IMSL®

C 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

バージョン 7.0

ドキュメント ID ： DC01149-01-1520-01

IMSL® C 数値関数統計ライブラリ

IMSL C 統計ライブラリは、科学的プログラミングで役立つ C 関数を収録したライブラリです。各関数は、研究活動における使用に加え、技術的スペシャリストによる使用も前提として設計され、文書化されています。

C、C#、Java™、Java™、および Fortran
アプリケーション開発ツール

内容

概要

目次

IMSL C 統計ライブラリ

IMSL C 統計ライブラリは、科学的プログラミングで役立つ C 関数を収録したライブラリです。各関数は、研究活動における使用に加え、技術的スペシャリストによる使用も前提として設計され、文書化されています。

このマニュアルの構成

このマニュアルでは、各関数について簡潔に説明します。特定の関数に関連した情報はすべて、各章内に一箇所にまとめてあります。

各章は序文で始まり、その後に、その章で説明する関数をリストした目次が続きます。関数に関する説明は、以下の情報で構成されます。

- セクション名 ： **通常は、*float* 型と *double* 型の関数の共通ルート。**
- 目的 ： **関数の目的に関する記述。**

- シノプシス ： **必須の引数を使用してサブプログラムを参照するための形式。**

  必須の引数 ： 必須の引数についてその出現順に説明したもの。

  入力 ： 引数の初期化が必要です。引数は関数によって変更されません。

  入力／出力 ： 引数の初期化が必要です。関数はこの引数を使って出力を返します。引数に定数または式を指定することはできません。

  出力 ： 初期化は必要ありません。引数に定数または式を指定することはできません。関数はこの引数を使って出力を返します。

- **戻り値** ： 関数によって返される値。
- **オプションの引数を使ったシノプシス** ： 必須の引数とオプションの引数の両方を使用して関数を参照するための形式。
- **オプションの引数** ： オプションの引数についてその出現順に説明したもの。
- **説明** ： アルゴリズムの説明と詳細情報への参照。ほとんどの場合、同様の関数または補完的な関数を持つ他の IMSL 関数について言及されます。
- **エラー** ： 特定の関数に関連して発生する可能性のあるエラーのリスト。エラーのタイプに関する説明は、リファレンス資料の「ユーザ・エラー」項に記載されています。エラーは次のようにタイプ別にリストされます。

  情報エラー ： 関数に関連して発生する可能性のある情報エラーのリスト。

  アラート・エラー ： 関数に関連して発生する可能性のあるアラート・エラーのリスト。

  警告エラー ： 関数に関連して発生する可能性のある警告エラーのリスト。

  致命的エラー ： 関数に関連して発生する可能性のある致命的なエラーのリスト。

  リファレンス ： リファレンスは著者名のアルファベット順にリストされます。

正しい関数を見つける方法

この C 統計ライブラリ・マニュアルは、複数の章に分けられています。各章には、計算機能または分析機能の点で類似した関数がまとめられています。特定の問題に対応した正しい関数を見つけるには、各章の冒頭部分にある目次を利用してください。

命名規則

ほとんどの関数は、*float* と *double* のいずれの型でも使用可能です。これは、これら 2 つの型の関数名が共通ルートを共有するためです。一部の関数は *int* 型でも使用できます。次の一覧は、複数の型で使用できる関数の関数名における、各型に対応するプレフィックスを示したものです。

種類	プレフィックス
float	`imsls_f_`
double	`imsls_d_`
int	`imsls_i_`

関数を見つけやすくするため、関数のセクション名には共通ルートのみが記載されます。

適切な場合は、C 統計ライブラリ全体で同じ変数名を統一して使用します。たとえば、`anova_table` は、分散統計の分析を格納する配列を示し、`y` は、従属変数の応答から成るベクトルを示します。

C 統計ライブラリにアクセスするプログラムを作成する場合は、IMSL 外部名と競合しない C 名を選択してください。名前を選択する際に次の規則に従うと、IMSL 名との競合を防ぐことができます。

- **大文字と小文字の組み合わせにかかわらず、"`imsls_`" で始まる名前は選択しないでください。**

エラー・ハンドリング、アンダフロー、オーバフロー

C 統計ライブラリに含まれている関数は、エラーおよび無効な入力の検出と報告を試みます。このエラー・ハンドリング機能により、ユーザはエラー状態の取り扱いに関する具体的な準備をしなくても、自動的に保護されます。エラーは重大度に基づいて分類され、コード番号を割り当てられます。デフォルトで、重大度が中程度以上のエラーが発生すると、メッセージが関数によって自動的に出力されます。さらに、重大度が最も高いエラーが発生すると、プログラムの実行が停止します。重大度レベルは、エラーの一般的な性質と共に、`IMSLS_FATAL` や `IMSLS_WARNING` などのシンボリック名を持つ「エラー・タイプ」によって指定されます。詳細については、リファレンス資料の「ユーザ・エラー」という項を参照してください。

一般的に、C 統計ライブラリのコードは、システム (ハードウェアまたはソフトウェア) がアンダフローを値 0 で置換することを条件として、計算がアンダフローの影響を受けないように記述されています。通常、アンダフローを示すシステム・エラー・メッセージは無視してかまいません。

IMSL コードは、オーバフローも防ぐように記述されています。オーバフローを示すシステム・エラー・メッセージを生成したプログラムについては、入力データの誤り、引数の型の不一致、不適切な次元などのプログラミング・エラーがないかどうかを調べる必要があります。

多くの場合、関数の説明には、アルゴリズムの不具合の原因となる一般的な注意点が指摘されています。

時系列と予測

ルーチン

使用法についての注釈

この章に示す関数は、時系列に欠測値が含まれていないことを前提としています。欠測値がある場合は、ルーチンによって適切なエラー・メッセージが返されるように、欠測値を NaN に設定する必要があります。モデルの適合を有効にするには、欠測値を適切な推定値で置換する必要があります。

時間領域法

データが定常的なものに変換されると、時間領域内の暫定的モデルがしばしば提示され、パラメータの推定、診断、および予測が実行されます。

ARIMA モデル (自己回帰和分移動平均)

定常な時系列モデルの小規模だが包括的なクラスは、次の式によって定義される非季節的 ARMA 過程で構成されます。

$$\phi(B)\ (W_t - \mu) = \theta(B)A_t, \qquad t \in Z$$

ここで、$Z = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$ は整数のセット、B は $B^k W_t = W_{t-k}$ によって定義される後退シフト作用素、μ は W_t の平均をそれぞれ表し、次の方程式は真です。

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p,\ p \geq 0$$

$$\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \ldots - \theta_q B^q,\ q \geq 0$$

このモデルは次数 (p, q) を持ち、ARMA (p, q) モデルと呼ばれます。

ARMA (p, q) モデルと同じバージョンは、次の式によって定義されます。

$$\phi(B)\ W_t = \theta_0 + \theta(B)A_t, \qquad t \in Z$$

ここで、θ_0 は次の式によって定義される定数項です。

$$\theta_0 = \mu\left(1 - \sum_{i=1}^{p}\phi_i\right)$$

この定数項の意味と効用については、『Box and Jenkins』(1976 年、92 ～ 93 ページ) を参照してください。

「生」データ {Zt} が同次かつ非定常な場合は、imsls_f_difference を使用した差分によって定常性が生じ、モデルは ARIMA (自己回帰和分移動平均) と呼ばれます。パラメータの推定は定常な時系列 $Wt, = \nabla dZt$ に対して実行されます。ここで、$d = (1 – B)$ は、周期 1 と次数 d ($d > 0$) を持つ後退差分演算子です。

通常、モーメント法では、暫定的なパラメータ推定のための関数 imsls_f_arma の呼び出しに引数 IMSLS_METHOD_OF_MOMENTS が使用されます。関数 imsls_f_arma の呼び出しに引数 IMSLS_LEAST_SQUARES を含めることにより、これらの推定値を最小二乗法に対する初期値として使用できます。ユーザによって指定されたその他の初期推定値が使用される場合もあります。最小二乗法を用いると、後戻り予測の長さに基づいて、パラメータの条件付最小二乗推定または無条件最小二乗推定を求めることができます。暫定的なパラメータ推定、最小二乗パラメータ推定、後戻り予測のための関数は、Box and Jenkins 法に従っています (1976 年、プログラム 2 ～ 4、498 ～ 509 ページ)。

arma

ARMA モデルのパラメータの最小二乗推定を計算します。

シノプシス

float *imsls_f_arma (*int* n_observations, *float* z[], *int* p, *int* q, ..., 0)

double 型の関数は imsls_d_arma です。

必須の引数

int n_*observations* (入力)
観測値の数。

float z[] (入力)
観測値を格納する、長さ n_observations の配列。

int p (入力)
自己回帰パラメータの数。

int q (入力)
移動平均パラメータの数。

戻り値

推定定数として AR および MA パラメータを持つ、長さ $1 + p + q$ の配列へのポインタ。`IMSLS_NO_CONSTANT` が指定されている場合、この配列の 0 番目の要素は 0.0 になります。

説明

関数 `imsls_f_arma` は、$t = 1, 2, \ldots, n$ (n = `n_observations`) に対する観測値 $\{W_t\}$ の標本を与えられたときに、非季節的 ARMA モデルのパラメータの推定値を求めます。モーメント法と最小二乗法という 2 つの方法のいずれかを選択できます。デフォルトはモーメント法です。

パラメータ推定法としては、モーメント法と最小二乗法の 2 種類が用意されています。モーメント法のアルゴリズムは、オプションの引数である `IMSLS_METHOD_OF_MOMENTS` と共に使用できます。最小二乗法のアルゴリズムは、`IMSLS_LEAST_SQUARES` を指定する場合に使用されます。最小二乗法のアルゴリズムを選択した場合、暫定的な推定は、デフォルトでモーメント法によって求められます。それ以外の場合は、オプションの引数 `IMSLS_INITIAL_ESTIMATES` を指定することにより、初期推定値を入力できます。次の表に、モーメント法、最小二乗アルゴリズム、およびその両方に適したオプションの引数を示します。

モーメント法のみ	最小二乗法のみ	モーメント法と最小二乗法の両方
IMSLS_METHOD_OF_MOMENTS	IMSLS_LEAST_SQUARES	IMSLS_RELATIVE_ERROR
	IMSLS_CONSTANT (または IMSLS_NO_CONSTANT)	IMSLS_MAX_ITERATIONS
	IMSLS_AR_LAGS	IMSLS_MEAN_ESTIMATE
	IMSLS_MA_LAGS	IMSLS_AUTOCOV(_USER)
	IMSLS_BACKCASTING	IMSLS_RETURN_USER
	IMSLS_CONVERGENCE_TOLERANCE	IMSLS_ARMA_INFO
	IMSLS_INITIAL_ESTIMATES	
	IMSLS_RESIDUAL (_USER)	
	IMSLS_PARAM_EST_COV (_USER)	
	IMSLS_SS_RESIDUAL	

モーメント法による推定

時系列 {*Z*t} が次の形式の ARMA (*p*, *q*) モデルによって生成されるとします。

$$\phi(B)Zt = \theta 0 + \theta(B)At$$

for $t \in \{0, \pm 1, \pm 2, \ldots\}$

$\hat{\mu}$ = z_mean が時系列 {*Z*t} の平均 μ の推定値だと仮定します。ここで、$\hat{\mu}$ は次の値と等しくなります。

$$\hat{\mu} = \begin{cases} \mu & \text{for } \mu \text{ known} \\ \frac{1}{n}\sum_{t=1}^{n} Z_t & \text{for } \mu \text{ unknown} \end{cases}$$

自己共分散関数は次の式によって推定されます。

$$\hat{\sigma}(k) = \frac{1}{n}\sum_{t=1}^{n-k}(Z_t - \hat{\mu})(Z_{t+k} - \hat{\mu})$$

for $k = 0, 1, \ldots, K$ (ここで $K = p + q$)。$\hat{\sigma}$ (0) は、標本分散の推定値です。

標本自己共分散が与えられている場合、この関数は、Yule-Walker 方程式の拡張版を使って、自己回帰パラメータのモーメント法推定を求めます。

$$\hat{\Sigma}\hat{\phi} = \hat{\sigma}$$

各パラメータの意味は次のとおりです。

$$\hat{\phi} = \left(\hat{\phi}_1, \ldots, \hat{\phi}_p\right)^T$$

$$\hat{\Sigma}_{ij} = \hat{\sigma}\left(|q + i - j|\right), \; i, j = 1, \ldots, p$$

$$\hat{\sigma}_i = \hat{\sigma}\left(q + i\right), \qquad i = 1, \ldots, p$$

ここで、定数項 $\theta 0$ は次の式によって推定されます。

$$\hat{\theta}_0 = \begin{cases} \hat{\mu} & \text{for } p = 0 \\ \hat{\mu}\left(1 - \sum_{i=1}^{p} \hat{\phi}_i\right) & \text{for } p > 0 \end{cases}$$

$K = p + q + 1$ 自己共分散、$k = 1, \ldots, K$ の場合の $\sigma(k)$、および $i = 1, \ldots, p$ の場合の p 自己回帰パラメータ ϕi が指定されている場合、移動平均パラメータは、非線形方程式系に基づいて推定されます。

$Z't = \phi(B)Zt$ と仮定します。ここから派生する移動平均過程 $Z't = \theta(B)At$ の自己共分散は、次の関係によって推定されます。

$$\hat{\sigma}'(k) = \begin{cases} \hat{\sigma}(k) & \text{for } p = 0 \\ \sum_{i=0}^{p} \sum_{j=0}^{p} \hat{\phi}_i \hat{\phi}_j \left(\hat{\sigma}\left(|k + i - j|\right)\right) & \text{for } p \geq 1, \hat{\phi}_0 \equiv -1 \end{cases}$$

移動平均パラメータを決定するための反復法は、次の関係に基づいています。

$$\sigma(k) = \begin{cases} \left(1 + \theta_1^2 + \ldots + \theta_q^2\right)\sigma_A^2 & \text{for } k = 0 \\ \left(-\theta_k + \theta_1\theta_{k+1} + \ldots + \theta_{q-k}\theta_q\right)\sigma_A^2 & \text{for } k \geq 1 \end{cases}$$

ここで、$\sigma(k)$ は、元の Zt 過程の自己共分散関数を示します。

$\tau = (\tau 0, \tau 1, \ldots, \tau q)T$ および $f = (f0, f1, \ldots, fq)T$ と仮定します。各パラメータの意味は次のとおりです。

$$\tau_j = \begin{cases} \sigma_A & \text{for } j = 0 \\ -\theta_j / \tau_0 & \text{for } j = 1, \ldots, q \end{cases}$$

および

$$f_j = \sum_{i=0}^{q-j} \tau_i \tau_{i+j} - \hat{\sigma}'(j) \qquad \text{for } j = 0, 1, \ldots, q$$

これにより、$(i+1)$ 回目の反復時の τ の値が次の式によって決定されます。

$$\tau^{i+1} = \tau^i - \left(T^i\right)^{-1} f^i$$

この推定手順は、次の初期値で開始されます。

$$\tau^0 = (\sqrt{\hat{\sigma}'(0)}, \quad 0, \ldots, 0)^T$$

そして、i 回目の反復時に $||f\,\mathrm{i}||$ が `relative_error` より小さいか、i が `max_iterations` と等しくなると、中断します。移動平均パラメータの推定値は、次のように設定することによって、τ の最終推定値から取得されます。

$$\hat{\theta}_j = -\tau_j / \tau_0 \ \text{for}\ j = 1, \ldots, q$$

確率ショック分散は次の式によって推定されます。

$$\hat{\sigma}_A^2 = \begin{cases} \hat{\sigma}(0) - \sum_{i=1}^{p} \hat{\phi}_i \hat{\sigma}(i) & \text{for } q = 0 \\ \tau_0^2 & \text{for } q \geq 0 \end{cases}$$

同様の計算を実行する関数の説明については、『Box and Jenkins』(1976 年、498 ～ 500 ページ)を参照してください。

最小二乗法による推定

時系列 $\{Zt\}$ が次の形式の非季節的 ARMA モデルによって生成されるとします。

$$\phi(B)\ (Zt - \mu) = \theta(B)At \qquad \text{for t} \in \{0, \pm 1, \pm 2, \cdots\}$$

ここで、B は後退シフト作用素、μ は Zt の平均です。

$$\phi\left(B\right) = 1 - \phi_1 B^{l_\phi(1)} - \phi_2 B^{l_\phi(2)} - \ldots - \phi_p B^{l_\phi(p)} \quad \text{for } p \geq 0$$
$$\theta\left(B\right) = 1 - \theta_1 B^{l_\theta(1)} - \theta_2 B^{l_\theta(2)} - \ldots - \theta_q B^{l_\theta(q)} \quad \text{for } q \geq 0$$

p と q は、それぞれ自己回帰パラメータと移動平均パラメータの数です。一般性を失わずに、次の条件が仮定されます。

$$1 \le l\mathrm{f}(1) \le l\mathrm{f}(2) \le \cdots \le l\mathrm{f}(p)$$

$$1 \le l\mathrm{q}(1) \le l\mathrm{q}(2) \le \cdots \le l\mathrm{q}(q)$$

これにより、この非季節的 ARMA モデルの次数は (p', q') になります。ここで、$p' = l\mathrm{q}(p)$、$q' = l\mathrm{q}(q)$ です。通常の階層的モデルでは、次のように仮定されます。

$$l\mathrm{f}(i) = i,\ 1 \le i \le p$$

$$l\mathrm{q}(j) = j,\ 1 \le j \le q$$

次のような平方和関数を考えてみましょう。

$$S_T(\mu,\phi,\theta) = \sum_{-T+1}^{n} [A_t]^2$$

各パラメータの意味は次のとおりです。

$$[A_t] = E\left[A_t \middle| (\mu,\phi,\theta,Z)\right]$$

T は後退原点です。確率ショック $\{A\mathrm{t}\}$ は、独立かつ等分布の次の確率変数であると仮定されます。

$$N\left(0,\sigma_A^2\right)$$

そのため、尤度関数は次のように表されます。

$$l(\mu,\phi,\theta,\sigma_A) = f(\mu,\phi,\theta) - n\ln(\sigma_A) - \frac{S_T(\mu,\phi,\theta)}{2\sigma_A^2}$$

ここで、$f(\mu, \phi, \theta)$ は μ、ϕ、および θ の関数です。

$T = 0$ の場合、尤度関数は、モデルを初期化するために必要となる Zt および At の両方の過去の値を条件とします。通常、これらの初期値の選択方法は、過渡的なバイアスをモデルで発生させる原因となります (『Box and Jenkins』1976 年、210 ～ 211 ページ)。$-T = \infty$ の場合、この依存性はなくなり、推定値が尤度関数の最大化に基づいて求められます。『Box and Jenkins』(1976 年、213 ページ) では、

$$S_\infty(\mu,\phi,\theta)/\left(2\sigma_A^2\right)$$

という項が次の項より優位であると解説されています。

$$l\left(\mu,\phi,\theta,\sigma_A^2\right)$$

平方和関数を最小化するパラメータ推定のことを、最小二乗推定と呼びます。n の値が大きい場合、無条件最小二乗推定は最尤推定と近似します。

実際的には、T に有限値を指定すると、無条件平方和関数の十分な近似が可能になります。無条件平方和を求めるために必要な $[AT]$ の値は、後戻り予測によって取得される Zt の初期値を使って反復的に計算されます。残差 (後戻り予測値を含む)、確率ショック分散の推定値、および最終パラメータ推定値の共分散行列も求められます。ARIMA パラメータは、imsls_f_difference を imsls_f_arma と共に使用して求めることができます。

警告エラー

IMSLS_LEAST_SQUARES_FAILED	最小二乗法によるパラメータの推定が収束できませんでした。“maxbc”、“tolerance”、“convergence_tolerance” の値を必要に応じて増やしてください。最終反復時のパラメータ推定値を新しい初期値として使用できます。

致命的エラー

IMSLS_TOO_MANY_CALLS	関数呼び出しの回数が “itmax”*(“n”+1) = %(i1) を超過しました。新しい初期推定値を使用してください。
IMSLS_INCREASE_ERRREL	相対誤差の誤差限界 “errrel” = %(r1) が小さすぎます。近似解の精度をこれ以上向上させることは不可能です。“errrel” の値を増やす必要があります。
IMSLS_NEW_INITIAL_GUESS	反復計算の進捗に問題があります。新しい初期推定値を使用してください。

max_arma

単変量 ARMA (自己回帰移動平均) 時系列モデルにあるパラメータの正確な最尤推定。

シノプシス

float *imsls_f_max_arma (*int* n_obs, *float* w[], *int* p, *int* q,…,0)

double 型の関数は imsls_d_max_arma です。

必須の引数

int n_obs (入力)
時系列内の観測値の数。

float w[] (入力)
時系列を格納する、長さ n_obs の配列。

int p (入力)
自己回帰パラメータの数を表す非負の値。

int q (入力)
移動平均パラメータの数を表す非負の値。

戻り値

推定定数として AR および MA パラメータを持つ、長さ 1+p+q の配列へのポインタ。値を算出できない場合は、NULL が返されます。

オプションの引数を使ったシノプシス

float *imsls_f_max_arma (*int* n_obs, *float* w[], *int* p, *int* q,
IMSLS_INITIAL_ESTIMATES, *float* init_ar[] *float* init_ma[],
IMSLS_PRINT_LEVEL, *int* iprint,
IMSLS_MAX_ITERATIONS, *int* maxit,
IMSLS_LOG_LIKELIHOOD, *float* *log_likeli,
IMSLS_VAR_NOISE, *float* *avar,
IMSLS_ARMA_INFO, *Imsls_f_arma* **arma_info,
IMSLS_MEAN_ESTIMATE, *float* *w_mean,
IMSLS_RETURN_USER, *float* *constant, *float* ar[], *float* ma[],
0)

オプションの引数

IMSLS_INITIAL_ESTIMATES, *float* init_ar[], *float* init_ma[] (入力)

この引数が指定されている場合、init_ar は、自己回帰パラメータの暫定的推定値を格納する、長さ p の配列、init_ma は、移動平均パラメータの暫定的推定値を格納する、長さ q の配列です。指定されない場合は、内部で計算されます。p=0 または q=0 の場合、対応する引数は無視されます。

IMSLS_PRINT_LEVEL, *int* iprint (入力)

出力オプション ：

0 ー 出力なし。

1 ー 最終結果のみ出力。

2 ー 中間結果と最終結果を出力。

デフォルト ： iprint = 0

IMSLS_MAX_ITERATIONS, *int* maxit (入力)

反復推定の最大数。

デフォルト ： maxit = 300

IMSLS_VAR_NOISE, *float* *avar (出力)

イノベーション分散の推定値。

IMSLS_LOG_LIKELIHOOD, *float* *log_likeli (出力)

適合モデルに対する -2*(ln(likelihood)) の値。

IMSLS_ARMA_INFO, *Imsls_f_arma* **arma_info (出力)

imsls_f_arma_forecast の呼び出しに必要な情報を格納する、型 *Imsls_f_arma* の内部割り当て構造へのポインタのアドレス。

IMSLS_MEAN_ESTIMATE, *float* *w_mean (入力／出力)

時系列 w の平均の推定値。出力時、w_mean には平均の更新式が格納されます。

デフォルト ： 時系列 w の中心が標本平均付近に置かれます。

IMSLS_RETURN_USER, *float* *constant, *float* ar[], *float* ma[] (出力)

この引数が指定されている場合、constant は、定数パラメータの推定値、ar は、自己回帰パラメータの最終推定値を格納する、長さ p の配列、ma は、移動平均パラメータの最終推定値を格納する、長さ q の配列です。

説明

関数 imsls_f_max_arma は、赤池、北川、荒畑、多田の 4 氏によって提案された最尤推定アルゴリズム (1979 年)、および TIMSAC-78 ライブラリで公開されている XSARMA ルーチンから派生したものです。

この章の冒頭で触れた時間領域法の表記法を使うと、平均 W_t を持つ定常時系列 μ は、非季節的自己回帰移動平均 (**ARMA**) モデルによって次の関係として表すことができます。

$$\phi(B)(W_t - \mu) = \theta(B)a_t$$

各パラメータの意味は次のとおりです。

$$t \in ZZ = \{\cdots, -2, -1, 0, 1, 2, \cdots\}$$

B は次の 3 つの関係によって定義される後退シフト作用素です。

$$B^k W_t = W_{t-k}$$

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p, \quad p \geq 0$$

および

$$\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q, \quad q \geq 0$$

関数 imsls_f_max_arma は、最尤推定を用いて、AR 係数 $\phi_1, \phi_2, \cdots, \phi_p$ と **MA** 係数 $\theta_1, \theta_2, \cdots, \theta_q$ を推定します。

関数 imsls_f_max_arma は、自己回帰係数と移動平均係数の両方の初期推定値を調べて、それぞれが定常な時系列と反転可能な時系列を表していることを確認します。

$$\phi_1, \phi_2, \cdots, \phi_p$$

上記の係数が定常系列の初期推定値である場合、次の多項式のすべての (複素) 根は単位円から外れます。

$$1 - \phi_1 z - \phi_2 z^2 - \cdots - \phi_p z^p$$

$$\theta_1, \theta_2, \cdots, \theta_q$$

上記の係数が反転可能系列の初期推定値である場合、次の多項式のすべての (複素) 根は単位円から外れます。

$$1-\theta_1 z-\theta_2 z^2-\cdots-\theta_q z^q$$

AR 係数と MA 係数の初期値は、ベクトル `init_ar` と `init_ma` によって供給されます。そうでない場合、推定値はモーメント法に基づいて内部で計算されます。`imsls_f_max_arma` は、陪多項式の根を求めます。**AR** 推定値が非定常な系列を表す場合、`imsls_f_max_arma` は警告メッセージを発行して、`init_ar` を定常な初期値で置換します。**MA** 推定値が反転不可能な系列を表す場合は、`imsls_f_max_arma` によってターミナル・エラーが発行され、新しい初期値を見つけることが必要となります。

`imsls_f_max_arma` は、**AR** 係数の最終推定値の検証も行って、推定値も定常系列を表していることを確認します。これは、内部の尤度最適化が非定常解に収束される可能性を防ぐために行われます。非定常推定が発生した場合、`imsls_f_max_arma` は致命的エラー・メッセージを発行します。

AIC の最小値が次のように設定された ARMA モデルを優先的に使用できます。

$$AIC=\texttt{log\_likeli}+2(\mathrm{p+q})$$

関数 `imsls_f_max_arma` は、ホワイト・ノイズ過程、つまり ARMA(0,0) 過程も処理できます。

auto_uni_ar

単変量自己回帰時系列モデルの自動選択と適合。このモデルのラグは、赤池情報量規準 (AIC) を用いて自動的に選択されます。最小の AIC を持つモデルの自己回帰パラメータの推定値は、モーメント法、最小二乗法、または最尤法に従って求められます。

シノプシス

float `*imsls_f_auto_uni_ar` (*int* `n_obs`, *float* `z[]`, *int* `maxlag`,
int `*p`,…,0)

double 型の関数は `imsls_d_auto_uni_ar` です。

必須の引数

int n_obs (入力)
  時系列内の観測値の数。

float z[] (入力)
  定常時系列を格納する、長さ n_obs の配列。

int maxlag (入力)
  要求された自己回帰パラメータの最大数。次の範囲に収まっている必要があります。
  1≤ maxlag ≤ n_obs/2

int *p (出力)
  最小の AIC を持つモデル内の自己回帰パラメータの数。

戻り値

最小の AIC を持つモデル内の定数パラメータと自己回帰パラメータの推定値を格納する、長さ 1+ maxlag のベクトル。推定値は、この配列の最初の 1+ p 個の位置に含まれています。

オプションの引数を使ったシノプシス

```
float *imsls_f_auto_uni_ar (int n_obs, float z[], int maxlag, int *p,
    IMSLS_PRINT_LEVEL, int iprint,
    IMSLS_MAX_ITERATIONS, int maxit,
    IMSLS_METHOD, int method,
    IMSLS_VAR_NOISE, float *avar,
    IMSLS_AIC, float *aic,
    IMSLS_MEAN_ESTIMATE, float *z_mean,
    IMSLS_RETURN_USER, float *constant, float ar[],
    0)
```

オプションの引数

IMSLS_PRINT_LEVEL, *int* iprint (入力)
  出力オプション ：
  0 ー 出力なし。
  1 ー 最終結果のみ出力。
  2 ー 中間結果と最終結果を出力。
  デフォルト ： iprint = 0

IMSLS_MAX_ITERATIONS, *int* maxit (入力)

反復推定の最大数。

デフォルト ： maxit = 300

IMSLS_METHOD, *int* method (入力)

推定法オプション ：

0 ― モーメント法

1 ― ハウスホルダー変換によって実現する最小二乗法

2 ― 最尤法

デフォルト ： method = 1

IMSLS_VAR_NOISE, *float* *avar (出力)

イノベーション分散の推定値。

IMSLS_AIC, *float* *aic (出力)

最小 AIC。

IMSLS_MEAN_ESTIMATE, *float* *z_mean (入力／出力)

時系列 z の平均の推定値。出力時、z_mean には平均の更新式が格納されます。

デフォルト ： 時系列 z の中心が標本平均付近に置かれます。

IMSLS_RETURN_USER, *float* *constant, *float* ar[] (出力)

この引数が指定されている場合、constant は、定数パラメータの推定値、ar は、最初の p 個の位置に自己回帰パラメータの最終推定値を格納する、長さ maxlag の配列です。

説明

関数 auto_uni_ar は、データに最も適合する AR モデルの次数を自動的に選択してから、AR 係数を求めます。auto_uni_ar で使われているアルゴリズムは、赤池弘次その他 (1979 年)、および北川・赤池両氏 (1978 年) による研究結果から導き出されたものです。このコードは、TIMSAC-78 ライブラリの一部として公開されている UNIMAR 手順に基づいて作成されました。

最適 AR モデルは、0, 1, 2, ..., maxlag という自己回帰係数を使って AR モデルを連続的に適合させることによって決定されます。各モデルの赤池情報量規準 (AIC) は、次の式に基づいて求められます。

$$AIC = -2\ln(likelihood) + 2(\mathrm{p}+1)$$

関数 auto_uni_ar は、尾崎、小田によって開発された次の式 (1979 年) に近似を適用します。

$$AIC = (\mathrm{n\_obs} - \mathrm{maxlag})\ln(\hat{\sigma}^2) + 2(\mathrm{p}+1) + (\mathrm{n\_obs} - \mathrm{maxlag})(\ln(2\pi)+1)$$

ここで、$\hat{\sigma}^2$ は、系列の残差分散の推定であり、時系列解析ではイノベーション分散として知られています。次の定数を削除すると、

$$(\mathrm{n\_obs} - \mathrm{maxlag})(\ln(2\pi)+1)$$

計算は次のように簡略化されます。

$$AIC = (\mathrm{n\_obs} - \mathrm{maxlag})\ln(\hat{\sigma}^2) + 2(\mathrm{p}+1)$$

最適モデルは、最小 AIC を持つモデルです。このモデル内のパラメータの数が、適合された最高次数の自己回帰モデル (つまり p=maxlag) と等しい場合は、maxlag の値を大きくしたときに、さらに小さい AIC を持つモデルが存在する可能性があります。その場合は、maxlag の値を大きくして、追加の自己回帰パラメータで AR モデルを検証することが認められる可能性があります。

method = 0 の場合、最小の AIC を持つモデルの自己回帰係数の推定は、モーメント法を使って計算されます。method =1 の場合、係数は、北川と赤池の両氏によって 1978 年に提案された形式の最小二乗法によって決定されます。それ以外の場合、つまり method =2 の場合、係数は最尤法を用いて推定されます。

ts_outlier_identification

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。

シノプシス

float *imsls_f_ts_outlier_identification (*int* n_obs, *int* model[],
float w[],…,0)

double 型の関数は imsls_d_ts_outlier_identification です。

必須の引数

int n_obs (入力)
　時系列内の観測値の数。

int model[] (入力)
　異常値のない系列が従う ARIMA $(p,0,q)\times(0,d,0)_s$ モデルの数値 *p*、*q*、*s*、*d* を格納する、長さ 4 のベクトル。

float w[] (入力)
　時系列を格納する、長さ n_obs の配列。

戻り値

異常値のない時系列を格納する、長さ n_obs の配列へのポインタ。
エラーが発生した場合は、NULL が返されます。

オプションの引数を使ったシノプシス

```
float *imsls_f_ts_outlier_identification (int n_obs,
    int model[], float w[],
    IMSLS_RETURN_USER, float x[],
    IMSLS_DELTA, float delta,
    IMSLS_CRITICAL, float critical,
    IMSLS_EPSILON, float epsilon,
    IMSLS_RELATIVE_ERROR, float relative_error,
    IMSLS_RESIDUAL, float **residual,
    IMSLS_RESIDUAL_USER, float residual[],
    IMSLS_RESIDUAL_SIGMA, float *res_sigma,
    IMSLS_NUM_OUTLIERS, int *num_outliers,
    IMSLS_OUTLIER_STATISTICS, int **outlier_stat,
    IMSLS_OUTLIER_STATISTICS_USER, int outlier_stat[],
    IMSLS_TAU_STATISTICS, float **tau_stat,
    IMSLS_TAU_STATISTICS_USER, float tau_stat[],
    IMSLS_OMEGA_WEIGHTS, float **omega,
    IMSLS_OMEGA_WEIGHTS_USER, float omega[],
    IMSLS_ARMA_PARAM, float **parameters,
    IMSLS_ARMA_PARAM_USER, float parameters[],
    IMSLS_AIC, float *aic,
    0)
```

オプションの引数

IMSLS_RETURN_USER, *float* x[] (出力)
: 異常値のない系列を格納する、長さ n_obs の配列。ユーザによって指定されます。

IMSLS_DELTA, *float* delta (入力)
: 一時的変化 (TC) の異常値を検出する際に使用される減衰効果パラメータ、0<delta < 1。
デフォルト : delta = 0.7

IMSLS_CRITICAL, *float* critical (入力)
: 異常値の検出のスレッショルドとして使用される臨界値、critical > 0。
デフォルト : critical = 3.0

IMSLS_EPSILON, *float* epsilon (入力)
: 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。
デフォルト : epsilon = 0.001

IMSLS_RELATIVE_ERROR, *float* relative_error (入力)
: 関数 imsls_f_arma で使われる非線形方程式ソルバーの停止基準。
デフォルト : relative_error = 10^{-10}

IMSLS_RESIDUAL, *float* **residual (出力)
: 異常値のない系列の残差を格納する、長さ n_obs の内部割り付け配列へのポインタのアドレス。

IMSLS_RESIDUAL_USER, *float* residual[] (出力)
: ユーザによって指定される配列 residual の記憶領域。「IMSLS_RESIDUAL」を参照してください。

IMSLS_RESIDUAL_SIGMA, *float* *res_sigma (出力)
: 異常値のない系列の残差標準誤差。

IMSLS_NUM_OUTLIERS, *int* *num_outliers (出力)
: 検出された異常値の数。

IMSLS_OUTLIER_STATISTICS, *int* **outlier_stat (出力)
: 異常値統計を格納する、長さ num_outliers × 2 の内部割り付け配列へのポインタのアドレス。最初のカラムには、異常値が観測された時刻 (t=1,2,...,n_obs) が格納され、2 番目のカラムには、観測された異常値のタイプを示す識別子が格納されます。
異常値のタイプは、次の 5 つに分類されます。

0　イノベーション異常値 (IO)

1　加法異常値 (AO)

2　水準変化異常値 (LS)

3　一時的変化異常値 (TC)

4　確定不能 (UI)

IMSLS_NUM_OUTLIERS (検出された異常値の数) を取得するには、IMSLS_NUM_OUTLIERS を使用します。
num_outliers = 0 の場合は、NULL が返されます。

IMSLS_OUTLIER_STATISTICS_USER, *int* outlier_stat[] (出力)
最初の num_outliers 個の位置の異常値統計を格納する、長さ n_obs × 2 のユーザ割り付け配列。「IMSLS_OUTLIER_STATISTICS」を参照してください。
num_outliers = 0 の場合、outlier_stat は不変のままです。

IMSLS_TAU_STATISTICS, *float* **tau_stat (出力)
検出された各異常値の t 値を格納する、長さ num_outliers の内部割り付け配列へのポインタのアドレス。
num_outliers = 0 の場合は、NULL が返されます。

IMSLS_TAU_STATISTICS_USER, *float* tau_stat[] (出力)
最初の num_outliers 個の位置で検出された各異常値の t 値を格納する、長さ n_obs のユーザ割り付け配列。
num_outliers = 0 の場合、tau_stat は不変のままです。

IMSLS_OMEGA_WEIGHTS, *float* **omega (出力)
検出された異常値に対して計算された重み ω を格納する、長さ num_outliers の内部割り付け配列へのポインタのアドレス。
num_outliers = 0 の場合は、NULL が返されます。

IMSLS_OMEGA_WEIGHTS_USER *float* omega[] (出力)
最初の num_outliers 個の位置で検出された異常値に対して計算された重み ω を格納する、長さ n_obs のユーザ割り付け配列。
num_outliers = 0 の場合、omega は不変のままです。

`IMSLS_ARMA_PARAM`, *float* `**parameters` (出力)
: 推定された不変 AR パラメータおよび MA パラメータを格納する、長さ 1+*p*+*q* の内部割り付け配列へのポインタのアドレス。

`IMSLS_ARMA_PARAM_USER` *float* `parameters[]` (出力)
: 推定された不変 AR および MA パラメータを格納する、長 1+*p*+*q* のユーザ割り付け配列。

`IMSLS_AIC`, *float* `*aic` (出力)
: 赤池情報量規準 (AIC)。

説明

次数 $\{Y_t\}$ の次の乗法季節的 ARIMA モデルによって説明できる、単変量時系列 $(p,0,q)\times(0,d,0)_s$ があるとします。

$$Y_t-\mu=\frac{\theta(B)}{\Delta_s^d\phi(B)}a_t,\ t=1,\ldots,n$$

ここで、$\Delta_s^d=(1-B^s)^d$ 、$\theta(B)=1-\theta_1B-\ldots-\theta_qB^q$, $\phi(B)=1-\phi_1B-\ldots-\phi_pB^p$ です。B はラグ演算子、$B^kY_t=Y_{t-k}$ 、$\{a_t\}$ はホワイト・ノイズ過程であり、μ は系列 $\{Y_t\}$ の平均を表します。

一般的に、$\{Y_t\}$ は、異常値の影響のため直接的には観測できません。Chen と Liu (1993 年) は、異常値を、イノベーション異常値 (IO)、加法異常値 (AO)、一時的変化 (TC)、水準変化 (LS) の 4 種類に分類しています。系列の最後の観測値として異常値が発生した場合、Chen と Liu のアルゴリズムでは、異常値の種類を特定できません。`imsls_f_ts_outlier_identification` では、このような異常値を UI (確定不能) と呼び、イノベーション異常値として処理します。

$t_1,t_2,\ldots,t_m$ という複数の時点で発生した複数の異常値の影響を考慮に入れるため、Chen と Liu は次のモデルを使用しています。

$$Y_t^*-\mu=\sum_{j=1}^m\omega_jL_j(B)I_t(t_j)+\frac{\theta(B)}{\Delta_s^d\phi(B)}a_t$$

ここで、$\{Y_t^*\}$ は観測された異常値の影響を受ける系列であり、ω_j と $L_j(B)$ はそれぞれ、異常値 j の大きさと動的パターンを表します。$I_t(t_j)$ は、異常値の影響の一時的経過を決定する指示関数であり、$I_{t_j}(t_j)=1$ または $I_t(t_j)=0$ の値をとります。$L_j(B)$ は I_t を介して $B^k I_t = I_{t-k}, k=0,1,\ldots$ に対して演算を実行します。

最後の式は、発生している異常値の影響をすべて取り除くことにより、異常値のない系列 $\{Y_t\}$ を元の系列 $\{Y_t^*\}$ から取得できることを示しています。

$$Y_t = Y_t^* - \sum_{j=1}^m \omega_j L_j(B) I_t(t_j)$$

異常値のタイプの違いに基づいて、$L_j(B)$ の値は次のように異なります。

1. $L_j(B) = \dfrac{\theta(B)}{\Delta_s^d \phi(B)}$ (イノベーション異常値)
2. $L_j(B) = 1$ (加法異常値)
3. $L_j(B) = (1-B)^{-1}$ (水準変化異常値)
4. $L_j(B) = (1-\delta B)^{-1}, 0 < \delta < 1,$ (一時的変化異常値)

関数 `imsls_f_ts_outlier_identification` は、Chen と Liu のアルゴリズムに基づいています。この関数は、観測対象の系列について、$\phi(B), \theta(B)$ の係数とモデル内の異常値の影響をまとめて 3 段階で特定します。異常値の影響の大きさは、最小二乗推定によって決定されます。異常値の検出そのものは、異常値の影響に関する標準化統計の最大値を検証することで行われます。詳細については、Chen と Liu の論文 (1993 年) を参照してください。

係数 $\phi(B)$ と $\theta(B)$ の中間推定値と最終推定値は、関数 `imsls_f_arma` と `imsls_f_max_arma` によって計算されます。$\phi(B)$ または $\theta(B)$ の根が単位円上または単位円内にある場合、アルゴリズムは適切なエラー・メッセージを発行して停止します。この場合は、*p* と *q* に対して別の値を指定する必要があります。

auto_arima

時系列異常値を自動的に識別し、乗法的季節 ARIMA $(p,0,q)\times(0,d,0)_s$ モデルのパラメータを決定し、系列の終端を超えて影響が持続する異常値の影響を組み込んで予測値を生成します。

シノプシス

float *imsls_f_auto_arima (*int* n_obs, *int* tpoints[], *float* x[],...,0)

double 型の関数 は imsls_d_auto_arima です。

必須の引数

int n_obs (入力)
元の時系列内の観測値の数。系列が時点 $t_1, t_2, \ldots t_{n\_obs}$ で定義されている場合、欠測値を含めた系列の実際の長さは $n = t_{\mathrm{n\_obs}} - t_1 + 1$ です。

int tpoints[] (入力)
時系列が観測された時点 $t_1, t_2, \ldots t_{n\_obs}$ を格納する、長さ n_obs のベクトル。
$t_1, t_2, \ldots t_{n\_obs}$ は、厳密に昇順で指定する必要があります。

float x[] (入力)
観測された時系列値 $t_i + 1, t_i + 2, \ldots, t_{i+1} - 1$ を格納する、長さ n_obs のベクトル。この系列には、異常値と欠測値を含めることができます。このルーチンによって異常値が識別され、ベクトル tpoints 内の時刻値によって欠損値が識別されます。2 つの連続した時点間の時間間隔が 1 より大きい場合 (つまり、$Y_1^*, Y_2^*, \cdots, Y_{n\_obs}^*$ である場合)、時間 $t_{i+1} - t_i = m > 1$ において、$m-1$ と t_i の間に欠損値 t_{i+1} が存在すると想定されます。したがって、等間隔の時点にはギャップのない系列が定義されると想定されます。欠損値は、異常値を識別して予測値を生成する前に自動的に推定されます。予測値は、欠損値と観測値の両方に対して生成されます。

戻り値

ARIMA $(p,0,q)\times(0,d,0)_s$ モデルを使用する異常値のない系列を適合するために使用される、推定定数として AR および MA パラメータを持つ、長さ $1+p+q$ の配列へのポインタ。処理の完了後に *d*=`model[3]`=0 となった場合、ARMA(*p, q*) モデルまたは AR(*p*) モデルは、異常値のない観測対象の系列 Y_t^* に適合されます。*d*=`model[3]`>0 となった場合、これらのパラメータは、季節ごとに調整される系列 $Z_t^* = \Delta_s^d \cdot Y_t^* = (1-B_s)^d \cdot Y_t^*$ の ARMA(*p,q*) 表現に対して計算されます ($B_s Y_t^* = Y_{t-s}^*$ および *s*=`model[2]`≥1)。

エラーが発生した場合は、`NULL` が返されます。

オプションの引数を使ったシノプシス

```
float *imsls_f_auto_arima (int n_obs, int tpoints[], float x[],
    IMSLS_METHOD, int method,
    IMSLS_MAX_LAG, int maxlag,
    IMSLS_MODEL, int model[],
    IMSLS_DELTA, float delta,
    IMSLS_CRITICAL, float critical,
    IMSLS_EPSILON, float epsilon,
    IMSLS_RESIDUAL_SIGMA, float *res_sigma,
    IMSLS_P_INITIAL, int n_p_initial, int p_initial[],
    IMSLS_Q_INITIAL, int n_q_initial, int q_initial[],
    IMSLS_S_INITIAL, int n_s_initial, int s_initial[],
    IMSLS_D_INITIAL, int n_d_initial, int d_initial[],
    IMSLS_OUTLIER_STATISTICS, int **outlier_stat,
    IMSLS_OUTLIER_STATISTICS_USER, int outlier_stat[],
    IMSLS_AIC, float *aic,
    IMSLS_AICC, float *aicc,
    IMSLS_BIC, float *bic,
    IMSLS_MODEL_SELECTION_CRITERION, int criterion,
    IMSLS_CONFIDENCE, float confidence,
    IMSLS_NUM_PREDICT, int n_predict,
    IMSLS_OUTLIER_FORECAST, float **outlier_forecast,
    IMSLS_OUTLIER_FORECAST_USER, float outlier_forecast[],
    0)
```

オプションの引数

IMSLS_METHOD, *int* method (入力)

モデルの選択に使用する方法 :

1 − 自動的な ARIMA $(p,0,0)\times(0,d,0)_s$ の選択

3 − 指定の ARIMA $(p,0,q)\times(0,d,0)_s$ モデル

引数 IMSLS_MODEL が必要。

デフォルト : method = 1

IMSLS_MAX_LAG, *int* maxlag (入力)

AR(*p*) モデルを適合する場合の最大ラグ。

デフォルト : maxlag = 10

IMSLS_MODEL, *int* model[] (入力／出力)

p、*q*、*s*、*d* の値を格納する、長さ 4 の配列。method=3 を選択した場合は、*p* と *q* の値を定義する必要があります。IMSLS_S_INITIAL と IMSLS_D_INITIAL を定義していない場合は、*s* と *d* も指定する必要があります。method=1 の場合、model は入力配列として無視されます。出力に関して、model は、model[0]、model[1]、および model[3] における *p*、*q*、*s*、*d* の最適値をそれぞれ格納します。

IMSLS_DELTA, *float* delta (入力)

一時的変化 (TC) の異常値を検出する際に使用される減衰効果パラメータ、0<delta < 1。

デフォルト : delta = 0.7

IMSLS_CRITICAL, *float* critical (入力)

異常値の検出のスレッショルドとして使用される臨界値、critical > 0。

デフォルト : critical = 3.0

IMSLS_EPSILON, *float* epsilon (入力)

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。

デフォルト : epsilon = 0.001

IMSLS_RESIDUAL_USER, float *residual[]* (出力)

ユーザによって指定される配列 residual の記憶領域。

IMSLS_RESIDUAL_SIGMA, *float* *res_sigma (出力)
: 異常値のない元の系列の残差標準誤差 (RSE)。

IMSLS_NUM_OUTLIERS, *int* *num_outliers (出力)
: 検出された異常値の数。

IMSLS_P_INITIAL, *int* n_p_initial, *int* p_initial[] (入力)
: n_p_initial 個の要素を持ち、最適値の選択対象である *p* の候補値を格納する配列。p_initial[] のすべての候補値は、負の値でなく、かつ n_p_initial ≥ 1 でなければなりません。method=2 である場合は、IMSLS_P_INITIAL を指定する必要があります。それ以外の場合、n_p_initial と p_initial は無視されます。

IMSLS_Q_INITIAL, *int* n_q_initial, *int* q_initial[] (入力)
: n_q_initial 個の要素を持ち、最適値の選択対象である *q* の候補値を格納する配列。q_initial[] のすべての候補値は、負の値でなく、かつ n_q_initial ≥ 1 でなければなりません。method=2 である場合は、IMSLS_Q_INITIAL を指定する必要があります。それ以外の場合n_q_initial と q_initial は無視されます。

IMSLS_S_INITIAL, *int* n_s_initial, *int* s_initial[] (入力)
: 最適値の選択対象である s の候補値を格納する、長さ n_s_initial のベクトル。s_initial[] のすべての候補値は、正の値で、かつ n_s_initial ≥ 1 でなければなりません。
デフォルト ： n_s_initial=1, s_initial={1}

IMSLS_D_INITIAL, *int* n_d_initial, *int* d_initial[] (入力)
: 最適値の選択対象である *d* の候補値を格納する、長さ n_d_initial のベクトル。d_initial[] のすべての候補値は、負ではない値で、かつ n_d_initial ≥ 1 でなければなりません。
デフォルト ： n_d_initial=1, d_initial={0}

IMSLS_OUTLIER_STATISTICS, *int* **outlier_stat (出力)
: 異常値統計を格納する、長さ num_outliers × 2 の内部割り付け配列へのポインタのアドレス。最初のカラムには、異常値が観測された時刻 ($t = t_1, t_1+1, t_1+2, \ldots, t_{\text{n\_obs}}$) が格納され、2 番目のカラムには、観測された異常値のタイプを示す識別子が格納されます。異常値のタイプは、次の 5 つに分類されます。

0	イノベーション異常値 (IO)
1	加法異常値 (AO)

2 水準変化異常値 (LS)

3 一時的変化異常値 (TC)

4 確定不能 (UI)

num_outliers=0 の場合は、NULL が返されます。

IMSLS_OUTLIER_STATISTICS_USER, *int* outlier_stat[] (出力)

最初の num_outliers 個のローの異常値統計を格納する、長さ $n \times 2$ のユーザ割り付け配列。ここで、$n = t_{\mathrm{n\_obs}} - t_1 + 1 \geq \mathrm{n\_obs}$ です。

num_outliers = 0 の場合、outlier_stat は不変のままです。

IMSLS_AIC, *float* *aic (出力)

最適なモデルに対する赤池情報量規準 (AIC) 値。系列のイノベーション分散の推定値に基づいた対数最尤推定の近似値を使用します。

IMSLS_AICC, *float* *aicc (出力)

最適なモデルに対する補正 AIC (AICC) 値。系列のイノベーション分散の推定値に基づいた対数最尤推定の近似値を使用します。

IMSLS_BIC, *float* *bic (出力)

最適なモデルに対するベイズ情報量規準 (BIC) 値。系列のイノベーション分散の推定値に基づいた対数最尤推定の近似値を使用します。

IMSLS_MODEL_SELECTION_CRITERION, *int* criterion (入力)

最適なモデル選択のために使用される情報量規準。

規準	選択される情報量規準
0	赤池情報量規準 (AIC)
1	赤池補正情報量規準 (AICC)
2	ベイズ情報量規準 (BIC)

デフォルト : criterion = 0.

IMSLS_OUT_FREE_SERIES_USER, *float* outfree_series[] (出力)

長さ n × 2 のユーザ割り付け配列 ($n = t_{\mathrm{n\_obs}} - t_1 + 1$)。

IMSLS_CONFIDENCE, *float* confidence (入力)

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。

デフォルト : confidence = 95.0

IMSLS_NUM_PREDICT, *int* n_predict (入力)

要求された予測の数。予測は、原点 $t_{\mathrm{n\_obs}}$ で (つまり、最後に観測された系列の値から) 生成されます。

デフォルト : n_predict = 0

IMSLS_OUTLIER_FORECAST, *float* **outlier_forecast (出力)

長さ n_predict × 3 の内部割り付け配列へのポインタのアドレス。最初のカラムには、$t = t_{\mathrm{n\_obs}}+1$, $t_{\mathrm{n\_obs}}+2$,..., $t_{\mathrm{n\_obs}}$+n_predict に対する元の系列の予測値が格納されます。

2 番目のカラムにはこれらの予測に対する標準エラーが格納され、3 番目のカラムにはモデルの無限次移動平均形式の ψ 重みが格納されます。

IMSLS_OUTLIER_FORECAST_USER, *float* outlier_forecast[] (出力)

長さ n_predict × 3 のユーザ割り付け配列。

IMSLS_RETURN_USER, *float* x[] (出力)

最初の 1+*p*+*q* 個の位置に、推定定数として AR および MA パラメータを格納するユーザ割り付け配列。値 *p* と *q* は、上限値によって推定できます。method=1 の場合、*p* の上限値は maxlag 、かつ *q*= 0 であると考えられます。method=2 の場合、*p* および *q* の上限値はそれぞれ、配列 p_initial および q_initial の最大値であると考えられます。method=3 の場合は、*p*= model[0]、*q*= model[1] となります。

説明

関数 imsls_f_auto_arima は、乗法季節的 ARIMA $(p,0,q)\times(0,d,0)_s$ モデルのパラメータを決定し、適合モデルを使用して異常値の識別と予測値の作成を行います。このモデルの次数は指定できますが、自動的に決定することもできます。

imsls_f_auto_arima によって処理される ARIMA $(p,0,q)\times(0,d,0)_s$ モデルの形式は、次のとおりです。

$$\phi(B)\Delta_s^d(Y_t - \mu) = \theta(B)a_t, \quad t = 1, 2, \ldots, n,$$

各パラメータの意味は次のとおりです。

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p, \quad \theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q, \quad \Delta_s^d = (1 - B^s)^d$$

および

$$B^k Y_t = Y_{t-k}$$

$\phi(B)$ と $\theta(B)$ のすべての根は、単位円の外部にあることが前提となります。明らかなことは、$s = 1$ である場合、これが従来の ARIMA(*p, d, q*) モデルに要約されることです。

Y_t は、平均 μ を持つ、非観測の、異常値のない時系列であり、ホワイト・ノイズは a_t です。このモデルは、異常値のない基礎モデルと呼ばれます。関数 `imsls_f_auto_arima` は、この系列を観測できることを前提としません。観測される値は 1 つ以上の異常値の影響を受ける可能性があり、その影響は異常値のない基礎系列に追加されることを前提とします。

$$Y_t^* = Y_t + outlier\_effect_t$$

異常値の識別では、Chen と Liu によって開発されたアルゴリズム (1993 年) を使用します。異常値は、次の 5 つのタイプのいずれかに分類されます。

1. イノベーション
2. 加法
3. 水準変化
4. 一時的変化
5. 確定不能

異常値が識別されると、`imsls_f_auto_arima` は、推定される異常値の影響を取り除くことにより、異常値のないデータ系列の表現である Y_t を推定します。

続いて、調整された ARIMA$(p,0,q)\times(0,d,0)_s$ モデルと取り除かれた異常値に関する情報に基づいて、異常値のない系列に対して予測値が作成されます。この予測値に異常値の影響を追加することにより、観測された系列の予測値 Y_t^* が生成されます。異常値がない場合、異常値のない系列と観測された系列の予測値は同一になります。

モデルの選択

ユーザは、p、q、s、d に固有の値を指定するか、imsls_f_auto_arima を使用して最適な値を自動的に決定するかを選択します。モデルの選択は、変数 method に基づいて、以下に示す 2 つの方法のいずれかを使用して行います。

方法 1 : 自動 ARIMA $(p,0,0)\times(0,d,0)_s$ 選択

この方法では、ノイズを含むデータの 最小 AIC が含まれる AR(p) 表現が最初に検索されます。ここで、p は 0,...,maxlag です。

IMSLS_D_INITIAL を定義している場合、s_initial と d_initial の値は検索対象となり、系列の最適な ARIMA $(p,0,0)\times(0,d,0)_s$ 表現が検索されます。ここでは、s_initial の p と s 、および d_initial の d の値のすべての可能な組み合わせが調べられます。その結果、検索された最適な ARIMA $(p,0,0)\times(0,d,0)_s$ が異常値検出ルーチンの入力として使用されます。

p、q、s、d の最適値は、それぞれ model[0]、model[1]、model[2]、model[3] 内で返されます。

方法 2 : 指定の ARIMA $(p,0,q)\times(0,d,0)_s$ モデル

2 番目の方法では、p、q、s、d に対して特定の値を指定します。p と q の値は、それぞれ model[0] と model[1] で定義する必要があります。IMSLS_S_INITIAL と IMSLS_D_INITIAL を定義していない場合は、model[2] と model[3] 内で値 $d \geq 0$ と $s > 0$ を指定する必要があります。IMSLS_S_INITIAL と IMSLS_D_INITIAL を定義している場合、s_initial と d_initial の入力値のすべての可能な組み合わせを使用して s と d の最適値のグリッド検索が実行されます。s および d の最適値は、それぞれ model[2] および model[3] 内で見つかります。

異常値

Chen と Liu のアルゴリズム (1993 年) は、異常値を識別するために使用されます。識別された異常値の数は、num_outliers に返されます。これらの異常値の時刻および分類は、outlier_stat[] に返されます。異常値は、各異常値タイプの標準化統計に基づいて、5 つのカテゴリのいずれかに分類されます。異常値の発生時刻は、outlier_stat の最初のカラムに示されます。次の表に、2 番目のカラムに返される異常値の識別子の概要を示します。

異常値識別子	名前	概要
0	(IO) イノベーション異常値	イノベーション異常値は持続します。つまり、異常値の発生時に初期影響があります。この影響は、その後のすべての観測で遅れて持続します。遅延係数は、基礎となる ARIMA $(p,0,q)\times(0,d,0)_s$ モデルの係数によって決まります。
1	(AO) 加法異常値	加法異常値は持続しません。名前が示すように、加法異常値は、異常値の発生時の観測だけに影響します。このため、加法異常値は、その後の予測値に影響を与えません。
2	(LS) 水準変化	水準変化異常値は持続します。異常値の発生時以降、系列の平均が上昇したり下降したりするという影響が現れます。この平均の変化は、突然発生して永続します。
3	(TC) 一時的変化	一時的変化異常値は持続し、1 つの重要な例外を除き、水準変化異常値に似ています。水準変化異常値と同様に、この異常値の発生時に、系列の平均に急激な変化が現れます。しかし、水準変化異常値とは異なり、この変化は持続しません。TC 異常値は徐々に衰え、系列の平均は、最終的には元の値に戻ります。この減衰率は、パラメータ `delta` を使用してモデル化されます。デフォルトの `delta=`0.7 は、Chen と Liu (1993 年) による一般的な用途での推奨値です。
4	(UI) 確定不能	最後の観測値として異常値が識別された場合、このアルゴリズムでは、異常値の種類を特定できません。予測では、UI 異常値は IO 異常値として処理されます。つまり、その効果は遅れて予測値に現れます。

加法異常値 (AO) を除いて、異常値の影響は、その異常値に続く観測値まで持続します。`imsls_f_auto_arima` によって生成される予測では、これが考慮されています。

difference

季節時系列と非季節時系列の差異を計算します。

シノプシス

float *imsls_f_difference (*int* n_observations, *float* z[], *int* n_differences, *int* periods[], ..., 0)

double 型の関数は imsls_d_difference です。

必須の引数

int n_observations (入力)
観測値の数。

float z[] (入力)
時系列を格納する、長さ n_observations の配列。

int n_differences (入力)
実行する差分の数。引数 n_differences には 1 以上の値を指定する必要があります。

int periods[] (入力)
z に対して差分を実行する期間を格納する、長さ n_differences の配列。

戻り値

差分された系列を格納する、長さ n_observations の配列へのポインタ。

オプションの引数を使ったシノプシス

float *imsls_f_difference (*int* n_observations, *float* z[], *int* n_differences, *int* periods[],
IMSLS_ORDERS, *int* orders[],
IMSLS_LOST, *int*v*n_lost,
IMSLS_EXCLUDE_FIRST, *or*
IMSLS_SET_FIRST_TO_NAN,
IMSLS_RETURN_USER, *float* w[],
0)

オプションの引数

IMSLS_ORDERS, *int* orders[] (入力)
期間に分けて指定された各差分の次数を格納する、長さ n_differences の配列。次数の要素数には 0 以上の値を指定する必要があります。

IMSLS_LOST, *int* *n_lost (出力)

時系列 z を差分したために失われた観測値の数。

IMSLS_EXCLUDE_FIRST または

IMSLS_SET_FIRST_TO_NAN

IMSLS_EXCLUDE_FIRST が指定されている場合は、最初の n_lost 個が差分のために w から除外されます。差分された系列 w の長さは n_observations − n_lost です。IMSLS_SET_FIRST_TO_NAN が指定されている場合は、最初の n_lost 個の観測値が NaN (非数) に設定されます。これは、IMSLS_EXCLUDE_FIRST と IMSLS_SET_FIRST_TO_NAN のいずれも指定されていない場合のデフォルトです。

IMSLS_RETURN_USER, *float* w[] (出力)

この引数が指定されている場合、w には差分された系列が格納されます。IMSLS_EXCLUDE_FIRST も指定されている場合、w の長さは n_observations になります。IMSLS_SET_FIRST_TO_NAN が指定されているか、IMSLS_EXCLUDE_FIRST と IMSLS_SET_FIRST_TO_NAN のいずれも指定されていない場合、w の長さは n_observations − n_lost です。

説明

関数 imsls_f_difference は、期間 s_i = periods [$i-1$] と次数 d_i = orders [$i-1$] である、m = n_differences 回の連続した後退差分を、n = n_observations 個の観測値 $\{Z_t\}$ に対して実行します。ここで、$i = 1, \ldots, m$、$t = 1, 2, \ldots, n$ です。

すべての k の値に対して次のように定義される後退シフト作用素 B があるとします。

$$B^k Z_t = Z_{t-k}$$

この場合、期間 s の後退シフト作用素は次のように定義されます。

$$\Delta_s Z_t = (1 - B^s) Z_t = Z_t - Z_{t-s} \quad \text{ここで} \quad s > 0$$

$B^s Z_t$ と $\Delta^s Z_t$ は $t = (s + 1), \ldots, n$ に対してのみ定義されています。期間 s の反復差分は次のように表されます。

$$\Delta_s^d Z_t = \left(1 - B^s\right)^d Z_t = \sum_{j=0}^{d} \frac{d!}{j!\left(d-j\right)!} \left(-1\right)^j B^{sj} Z_t$$

ここで、$d \geq 0$ は差分の次数です。

$$\Delta_s^d Z_t$$

が $t = (sd + 1), \ldots, n$ に対してのみ定義されていることに注意してください。

関数 `imsls_f_difference` で使われる一般的な差分式は次のようになります。

$$W_t = \begin{cases} \text{NaN} & \text{for } t = 1, \ldots, n_L \\ \Delta_{s_1}^{d_1} \Delta_{s_2}^{d_2} \ldots \Delta_{s_m}^{d_m} Z_t & \text{for } t = n_L + 1, \ldots, n \end{cases}$$

ここで、nL は、差分が原因で『失われた』観測値の数を表し、NaN は、欠損値のコードを表します。k が 0 の場合は、次のように定義されます。

$$n_L = \sum_j s_j d_j$$

同次の定常時系列は、同次の非定常時系列を適切に差分することによって得られます (『Box and Jenkins』1976 年、85 ページ)。系列に適切な変換を予備的に適用した後で差分を実行すると、同次かつ定常な自己回帰移動平均モデルのクラスにおける、モデルの同定とパラメータの推定が可能になります。

致命的エラー

`IMSLS_PERIODS_LT_ZERO`	“period[#]” = #。“period” のすべての要素を 0 よりも大きくする必要があります。
`IMSLS_ORDER_NEGATIVE`	“order[#]” = #。“order” のすべての要素を非負の値にする必要があります。
`IMSLS_Z_CONTAINS_NAN`	“z[#]” = NaN。“z” には欠損値を格納できません。“z” の他の要素が NaN と等しくなる場合もあります。

box_cox_transform

前方または逆 Box-Cox ベキ変換を実行します。

シノプシス

float *imsls_f_box_cox_transform (*int* n_observations, *float* z[], *float* power, ..., 0)

double 型の関数は imsls_d_box_cox_transform です。

必須の引数

int n_observations (入力)
z 内の観測値の数。

float z[] (入力)
観測値を格納する、長さ n_observations の配列。

float power (入力)
Box-Cox ベキ変換における指数パラメータ。

戻り値

変換されたデータを格納する、長さ n_observations の内部割り付け配列へのポインタ。このスペースを解放するには、imsls_free を使用します。値を算出できない場合は、NULL が返されます。

オプションの引数を使ったシノプシス

float *imsls_f_box_cox_transform (*int* n_observations, *float* z[], *float* power,
IMSLS_SHIFT, *float* shift,
IMSLS_INVERSE_TRANSFORM,
IMSLS_RETURN_USER, *float* x[],
0)

オプションの引数

`IMSLS_SHIFT`, *float* `shift` (入力)

Box-Cox ベキ変換におけるシフト・パラメータ。パラメータ shift は min ($z(i)$) + `shift` > 0 という関係を満たす必要があります。

デフォルト ： `shift` = 0.0

`IMSLS_INVERSE_TRANSFORM`

`IMSLS_INVERSE_TRANSFORM` が指定されている場合は、逆変換が実行されます。

`IMSLS_RETURN_USER`, *float* `x[]` (出力)

変換されたデータを格納する、長さ `n_observations` のユーザ割り付け配列。

説明

関数 `imsls_f_box_cox_transform` は、n = `n_observations` 個の観測値 $\{Z_t\}$ に対して、前方または逆 Box-Cox ベキ変換を実行します。ここで、$t = 1, 2, \ldots, n$ です。

前方変換は、線形モデル、または非正規誤差や非一定分散を持つモデルの解析に有用です (『Draper and Smith』1981 年、222 ページ)。時系列解析の場合、系列を適切に変換した後で差分すると、同次かつ定常な自己回帰移動平均モデルのクラスにおける、モデルの同定とパラメータの推定が可能になります。予測や予測限界などの特定の解析結果に対して後で逆変換を適用すると、元のデータの尺度で結果を表現することが可能になります。時系列モデルで使用できる変換については、『Box and Jenkins』(1976 年、328 ページ) で概説されています。

Box および Cox によって 1964 年に提案されたベキ変換のクラスは、次のように定義されます。

$$X_t = \begin{cases} \dfrac{\left(Z_t + \xi\right)^{\lambda} - 1}{\lambda} & \lambda \neq 0 \\ \ln\left(Z_t + \xi\right) & \lambda = 0 \end{cases}$$

この場合、t の値にかかわらず $Z_t + \xi > 0$ です。

$$\lim_{\lambda \to 0} \frac{\left(Z_t + \xi\right)^{\lambda} - 1}{\lambda} = \ln\left(Z_t + \xi\right)$$

上記の条件が真であるため、ベキ変換は継続的です。

λ = `power` および ξ = `shift` と仮定した場合、`imsls_f_box_cox_transform` によって使用される計算式は次のように表されます。

$$X_t = \begin{cases} (Z_t + \xi)^{\lambda} & \lambda \neq 0 \\ \ln(Z_t + \xi) & \lambda = 0 \end{cases}$$

この場合、t の値にかかわらず Zt + $\xi > 0$ です。計算式と Box-Cox 式の違いは、変換対象データの尺度と取得元のみです。そのため、データの一般的な解析は影響を受けません (『Draper and Smith』1981 年。225 ページ)。

逆変換は次のように計算されます。

$$X_t = \begin{cases} Z_t^{1/\lambda} - \xi & \lambda \neq 0 \\ exp(Z_t) - \xi & \lambda = 0 \end{cases}$$

ここで、{Zt} は、パラメータ λ と ξ を使って `imsls_f_box_cox_transform` によって計算された、元のデータに対する前方変換の結果を表します。

致命的エラー

`IMSLS_ILLEGAL_SHIFT`	“shift” = # であり、“z” の最小要素が “z[#]” = # です。“shift” = # です。*i* = 1, ..., “n_observations” の場合、“shift” + “z[i] ” には、0 より大きい値を指定する必要があります。“n_observations” = # です。
`IMSLS_BCTR_CONTAINS_NAN`	NaN (非数) と等しい “z” の要素が 1 つまたは複数あります。欠損値があってはなりません。NaN と等しい “z” の要素の最小インデックスが # です。
`IMSLS_BCTR_F_UNDERFLOW`	前方変換。“power” = # です。“shift” = # です。“z” の最小要素が “z” is “z[#]” = # です。(“z[#]”+ “shift”) ^ “power” がアンダフローします。
`IMSLS_BCTR_F_OVERFLOW`	前方変換。“power” = # です。“shift” = # です。“z” の最大要素が “z[#]” = # です。(“z[#]” + “shift”) ^ “power” がオーバフローします。

IMSLS_BCTR_I_UNDERFLOW	逆変換。"power" = # です。"z" の最小要素が "power" = # です。exp("z[#]") がアンダフローします。
IMSLS_BCTR_I_OVERFLOW	逆変換。"power" = # です。"z" の最大要素が "z[#]" = # です。exp("z[#]") がオーバフローします。
IMSLS_BCTR_I_ABS_UNDERFLOW	逆変換。"power" = # です。最小絶対値を持つ "z" の要素が "power" = # です。"z[#]" = # がアンダフローします。
IMSLS_BCTR_I_ABS_OVERFLOW	逆変換。"power" = # です。最大絶対値を持つ "z" の要素が "power" = # です。"z[#]" ^ (1/ "power") がオーバフローします。

autocorrelation

定常時系列のサンプル自己相関関数を計算します。

シノプシス

float `*imsls_f_autocorrelation` (*int* `n_observations`, *float* `x[]`,
int `lagmax`, ...0)

double 型の関数は `imsls_d_autocorrelation` です。

必須の引数

int `n_observations` (入力)
時系列 `x` 内の観測値の数。`n_observations` には、2 以上の値を指定する必要があります。

float `x[]` *(*入力)
時系列を格納する、長さ `n_observations` の配列。

int `lagmax` (入力)
計算する自己共分散、自己相関、および自己相関の標準誤差の最大ラグ。`lagmax` には、1 以上 `n_observations` 未満の値を指定する必要があります。

戻り値

時系列 `x` の自己相関を格納する、長さ `lagmax` + 1 の配列へのポインタ。この配列の *0* 番目の要素は 1 です。この配列の *k* 番目の要素には、ラグ *k* の自己相関が格納されます。ここで、*k* は 1, ..., `lagmax` です。

オプションの引数を使ったシノプシス

```
float *imsls_f_autocorrelation (int n_observations, float x[],
    int lagmax,
    IMSLS_PRINT_LEVEL, int iprint,
    IMSLS_X_MEAN_IN, float x_mean_in,
    IMSLS_X_MEAN_OUT, float *x_mean_out,
    IMSLS_ACV, float **autocovariances,
    IMSLS_ACV_USER, float autocovariances[],
    IMSLS_SEAC, float **standard_errors, int se_option,
    IMSLS_SEAC_USER, float standard_errors[], int se_option,
    IMSLS_RETURN_USER, float autocorrelations[],
    0)
```

オプションの引数

IMSLS_PRINT_LEVEL, *int* iprint （入力)

出力オプション ：
デフォルトは 0 です。

入力	対処法
0	出力は実行されません。
1	平均と分散を出力します。
2	平均、分散、および自己共分散を出力します。
3	平均、分散、自己共分散、自己相関、および自己相関の標準誤差を出力します。

IMSLS_X_MEAN_IN, *float* x_mean_in (入力)

ユーザによって入力された、時系列 x の推定値。

IMSLS_X_MEAN_OUT, *float* *x_mean_out (出力)

この引数が指定されている場合、x_mean_out は、
時系列 x の平均の推定値です。

IMSLS_ACV, *float* **autocovariances （出力)

時系列 x の分散と自己共分散を格納する、長さ lagmax + 1 の配列へのポインタのアドレス。この配列の *0* 番目の要素は、時系列 x の分散です。*k* 番目の要素には、ラグ *k* の自己共分散が格納されます。ここで、*k* は 1, ..., lagmax です。

IMSLS_ACV_USER, *float* autocovariances[] (出力)

この引数が指定されている場合、autocovariances は、時系列 x の分散と自己共分散を格納する、長さ lagmax + 1 の配列です。

IMSLS_SEAC, *float* **standard_errors, *int* se_option (出力)

時系列 x の自己相関の標準誤差を格納する、長さ lagmax の配列へのポインタのアドレス。自己相関の標準誤差の計算方法は、se_option によって選択されます。

se_option	**対処法**
1	Barlett の公式を使って autocorrelations の標準誤差を計算します。
2	Moran の公式を使って autocorrelations の標準誤差を計算します。

IMSLS_SEAC_USER, *float* standard_errors[], *int* se_option (出力)

この引数が指定されている場合、autocovariances は、時系列 x の自己相関の標準誤差を格納する、長さ lagmax の配列です。

IMSLS_RETURN_USER, *float* autocorrelations[] (出力)

この引数が指定されている場合、autocorrelations は、時系列 x の自己相関を格納する、長さ lagmax + 1 の配列です。この配列の *o* 番目の要素は 1 です。この配列の *k* 番目の要素には、ラグ ***k*** の自己相関が格納されます。ここで、***k*** は 1, ..., lagmax です。

説明

関数 `imsls_f_autocorrelation` は、`n_observations` 個の観測値 {*X*t} から成る標本を与えられたときに、定常時系列の自己相関関数を推定します。ここで、*t* は 1, 2, …, *n* です。

時系列 {Xt} の平均　の推定値を次のように仮定します。

$$\hat{\mu} = \texttt{x\_mean}$$

μ各パラメータの意味は次のとおりです。

$$\hat{\mu} = \begin{cases} \mu, & \mu \text{ known} \\ \dfrac{1}{n}\sum_{t=1}^{n} X_t & \mu \text{ unknown} \end{cases}$$

自己共分散関数 σ(k) は次の式によって推定されます。

$$\hat{\sigma}(k) = \frac{1}{n}\sum_{t=1}^{n-k}(X_t - \hat{\mu})(X_{t+k} - \hat{\mu}), \quad k = 0, 1, \ldots, K$$

ここで、*K* は `lagmax` です。k が 0 の場合は、次のように定義されます。

$$\hat{\sigma}(0)$$

自己相関関数 ρ(*k*) は次の式によって推定されます。

$$\hat{\rho}(k) = \frac{\hat{\sigma}(k)}{\hat{\sigma}(0)}, \quad k = 0, 1, \ldots, K$$

k が 0 の場合は、次のように定義されます。

$$\hat{\rho}(0) \equiv 1$$

標本自己相関の標準誤差は、オプションの引数である `IMSLS_SEAC` の引数 `se_option` に基づいて計算することもできます。これを行う 1 つの方法 (Bartlett 1946) は、独立かつ等分布の正規誤差を持つ定常時系列を対象としたもので、標本自己相関係数の分散に関連した一般的な漸近表現に基づいています。理論式は次のとおりです。

$$\operatorname{var}\{\hat{\rho}(\mathrm{k})\} = \frac{1}{n}\sum_{i=-\infty}^{\infty}\left[\rho^2(i) + \rho(i-k)\rho(i+k) - 4\rho(i)\rho(k)\rho(i-k) + 2\rho^2(i)\rho^2(k)\right]$$

各パラメータの意味は次のとおりです。

$$\hat{\rho}(k)$$

は、μ が不明であることを仮定します。計算目的上、自己相関係数 r(k) は、その推定値である次の値で置換されます (|k| > K の場合)。

$$\hat{\rho}(k)$$

また、|k| ≤ K となるように、すべての k に対して r(k) は 0 であると仮定されているため、総和の制限が適用されます。

2 番目の方法 (Moran 1947) は、独立かつ等分布の正規誤差を持つ無作為過程を対象としたもので、標本自己相関係数の分散に関連した精密な公式を利用します。理論式は次のとおりです。

$$\operatorname{var}\left\{\hat{\rho}\left(k\right)\right\} = \frac{n-k}{n\left(n+2\right)}$$

ここで、μ はゼロと等しいものと仮定されます。この式は、自己相関関数に依存しません。

partial_autocorrelation

定常時系列のサンプル偏自己相関関数を計算します。

シノプシス

float `*imsls_f_partial_autocorrelation` (*int* `lagmax`, *int* `cf[]`,…, 0)

double 型の関数は `imsls_d_partial_autocorrelation` です。

必須の引数

int `lagmax` (入力)
計算する偏自己相関の最大ラグ。

float `cf[]` (入力)
時系列 `x` の自己相関を格納する、長さ `lagmax + 1` の配列。

戻り値

時系列 `x` の偏自己相関を格納する、長さ `lagmax` の配列へのポインタ。

オプションの引数を使ったシノプシス

float `*imsls_f_partial_autocorrelation` (*int* `lagmax`, *float* `cf[]`,
`IMSLS_RETURN_USER`, *float* `partial_autocorrelations[]`,
0)

オプションの引数

`IMSLS_RETURN_USER`, *float* `partial_autocorrelations[]` (出力)
この引数が指定されている場合、偏自己相関は、ユーザによって指定された長さ `lagmax` の配列に格納されます。

説明

関数 `imsls_f_partial_autocorrelation` は、K = `lagmax` の標本自己相関が指定されている場合に、定常時系列の偏自己相関を推定します。

$$\hat{\rho}(k)$$

ここで、k は 0, 1, ⋯, K です。次の式で定義される AR(k) 過程があるとします。

$$X_t = \phi_{k1} X_{t-1} + \phi_{k2} X_{t-2} + \ldots + \phi_{kk} X_{t-k} + A_t$$

ここで、ϕkj は、過程内の j 番目の係数を表します。次の推定値のセットは、k が 1, ⋯, K の場合、標本偏自己相関関数になります。

$$\left\{\hat{\phi}_{kk}\right\}$$

次の自己回帰パラメータは、連続的な AR(k) モデルの Yule-Walker 推定によって近似されます。

$$\left\{\hat{\phi}_{kj}\right\}$$

ここで、j は 1, ⋯, k、k は 1, ⋯, K です。標本の Yule-Walker 方程式は、次のようになります。

$$\hat{\rho}(j) = \hat{\phi}_{k1}\hat{\rho}(j-1) + \hat{\phi}_{k2}\hat{\rho}(j-2) + \ldots + \hat{\phi}_{kk}\hat{\rho}(j-k), \quad j = 1, 2, \ldots, k$$

この方程式に基づき、Durbin は、k = 1, ⋯, K の場合の再帰的な関係を次の方程式によって表しました (1960 年)。

$$\hat{\phi}_{kk} = \begin{cases} \hat{\rho}(1) & k = 1 \\ \dfrac{\hat{\rho}(k) - \sum_{j=1}^{k-1} \hat{\phi}_{k-1,j}\hat{\rho}(k-j)}{1 - \sum_{j=1}^{k-1} \hat{\phi}_{k-1,j}\hat{\rho}(j)} & k = 2, \ldots, K \end{cases}$$

および

$$\hat{\phi}_{kk} = \begin{cases} \hat{\rho}(1) & k = 1 \\ \dfrac{\hat{\rho}(k) - \sum_{j=1}^{k-1} \hat{\phi}_{k-1,j}\hat{\rho}(k-j)}{1 - \sum_{j=1}^{k-1} \hat{\phi}_{k-1,j}\hat{\rho}(j)} & k = 2, \ldots, K \end{cases}$$

この手順は、丸め誤差に対する感度が高いため、パラメータが非定常境界に近い場合は使用しないでください。これに代わる方法としては、最小二乗法または最尤法を使用して、連続 AR(*k*) モデルの {ϕkk} を推定することが挙げられます。真の過程が AR(*p*) であるという仮説に基づいて、『Box and Jenkins』(1976 年、65 ページ) では、次のように解説されています。

$$\mathrm{var}\{\hat{\phi}_{kk}\} \simeq \frac{1}{n} \quad k \ge p+1$$

偏自己相関関数の詳細については、『Box and Jenkins』(1976 年、82 〜 84 ページ) を参照してください。

lack_of_fit

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。

シノプシス

float `*imsls_lack_of_fit` (*int* `n_observations`, *float* `cf[]`,
int `lagmax`, *int* `npfree`,..., 0)

必須の引数

int `n_observations` (入力)
定常時系列の観測値の数。

float `cf[]` (入力)
相関関数を格納する、長さ `lagmax`+1 の配列。

int `lagmax` (入力)
相関関数の最大ラグ。

int `npfree` (入力)
時系列モデルの公式における自由パラメータの数。`npfree` には、0 以上 `lagmax` 未満の値を指定する必要があります。Woodfield は `npfree = p + q` を推奨しています (1990 年)。

戻り値

検定統計量 Q とその *p* 値である *p* を持つ、長さ 2 の配列へのポインタ。帰無仮説では、Q は、自由度 `lagmax-lagmin+1-npfree` の近似カイ二乗分布を持ちます。

オプションの引数を使ったシノプシス

```
#include <imsls.h>

float *imsls_f_lack_of_fit (int n_observations, float cf[], int lagmax,
        int npfree,
        IMSLS_LAGMIN, int lagmin,
        IMSLS_RETURN_USER, float stat[],
        0)
```

オプションの引数

`IMSLS_LAGMIN`, *int* `lagmin` (入力)
: 相関関数の最小ラグ。`lagmin` は、適合度検定統計量がない場合の総和の下限に対応します。デフォルト値 は 1 です。

`IMSLS_RETURN_USER`, *float* `stat[]` (出力)
: 適合度検定の統計量を格納する、ユーザ定義の配列。

説明

ルーチン `imsls_f_lack_of_fit` は、ARMA モデルと伝達関数モデルの両方において適合度を診断するために使用されます。各モデルの典型的な引数は次のとおりです。

モデル	LAGMIN	LAGMAX	NPFREE
ARMA (*p, q*)	1	$\sqrt{\text{NOBS}}$	$p + q$
伝達関数	0	$\sqrt{\text{NOBS}}$	$r + s$

関数 `imsls_f_lack_of_fit` は、適切な標本相関関数が指定されている場合に、n 個の観測値を含む時系列関数または伝達関数に対してポルトマント検定を実行します。

$$\hat{\rho}(k)$$

$k = L, L + 1, \ldots, K$ であり、L は `lagmin`、K は `lagmax` です。

検定統計量 Q の基本的な形式は次のとおりです。

$$Q = n(n+2)\sum_{k=L}^{K}(n-k)^{-1}\hat{\rho}(k)$$

ただし、$L = 1$ であり、次の要素が自己相関関数であることが前提となります。

$$\hat{\rho}(k)$$

モデルが適切であることを前提とすると、Q は、自由度 $K - L + 1 - m$ のカイ二乗分布を持ちます。ここで、m = `npfree` は、モデル内の推定パラメータの数を示します。時系列の平均が推定される場合、Woodfield (1990 年) は、モデル内の推定パラメータの数に時系列の平均を含めることを推奨しています。そのため、ARMA(p, q) モデルの場合は、平均が推定されるかどうかにかかわらず `npfree`= $p + q$ と設定します。時系列モデルの元の導出は、Box and Pierce (1970 年) によるものであり、Ljung と Box (1978 年) によって提案された上記の修正版に基づいています。伝達関数モデルを検定するための拡張機能については、Box と Jenkins によって解説されています (1976 年、394 ～ 395 ページ)。

estimate_missing

時系列内の欠損値を推定します。

シノプシス

float *imsls_f_estimate_missing(*int* n_obs, *int* tpoints[],
float z[], …, 0)

double 型の関数は imsls_d_estimate_missing です。

必須の引数

int n_obs（入力)
時系列内の欠損でない観測値の数。時系列には、4 個以上の欠損値を持つギャップがあってはなりません。

int tpoints[] (入力)
時系列値が観測された時点 $t_1, \ldots, t_{n\_obs}$ を格納する、長さ n_obs のベクトル。時点は狭義増加でなければなりません。欠損値に相当する時点は、開区間 (t_1, t_{n_obs}) に含まれる必要があります。

float z[] (入力)
時系列値を格納する、長さ n_obs のベクトル。値の順序は、ベクトル tpoints 内の値に基づいて設定する必要があります。欠損値の推定の後の時系列には、2 つの連続した時点間の距離が 1 になるような、等間隔時点における値が格納されることが前提となっています。欠損でない時系列値が時点 $t_1, \ldots, t_{n\_obs}$ で観測される場合、t_i と t_{i+1} の間の欠損値は、$i = 1, \ldots, \text{n\_obs} - 1$ のときに存在します (ここで、$t_{i+1} - t_i > 1$ です)。t_i と t_{i+1} の間のギャップのサイズは、$t_{i+1} - t_i - 1$ になります。欠損でない値と推定された欠損値を持つ時系列の合計長は、$t_{n\_obs} - t_1 + 1$、つまり tpoints[n_obs-1]-tpoints[0]+1 です。

戻り値

欠損値の推定と共に時系列を格納する、長さ (tpoints[n_obs-1]-tpoints[0]+1) の配列へのポインタ。エラーが発生した場合は、NULL が返されます。

オプションの引数を使ったシノプシス

```
float  *imsls_f_estimate_missing (int n_obs, int tpoints[], float z[],
        IMSLS_METHOD, int method,
        IMSLS_MAX_LAG, int maxlag,
        IMSLS_NTIMES, int *ntimes,
        IMSLS_MEAN_ESTIMATE, float mean_estimate,
        IMSLS_CONVERGENCE_TOLERANCE, float convergence_tolerance,
        IMSLS_RELATIVE_ERROR, float relative_error,
        IMSLS_MAX_ITERATIONS, int max_iterations,
        IMSLS_TIMES_ARRAY, int **times,
        IMSLS_TIMES_ARRAY_USER, int times[],
        IMSLS_MISSING_INDEX, int **missing_index,
        IMSLS_MISSING_INDEX_USER, int missing_index[],
        IMSLS_RETURN_USER, float u_z[],
        0)
```

オプションの引数

IMSLS_METHOD, *int* method (入力)

欠損値の推定に使われる方法 ：

0 ー 平均を使用。
1 ー 3 次スプライン補間法を使用。
2 ー AR(1) モデルの短期予測を使用。
3 ー AR(p) モデルの短期予測を使用。

デフォルト ： method = 3

method = 2 が選択されている場合、時点 t_1+1 または t_1+2 以降のギャップのすべての値は、method 0 によって推定されます。method = 3 が選択され、最初のギャップが t_1+1 で始まる場合は、このギャップの値も method 0 によって推定されます。あるギャップの前の系列の長さ len が 1 より大きく 2·maxlag より小さい場合は、このギャップ内の欠損値を計算するため、maxlag が len/2 に削減されます。

IMSLS_MAX_LAG, *int* maxlag (入力)

method = 3 が選択されたときの最大ラグ数。

デフォルト : maxlag = 10

IMSLS_NTIMES, *int* *ntimes (出力)

推定された欠損値を持つ系列内の要素の数。ntimes は tpoints[n_obs-1]-tpoints[0]+1 です。

IMSLS_MEAN_ESTIMATE, *float* mean_estimate (入力)

時系列の平均の推定値。

IMSLS_CONVERGENCE_TOLERANCE, *float* convergence_tolerance (入力)

方法 2 で使われる非線形最小二乗アルゴリズムの収束性を判定する許容度。引数 convergence_tolerance は、収束性を判定するために必要な 2 回の反復処理間の平方和における最小の相対的減少を表します。そのため、convergence_tolerance には、0 以上の値を指定する必要があります。

デフォルト : $\max\{10^{-10}, \mathrm{eps}^{2/3}\}$ (単精度の場合)、$\max\{10^{-20}, \mathrm{eps}^{2/3}\}$ (倍精度の場合)。

ここで、eps は単精度の場合に =imsls_f_machine(4)、

倍精度の場合に =imsls_d_machine(4) です。

IMSLS_RELATIVE_ERROR, *float* relative_error (入力)

方法 2 で使われる非線形方程式ソルバーの停止基準。

デフォルト : relative_error = 100 × imsls_f_machine(4) (単 精度の場合)、

relative_error = 100 × imsls_d_machine(4) (倍精度の場合)。

IMSLS_MAX_ITERATIONS, *int* max_iterations (入力)

方法 2 で使われる非線形方程式ソルバーの最大許容反復回数。

デフォルト : max_iterations = 200

IMSLS_TIMES_ARRAY, *int* **times (出力)

欠損値の推定を持つ時系列の時点を格納する、長さ

ntimes = tpoints[n_obs-1]-tpoints[0]+1 の内部割り付け配列へのポインタのアドレス。

IMSLS_TIMES_ARRAY_USER, *int* times[] (出力)

配列 times の記憶領域はユーザによって指定されます。「IMSLS_TIMES_ARRAY」を参照してください。

IMSLS_MISSING_INDEX, *int* **missing_index (出力)

配列 times 内の欠損値のインデックスを格納する、長さ (ntimes-n_obs) の内部割り付け配列へのポインタのアドレス。ntimes-n_obs が 0 の場合は、欠損値が見つからず、NULL が返されます。

IMSLS_MISSING_INDEX_USER, *int* missing_index[] (出力)

配列 missing_index の記憶領域はユーザによって指定されます。「IMSLS_MISSING_INDEX」を参照してください。

IMSLS_RETURN_USER, *float* u_z[] (出力)

この引数が指定されている場合、u_z は、欠損値の推定と共に時系列値を格納する、長さ tpoints[n_obs-1]-tpoints[0]+1 のベクトルです。

説明

Box、Jenkins、および Reinsel (1994 年) によって提案された従来の時系列解析では、$t_1, t_1+1, t_1+2, \ldots, t_n$ という等間隔時点で観測を行うことが必要です。観測値が欠損していると、適切な推定値を決定するうえで問題が発生します。関数 imsls_f_estimate_missing では、4 とおりの推定方法を使用できます。方法 0 は、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 個の値の平均を求めることで、ギャップ内の欠測値を推定します。ギャップの前後に十分な数の値がない場合は、値は適宜削減されます。この方法は非常に高速で単純ですが、異常値と水準変化のないエルゴード的な定常系列に対してしか使用できません。

方法 1 は、3 次スプライン補間法を使って欠損値を推定します。この場合も、補間は、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 個の値に対して行われます。欠損値は、補間によって推定されます。この方法では、欠損値全体にわたる円滑な遷移が可能になります。
方法 2 は、ギャップの前の時系列を AR(1) 過程によって説明できることを前提としています。ギャップの前の最後の観測が時点 t_m で行われる場合は、原点 $t_1, t_1+1, \ldots, t_m$ における短期予測を求めるために、t_m の時系列値が使用されます。この値は、時点 t_{m+1} における欠損値の推定値として取得されます。t_{m+1} における値も欠損している場合は、時点 $t_1, t_1+1, \ldots, t_{m+1}$ の値を使用して、AR(1) モデルの再計算と t_{m+2} における値の推定が行われます。AR(1) モデルの係数 ϕ_1 は、最小二乗法によってルーチン imsls_f_arma から内部で計算されます。

最後の方法 3 は、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。まず、欠損値の前の時系列に適用される関数 imsls_f_auto_uni_ar を使用して、有効な値のセット {0, 1, …, max_lag} から最適な p が判定され、結果として生じる AR(p) モデルのパラメータ $\phi_1,\ldots,\phi_p$ が計算されます。これらのパラメータは、北川・赤池の両氏 (1978 年) によって提案されたハウスホルダー変換に基づく最小二乗法によって推定されます。系列 $y_{t_1}, y_{t_1+1},\ldots,y_{t_m}$ の平均を μ で表すと、原点 t_m における短期予測 $\hat{y}_{t_m}(1)$ は、次の式によって計算できます。

$$\hat{y}_{t_m}(1) = \mu(1-\sum\nolimits_{j=1}^{p}\phi_j) + \sum\nolimits_{j=1}^{p}\phi_j y_{t_m+1-j}$$

この値は、欠損値の推定値として使用されます。次に、ギャップ内のその他の欠損値ごとに、imsls_f_auto_uni_ar で始まる手順が繰り返されます。4 つのいずれの推定方法でも、欠損値のギャップは時間の昇順で処理されます。

garch

GARCH(*p*,*q*) モデルのパラメータの予測値を計算します。

シノプシス

float *imsls_f_garch (*int* p, *int* q, *int* m, *float* y[], *float* xguess[], …, 0)

double 型の関数は imsls_d_garch です。

必須の引数

int p (入力)
GARCH パラメータの数。

int q (入力)
ARCH パラメータの数。

int m (入力)
観測された時系列の長さ。

float y[] (入力)
観測された時系列データを格納する、長さ m の配列。

float xguess[] (入力)
パラメータ配列 x[] の初期値を格納する、長さ p + q + 1 の配列。

戻り値

q ARCH パラメータと p GARCH パラメータが後に続き、シグマ 2 乗推定値を格納する、長さ p + q + 1 のパラメータ配列 x[] へのポインタ。

オプションの引数を使ったシノプシス

float *imsls_f_garch (*int* p, *int* q, *int* m, *float* y[], *float* xguess[],
IMSLS_MAX_SIGMA, *float* max_sigma,
IMSLS_A, *float* *a,
IMSLS_AIC, *float* *aic,
IMSLS_RETURN_USER, *float* x[],
0)

オプションの引数

IMSLS_MAX_SIGMA, *float* max_sigma, (入力)
返された推定係数の配列の最初の要素 (シグマ) の上限値。デフォルトは 10 です。

IMSLS_A, *float* *a, (出力)
推定されたパラメータ配列 x で評価された尤度関数の値。

IMSLS_AIC, *float* *aic, (出力)
推定されたパラメータ配列 x で評価された赤池情報量規準の値。

IMSLS_RETURN_USER, *float* x[], (出力)
この引数が指定されている場合、x は、q ARCH パラメータと p GARCH パラメータが後に続き、シグマ 2 乗推定値を格納する、長さ $p + q + 1$ の配列を返します。推定されたパラメータ配列 x の記憶領域はユーザによって指定されます。

説明

時系列の GARCH (一般化自己回帰条件付き分散不均一) モデル $\{w_t\}$ は、次のように定義されます。

$$w_t = z_t \sigma_t$$

$$\sigma_t^2 = \sigma^2 + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2 + \sum_{i=1}^{q} \alpha_i w_{t-i}^2$$

ここで、z_t は、独立かつ等分布の次の標準正規ランダム変数です。

$$0 < \sigma^2 < \texttt{max\_sigma},\ \beta_i \ge 0,\ \alpha_i \ge 0$$

$$\sum_{i=2}^{p+q+1} x(i) = \sum_{i=1}^{p} \beta_i + \sum_{i=1}^{q} \alpha_i < 1$$

上記のモデルは GARCH(p,q) とし示されます。係数 β_i および α_i は、それぞれ GARCH 係数および ARCH 係数と呼ばれます。$\beta_i = 0$、$i = 1,2,\cdots,p$ の場合、上記のモデルは、Engle (1982 年) によって提唱された ARCH(q) に要約されます。パラメータに対する非負制条件は非負変数を意味し、広義の定常性を得るために β_i と α_i' の和に対する条件が必要になります。

観測データの経験的分析において、GARCH(1,1) または GARCH(1,2) モデルが条件付き分散不均一 (Palm 1996 年) を適切に表すことが判明することがあります。これは、ARMA モデルに基づいた線形時系列分析に似ています。

上記モデルの場合は、非負変数に対して、過去の正または負の値が対称効果をもたらすことに気付くことが重要です。実際には、非負変数に対して、多数の系列が強い対称効果をもたらす可能性があります。この現象を考慮に入れるため、Nelson (1991 年) は指数 GARCH (EGARCH) を提唱しました。また、Lai (1998 年) は、ARCH および GARCH における条件変数の線形関係を非線形形式に拡張する、一般的なモデル・クラスのいくつかのプロパティを提案し、研究しました。

GARCH(p,q) 内のパラメータを推定する際には、最尤法が使用されます。長さ m = `nobs` を持つ観測対象の系列 $\{w_t\}$ に対するモデルの対数尤度は、次のとおりです。

$$\log(L) = -\frac{m}{2}\log(2\pi) - \frac{1}{2}\sum_{t=1}^{m} y_t^2 / \sigma_t^2 - \frac{1}{2}\sum_{t=1}^{m}\log\sigma_t^2$$

$$where\ \sigma_t^2 = \sigma^2 + \sum_{i=1}^{p}\beta_i\sigma_{t-i}^2 + \sum_{i=1}^{q}\alpha_i w_{t-i}^2$$

したがって、$\log(L)$ は、α_i、β_i、σ の制約を最大限受けます。

このモデルでは、$q = 0$ の場合、推定されるヘッセ行列は特異行列なので、GARCH モデルも特異行列です。

ベクトル `xguess` に入力するパラメータ・ベクトル `x` の初期値は、特定の制約を満たす必要があります。`xguess` の最初の要素は σ^2 を表しており、0 よりも大きく、`max_sigma` よりも小さい必要があります。残りの $p+q$ の初期値は、それぞれが 0 以上で、和が 1 未満の値でなければなりません。

モデルの適合における定常性を保証するには、次の式を使用します。

$$\sum_{i=2}^{p+q+1} x(i) = \sum_{i=1}^{p}\beta_i + \sum_{i=1}^{q}\alpha_i < 1$$

この式を内部検査します。初期値には、0 ～ 1 の範囲の値を選択します。

`AIC` は、次の式を使用して計算します。

$$-2\log(L) + 2(p+q+1)$$

ここで、$\log(L)$ は尤度関数の値です。

統計的推論は、尤度関数 (`A`) の出力、赤池情報量規準 (AIC)、および変数の共分散行列 (`VAR`) に基づいて、`GARCH` ルーチンの外部で実行できます。

リファレンス資料

ユーザ・エラー

IMSL 関数は、ユーザ・エラーを検出すると、できるかぎり多くの情報をユーザに提供することによってそれらのエラーを処理します。これを行うため、さまざまな重大度のエラーが認識されるだけでなく、各エラーの範囲についても関数の目的と照らし合わせて検討されます。ある状況では軽微なエラーが別の状況では深刻な影響をもたらす場合があるからです。IMSL は、合理的に検出できる最大限のエラーの報告を試みます。最初のエラーが検出されると、入力は不確実なコンテキストで解釈されるため、複数のエラーがあると、エラーの検出はさらに困難になります。

エラーの重大度の決定基準

ユーザによる入力が数学的に正しくても、コンピュータによる算術の制約や使われるアルゴリズムの制約のため、答えを正確に計算することが不可能な場合があります。この場合は、評価された正確度によってエラーの重大度が決まります。関数によって複数の出力数量が計算される場合に、一部の数量は計算不可能であるが、大半の数量が計算可能であるときは、エラー状態が存在します。エラーの重大度は、そのエラーの全体的な影響がどのように評価されるかによって決まります。

エラーの種類とデフォルト・アクション

IMSL C/Stat/Library では、5 レベルの重大度のエラーが定義されています。各レベルには、PRINT 属性と STOP 属性が関連付けられています。これらの属性にはデフォルト設定 (YES または NO) がありますが、ユーザによる設定も可能です。複数の種類のエラーを定義している目的は、重大度の異なる各エラーに対して実行すべきアクションを個別に制御できるようにすることです。IMSL 関数から値が返されるたびに、1 個のエラー状態の存在が報告されます (コード 0 “error” は、エラーがないことを意味します)。複数の情報エラーが発生た場合でも、出力されるメッセージは 1 つだけです (PRINT 属性が YES に設定されている場合)。呼び出し元プログラム内での是正アクションを実行することが合理的でないか、必要でないエラーが複数ある場合は、複数のメッセージが出力されます (重大度レベルの PRINT 属性が YES に設定されている場合)。重大度レベルにかかわらず、`IMSLS_TERMINAL` 以外のエラーは情報エラーである場合があります。インクルード・ファイル *imsls.h* は、`IMSLS_NOTE`、`IMSLS_ALERT`、`IMSLS_WARNING`、`IMSLS_FATAL`、`IMSLS_TERMINAL`、`IMSLS_WARNING_IMMEDIATE`、および `IMSLS_FATAL_IMMEDIATE` のそれぞれを、列挙データ型の *Imsls_error* として定義します。

`IMSLS_NOTE`。軽微なエラーの可能性を示すため、または計算に関する情報を単に提供するための「ノート」が発行されます。
デフォルト属性 ： PRINT=NO、STOP=NO

IMSLS_ALERT。「アラート」は、アンダフローが原因で関数値が 0 に設定されたことを示します。
デフォルト属性 ： PRINT=NO、STOP=NO

IMSLS_WARNING。「警告」は、ユーザまたは呼び出し元関数による是正アクションを必要とする状態が存在することを示します。警告エラーが発行される理由としては、結果が正確な桁数が少数しかないこと、出力の大半は正確であるが、出力の一部に誤りがあること、解析手法の基本となる仮説に違反があることが挙げられます。通常、是正アクションは不要であり、この状態は無視してかまいません。
デフォルト属性 ： PRINT=YES、STOP=NO

IMSLS_FATAL。「致命的エラー」は、深刻な影響を及ぼす可能性のある状態の存在を示します。ほとんどの場合、ユーザまたは呼び出し元関数が修復のための是正アクションを実行する必要があります。
デフォルト属性 ： PRINT=YES、STOP=YES

IMSLS_TERMINAL。「ターミナル・エラー」は深刻なエラーです。通常は、式の数として負数を指定するなど、指定を誤った結果として生じます。また、C では正確に診断できない、さまざまなプログラミング・エラーが原因で発生することもあります。このような場合は、関数に対して渡された実際の引数を、マニュアルで説明されているダミー引数と慎重に比較してください。引数の順番とデータ型のチェックには、特に注意を払う必要があります。

プログラム内での是正アクションは一般的に合理的でないため、ターミナル・エラーは情報エラーではありません。通常の使用においては、ターミナル・エラーが発生すると、関数の実行はただちに終了されます。複数のターミナル・エラーが発生すると、それに関連したメッセージが出力されます。
デフォルト属性 ： PRINT=YES、STOP=YES

IMSLS_WARNING_IMMEDIATE。「即時警告エラー」は、ただちに出力される点を除いて警告エラーとまったく同じです。
デフォルト属性 ： PRINT=YES、STOP=NO

IMSLS_FATAL_IMMEDIATE。「即時致命的エラー」は、ただちに出力される点を除いて致命的エラーとまったく同じです。
デフォルト属性 ： PRINT=YES、STOP=YES

ユーザは、imsls_error_options 関数を呼び出すことによって、PRINT 属性と STOP 属性を設定できます。

製品サポート

Visual Numerics サポートへの問い合わせ

有効なサポート契約をお持ちのユーザの方は、IMSL C Numerical Library の使用法について Visual Numerics に問い合わせいただくことができます。Visual Numerics では、次のトピックに関する質問にお答えします。

- マニュアルの内容の明確化
- Visual Numerics に関連したプログラミングの問題
- 特定の問題に適用できる IMSL ライブラリ関数または手順

数学/統計関連のご相談やお使いのプログラムのデバッグは受け付けておりません。

Visual Numerics 製品サポートの連絡先については、次のページを参照してください。

- http://www.vni.com/tech/imsl/phone.php

Visual Numerics への問い合わせ時には、次のガイドラインに従ってください。

1. Visual Numerics ライセンス番号を明示する
2. 製品名とバージョン番号 (IMSL C Numerical Library Version 7.0) を明示する
3. コンパイラとオペレーティング・システムのバージョン番号を明示する
4. サポートが必要なルーチンの名前を明示し、問題についての説明を含める

付録 A

References

Abe

Abe, S. (2001) Pattern Classification： Neuro-Fuzzy Methods and their Comparison, Springer-Verlag.

Abramowitz and Stegun

Abramowitz, Milton and Irene A. Stegun (editors) (1964), *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, National Bureau of Standards, Washington.

Afifi and Azen

Afifi, A.A. and S.P. Azen (1979), *Statistical Analysis：A Computer Oriented Approach*, 2d ed., Academic Press, New York.

Agresti, Wackerly, and Boyette

Agresti, Alan, Dennis Wackerly, and James M. Boyette (1979), Exact conditional tests for cross-classifications： Approximation of attained significance levels, *Psychometrika*, **44**, 75-83.

Aha

Aha, D. W. (1991). Incremental constructive induction： An instance-based approach.

In Proceedings of the Eighth International Workshop on Machine Learning (pp. 117--121). Evanston, ILL： Morgan Kaufmann.

Ahrens and Dieter

Ahrens, J.H. and U. Dieter (1974), Computer methods for sampling from gamma, beta, Poisson, and binomial distributions, *Computing,* **12**, 223–246.

Ahrens, J.H., and U. Dieter (1985), Sequential random sampling, *ACM Transactions on Mathematical Software*, **11**, 157—169.

Akaike

Akaike, H., (1978), *Covariance Matrix Computation of the State Variable of a Stationary Gaussian Process*, Ann. Inst. Statist. Math. 30 , Part B, 499-504.

Akaike et al

Akaike, H. , Kitagawa, G., Arahata, E., Tada, F., (1979), Computer Science Monographs No. 13, The Institute of Statistical Mathematics, Tokyo.

Anderberg

Anderberg, Michael R. (1973), *Cluster Analysis for Applications*, Academic Press, New York.

Anderson

Anderson, T.W. (1971), *The Statistical Analysis of Time Series*, John Wiley & Sons, New York.

Anderson, T. W. (1994) *The Statistical Analysis of Time Series*, John Wiley & Sons, New York.

Anderson and Bancroft

Anderson, R.L. and T.A. Bancroft (1952), *Statistical Theory in Research*, McGraw-Hill Book Company, New York.

Asuncion and Newman

Asuncion, A.and Newman, D.J. (2007), UCI Machine Learning Repository, http://www.ics.uci.edu/~mlearn/MLRepository.html. Irvine, CA: University of California, School of Information and Computer Science.

Atkinson

Atkinson, A.C. (1979), A Family of Switching Algorithms for the Computer Generation of Beta Random Variates, *Biometrika*, **66**, 141–145.

Atkinson, A.C. (1985), *Plots, Transformations, and Regression*, Claredon Press, Oxford.

Baker

Baker, J. E. (1987), Reducing Bias and Inefficiency in the Selection Algorithm. *Genetic Algorithms and their Applications: Proceeding of the Second international Conference on Genetic Algorithms*, 14-21.

Barrodale and Roberts

Barrodale, I., and F.D.K. Roberts (1973), An improved algorithm for discrete L_1 approximation, *SIAM Journal on Numerical Analysis*, **10**, 839–848.

Barrodale, I., and F.D.K. Roberts (1974), Solution of an overdetermined system of equations in the l_1 norm, *Communications of the ACM*, **17**, 319–320.

Barrodale, I., and C. Phillips (1975), Algorithm 495. Solution of an overdetermined system of linear equations in the Chebyshev norm, *ACM Transactions on Mathematical Software*, **1**, 264–270.

Bartlett, M. S.

Bartlett, M.S. (1935), Contingency table interactions, *Journal of the Royal Statistics Society Supplement*, **2**, 248–252.

Bartlett, M. S. (1937) Some examples of statistical methods of research in agriculture and applied biology*, Supplement to the Journal of the Royal Statistical Society*, **4**, 137-183.

Bartlett, M. (1937), The statistical conception of mental factors, *British Journal of Psychology*, **28**, 97–104.

Bartlett, M.S. (1946), On the theoretical specification and sampling properties of autocorrelated time series, *Supplement to the Journal of the Royal Statistical Society*, **8**, 27–41.

Bartlett, M.S. (1978), *Stochastic Processes,* 3rd. ed., Cambridge University Press, Cambridge.

Bays and Durham

Bays, Carter and S.D. Durham (1976), Improving a poor random number generator, *ACM Transactions on Mathematical Software*, **2**, 59–64.

Bendel and Mickey

Bendel, Robert B., and M. Ray Mickey (1978), Population correlation matrices for sampling experiments, *Communications in Statistics*, **B7**, 163–182.

Berry

Berry, M. J. A. and Linoff, G. (1997) *Data Mining Techniques*, John Wiley & Sons, Inc.

Best and Fisher

Best, D.J., and N.I. Fisher (1979), Efficient simulation of the von Mises distribution, *Applied Statistics*, **28**, 152–157.

Bishop

Bishop, C. M. (1995) *Neural Networks for Pattern Recognition*, Oxford University Press.

Bishop et al

Bishop, Yvonne M.M., Stephen E. Feinberg, and Paul W. Holland (1975), *Discrete Multivariate Analysis : Theory and Practice*, MIT Press, Cambridge, Mass.

Bjorck and Golub

Bjorck, Ake, and Gene H. Golub (1973), Numerical Methods for Computing Angles Between Subspaces, *Mathematics of Computation,* **27**, 579–594.

Blom

Blom, Gunnar (1958), *Statistical Estimates and Transformed Beta-Variables*, John Wiley & Sons, New York.

Bosten and Battiste

Bosten, Nancy E. and E.L. Battiste (1974), Incomplete beta ratio, *Communications of the ACM*, **17**, 156s–157.

Box and Jenkins

Box, George E.P. and Gwilym M. Jenkins (1970) *Time Series Analysis : Forecasting and Control*, Holden-Day, Inc.

Box, George E.P. and Gwilym M. Jenkins (1976), *Time Series Analysis : Forecasting and Control*, revised ed., Holden-Day, Oakland.

Box and Pierce

Box, G.E.P., and David A. Pierce (1970), Distribution of residual autocorrelations in autoregressive-integrated moving average time series models, *Journal of the American Statistical Association*, **65**, 1509–1526.

Box and Tidwell

Box, G.E.P. and P.W. Tidwell (1962), Transformation of the Independent Variables, *Technometrics*, **4**, 531–550.

Box et al.

Box, George E.P., Jenkins,Gwilym M. and Reinsel G.C., (1994) *Time Series Analysis*, Third edition, Prentice Hall, Englewood Cliffs, New Jersey.

Boyette

Boyette, James M. (1979), Random RC tables with given row and column totals, *Applied Statistics*, **28**, 329—332.

Bradley

Bradley, J.V. (1968), *Distribution-Free Statistical Tests*, Prentice-Hall, New Jersey.

Breiman

Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. (1984) *Classification and Regression Trees*, Chapman & Hall. For the latest information on CART visit： http：//www.salford-systems.com/cart.php.

Breslow

Breslow, N.E. (1974), Covariance analysis of censored survival data, *Biometrics*, **30**, 89–99.

Bridel

Bridle, J. S. (1990) Probabilistic Interpretation of Feedforward Classification Network Outputs, with relationships to statistical pattern recognition, in F. Fogelman Soulie and J. Herault (Eds.), *Neuralcomputing：Algorithms, Architectures and Applications*, Springer-Verlag, 227-236.

Brown

Brown, Morton E. (1983), MCDP4F, two-way and multiway frequency tables-measures of association and the log-linear model (complete and incomplete tables), in *BMDP Statistical Software, 1983 Printing with Additions*, (edited by W.J. Dixon), University of California Press, Berkeley.

Brown and Benedetti

Brown, Morton B. and Jacqualine K. Benedetti (1977), Sampling behavior and tests for correlation in two-way contingency tables, *Journal of the American Statistical Association*, **42**, 309–315.

Calvo

Calvo, R. A. (2001) Classifying Financial News with Neural Networks, *Proceedings of the 6th Australasian Document Computing Symposium*.

Chen and Liu

Chen, C. and Liu, L., *Joint Estimation of Model Parameters and Outlier Effects in Time Series*, Journal of the American Statistical Association, Vol. 88, No.421, March 1993.

Cheng

Cheng, R.C.H. (1978), Generating beta variates with nonintegral shape parameters, *Communications of the ACM*, **21**, 317–322.

Chiang

Chiang, Chin Long (1968), *Introduction to Stochastic Processes in Statistics*, John Wiley & Sons, New York.

Clarkson and Jenrich

Clarkson, Douglas B. and Robert B Jenrich (1991), Computing extended maximum likelihood estimates for linear parameter models, submitted to *Journal of the Royal Statistical Society*, Series B, **53**, 417-426.

Coley

Coley, D. A. (1999), *An Introduction to Genetic Algorithms for Scientists and Engineers*, World Scientific Publishing Co.

Conover

Conover, W.J. (1980), *Practical Nonparametric Statistics*, 2d ed., John Wiley & Sons, New York.

Conover and Iman

Conover, W.J. and Ronald L. Iman (1983), *Introduction to Modern Business Statistics*, John Wiley & Sons, New York.

Conover, W. J., Johnson, M. E., and Johnson, M. M

Conover, W. J., Johnson, M. E., and Johnson, M. M. (1981) A comparative study of tests for homogeneity of variances, with applications to the outer continental shelf bidding data, *Technometrics*, **23**, 351-361.

Cook and Weisberg

Cook, R. Dennis and Sanford Weisberg (1982), *Residuals and Influence in Regression*, Chapman and Hall, New York.

Cooper

Cooper, B.E. (1968), Algorithm AS4, An auxiliary function for distribution integrals, *Applied Statistics*, **17**, 190–192.

Cox

Cox, David R. (1970), *The Analysis of Binary Data*, Methuen, London.

Cox, D.R. (1972), Regression models and life tables (with discussion), *Journal of the Royal Statistical Society*, Series B, *Methodology*, **34**, 187−220.

Cox and Lewis

Cox, D.R., and P.A.W. Lewis (1966), *The Statistical Analysis of Series of Events*, Methuen, London.

Cox and Oakes

Cox, D.R., and D. Oakes (1984), *Analysis of Survival Data*, Chapman and Hall, London.

Cox and Stuart

Cox, D.R., and A. Stuart (1955), Some quick sign tests for trend in location and dispersion, *Biometrika*, **42**, 80–95.

Cranley and Patterson

Cranley, R. and Patterson, T.N.L. (1976), Randomization of Number Theoretic Methods for Multiple Integration, *SIAM Journal of Numerical Analysis*, **13**, 904-914.

D'Agostino and Stevens

D'Agostino, Ralph B. and Michael A. Stevens (1986), *Goodness-of-Fit Techniques*, Marcel Dekker, New York.

Dallal and Wilkinson

Dallal, Gerald E. and Leland Wilkinson (1986), An analytic approximation to the distribution of Lilliefor's test statistic for normality, *The American Statistician*, **40**, 294–296.

Davis and Rabinowitz

Davis, P.J. and Rabinowitz, P. (1984), *Methods of Numerical Integration*, Academic Press, 482-483.

De Jong

De Jong, K. A. (1975), An Analysis of the Behavior of a Class of Genetic Adaptive Systems. (Doctorial dissertation, Univ. of Michigan). *Dissertation Abstracts International* 36(10), 5140B. (University Microfilms No. 76-9381).

Demiroz et al.

Demiroz, G., H. A. Govenir, and N. Ilter (1988),“Learning Differential Diagnosis of Eryhemato-Squamous Diseases using Voting Feature Intervals”, Artificial Intelligence in Medicine.

Dennis and Schnabel

Dennis, J.E., Jr. and Robert B. Schnabel (1983), *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, New Jersey.

Devore

Devore, Jay L (1982), *Probability and Statistics for Engineering and Sciences*, Brooks/Cole Publishing Company, Monterey, Calif.

Doornik

Doornik, J.A. (2005), An Improved Ziggurat Method to Generate Normal Random Samples, http：//www.doornik.com/research/ziggurat.pdf., University of Oxford.

Draper and Smith

Draper, N.R. and H. Smith (1981), *Applied Regression Analysis*, 2d ed., John Wiley & Sons, New York.

Dunnett and Sobel

Dunnett, C. W. and Sobel, M. (1955), Approximations to the Probability Integral and Certain Percentage Points of a Multivariate Analogue of Student's *t*-distribution. *Biometrika*, **42**, 258-260.

Durbin

Durbin, J. (1960), The fitting of time series models, *Revue Institute Internationale de Statistics*, **28**, 233–243.

Efroymson

Efroymson, M.A. (1960), Multiple regression analysis, *Mathematical Methods for Digital Computers*, Volume 1, (edited by A. Ralston and H. Wilf), John Wiley & Sons, New York, 191–203.

Ekblom

Ekblom, Hakan (1973), Calculation of linear best L_p-approximations, *BIT*, **13**, 292—300.

Ekblom, Hakan (1987), The L_1-estimate as limiting case of an L_p or Huber-estimate, in *Statistical Data Analysis Based on the L_1-Norm and Related Methods* (edited by Yadolah Dodge), North-Holland, Amsterdam, 109—116.

Elandt-Johnson and Johnson

Elandt-Johnson, Regina C., and Norman L. Johnson (1980), *Survival Models and Data Analysis*, John Wiley & Sons, New York, 172–173.

Elman

Elman, J. L. (1990) Finding Structure in Time, *Cognitive Science,* 14, 179-211.

Emmett

Emmett, W.G. (1949), Factor analysis by Lawless method of maximum likelihood, *British Journal of Psychology, Statistical Section*, **2**, 90–97.

Engle

Engle, C. (1982), Autoregressive conditional heteroskedasticity with estimates of the variance of U.K. inflation, *Econometrica ,* **50**, 987–1008.

Fisher

Fisher, R.A. (1936), The use of multiple measurements in taxonomic problems, *The Annals of Eugenics*, **7**, 179–188.

Fishman

Fishman, George S. (1978), *Principles of Discrete Event Simulation*, John Wiley & Sons, New York.

Fishman and Moore

Fishman, George S. and Louis R. Moore (1982), A statistical evaluation of multiplicative congruential random number generators with modulus , *Journal of the American Statistical Association*, **77**, 129–136.

Forsythe

Forsythe, G.E. (1957), Generation and use of orthogonal polynomials for fitting data with a digital computer, *SIAM Journal on Applied Mathematics*, **5**, 74–88.

Frey and Slate

Frey, P. W. and D. J. Slate. (1991), “Letter Recognition using Holland-style Adaptive Classifiers”. (Machine Learning *Vol* 6 #2).

Fuller

Fuller, Wayne A. (1976), *Introduction to Statistical Time Series*, John Wiley & Sons, New York.

Furnival and Wilson

Furnival, G.M. and R.W. Wilson, Jr. (1974), Regressions by leaps and bounds, *Technometrics*, **16**, 499–511.

Fushimi

Fushimi, Masanori (1990), Random number generation with the recursion $X_t = X_{t-3p} \oplus X_{t-3q}$, *Journal of Computational and Applied Mathematics*, **31**, 105–118.

Gentleman

Gentleman, W. Morven (1974), Basic procedures for large, sparse or weighted linear least squares problems, *Applied Statistics*, **23**, 448–454.

Genz

Genz, A. (1992), Numerical Computation of Multivariate Normal Probabilities. *J. Comp. Graph Stat*., **1**, 141-149.

Gibbons

Gibbons, J.D. (1971), *Nonparametric Statistical Inference*, McGraw-Hill, New York.

Girschick

Girschick, M.A. (1939), On the Sampling Theory of Roots of Determinantal Equations, *Annals of Mathematical Statistics*, **10**, 203–224.

Goldberg

Goldberg, D. E. (1989), Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Publishing Co.

Goldberg, D. E. and Deb, K. (1991), A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. In G. Rawlins, Ed., *Foundations of Genetic Algorithms*. Morgan Kaufmann.

Golub and Van Loan

Golub, Gene H. and Charles F. Van Loan (1983), *Matrix Computations*, Johns Hopkins University Press, Baltimore, Md.

Gonin and Money

Gonin, Rene, and Arthur H. Money (1989), *Nonlinear L_p-Norm Estimation*, Marcel Dekker, New York.

Goodnight

Goodnight, James H. (1979), A tutorial on the SWEEP operator, *The American Statistician*, **33**, 149–158.

Graybill

Graybill, Franklin A. (1976), *Theory and Application of the Linear Model*, Duxbury Press, North Scituate, Mass.

Griffin and Redish

Griffin, R. and K.A. Redish (1970), Remark on Algorithm 347：An efficient algorithm for sorting with minimal storage, *Communications of the ACM*, **13**, 54.

Gross and Clark

Gross, Alan J., and Virginia A. Clark (1975), *Survival Distributions：Reliability Applications in the Biomedical Sciences*, John Wiley & Sons, New York.

Gruenberger and Mark

Gruenberger, F., and A.M. Mark (1951), The d^2 test of random digits, *Mathematical Tables and Other Aids in Computation*, **5**, 109–110.

Guerra et al.

Guerra, Victor O., Richard A. Tapia, and James R. Thompson (1976), A random number generator for continuous random variables based on an interpolation procedure of Akima, in *Proceedings of the Ninth Interface Symposium on Computer Science and Statistics*, (edited by David C. Hoaglin and Roy E. Welsch), Prindle, Weber & Schmidt, Boston, 228─230.

Giudici

Giudici, P. (2003) *Applied Data Mining：Statistical Methods for Business and Industry*, John Wiley & Sons, Inc.

Haldane

Haldane, J.B.S. (1939), The mean and variance of when used as a test of homogeneity, when expectations are small, *Biometrika*, **31**, 346.

Hamilton

Hamilton, James D., *Time Series Analysis*, Princeton University Press, Princeton (NewJersey), 1994.

Harman

Harman, Harry H. (1976), *Modern Factor Analysis*, 3d ed. revised, University of Chicago Press, Chicago.

Hart et al

Hart, John F., E.W. Cheney, Charles L. Lawson, Hans J. Maehly, Charles K. Mesztenyi, John R. Rice, Henry G. Thacher, Jr., and Christoph Witzgall (1968), *Computer Approximations*, John Wiley & Sons, New York.

Hartigan

Hartigan, John A. (1975), *Clustering Algorithms*, John Wiley & Sons, New York.

Hartigan and Wong

Hartigan, J.A. and M.A. Wong (1979), Algorithm AS 136： A *K*-means clustering algorithm, *Applied Statistics*, **28**, 100–108.

Hayter

Hayter, Anthony J. (1984), A proof of the conjecture that the Tukey-Kramer multiple comparisons procedure is conservative, *Annals of Statistics*, **12**, 61–75.

Hebb

Hebb, D. O. (1949) *The Organization of Behaviour： A Neuropsychological Theory*, John Wiley.

Heiberger

Heiberger, Richard M. (1978), Generation of random orthogonal matrices, *Applied Statistics*, **27**, 199─206.

Hemmerle.

Hemmerle, William J. (1967), *Statistical Computations on a Digital Computer*, Blaisdell Publishing Company, Waltham, Mass.

Herraman

Herraman, C. (1968), Sums of squares and products matrix, *Applied Statistics*, **17**, 289–292.

Hill

Hill, G.W. (1970), Student's *t*-distribution, *Communications of the ACM*, **13**, 617–619.

Hill, G.W. (1970), Student's *t*-quantiles, *Communications of the ACM*, **13**, 619–620.

Hinkelmann, K and Kemthorne

Hinkelmann, K and Kemthorne, O (1994) *Design and Analysis of Experiments − Vol 1*, John Wiley.

Hinkley

Hinkley, David (1977), On quick choice of power transformation, *Applied Statistics*, **26**, 67–69.

Hoaglin and Welsch

Hoaglin, David C. and Roy E. Welsch (1978), The hat matrix in regression and ANOVA, *The American Statistician*, **32**, 17–22.

Hocking

Hocking, R.R. (1972), Criteria for selection of a subset regression： Which one should be used?, *Technometrics*, **14**, 967–970.

Hocking, R.R. (1973), A discussion of the two-way mixed model, *The American Statistician*, **27**, 148–152.

Hocking, R.R. (1985), *The Analysis of Linear Models*, Brooks/Cole Publishing Company, Monterey, California.

Hopfield

Hopfield, J. J. (1987) Learning Algorithms and Probability Distributions in Feed-Forward and Feed-Back Networks, *Proceedings of the National Academy of Sciences*, 84, 8429-8433.

Holland

Holland, J.H. (1975), *Adaptation in Natural and Artificial Systems*. Ann Arbor： The University of Michigan Press.

Huber

Huber, Peter J. (1981), *Robust Statistics*, John Wiley & Sons, New York.

Hutchinson

Hutchinson, J. M. (1994) *A Radial Basis Function Approach to Financial Timer Series Analysis*, Ph.D. dissertation, Massachusetts Institute of Technology.

Hughes and Saw

Hughes, David T., and John G. Saw (1972), Approximating the percentage points of Hotelling's generalized T_0^2 statistic, *Biometrika*, **59**, 224–226.

Hwang

Hwang, J. T. G. and Ding, A. A. (1997) Prediction Intervals for Artificial Neural Networks, *Journal of the American Statistical Society*, 92(438) 748-757.

Iman and Davenport

Iman, R.L., and J.M. Davenport (1980), Approximations of the critical region of the Friedman statistic, *Communications in Statistics*, **A9(6)**, 571–595.

Jacobs

Jacobs, R. A., Jorday, M. I., Nowlan, S. J., and Hinton, G. E. (1991) Adaptive Mixtures of Local Experts, *Neural Computation*, 3(1), 79-87.

Jennrich and Robinson

Jennrich, R.I. and S.M. Robinson (1969), A Newton-Raphson algorithm for maximum likelihood factor analysis, *Psychometrika*, **34**, 111–123.

Jennrich and Sampson

Jennrich, R.I. and P.F. Sampson (1966), Rotation for simple loadings, *Psychometrika*, **31**, 313–323.

John

John, Peter W.M. (1971), *Statistical Design and Analysis of Experiments*, Macmillan Company, New York.

Jöhnk

Jöhnk, M.D. (1964), Erzeugung von Betaverteilten und Gammaverteilten Zufallszahlen, *Metrika*, **8**, 5–15.

Johnson and Kotz

Johnson, Norman L., and Samuel Kotz (1969), *Discrete Distributions*, Houghton Mifflin Company, Boston.

Johnson, Norman L., and Samuel Kotz (1970a), *Continuous Univariate Distributions*-1, John Wiley & Sons, New York.

Johnson, Norman L., and Samuel Kotz (1970b), *Continuous Univariate Distributions*-2, John Wiley & Sons, New York.

Johnson and Kotz

Johnson, N.L. and Kotz, S. (1972), *Distributions in Statistics： Continuous Multivariate Distributions*, John Wiley & Sons, Inc., New York.

Johnson and Welch

Johnson, D.G., and W.J. Welch (1980), The generation of pseudo-random correlation matrices, *Journal of Statistical Computation and Simulation*, **11**, 55─69.

Jonckheere

Jonckheere, A.R. (1954), A distribution-free *k*-sample test against ordered alternatives, *Biometrika*, **41**, 133–143.

Jöreskog

Jöreskog, K.G. (1977), Factor analysis by least squares and maximum-likelihood methods, *Statistical Methods for Digital Computers*, (edited by Kurt Enslein, Anthony Ralston, and Herbert S. Wilf), John Wiley & Sons, New York, 125–153.

Kachitvichyanukul

Kachitvichyanukul, Voratas (1982), *Computer generation of Poisson, binomial, and hypergeometric random variates*, Ph.D. dissertation, Purdue University, West Lafayette, Indiana.

Kaiser

Kaiser, H.F. (1963), Image analysis, *Problems in Measuring Change*, (edited by C. Harris), University of Wisconsin Press, Madison, Wis.

Kaiser and Caffrey

Kaiser, H.F. and J. Caffrey (1965), Alpha factor analysis, *Psychometrika*, **30**, 1–14.

Kalbfleisch and Prentice

Kalbfleisch, John D., and Ross L. Prentice (1980), *The Statistical Analysis of Failure Time Data*, John Wiley & Sons, New York.

Keast

Keast, P. (1973) Optimal Parameters for Multidimensional Integration, *SIAM Journal of Numerical Analysis*, **10**, 831-838.

Kemp

Kemp, A.W., (1981), Efficient generation of logarithmically distributed pseudo-random variables, *Applied Statistics,* **30**, 249–253.

Kendall and Stuart

Kendall, Maurice G. and Alan Stuart (1973), *The Advanced Theory of Statistics*, Volume 2： *Inference and Relationship*, 3d ed., Charles Griffin & Company, London.

Kendall, Maurice G. and Alan Stuart (1979), *The Advanced Theory of Statistics*, Volume 2： *Inference and Relationship*, 4th ed., Oxford University Press, New York.

Kendall et al.

Kendall, Maurice G., Alan Stuart, and J. Keith Ord (1983), *The Advanced Theory of Statistics*, Volume3： *Design and Analysis, and Time Series*, 4th. ed., Oxford University Press, New York.

Kennedy and Gentle

Kennedy, William J., Jr. and James E. Gentle (1980), *Statistical Computing*, Marcel Dekker, New York.

Kohonen

Kohonen, T. (1995), Self-Organizing Maps, Third Edition. Springer Series in Information Sciences., New York.

Kuehl, R. O.

Kuehl, R. O. (2000) *Design of Experiments： Statistical Principles of Research Design and Analysis*, 2nd edition, Duxbury Press.

Kim and Jennrich

Kim, P.J., and R.I. Jennrich (1973), Tables of the exact sampling distribution of the two sample Kolmogorov-Smirnov criterion D_{mn} ($m < n$), in *Selected Tables in Mathematical Statistics*, Volume 1, (edited by H. L. Harter and D.B. Owen), American Mathematical Society, Providence, Rhode Island.

Kinderman and Ramage

Kinderman, A.J., and J.G. Ramage (1976), Computer generation of normal random variables, *Journal of the American Statistical Association*, **71**, 893–896.

Kinderman et al.

Kinderman, A.J., J.F. Monahan, and J.G. Ramage (1977), Computer methods for sampling from Student's *t* distribution, *Mathematics of Computation* **31**, 1009—1018.

Kinnucan and Kuki

Kinnucan, P. and H. Kuki (1968), *A Single Precision INVERSE Error Function Subroutine*, Computation Center, University of Chicago.

Kirk

Kirk, Roger E. (1982), *Experimental Design：Procedures for the Behavioral Sciences*, 2d ed., Brooks/Cole Publishing Company, Monterey, Calif.

Kitagawa and Akaike

Kitagawa, G. and Akaike, H., A Procedure for the modeling of non-stationary time series, Ann. Inst. Statist. Math. 30 (1978), Part B, 351-363.

Konishi and Kitagawa

Konishi, S. and Kitagawa, G (2008), Information Criteria and Statistical Modeling, Springer, New York.

Knuth

Knuth, Donald E. (1981), *The Art of Computer Programming*, Volume 2：*Seminumerical Algorithms*, 2d ed., Addison-Wesley, Reading, Mass.

Kshirsagar

Kshirsagar, Anant M. (1972), *Multivariate Analysis*, Marcel Dekker, New York.

Lachenbruch

Lachenbruch, Peter A. (1975), *Discriminant Analysis*, Hafner Press, London.

Lai

Lai, D. (1998a), Local asymptotic normality for location-scale type processes. *Far East Journal of Theorectical Statistics*, (in press).

Lai, D. (1998b), Asymptotic distributions of the correlation integral based statistics. *Journal of Nonparametric Statistics*, (in press).

Lai, D. (1998c), Asymptotic distributions of the estimated BDS statistic and residual analysis of AR Models on the Canadian lynx data. *Journal of Biological Systems*, (in press).

Laird and Oliver

Laird, N.M., and D. Fisher (1981), Covariance analysis of censored survival data using log-linear analysis techniques, *JASA* **76**, 1231─1240.

Lawless

Lawless, J.F. (1982), *Statistical Models and Methods for Lifetime Data*, John Wiley & Sons, New York.

Lawley and Maxwell

Lawley, D.N. and A.E. Maxwell (1971), *Factor Analysis as a Statistical Method*, 2d ed., Butterworth, London.

Lawrence et al

Lawrence, S., Giles, C. L, Tsoi, A. C., Back, A. D. (1997) Face Recognition：A Convolutional Neural Network Approach, *IEEE Transactions on Neural Networks, Special Issue on Neural Networks and Pattern Recognition*, 8(1), 98-113.

Learmonth and Lewis

Learmonth, G.P. and P.A.W. Lewis (1973), *Naval Postgraduate School Random Number Generator Package LLRANDOM, NPS55LW73061A*, Naval Postgraduate School, Monterey, Calif.

Lee

Lee, Elisa T. (1980), *Statistical Methods for Survival Data Analysis*, Lifetime Learning Publications, Belmont, Calif.

Lehmann

Lehmann, E.L. (1975), *Nonparametrics：Statistical Methods Based on Ranks*, Holden-Day, San Francisco.

Levenberg

Levenberg, K. (1944), A method for the solution of certain problems in least squares, *Quarterly of Applied Mathematics*, **2**, 164–168.

Levene, H.

Levene, H. (1960) In *Contributions to Probability and Statistics：Essays in Honor of Harold Hotelling*, I. Olkin et al. editors, Stanford University Press, 278-292.

Lewis et al.

Lewis, P.A.W., A.S. Goodman, and J.M. Miller (1969), A pseudorandom number generator for the System/360, *IBM Systems Journal*, **8**, 136–146.

Li

Li, L. K. (1992) Approximation Theory and Recurrent Networks, Proc. Int. Joint Conf. On Neural Networks, vol. II, 266-271.

Liffiefors

Lilliefors, H.W. (1967), On the Kolmogorov-Smirnov test for normality with mean and variance unknown, *Journal of the American Statistical Association*, **62**, 534–544.

Lippmann

Lippmann, R. P. (1989) Review of Neural Networks for Speech Recognition, Neural Computation, I, 1-38.

Ljung and Box

Ljung, G.M., and G.E.P. Box (1978), On a measure of lack of fit in time series models, *Biometrika*, **65**, 297–303.

Loh

Loh, W.-Y. and Shih, Y.-S. (1997) Split Selection Methods for Classification Trees, *Statistica Sinica*, 7, 815-840. For information on the latest version of QUEST see：http：//www.stat.wisc.edu/~loh/quest.html.

Longley

Longley, James W. (1967), An appraisal of least-squares programs for the electronic computer from the point of view of the user, *Journal of the American Statistical Association*, **62**, 819–841.

Matsumoto and Nishimure

Makoto Matsumoto and Takuji Nishimura, ACM Transactions on Modeling and Computer Simulation, Vol. 8, No. 1, January 1998, Pages 3–30.

Mandic

Mandic, D. P. and Chambers, J. A. (2001) *Recurrent Neural Networks for Prediction*, John Wiley & Sons, LTD.

Manning

Manning, C. D. and Schütze, H. (1999) *Foundations of Statistical Natural Language Processing,* MIT Press.

Marsaglia

Marsaglia, George (1964), Generating a variable from the tail of a normal distribution, *Technometrics*, **6**, 101—102.

Marsaglia, G. (1968), Random numbers fall mainly in the planes, *Proceedings of the National Academy of Sciences*, **61**, 25—28.

Marsaglia, G. (1972), The structure of linear congruential sequences, in *Applications of Number Theory to Numerical Analysis*, (edited by S. K. Zaremba), Academic Press, New York, 249—286.

Marsaglia, George (1972), Choosing a point from the surface of a sphere, *The Annals of Mathematical Statistics*, **43**, 645—646.

Marsaglia and Tsang

Marsaglia, G. and Tsang, W. W. (2000), The Ziggurat Method for Generating Random Variables, *Journal of Statistical Software*, **5-8**, 1—7.

McCulloch

McCulloch, W. S. and Pitts, W. (1943), A Logical Calculus for Ideas Imminent in Nervous Activity, *Bulletin of Mathematical Biophysics*, 5, 115-133.

McKean and Schrader

McKean, Joseph W., and Ronald M. Schrader (1987), Least absolute errors analysis of variance, in *Statistical Data Analysis Based on the* L_1*-Norm and Related Methods* (edited by Yadolah Dodge), North-Holland, Amsterdam, 297—305.

McKeon

McKeon, James J. (1974), *F* approximations to the distribution of Hotelling's T_0^2, *Biometrika*, **61**, 381–383.

McCullagh and Nelder

McCullagh, P., and J.A. Nelder, (1983), *Generalized Linear Models*, Chapman and Hall, London.

Maindonald

Maindonald, J.H. (1984), *Statistical Computation*, John Wiley & Sons, New York.

Marazzi

Marazzi, Alfio (1985), Robust affine invariant covariances in ROBETH, ROBETH-85 document No. 6, Division de Statistique et Informatique, Institut Universitaire de Medecine Sociale et Preventive, Laussanne.

Mardia et al.

Mardia, K.V. (1970), Measures of multivariate skewness and kurtosis with applications, *Biometrics*, **57**, 519–530.

Mardia, K.V., J.T. Kent, J.M. Bibby (1979), *Multivariate Analysis*, Academic Press, New York.

Mardia and Foster

Mardia, K.V. and K. Foster (1983), Omnibus tests of multinormality based on skewness and kurtosis, *Communications in Statistics A, Theory and Methods*, **12**, 207–221.

Marquardt

Marquardt, D. (1963), An algorithm for least-squares estimation of nonlinear parameters, *SIAM Journal on Applied Mathematics*, **11**, 431–441.

Marsaglia

Marsaglia, George (1964), Generating a variable from the tail of a normal distribution, *Technometrics*, **6**, 101–102.

Marsaglia and Bray

Marsaglia, G. and T.A. Bray (1964), A convenient method for generating normal variables, *SIAM Review*, **6**, 260–264.

Marsaglia et al.

Marsaglia, G., M.D. MacLaren, and T.A. Bray (1964), A fast procedure for generating normal random variables, *Communications of the ACM*, **7**, 4—10.

Merle and Spath

Merle, G., and H. Spath (1974), Computational experiences with discrete L_p approximation, *Computing*, **12**, 315—321.

Miller

Miller, Rupert G., Jr. (1980), *Simultaneous Statistical Inference*, 2d ed., Springer-Verlag, New York.

Milliken and Johnson

Milliken, George A. and Dallas E. Johnson (1984), *Analysis of Messy Data*, *Volume 1： Designed Experiments*, Van Nostrand Reinhold, New York.

Mitchell

Mitchell, M. (1996), *An Introduction to Genetic Algorithms*, MIT Press.

Moran

Moran, P.A.P. (1947), Some theorems on time series I, *Biometrika*, **34**, 281–291.

Moré et al.

Moré, Jorge, Burton Garbow, and Kenneth Hillstrom (1980), *User Guide for [4] MINPACK-1*, Argonne National Laboratory Report ANL-80_74, Argonne, Ill.

Morrison

Morrison, Donald F. (1976), *Multivariate Statistical Methods*, 2nd. ed. McGraw-Hill Book Company, New York.

Muller

Muller, M.E. (1959), A note on a method for generating points uniformly on N-dimensional spheres, *Communications of the ACM*, **2**, 19—20.

Nelson

Nelson, D. B. (1991), Conditional heteroskedasticity in asset returns：A new approach. *Econometrica*, , **59**, 347–370.

Nelson

Nelson, Peter (1989), Multiple Comparisons of Means Using Simultaneous Confidence Intervals, *Journal of Quality Technology*, **21**, 232–241.

Neter

Neter, John (1983), *Applied Linear Regression Models*, Richard D. Irwin, Homewood, Ill.

Neter and Wasserman

Neter, John and William Wasserman (1974), *Applied Linear Statistical Models*, Richard D. Irwin, Homewood, Ill.

Noether

Noether, G.E. (1956), Two sequential tests against trend, *Journal of the American Statistical Association*, **51**, 440–450.

Owen

Owen, D.B. (1962), *Handbook of Statistical Tables*, Addison-Wesley Publishing Company, Reading, Mass.

Owen, D.B. (1965), A Special Case of the Bivariate Non-central *t* Distribution, *Biometrika*, **52**, 437–446.

Ozaki and Oda

Ozaki, T and Oda H (1978) Nonlinear time series model identification by Akaike's information criterion. *Information and Systems, Dubuisson eds*, Pergamon Press. 83-91.

Pao

Pao, Y. (1989) *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley Publishing.

Palm

Palm, F. C. (1996), GARCH models of volatility. In *Handbook of Statistics*, Vol. 14, 209-240. Eds：Maddala and Rao. Elsevier,New York.

Parker

Parker, D. B., (1985), *Learning Logic*. Technical Report TR-47, Cambridge, MA： MIT Center for Research in computational Economics and Management Science.

Patefield

Patefield, W.M. (1981), An efficient method of generating $R \times C$ tables with given row and column totals, *Applied Statistics*, **30**, 91–97.

Patefield and Tandy

Patefield, W.M. (1981), and Tandy D. (2000) Fast and Accurate Calculation of Owen’s T-Function, *J. Statistical Software*, **5**, Issue 5.

Peixoto

Peixoto, Julio L. (1986), Testable hypotheses in singular fixed linear models, Communications in Statistics： *Theory and Methods*, **15**, 1957–1973.

Petro

Petro, R. (1970), Remark on Algorithm 347： An efficient algorithm for sorting with minimal storage, *Communications of the ACM*, **13**, 624.

Pillai

Pillai, K.C.S. (1985), Pilla i’s trace, in *Encyclopedia of Statistical Sciences*, *Volume 6*, (edited by Samuel Kotz and Norman L. Johnson), John Wiley & Sons, New York, 725–729.

Poli

Poli, I. and Jones, R. D. (1994) A Neural Net Model for Prediction, *Journal of the American Statistical Society*, 89(425) 117-121.

Pregibon

Pregibon, Daryl (1981), Logistic regression diagnostics, *The Annals of Statistics*, **9**, 705–724.

Prentice

Prentice, Ross L. (1976), A generalization of the probit and logit methods for dose response curves, *Biometrics*, **32**, 761–768.

Priestley

Priestley, M.B. (1981), *Spectral Analysis and Time Series*, Volumes 1 and 2, Academic Press, New York.

Quinlan

Quinlan, J. R. (1993). C4.5 Programs for Machine Learning, Morgan Kaufmann. For the latest information on Quinlan's algorithms see http://www.rulequest.com/.

Quinlan (1987). *Simplifying Decision Trees.* Int J Man-Machine Studies 27, pp. 221-234.

Rao

Rao, C. Radhakrishna (1973), *Linear Statistical Inference and Its Applications*, 2d ed., John Wiley & Sons, New York.

Reed

Reed, R. D. and Marks, R. J. II (1999) *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*, The MIT Press, Cambridge, MA.

Ripley

Ripley, B. D. (1994) Neural Networks and Related Methods for Classification, *Journal of the Royal Statistical Society B*, 56(3), 409-456.

Ripley, B. D. (1996) *Pattern Recognition and Neural Networks*, Cambridge University Press.

Robinson

Robinson, Enders A. (1967), *Multichannel Time Series Analysis with Digital Computer Programs*, Holden-Day, San Francisco.

Rosenblatt

Rosenblatt, F. (1958) The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, *Psychol. Rev.*, 65, 386-408.

Royston

Royston, J.P. (1982a), An extension of Shapiro and Wilk's *W* test for normality to large samples, *Applied Statistics*, **31**, 115–124.

Royston, J.P. (1982b), The *W* test for normality, *Applied Statistics*, **31**, 176–180.

Royston, J.P. (1982c), Expected normal order statistics (exact and approximate), *Applied Statistics*, **31**, 161–165.

Royston, J. P. (1991), Approximating the Shapiro-Wilk W-test for non-normality, *Statistics and Computing*, **2**, 117-119.

Rumelhart

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986) Learning Representations by Back-Propagating Errors, *Nature*, 323, 533-536.

Rumelhart, D. E. and McClelland, J. L. eds. (1986) *Parallel Distributed Processing：Explorations in the Microstructure of Cognition*, 1, 318-362, MIT Press.

Sallas

Sallas, William M. (1990), An algorithm for an L_p norm fit of a multiple linear regression model, *American Statistical Association 1990 Proceedings of the Statistical Computing Section*, 131─136.

Sallas and Lionti

Sallas, William M. and Abby M. Lionti (1988), *Some useful computing formulas for the nonfull rank linear model with linear equality restrictions*, IMSL Technical Report 8805, IMSL, Houston.

Savage

Savage, I. Richard (1956), Contributions to the theory of rank order statistics-the two-sample case, *Annals of Mathematical Statistics*, **27**, 590–615.

Scheffe

Scheffe, Henry (1959), *The Analysis of Variance*, John Wiley & Sons, New York.

Schmeiser

Schmeiser, Bruce (1983), Recent advances in generating observations from discrete random variates, *Computer Science and Statistics : Proceedings of the Fifteenth Symposium on the Interface*, (edited by James E. Gentle), North-Holland Publishing Company, Amsterdam, 154–160.

Schmeiser and Babu

Schmeiser, Bruce W. and A.J.G. Babu (1980), Beta variate generation via exponential majorizing functions, *Operations Research*, **28**, 917–926.

Schmeiser and Kachitvichyanukul

Schmeiser, Bruce and Voratas Kachitvichyanukul (1981), *Poisson Random Variate Generation*, Research Memorandum 81–4, School of Industrial Engineering, Purdue University, West Lafayette, Ind.

Schmeiser and Lal

Schmeiser, Bruce W. and Ram Lal (1980), Squeeze methods for generating gamma variates, *Journal of the American Statistical Association*, **75**, 679–682.

Searle

Searle, S.R. (1971), *Linear Models*, John Wiley & Sons, New York.

Seber

Seber, G.A.F. (1984), *Multivariate Observations*, John Wiley & Sons, New York.

Snedecor and Cochran

Snedecor and Cochran (1967) *Statistical Methods*, 6th edition, Iowa State University Press.

Snedecor and Cochran

Snedecor, George W. and Cochran, William G. (1967) *Statistical Methods*, 6th edition, Iowa State University Press, 296-298.

Snedecor, George W. and Cochran, William G. (1967) *Statistical Methods*, 6th edition, Iowa State University Press, 432–436.

Shampine

Shampine, L.F. (1975), Discrete least-squares polynomial fits, *Communications of the ACM*, **18**, 179–180.

Siegal

Siegal, Sidney (1956), *Nonparametric Statistics for the Behavioral Sciences*, McGraw-Hill, New York.

Singleton

Singleton, R.C. (1969), Algorithm 347： An efficient algorithm for sorting with minimal storage, *Communications of the ACM*, **12**, 185–187.

Smirnov

Smirnov, N.V. (1939), Estimate of deviation between empirical distribution functions in two independent samples (in Russian), *Bulletin of Moscow University*, **2**, 3–16.

Smith and Dubey

Smith, H., and S. D. Dubey (1964), “Some reliability problems in the chemical industry”, Industrial Quality Control, 21 (2), 1964, 64-70.

Smith

Smith, M. (1993) *Neural Networks for Statistical Modeling*, New York： Van Nostrand Reinhold.

Snedecor and Cochran

Snedecor, George W. and William G. Cochran (1967), *Statistical Methods*, 6th ed., Iowa State University Press, Ames, Iowa.

Sposito

Sposito, Vincent A. (1989), Some properties of L_p-estimators, in *Robust Regression： Analysis and Applications* (edited by Kenneth D. Lawrence and Jeffrey L. Arthur), Marcel Dekker, New York, 23–58.

Spurrier and Isham

Spurrier, John D. and Steven P. Isham (1985), Exact simultaneous confidence intervals for pairwise comparisons of three normal means, *Journal of the American Statistical Association*, **80**, 438–442.

Stablein, Carter, and Novak

Stablein, D.M, W.H. Carter, and J.W. Novak (1981), Analysis of survival data with nonproportional hazard functions, *Controlled Clinical Trials*, **2**, 149–159.

Stahel

Stahel, W. (1981), Robuste Schatzugen： Infinitesimale Opimalitat und Schatzugen von Kovarianzmatrizen, Dissertation no. 6881, ETH, Zurich.

Steel and Torrie

Steel and Torrie (1960) *Principles and Procedures of Statistics*, McGraw-Hill.

Stephens

Stephens, M.A. (1974), EDF statistics for goodness of fit and some comparisons, *Journal of the American Statistical Association*, **69**, 730–737.

Stirling

Stirling, W.D. (1981), Least squares subject to linear constraints, *Applied Statistics*, **30**, 204–212. (See correction, p. 357.)

Stoline

Stoline, Michael R. (1981), The status of multiple comparisons： simultaneous estimation of all pairwise comparisons in one-way ANOVA designs, *The American Statistician*, **35**, 134–141.

Strecok

Strecok, Anthony J. (1968), On the calculation of the inverse of the error function, *Mathematics of Computation*, **22**, 144–158.

Studenmund

Studenmund, A. H. (1992) Using Economics： A Practical Guide, New York： Harper Collins.

Swingler

Swingler, K. (1996) Applying Neural Networks： A Practical Guide, Academic Press.

Tanner and Wong

Tanner, Martin A., and Wing H. Wong (1983), The estimation of the hazard function from randomly censored data by the kernel method, *Annals of Statistics*, **11**, 989–993.

Tanner, Martin A., and Wing H. Wong (1984), Data-based nonparametric estimation of the hazard function with applications to model diagnostics and exploratory analysis, *Journal of the American Statistical Association*, **79**, 123–456.

Taylor and Thompson

Taylor, Malcolm S., and James R. Thompson (1986), Data based random number generation for a multivariate distribution via stochastic simulation, *Computational Statistics & Data Analysis*, **4**, 93─101.

Tesauro

Tesauro, G. (1990) Neurogammon Wins Computer Olympiad, *Neural Computation*, 1, 321-323.

Tezuka

Tezuka, S. (1995), *Uniform Random Numbers：Theory and Practice*. Academic Publishers, Boston.

Thompson

Thompson, James R, (1989), *Empirical Model Building*, John Wiley & Sons, New York.

Tong

Tong, Y. L. (1990), *The Multivariate Normal Distribution*, Springer-Verlag, New York.

Tucker and Lewis

Tucker, Ledyard and Charles Lewis (1973), A reliability coefficient for maximum likelihood factor analysis, *Psychometrika*, **38**, 1–10.

Tukey

Tukey, John W. (1962), The future of data analysis, *Annals of Mathematical Statistics,* **33**, 1–67.

Velleman and Hoaglin

Velleman, Paul F. and David C. Hoaglin (1981), *Applications, Basics, and Computing of Exploratory Data Analysis*, Duxbury Press, Boston.

Verdooren

Verdooren, L. R. (1963), Extended tables of critical values for Wilcoxon's test statistic, *Biometrika*, **50**, 177–186.

Wallace

Wallace, D.L. (1959), Simplified Beta-approximations to the Kruskal-Wallis H-test, *Journal of the American Statistical Association*, **54**, 225–230.

Warner

Warner, B. and Misra, M. (1996) Understanding Neural Networks as Statistical Tools, *The American Statistician*, 50(4) 284-293.

Weisberg

Weisberg, S. (1985), *Applied Linear Regression*, 2d ed., John Wiley & Sons, New York.

Werbos

Werbos, P. (1974) *Beyond Regression： New Tools for Prediction and Analysis in the Behavioral Science*, PhD thesis, Harvard University, Cambridge, MA.

Werbos, P. (1990) Backpropagation Through Time： What It Does and How to do It, *Proc. IEEE*, 78, 1550-1560.

Wetzel

Wetzel, A. (1983), *Evaluation of the Effectiveness of Genetic Algorithms in Combinatorial Optimization*, Unpublished manuscript, Univ. of Pittsburg, Pittsburg.

Williams

Williams, R. J. and Zipser, D. (1989) A Learning Algorithm for Continuously Running Fully Recurrent Neural Networks, *Neural Computation,* 1, 270-280.

Witten

Witten, I. H. and Frank, E. (2000) *Data Mining： Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann Publishers.

Woodfield

Woodfield, Terry J. (1990), Some notes on the Ljung-Box portmanteau statistic, *American Statistical Association 1990 Proceedings of the Statistical Computing Section*, 155–160.

Wu

Wu, S-I (1995) Mirroring Our Thought Processes, *IEEE Potentials*, 14, 36-41.

Yates, F.

Yates, F. (1936) A new method of arranging variety trials involving a large number of varieties. *Journal of Agricultural Science*, **26**, 424-455.