。

示例

此示例显示了一个包含 **TS_AUTO_UNI_AR** 函数的 SQL 语句, 以及该函数返回的数据值。此示例使用示例输入数据表 (名为 *DATASET*) 作为输入数据。请参见 *DATASET* 示例输入数据 (第 72 页)。

下面的 SQL 语句从包含 *data* 列中两个元素的数组中返回第一个元素:

```
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 返回 50 行, 每行都包含相同的值:

表 13. **TS_AUTO_UNI_AR** 返回的值

res
0.883453
0.883453
0.883453

res
0.883453
0.883453
0.883453
0.883453
0.883453
0.883453
...
0.883453

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

《系统管理指南：第二卷》 > “使用 OLAP”

IMSL C Numerical Library User’s Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_BOX_COX_XFORM 函数 [集合]

执行正向和反向 Box-Cox 幂转换。

语法

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

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **timeseries_expression** - 包含时序中元素的数值表达式 (通常为列名)。
- **power** - 表示 Box-Cox 幂转换中指数参数的双精度浮点值。
- **shift** - (可选) 表示漂移参数的双精度浮点值。此值必须满足以下关系：
 $\min(\text{timeseries}) + \text{shift} > 0$ 。Shift 缺省值为 0.0。

- **inverse** - (可选) 一个 tinyint 值; 如果设置为 1, 则 Sybase IQ 会执行反向转换。如果为 0 或 null, 则 Sybase IQ 会执行正向转换。缺省值是 0。
- **window-spec** - **TS_BOX_COX_XFORM** 是需要具有未受限制窗口的 **OVER ()** 子句的 OLAP 函数。**TS_BOX_COX_XFORM** 不支持基于值的窗口; 例如, 在 **OVER ()** 子句中不能使用范围分类符。

用法

TS_BOX_COX_XFORM 为时序中的每个元素返回相应的计算转换值; 它调用 IMSL 库中的函数 **imsls_d_box_cox_transform**。

IMSL 映射

TS_BOX_COX_XFORM 的参数映射到 IMSL 库函数 **imsls_d_box_cox_transform** 的方式如下:

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

- **n_objs** - 包含当前窗口架构中的行数。
- **z[]** - 包含当前窗口架构的 *timeseries_expression* 值。
- **power** - 映射到用户定义的集合函数参数 *power*。
- **shift** - 映射到用户定义的集合函数参数 *shift*。
- **IMSL_INVERSE** - 如果用户定义的集合函数参数 *inverse* 为 1, 则 Sybase IQ 将使用 **IMSL_INVERSE** 调用 Box-Cox 转换, 否则函数调用将不考虑此参数。

有关函数 **imsls_d_box_cox_transform** 如何执行时序计算的详细信息, 请参见 IMSL C Numerical Library User's Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南: 第二卷 (共两卷) C Stat 库》)。

示例

此示例显示了一个输入数据表、一个包含 **TS_BOX_COX_XFORM** 函数的 SQL 语句, 以及该函数返回的数据值。此示例使用下表 (名为 **BOX_COX_XFORM_DATASET**) 作为输入数据。BOX_COX_XFORM_DATASET 表包含 13 行时序数据:

表 14. 输入数据表 **BOX_COX_XFORM_DATASET**

rownum	data
1	7
2	26
3	6
4	60
5	78.5

rownum	data
6	1
7	29
8	15
9	52
10	74.3
11	11
12	56
13	8

下面的 SQL 语句从 *data* 列中返回 Box-Cox 幂转换：

```
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 返回以下 13 行：

表 15. TS_BOX_COX_XFORM 返回的值

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

res
9

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

《系统管理指南：第二卷》> “使用 OLAP”

IMSL C Numerical Library User's Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_DIFFERENCE 函数 [集合]

区分季节性或非季节性时序。

语法

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

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **timeseries_expression** - 包含要区分的时序中元素的数值表达式 (通常为列名)。
- **period1 ... period10** - 每个时间段都是一个整数表达式, 其中包含要在其中区分时序的时间段。必须至少指定一个时间段, 最多可以指定 10 个时间段。
- **window-spec** - **TS_DIFFERENCE** 是一个需要具有未受限制窗口的 **OVER ()** 子句的 OLAP 函数。此函数不支持基于值的窗口; 例如, 在 **OVER ()** 子句中不能使用范围分类符。

用法

对于时序中的每个元素, **TS_DIFFERENCE** 都会返回相应的时序计算区分值; 它调用 IMSL 库中的函数 **imsls_d_difference**。

IMSL 映射

TS_DIFFERENCE 的参数映射到 IMSL 库函数 **imsls_d_difference** 的方式如下:

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

- **n_objs** - 包含当前窗口架构中的行数。
- **z[]** - 包含当前窗口架构的 *timeseries_expression* 值。
- **n_differences** - 映射到 **TS_DIFFERENCE** 中定义的 *period* 参数。
- **period** - **TS_DIFFERENCE** 中定义的 *period* 参数的数组。

有关函数 **imsls_d_difference** 如何执行时序计算的详细信息，请参见 **IMSL C Numerical Library User's Guide: Volume 2 of 2C Stat Library** (《IMSL C 数字库用户指南: 第二卷 (共两卷) C Stat 库》)。

示例

此示例显示了一个包含 **TS_DIFFERENCE** 函数的 SQL 语句，以及该函数返回的数据值。此示例使用示例输入数据表 (名为 *DATASET*) 作为输入数据。请参见 **DATASET** 示例输入数据 (第 72 页)。

下面的 SQL 语句区分 *data* 列中的数据:

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

Sybase IQ 返回 50 行:

表 16. TS_DIFFERENCE 返回的值

res
空值
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

res
0.027546
1.45734
-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

res
0.330772
-1.34311
1.23124
0.209869
0.791146
-0.259155
0.15124
-0.963484
0.383186

注意： 结果的第一行为 NULL，这是因为 IMSL 库为该行返回了一个 *not a number*(NaN) 值。

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

《系统管理指南：第二卷》 > “使用 OLAP”

IMSL C Numerical Library User's Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_DOUBLE_ARRAY 函数 [标量]

TS_GARCH 的支持函数。构造包含 3 - 10 个常量双精度浮点值的逻辑数组，并返回单个 varbinary 值。

语法

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

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **xguess** - 一组常量双精度浮点值。根据 p 和 q 的 **TS_GARCH** 值，生成的 varbinary 编码的逻辑值数组中将有 3 - 10 个值。

用法

支持 **TS_GARCH** 的标量函数。为 **TS_GARCH** *xguess_binary_encoding* 参数生成 *xguess_array*。

IMSL 映射

映射到 **imsls_d_garch** 的 *float xguess[]* 参数。

示例

请参见 **TS_GARCH** 的示例。

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

- 《系统管理指南：第二卷》> “使用 OLAP”
- IMSL C Numerical Library User’s Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_ESTIMATE_MISSING 函数 [集合]

估计时序中的缺失值并将它们作为散布在原始时序中的新时序返回。

语法

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

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

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **timeseries_expression** - 包含要区分的时序中元素的数值表达式 (通常为列名)。如果提供空值，则假定该值反映时序中的间隔，其值将由函数计算。
- **method** - (可选) 用于指定确定缺失值时要使用的方法的整数：

- 0 (缺省值) - 用间隔之前的最后四个时序值和间隔之后的前四个值的中位数估计间隔中的缺失时序观察值。
- 1 - 使用三次样条内插法估计缺失值。在此方法中，同样对间隔之前的最后四个时序值和间隔之后的前四个值执行了内插法。
- 2 - 假定 AR(1) 过程可以很好地描述间隔之前的时序。
- 3 - 使用 AR(p) 模型用一步向前预测估计缺失值。
- **window-spec** - **TS_ESTIMATE_MISSING** 是需要具有未受限制窗口的 **OVER ()** 子句的 **OLAP** 函数。此函数不支持基于值的窗口；例如，在 **OVER ()** 子句中不能使用范围分类符。

用法

使用 **TS_ESTIMATE_MISSING** 通过四个估计方法中的一个来估计任何缺失等距时间点。**TS_ESTIMATE_MISSING** 调用 IMSL 库中的函数 **imsls_d_estimate_missing**

如果时间点集中存在两个以上连续空值，将无法使用 **TS_ESTIMATE_MISSING**。如果时间点集中的前两个值或最后两个值为 NULL，函数将返回 NULL。

IMSL 映射

TS_ESTIMATE_MISSING 的参数映射到 IMSL 库函数 **imsls_d_estimate_missing** 的方式如下：

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

- **n_objs** - 包含当前窗口架构中的行数。
- **tpoints** - 用于指定时间点序列中缺失值的索引数组。
- **z[]** - 累积的 *timeseries_expression*，在调用 *next_value* 时获得。
- **method** - 映射到 **TS_ESTIMATE_MISSING** 中定义的 *method* 参数。

有关函数 **imsls_d_estimate_missing** 如何执行时序计算的详细信息，请参见 **IMSL C Numerical Library User's Guide: Volume 2 of 2 C Stat Library** (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)。

示例

此示例显示一个输入数据表 (即包含 **TS_ESTIMATE_MISSING** 函数的 SQL 语句) 以及该函数返回的数据值。此示例使用下表 (名为 *EST_MISSING_DATASET*) 作为输入数据。**EST_MISSING_DATASET** 表包含 9 行时序数据：

表 17. 输入数据表 EST_MISSING_DATASET

rownum	data
1	2.8223
2	-0.5721

rownum	data
3	2.2771
4	空值
5	1.2648
6	1.0278
7	0.6991
8	-1.7539
9	-2.8875

下面的 SQL 语句估计第 4 行中缺失数据的值：

```
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 返回以下 9 行，并将空值替换为 1.0278：

表 18. TS_ESTIMATE_MISSING 返回的值

res
2.8223
-0.5721
2.2771
1.0278
1.2648
1.0278
0.6991
-1.7539
-2.8875

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

《系统管理指南：第二卷》> “使用 OLAP”

IMSL C Numerical Library User's Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南: 第二卷 (共两卷) C Stat 库》)

TS_GARCH 函数 [集合]

用于分析和预测时序数据的变动性。**TS_GARCH** 计算 GARCH(p, q) 模型的参数估计值。GARCH (广义自回归条件异方差) 是 ARCH 的广义模型; ARCH 计算将误差方差与上一时间段的误差的平方相关联。

语法

```

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

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **timeseries_expression** - 包含要区分的时序中元素的数值表达式 (通常为列名)。
- **garch_count** - 用于计算 GARCH(p,q) 的 p GARCH 参数的常量整数。
- **arch_count** - 用于计算 GARCH(p,q) 的 q ARCH 参数的常量整数。GARCH 计数与 ARCH 计数之和不能超过 9。
- **xguess_binary_encoding** - 表示用于确定模型的种子值 (又称为 *xguess array*) , 编码为通过调用 **TS_DOUBLE_ARRAY** 标量函数生成的常量二进制表达式。数组的长度为 $p+q+1$, 且其中包含 **TS_GARCH_RESULT_USER** 中使用的 x 参数的初始值。第一个元素 (平方值之和的种子) 必须为小于 **max_sigma** 的值的非零正双精度浮点值。其余 p 和 q 种子值必须为大于或等于零的双精度浮点值。它们的和必须小于 1.0。
- **max_sigma** - (可选) 平方值之和的常量双精度浮点值上限。必须是正值。缺省值为 10.0。
- **window-spec** - **TS_GARCH** 是需要 **OVER()** 子句的 OLAP 函数, 该子句包含 **ORDER BY** 子句。**OVER()** 子句中不允许 **ROWS** 或 **RANGE** 分类符。

用法

作为 OLAP 样式的集合函数，**TS_GARCH** 生成单个 SQL 结果 — 一个专门编码的可变长度的二进制结果值。支持的标量函数接受二进制组合输出值并返回其中的各个标量结果值。

因为 OLAP 样式的集合函数为每个输入元组返回一个结果值，所以为分区中的每行返回相同的二进制组合结果。如果未指定 **PARTITION BY** 子句（在 **OVER** 子句中），请使用 **SELECT FIRST** 将结果减少到包含二进制组合结果的单个元组。如果在 **OVER** 子句中指定了 **PARTITION BY** 子句，请使用 **SELECT DISTINCT** 为每个分区只留下其中一个元组，而删除所有其它元组。

IMSL 映射

TS_GARCH_RESULT 支持的标量函数将执行到外部 VNI 库中的 IMSL C 函数参数的映射。

示例

此示例分别为四家指定公司中每一家的股票价格计算 **GARCH(1,2)** 模型。然后，查询使用 **DISTINCT** 限定符将元组集减少为每个股票符号一个元组。最后，返回一个输出行，其中包含股票符号和介绍为该股票计算的 **GARCH(1,2)** 模型的所有相关信息。

TS_GARCH 函数的第二个和第三个参数是 p 和 q 值，指定要使用的模型的 **GARCH(p,q)** 类型。这些值必须为正整数常量或常量表达式。

第四个参数是用于确定 **GARCH(p,q)** 模型的种子值集，编码为必须通过支持的标量函数 **TS_DOUBLE_ARRAY** 生成的二进制常量表达式。

```
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' , 'ZXZ' , 'ZZZ' )
       as dt1
```

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

- 《系统管理指南：第二卷》> “使用 OLAP”
- IMSL C Numerical Library User’s Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_GARCH_RESULT_A 函数 [标量]

TS_GARCH 的支持函数。检索 **TS_GARCH** 集合函数生成的对数似然输出参数 *A*。

语法

```
TS_GARCH_RESULT_A (  
    ts_garch_result  
)
```

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **ts_garch_result** - 通过调用 **TS_GARCH** 集合函数生成的 varbinary 结果参数。

用法

返回生成的对数似然输出参数的双精度浮点值。

IMSL 映射

映射到 **imsls_d_garch** 的 *IMSLA_A, float *a*, 参数。

示例

请参见 **TS_GARCH** 的示例。

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

- 《系统管理指南：第二卷》> “使用 OLAP”
- IMSL C Numerical Library User’s Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_GARCH_RESULT_AIC 函数 [标量]

TS_GARCH 的支持函数。检索 **TS_GARCH** 集合函数生成的阿凯克信息准则输出参数 *AIC*。

语法

```
TS_GARCH_RESULT_AIC (
    ts_garch_result
)
```

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **ts_garch_result** - 通过调用 **TS_GARCH** 集合函数生成的 varbinary 结果参数。

用法

返回生成的阿凯克信息准则输出参数的双精度浮点值。

IMSL 映射

映射到 **imsls_d_garch** 的 *IMSL_AIC*, *float *aic*, 参数。

示例

请参见 **TS_GARCH** 的示例。

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

- 《系统管理指南：第二卷》 > “使用 OLAP”
- IMSL C Numerical Library User’s Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_GARCH_RESULT_USER [标量]

TS_GARCH 的支持函数。访问介绍 GARCH(p,q) 模型的逻辑数组中的每个元素。

语法

```
TS_GARCH_RESULT_USER (
    ts_garch_result
```

```

        ,
        model_element_number
    )

```

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **ts_garch_result** - 通过调用 **TS_GARCH** 集合函数生成的 varbinary 结果参数。
- **model_element_number** - 位于 1 至 $(1+p+q)$ 范围内的整数常量表达式值。

用法

返回指定的模型说明输出值的双精度浮点值。输出集中返回的元素数量为 $p + q + 1$ 。输出集中的第一个元素是生成的平方值之和。随后的 q 值是计算的 ARCH 参数。最后的 p 值是确定的 GARCH 参数。

虽然输出集的大小是可变的（由于 p 和 q 值必须作为常量输入参数传递给 **TS_GARCH**），但模型说明值的数量对于任何特定 **TS_GARCH** 调用都是固定的。

IMSL 映射

映射到 *IMSL_RETURN_USER, float x[], 参数*（属于 **imsls_d_garch**）。

示例

请参见 **TS_GARCH** 的示例。

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

- 《系统管理指南：第二卷》 > “使用 OLAP”
- IMSL C Numerical Library User’s Guide: Volume 2 of 2 C Stat Library（《IMSL C 数字库用户指南：第二卷（共两卷）C Stat 库》）

TS_INT_ARRAY 函数 [标量]

TS_AUTO_ARIMA 和 **TS_AUTO_ARIMA_OUTLIER** 的支持函数。

语法

```

    TS_INT_ARRAY(
        int1
        ,
        int2
    )

```

```

,
  int3
,
  int4
, [ ... [ ,
    int10
  ] ... ] ]

```

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **int1 ... int10** - 一组常量整数值。如果指定了 **TS_AUTO_ARIMA** 的 *model* 参数，则提供四个整数。该组的最大值为十个整数。

注意： 整数值是必需的。将非整数值传递给函数可能会导致意外结果。

用法

返回将指定的整数输入元素编码为逻辑值数组的 **varbinary** 值。

IMSL 映射

如果提供了四个整数并将结果传递给 **TS_AUTO_ARIMA** 或 **TS_AUTO_ARIMA_OUTLIER**，则 **TS_INT_ARRAY** 映射到 IMSL 库中 *imsls_d_auto_arima* 的 (*IMSL_METHOD*, *int method*) 输入参数的 *method 3*。

示例

请参见 **TS_AUTO_ARIMA** 的示例。

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

- 《系统管理指南：第二卷》> “使用 OLAP”
- IMSL C Numerical Library User’s Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_LACK_OF_FIT 函数 [集合]

在给定的适当的相关函数的情况下针对单变量时序或传递函数执行失拟检验。

语法

```

TS_LACK_OF_FIT (timeseries_expression

```

```
, p_value, q_value, lagmax,
[ tolerance])
```

OVER (*window-spec*)

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **timeseries_expression** - 包含时序中元素的数值表达式（通常为列名）。
- **p_value** - 包含自回归参数数量的整数。
- **q_value** - 包含移动平均参数数量的整数。
- **lagmax** - 包含相关函数的最大滞后的整数。
- **tolerance** - （可选）用于确定非线性最小二乘法收敛的浮点值级别。缺省值是 0。
- **window-spec** - **TS_LACK_OF_FIT** 是一个需要 **OVER ()** 子句的 OLAP 函数。

用法

此函数返回包含时序失拟统计 (q) 的双精度浮点值。**TS_LACK_OF_FIT** 调用 IMSL 库中的函数 **imsls_d_lack_of_fit** 。

IMSL 映射

TS_LACK_OF_FIT 的参数映射到 IMSL 库函数 **imsls_d_lack_of_fit** 的方式如下：

```
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** - 包含当前窗口架构中的行数。
- **z[]** - 包含当前窗口架构的 *timeseries_expression* 值。
- **p** - 映射到 **TS_LACK_OF_FIT** 中定义的 *p_value* 参数。
- **q** - 映射到 **TS_LACK_OF_FIT** 中定义的 *q_value* 参数。
- **lagmax** - 映射到 **TS_LACK_OF_FIT** 中定义的 *lagmax* 参数。
- **npfree** - 从 p + q 派生。
- **tolerance** - 使用 **IMSL_CONV_TOL** 的可选参数。如果为 null, IMSL 库会应用缺省值, 不会使用 **IMSL_CONV_TOL**。

有关 IMSL 函数 **imsls_d_lack_of_fit** 如何执行时序计算的详细信息, 请参见 IMSL C Numerical Library User's Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南: 第二卷 (共两卷) C Stat 库》)。

示例

此示例显示了一个包含 **TS_LACK_OF_FIT** 函数的 SQL 语句，以及该函数返回的数据值。此示例使用示例输入数据表（名为 *DATASET*）作为输入数据。请参见 *DATASET* 示例输入数据（第 72 页）。

下面的 SQL 语句返回 *data* 列中数据的失拟统计：

```
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 返回 50 行，每行都包含相同的值：

表 19. TS_LACK_OF_FIT 返回的值

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

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

《系统管理指南：第二卷》> “使用 OLAP”

IMSL C Numerical Library User’s Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_LACK_OF_FIT_P 函数 [集合]

对单变量时序执行失拟检验。此函数与 **TS_LACK_OF_FIT** 相同，只不过 **TS_LACK_OF_FIT_P** 返回的是 *q* 的 *p* 值，而不是返回 *q*。

语法

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

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

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **timeseries_expression** - 包含时序中元素的数值表达式（通常为列名）。
- **p_value** - 包含自回归参数数量的整数。
- **q_value** - 包含移动平均参数数量的整数。
- **lagmax** - 包含相关函数的最大滞后的整数。
- **tolerance** - （可选）用于确定非线性最小二乘法收敛的浮点值级别。缺省值是 0。
- **window-spec** - **TS_LACK_OF_FIT_P** 是一个需要 **OVER ()** 子句的 OLAP 函数。

用法

此函数返回包含时序失拟统计 (*q*) 的 *p* 值的双精度浮点值。**TS_LACK_OF_FIT_P** 调用 IMSL 库中的函数 **imsls_d_lack_of_fit**。

IMSL 映射

TS_LACK_OF_FIT_P 的参数映射到 IMSL 库函数 **imsls_d_lack_of_fit** 的方式如下：

```
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** - 包含当前窗口架构中的行数。
- **z[]** - 包含当前窗口架构的 *timeseries_expression* 值。
- **p** - 映射到 **TS_LACK_OF_FIT_P** 中定义的 *p_value* 参数。
- **q** - 映射到 **TS_LACK_OF_FIT_P** 中定义的 *q_value* 参数。
- **lagmax** - 映射到 **TS_LACK_OF_FIT_P** 中定义的 *lagmax* 参数。
- **npfree** - 从 *p* + *q* 派生。

- **tolerance** – 使用 **IMSL_CONVERGENCE_TOLERANCE** 的可选参数。如果为 null，IMSL 库会应用缺省值，不会使用 **IMSL_CONVERGENCE_TOLERANCE**。

有关 IMSL 函数 **imsls_d_lack_of_fit** 如何执行时序计算的详细信息，请参见 **IMSL C Numerical Library User's Guide: Volume 2 of 2 C Stat Library** (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)。

示例

此示例显示了一个包含 **TS_LACK_OF_FIT_P** 函数的 SQL 语句，以及该函数返回的数据值。此示例使用示例输入数据表 (名为 **DATASET**) 作为输入数据。请参见 **DATASET 示例输入数据** (第 72 页)。

下面的 SQL 语句返回 **data** 列中数据的失拟统计 p 值：

```
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 返回 50 行，每行都包含相同的值：

表 20. TS_LACK_OF_FIT_P 返回的值

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

标准和兼容性

- **SQL** – 符合 ISO/ANSI SQL 标准

- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

《系统管理指南：第二卷》> “使用 OLAP”

IMSL C Numerical Library User's Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_MAX_ARMA_AR 函数 [集合]

计算单变量 ARMA (自回归移动平均值) 时序模型中参数的精确最大似然估计值，并返回请求的自回归估计值。

语法

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

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

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **timeseries_expression** - 包含时序中元素的数值表达式 (通常为列名)。
- **ar_count** - 包含要计算的自回归值数量的整数。
- **ar_elem** - 用于标识将返回已计算的自回归数组中哪个元素的整数。该整数必须大于 0，并小于或等于 *ar_count*。
- **window-spec** - **TS_MAX_ARMA_AR** 是一个需要 **OVER ()** 子句的 OLAP 函数。

用法

此函数会返回包含自回归估计值的双精度浮点值。TS_MAX_ARMA_AR 调用 IMSL 库中的函数 **imsls_d_max_arma**。

IMSL 映射

TS_MAX_ARMA_AR 的参数映射到 IMSL 库函数 **imsls_d_max_arma** 的方式如下：

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

- **n_objs** - 包含当前窗口架构中的行数。
- **z[]** - 包含当前窗口架构的 *timeseries_expression* 值。
- **p** - 映射到 *ar_count* 参数。
- **q** - =1.

有关 IMSL 函数 `imsls_d_max_arma` 如何执行时序计算的详细信息，请参见 IMSL C Numerical Library User's Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)。

示例 1

此示例显示了一个包含 `TS_MAX_ARMA_AR` 函数的 SQL 语句，以及该函数返回的数据值。此示例使用示例输入数据表 (名为 `DATASET`) 作为输入数据。请参见 `DATASET` 示例输入数据 (第 72 页)。

下面的 SQL 语句从包含 `data` 列中数据的两个自回归估计值的数组中返回第二个元素：

```
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 返回 50 行，每行都包含相同的值：

表 21. TS_MAX_ARMA_AR 示例 1 返回的值

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

示例 2

此示例提供一个示例查询，它返回 `DATASET` 表的两列结果，即，自回归估计值的第一个元素和第二个元素。

```
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 返回 50 行数据，每行都包含相同的两个值：

表 22. TS_MAX_ARMA_AR 示例 2 返回的值

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

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

《系统管理指南：第二卷》 > “使用 OLAP”

IMSL C Numerical Library User's Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_MAX_ARMA_CONST 函数 [集合]

计算单变量 ARMA (自回归移动平均值) 时序模型中参数的精确最大似然估计值, 并返回常量估计值。

语法

```
TS_MAX_ARMA_CONST (timeseries_expression
)
```

OVER (*window-spec*)

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **timeseries_expression** - 包含时序中元素的数值表达式（通常为列名）。
- **window-spec** - **TS_MAX_ARMA_CONST** 是一个需要 **OVER ()** 子句的 OLAP 函数。

用法

此函数返回包含常量估计值的双精度浮点值。**TS_MAX_ARMA_CONST** 调用 IMSL 库中的函数 **imsls_d_arma**。

IMSL 映射

TS_MAX_ARMA_CONST 的参数映射到 IMSL 库函数 **imsls_d_arma** 的方式如下：

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

- **n_objs** - 包含当前窗口架构中的行数。
- **z[]** - 包含当前窗口架构的 **timeseries_expression** 值。
- **p** - = 1.
- **q** - = 1.

有关 IMSL 函数 **imsls_d_arma** 如何执行时序计算的详细信息，请参见 **IMSL C Numerical Library User's Guide: Volume 2 of 2 C Stat Library**（《IMSL C 数字库用户指南：第二卷（共两卷）C Stat 库》）。

示例

此示例显示了一个包含 **TS_MAX_ARMA_CONST** 函数的 SQL 语句，以及该函数返回的数据值。此示例使用示例输入数据表（名为 **DATASET**）作为输入数据。请参见 **DATASET** 示例输入数据（第 72 页）。

下面的 SQL 语句返回对 **data** 列的数据进行最大似然自回归计算的常量估计值：

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

Sybase IQ 返回 50 行，每行都包含相同的值：

表 23. TS_MAX_ARMA_CONST 返回的值

res
0.107555

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

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

《系统管理指南：第二卷》 > “使用 OLAP”

IMSL C Numerical Library User’s Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_MAX_ARMA_LIKELIHOOD 函数 [集合]

计算单变量 ARMA (自回归移动平均值) 时序模型中参数的精确最大似然估计值, 并返回拟合模型的似然值 (ln)。

语法

```

TS_MAX_ARMA_LIKELIHOOD (timeseries_expression
)
OVER (window-spec)
    
```

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **timeseries_expression** - 包含时序中元素的数值表达式（通常为列名）。
- **window-spec** - **TS_MAX_ARMA_LIKELIHOOD** 是一个需要 **OVER ()** 子句的 OLAP 函数。

用法

此函数返回包含 $-2 * (\ln(\text{likelihood}))$ 值的双精度浮点值。**TS_MAX_ARMA_LIKELIHOOD** 调用 IMSL 库中的函数 **imsls_d_max_arma**。

IMSL 映射

TS_MAX_ARMA_LIKELIHOOD 的参数映射到 IMSL 库函数 **imsls_d_max_arma** 的方式如下：

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

- **n_objs** - 包含当前窗口架构中的行数。
- **z[]** - 包含当前窗口架构的 **timeseries_expression** 值
- **p** - = 1.
- **q** - = 1.
- **likelihood** - 通过函数调用提供。 包含日志似然结果。

有关 IMSL 函数 **imsls_d_max_arma** 如何执行时序计算的详细信息，请参见 IMSL C Numerical Library User's Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷（共两卷）C Stat 库》)。

示例

此示例显示了一个包含 **TS_MAX_ARMA_LIKELIHOOD** 函数的 SQL 语句，以及该函数返回的数据值。此示例使用示例输入数据表（名为 **DATASET**）作为输入数据。请参见 **DATASET** 示例输入数据（第 72 页）。

下面的 SQL 语句返回对 **data** 列的数据进行最大似然估计的似然值：

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

Sybase IQ 返回 50 行，每行都包含相同的值：

表 24. **TS_MAX_ARMA_LIKELIHOOD** 返回的值

res
-11.7818
-11.7818

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

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

《系统管理指南：第二卷》 > “使用 OLAP”

IMSL C Numerical Library User’s Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_MAX_ARMA_MA 函数 [集合]

计算单变量 ARMA (自回归移动平均值) 时序模型中参数的精确最大似然估计值，并返回请求的移动平均估计值。

语法

```

TS_MAX_ARMA_MA (timeseries_expression
                , ma_count, ma_elem
                )

OVER (window-spec)
    
```

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **timeseries_expression** - 包含时序中元素的数值表达式（通常为列名）。
- **ma_count** - 包含要计算的自回归值的数量的整数。
- **ma_elem** - 用于指定要返回的已计算移动平均值数组中元素的整数。该整数必须大于 0，并小于或等于 *ma_count*。
- **window-spec** - **TS_MAX_ARMA_MA** 是一个需要 **OVER ()** 子句的 OLAP 函数。

用法

此函数返回包含自回归估计值的双精度浮点值。**TS_MAX_ARMA_MA** 调用 IMSL 库中的函数 **imsls_d_max_arma** 。

IMSL 映射

TS_MAX_ARMA_MA 的参数映射到 IMSL 库函数 **imsls_d_max_arma** 的方式如下：

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

- **n_objs** - 包含当前窗口架构中的行数。
- **z[]** - 包含当前窗口架构的 **timeseries_expression** 值。
- **p** - =1.
- **q** - 映射到 **TS_MAX_ARMA_MA** 参数 *ma_count*。

有关 IMSL 函数 **imsls_d_max_arma** 如何执行时序计算的详细信息，请参见 IMSL C Numerical Library User's Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷（共两卷）C Stat 库》)。

示例

此示例显示了一个包含 **TS_MAX_ARMA_MA** 函数的 SQL 语句，以及该函数返回的数据值。此示例使用示例输入数据表（名为 *DATASET*）作为输入数据。请参见 *DATASET* 示例输入数据表（第 72 页）。

下面的 SQL 语句返回对 *data* 列的数据进行最大似然估计的移动平均值：

```
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 返回 50 行，每行都包含相同的值：

表 25. **TS_MAX_ARMA_MA** 返回的值

res
-0.035006
-0.035006
-0.035006

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

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

《系统管理指南：第二卷》> “使用 OLAP”

IMSL C Numerical Library User’s Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_OUTLIER_IDENTIFICATION 函数 [集合]

检测并确定离群值，同时估计某个时序中的模型参数的值，其中，基础无离群值序列遵循常规季节性或非季节性 ARMA 模型。

语法

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

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

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **timeseries_expression** - 包含时序中元素的数值表达式（通常为列名）。
- **p_value** - 包含无离群值序列遵循的自回归集成移动平均值 (ARIMA) $(p, 0, q)x(0, d, 0)_s$ 模型 p 部分的整数。
- **q_value** - 包含无离群值序列遵循的 ARIMA $(p, 0, q)x(0, d, 0)_s$ 模型 q 部分的整数。
- **s_value** - 包含无离群值序列遵循的 ARIMA $(p, 0, q)x(0, d, 0)_s$ 模型 s 部分的整数。
- **d_value** - 包含无离群值序列遵循的 ARIMA $(p, 0, q)x(0, d, 0)_s$ 模型 d 部分的整数。
- **delta_value** - （可选）包含检测临时变化离群时使用的阻尼影响参数的双精度浮点值。该整数必须大于 0 并小于 1。缺省值为 0.7。
- **critical_value** - （可选）用作离群检测阈值的双精度浮点值。缺省值为 3.0。
- **window-spec** - **TS_OUTLIER_IDENTIFICATION** 是需要具有未受限制窗口的 **OVER ()** 子句的 OLAP 函数。此函数不支持基于值的窗口；例如，在 **OVER ()** 子句中不能使用范围分类符。

用法

此函数返回一个无离群值时序。**TS_OUTLIER_IDENTIFICATION** 调用 IMSL 库中的函数 **imsls_d_ts_outlier_identification**。

IMSL 映射

TS_OUTLIER_IDENTIFICATION 的参数映射到 IMSL 库函数 **imsls_d_ts_outlier_identification** 的方式如下：

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

- **n_objs** - 包含当前窗口架构中的行数。
- **model** - 包含 **TS_OUTLIER_IDENTIFICATION** 参数 *p_value*、*s_value*、*q_value*、*d_value* 的数组：`model[0] = p_value`；`model[1] = s_value`；`model[2] = q_value`；`model[3] = d_value`；
- **z[]** - 包含当前窗口架构的 *timeseries_expression* 值。

如果 *delta_value* 非空，则 **TS_OUTLIER_IDENTIFICATION** 的参数映射到 IMSL 库函数 **imsls_d_ts_outlier_identification** 的方式如下：

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

如果 *critical_value* 非空，则 **TS_OUTLIER_IDENTIFICATION** 的参数映射到 IMSL 库函数 **imsls_d_ts_outlier_identification** 的方式如下：

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

如果 *delta_value* 和 *critical_value* 都非空，则 **TS_OUTLIER_IDENTIFICATION** 的参数映射到 IMSL 库函数 **imsls_d_ts_outlier_identification** 的方式如下：

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

有关 IMSL 函数 `imsls_d_ts_outlier_identification` 如何执行时序计算的详细信息，请参见 IMSL C Numerical Library User's Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)。

示例

此示例显示了一个包含 `TS_OUTLIER_IDENTIFICATION` 函数的 SQL 语句，以及该函数返回的数据值。此示例使用示例输入数据表 (名为 `DATASET`) 作为输入数据。请参见 `DATASET` 示例输入数据表 (第 72 页)。

下面的 SQL 语句检测并确定 `data` 列中数据的离群值：

```
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 返回 50 行：

表 26. `ts_outlier_identification` 返回的值

res
0.315523
0.485859
0.676886
1.97381
2.77555
2.73657
2.64233
4.26118
3.13641
4.16566
2.95952
2.14504
1.98799
0.805859
0.833405

res
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
0.614096
0.364668
1.16043
-0.654063
0.616094
2.00875
1.86696
2.80171
3.78422
4.11499

res
2.77188
4.00312
4.21298
5.00413
4.74498
4.89621
3.93273
4.31592

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

《系统管理指南：第二卷》> “使用 OLAP”

IMSL C Numerical Library User’s Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_PARTIAL_AUTOCORRELATION 函数 [集合]

计算固定时序的示例偏自动相关函数。

语法

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

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

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **timeseries_expression** - 包含时序中元素的数值表达式 (通常为列名)。

- **lagmax** - 一个整数，包含要计算的自协方差、自动相关以及自动相关的标准误差的最大滞后。该整数必须大于或等于 1，并小于时序中元素的数量。
- **lag_elem** - 用于标识要返回的自动相关数组中元素的整数。该整数必须大于 0，并小于或等于 *lagmax*。
- **window-spec** - **TS_PARTIAL_AUTOCORRELATION** 是一个需要 **OVER ()** 子句的 OLAP 函数。

用法

此函数返回一个无离群值时序。**TS_PARTIAL_AUTOCORRELATION** 调用 IMSL 库中的函数 **imsls_d_autocorrelation** 和 **imsls_d_partial_autocorrelation**。

IMSL 映射

TS_PARTIAL_AUTOCORRELATION 的参数映射到 IMSL 库函数

imsls_d_autocorrelation 和 **imsls_d_partial_autocorrelation** 的方式如下：

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

- **n_objs** - 包含当前窗口架构中的行数。
- **z[]** - 包含当前窗口架构的 *timeseries_expression* 值。
- **lagmax** - 映射到 **TS_PARTIAL_AUTOCORRELATION** 参数 *lagmax*。

有关 IMSL 函数 **imsls_d_autocorrelation** 和 **imsls_d_partial_autocorrelation** 如何执行时序计算的详细信息，请参见 IMSL C Numerical Library User's Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)。

示例

此示例显示了包含 **TS_PARTIAL_AUTOCORRELATION** 函数的 SQL 语句，以及该函数返回的数据值。此示例使用示例输入数据表 (名为 *DATASET*) 作为输入数据。请参见 *DATASET* 示例输入数据表 (第 72 页)。

下面的 SQL 语句从包含 *data* 列中数据的偏自动相关的数组中返回第一个元素：

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

Sybase IQ 返回 50 行，每行都包含相同的值：

表 27. **TS_PARTIAL_AUTOCORRELATION** 返回的值

res
0.883453
0.883453
0.883453

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

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

《系统管理指南：第二卷》 > “使用 OLAP”

IMSL C Numerical Library User’s Guide: Volume 2 of 2 C Stat Library (《IMSL C 数字库用户指南：第二卷 (共两卷) C Stat 库》)

TS_VWAP 函数 [集合]

VWAP 表示加权平均价。**TS_VWAP** 计算特定时间范围内交易价值与交易总量之间的比率。VWAP 是已定义交易范围的平均股价的测量值。可以使用 **TS_VWAP** 作为简单集合函数和 OLAP 样式集合函数。与其它时序函数不同，**TS_VWAP** 不调用 IMSL 库。

语法 1

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

语法 2

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

OVER (*window-spec*)

许可前提条件

仅可用于 RAP - Trading Edition Enterprise。

参数

- **price_expression** - 指定要并入加权平均值的价格的数值表达式。
- **volume_expression** - 指定计算加权平均值时要使用的量的数值表达式。
- **window-spec** - 如果使用语法 2, **TS_VWAP** 为需要 **OVER ()** 子句的 OLAP 函数。

用法

Sybase IQ 使用以下公式计算 **TS_VWAP**：

图 1: VWAP 计算

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

P_{vwap} = 加权平均价。 P_j = 交易 j 的价格。 Q_j = 交易 j 的数量。 j = 该时间范围内发生的单笔交易。

示例

此示例显示了一个输入数据表、一个包含 **TS_VWAP** 函数的 SQL 语句，以及该函数返回的数据值。此示例使用下表（名为 *VWAP_DATASET*）作为输入数据。

VWAP_DATASET 表包含 3 行时序数据：

表 28. 输入数据表 *VWAP_DATASET*

rownum	price	volume
1	1	1
2	2	2
3	5	1

下面的 SQL 语句计算加权平均价：

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

Sybase IQ 返回 3 行：

表 29. TS_VWAP 返回的值

res
2.5
2.5
2.5

标准和兼容性

- **SQL** - 符合 ISO/ANSI SQL 标准
- **Sybase** - 与 SQL Anywhere 或 Adaptive Server Enterprise 不兼容

另请参见

《系统管理指南：第二卷》> “使用 OLAP”

DATASET 示例输入数据表

时序和预测分析函数示例使用以下表（名为 *DATASET*）作为输入数据。DATASET 表包含 50 行时序数据。

表 30. 输入数据表 DATASET

rownum	data
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

rownum	data
13	1.98799
14	0.805859
15	0.833405
16	2.29075
17	1.30045
18	0.467122
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

rownum	data
41	3.78422
42	4.11499
43	2.77188
44	4.00312
45	4.21298
46	5.00413
47	4.74498
48	4.89621
49	3.93273
50	4.31592

索引

C

错误处理

IMSL 库 6

错误日志记录

IMSL 库 6

H

函数 9

TS_ARMA_AR 函数 11

TS_ARMA_CONST 函数 13

TS_ARMA_MA 函数 16

TS_AUTO_ARIMA 20

TS_AUTO_ARIMA_OUTLIER 23

TS_AUTO_ARIMA_OUTLIER 函数 23

TS_AUTO_ARIMA_RESULT_AIC 25

TS_AUTO_ARIMA_RESULT_AICC 26

TS_AUTO_ARIMA_RESULT_BIC 27

TS_AUTO_ARIMA_RESULT_FORECAST_ERROR
28

TS_AUTO_ARIMA_RESULT_FORECAST_VALUE
29

TS_AUTO_ARIMA_RESULT_MODEL_D 32

TS_AUTO_ARIMA_RESULT_MODEL_P 30

TS_AUTO_ARIMA_RESULT_MODEL_Q 31

TS_AUTO_ARIMA_RESULT_MODEL_S 32

TS_AUTO_ARIMA_RESULT_MODEL_S 函数 33

TS_AUTO_ARIMA_RESULT_RESIDUAL_SIGMA
33

TS_AUTO_UNI_AR 函数 34

TS_AUTOCORRELATION 函数 18

TS_BOX_COX_XFORM 函数 36

TS_DIFFERENCE 函数 39

TS_DOUBLE_ARRAY 42

TS_ESTIMATE_MISSING 函数 43

TS_GARCH 函数 46

TS_GARCH_RESULT_A 48

TS_GARCH_RESULT_AIC 49

TS_GARCH_RESULT_USER 49

TS_INT_ARRAY 50

TS_LACK_OF_FIT 函数 51

TS_LACK_OF_FIT_P 函数 54

TS_MAX_ARMA_AR 函数 56

TS_MAX_ARMA_CONST 函数 58

TS_MAX_ARMA_LIKELIHOOD 函数 60

TS_MAX_ARMA_MA 函数 62

TS_OUTLIER_IDENTIFICATION 函数 64

TS_PARTIAL_AUTOCORRELATION 函数 68

TS_VWAP 函数 70

时序 9

I

IMSL 库

错误处理 6

错误日志记录 6

连接 5

K

库

IMSL 错误处理 6

IMSL 错误日志记录 6

L

连接

IMSL 库 5

S

时序函数 5, 9

IMSL 库 5

错误处理 6

错误日志记录 6

示例输入数据 72

输入数据 72

数据集 72

T

TS_ARMA_AR 函数 11

TS_ARMA_CONST 函数 13

TS_ARMA_MA 函数 16

TS_AUTO_ARIMA 10

TS_AUTO_ARIMA 函数 20

TS_AUTO_ARIMA_OUTLIER 10

TS_AUTO_ARIMA_RESULT_AIC 函数 25
TS_AUTO_ARIMA_RESULT_AICC 函数 26
TS_AUTO_ARIMA_RESULT_BIC 函数 27
TS_AUTO_ARIMA_RESULT_FORECAST_ERROR 函数 28
TS_AUTO_ARIMA_RESULT_FORECAST_VALUE 函数 29
TS_AUTO_ARIMA_RESULT_MODEL_D 函数 32
TS_AUTO_ARIMA_RESULT_MODEL_P 函数 30
TS_AUTO_ARIMA_RESULT_MODEL_Q 函数 31
TS_AUTO_ARIMA_RESULT_RESIDUAL_SIGMA 函数 33
TS_AUTO_UNI_AR 函数 34
TS_AUTOCORRELATION 函数 18
TS_BOX_COX_XFORM 函数 36
TS_DIFFERENCE 函数 39
TS_DOUBLE_ARRAY 函数 42
TS_ESTIMATE_MISSING 函数 43
TS_GARCH 10

TS_GARCH 函数 46
TS_GARCH_RESULT_A 函数 48
TS_GARCH_RESULT_AIC 函数 49
TS_GARCH_RESULT_USER 函数 49
TS_INT_ARRAY 函数 50
TS_LACK OF FIT 函数 51
TS_LACK_OF_FIT_P 函数 54
TS_MAX_ARMA_AR 函数 56
TS_MAX_ARMA_CONST 函数 58
TS_MAX_ARMA_LIKELIHOOD 函数 60
TS_MAX_ARMA_MA 函数 62
TS_OUTLIER_IDENTIFICATION 函数 64
TS_PARTIAL_AUTOCORRELATION 函数 68
TS_VWAP 函数 70

W

文档

SQL Anywhere 4

Sybase IQ 3