

# Package ‘stats4’

September 23, 2016

**Title** Statistical Functions using S4 Classes

**Version** 3.3.1

**Priority** base

**Author** R Core Team and contributors worldwide

**Description** Statistical Functions using S4 classes.

**Maintainer** R Core Team <R-core@r-project.org>

**Imports** graphics, methods, stats

**License** Part of R 3.3.1

## R topics documented:

stats4-package . . . . .	1
coef-methods . . . . .	2
confint-methods . . . . .	2
logLik-methods . . . . .	3
mle . . . . .	3
mle-class . . . . .	5
plot-methods . . . . .	6
profile-methods . . . . .	6
profile.mle-class . . . . .	7
show-methods . . . . .	8
summary-methods . . . . .	8
summary.mle-class . . . . .	9
update-methods . . . . .	9
vcov-methods . . . . .	10

<b>Index</b>	<b>11</b>
--------------	-----------

---

stats4-package	S4 クラスを用いた統計関数
----------------	----------------

---

## Description

S4 クラスを用いた統計関数.

## Details

このパッケージは [S version 4](#) クラスシステムを使った統計に対する関数とクラスを含む。

メソッドは現在最大尤度(クラス "mle" を返す関数 `mle()`) をサポートし, [AIC](#) と共に使う `logLik` に対するメソッドを含む。

## Author(s)

R Core チームと世界中の貢献者.

保守者 : R Core Team <R-core@r-project.org>

---

coef-methods	最尤推定オブジェクトから係数を取り出す
--------------	---------------------

---

## Description

"mle" オブジェクトから係数ベクトルを取り出す。

## メソッド

`signature(object = "ANY")` 総称的関数 : `coef` を見よ。

`signature(object = "mle")` 当てはめから完全な係数ベクトルを取り出す (固定係数を含む)。

`signature(object = "summary.mle")` 当てはめの要約から係数ベクトルと標準誤差を取り出す。

---

confint-methods	信頼区間メソッド
-----------------	----------

---

## Description

信頼区間を作る。

## メソッド

`signature(object = "ANY")` 総称的関数 : `confint` を見よ。

`signature(object = "mle")` 先ずプロファイル, そして次にプロファイルから信頼区間を作る。

`signature(object = "profile.mle")` 尤度プロファイルから信頼区間を作る。

---

logLik-methods	最大対数尤度を取り出す
----------------	-------------

---

### Description

"mle" オブジェクトから最大対数尤度を取り出す。

### メソッド

signature(object = "ANY") 総称的関数 : [logLik](#) を見よ。

signature(object = "mle") 当てはめから対数尤度を取り出す。

### Note

mle メソッドは `nobs` が指定されていない限り観測値の数を知らないのので、[BIC](#) と共に使うには適していないかもしれない。

---

mle	最尤推定
-----	------

---

### Description

最尤法でパラメータを推定する。

### Usage

```
mle(minuslogl, start = formals(minuslogl), method = "BFGS",
    fixed = list(), nobs, ...)
```

### Arguments

<code>minuslogl</code>	負の対数尤度を計算する関数。
<code>start</code>	名前付きリスト。最適器に対する初期値。
<code>method</code>	使用する最適化手法。 <a href="#">optim</a> を見よ。
<code>fixed</code>	名前付きリスト。最適化の過程で固定されるべきパラメータ。
<code>nobs</code>	オプションの整数：観測値数で例えば <a href="#">BIC</a> を計算するのに使われる。
<code>...</code>	<a href="#">optim</a> に渡される追加引数。

### Details

最適化関数 [optim](#) が負の対数尤度の最小値を見つけるために使われる。パラメータに対する近似共分散行列が最適値での Hessian 行列の逆行列から得られる。

### Value

クラス `mle-class` のオブジェクト。

**Note**

引数は  $-\log L$  ( $-2 \log L$  でない)ことを注意する。これは尤度が正確であること、漸近的尤度推測が正確であることをユーザに保証するためである。

**See Also**

[mle-class](#)

**Examples**

```
## 不要な正確さのプリントを避ける
od <- options(digits = 5)
x <- 0:10
y <- c(26, 17, 13, 12, 20, 5, 9, 8, 5, 4, 8)

## 簡単な一次元 MLE :
nLL <- function(lambda) -sum(stats::dpois(y, lambda, log = TRUE))
fit0 <- mle(nLL, start = list(lambda = 5), nobs = NROW(y))
# 一次元に対してはこれが好ましい :
fit1 <- mle(nLL, start = list(lambda = 5), nobs = NROW(y),
           method = "Brent", lower = 1, upper = 20)
stopifnot(nobs(fit0) == length(y))

## これは制約付きパラメータ空間を必要とする : 殆どのメソッドは NA を受け入れる
ll <- function(ymax = 15, xhalf = 6) {
  if(ymax > 0 && xhalf > 0)
    -sum(stats::dpois(y, lambda = ymax/(1+x/xhalf), log = TRUE))
  else NA
}
(fit <- mle(ll, nobs = length(y)))
mle(ll, fixed = list(xhalf = 6))
## 最適化の限界を使う別法
ll2 <- function(ymax = 15, xhalf = 6)
  -sum(stats::dpois(y, lambda = ymax/(1+x/xhalf), log = TRUE))
mle(ll2, method = "L-BFGS-B", lower = rep(0, 2))

AIC(fit)
BIC(fit)

summary(fit)
logLik(fit)
vcov(fit)
plot(profile(fit), absVal = FALSE)
confint(fit)

## 限界付き最適化
## 下限は実際に > 0,
## しかしプロファイルのストレステストのために >=0 を使う
(fit2 <- mle(ll, method = "L-BFGS-B", lower = c(0, 0)))
plot(profile(fit2), absVal = FALSE)

## より良いパラメータ化 :
ll3 <- function(lymax = log(15), lxhalf = log(6))
  -sum(stats::dpois(y, lambda = exp(lymax)/(1+exp(lxhalf)), log = TRUE))
(fit3 <- mle(ll3))
plot(profile(fit3), absVal = FALSE)
```

```
exp(confint(fit3))
```

```
options(od)
```

---

mle-class

最尤推定クラス

---

## Description

このクラスは総称的な最尤法の結果を要約する。

### クラスからのオブジェクト

オブジェクトは形式 `new("mle", ...)` の呼び出しで作ることができるが、最もしばしば `mle` の呼び出しの結果として得られる。

### スロット

`call`: クラス "language" のオブジェクト。 `mle` への呼び出し。

`coef`: Object of class クラス "numeric" のオブジェクト。 推定パラメータ。

`fullcoef`: クラス "numeric" のオブジェクト。 固定と推定パラメータ。

`vcov`: Object of class クラス "matrix" のオブジェクト。 近似分散共分散行列。

`min`: クラス "numeric" のオブジェクト。 目的関数の最小値。

`details`: `optim` が返すような "list"。

`minuslogl`: クラス "function" のオブジェクト。 負の尤度関数。

`nobs`: 長さ 1 の "integer"。 観測値数(呼び出しで正確に設定されなければしばしば NA)。

`method`: クラス "character" のオブジェクト。 使われた最適化手法。

### メソッド

**confint** signature(object = "mle"): 尤度プロファイルからの信頼区間。

**logLik** signature(object = "mle"): とり出された最大対数尤度。

**profile** signature(fitted = "mle"): 尤度プロファイル生成。

**nobs** signature(object = "mle"): 観測値数で、ここでは単に上で述べた nobs スロットにアクセスする。

**show** signature(object = "mle"): オブジェクトを簡略に表示。

**summary** signature(object = "mle"): オブジェクトの要約を生成する。

**update** signature(object = "mle"): 当てはめを更新。

**vcov** signature(object = "mle"): 分散共分散行列を取り出す。

---

 plot-methods

 パッケージ *stats4* 中の *plot* 関数に対するメソッド
 

---

### Description

"mle" オブジェクトに対するプロファイル尤度のプロット.

### Usage

```
## S4 method for signature 'profile.mle,missing'
plot(x, levels, conf = c(99, 95, 90, 80, 50)/100, nseg = 50,
     absVal = TRUE, ...)
```

### Arguments

x	クラス "profile.mle" のオブジェクト.
levels	水準, t 統計量の絶対値のスケールで, そこで区間を補間する. 普通 conf が levels を明示的に与える代わりに使われる.
conf	パラメータに関するプロファイルに基づく信頼区間の信頼係数の数値クトル.
nseg	プロファイル t 曲線のスプライン補間で使われる線分の数を与える整数.
absVal	プロットはプロファイル t の絶対値のスケールで行われるべきかどうかを指示する論理値. 既定では TRUE.
...	plot 関数への他の引数はここで渡すことができる.

### メソッド

signature(x = "ANY", y = "ANY") 総称的関数: [plot](#) を見よ.

signature(x = "profile.mle", y = "missing") x に対する尤度プロファイルのプロット.

---

 profile-methods

 パッケージ *stats4* 中の *profile* 関数へのメソッド
 

---

### Description

"mle" オブジェクトに対するプロファイル尤度.

### Usage

```
## S4 method for signature 'mle'
profile(fitted, which = 1:p, maxsteps = 100, alpha = 0.01,
       zmax = sqrt(qchisq(1 - alpha, 1L)), del = zmax/5,
       trace = FALSE, ...)
```

**Arguments**

fitted	プロファイルされるオブジェクト.
which	オプションでプロファイルされるパラメータの一部を選ぶ.
maxsteps	zmax をひとまとめにするステップ数の最大値.
alpha	zmax に対応する信頼水準で, Scheffe スタイルの多重検定区間に基づく. もし zmax が指定されると無視される.
zmax	符号付きルート尤度のプロファイル値に対するカットオフ値.
del	ルート尤度のスケールでの初期ステップサイズ.
trace	論理値. 結果を即座にプリントする.
...	現在使われない.

**Details**

プロファイリングアルゴリズムは尤度関数のルートをカバーする近似的に等間隔の(最適値から各方向の)少なくとも五つのパラメータ値のセットを見つけようとする. パラメータの曲率が大きい時意味のある結果を試み得るためにある注意がなされる. 尤度がなだらかになるかもしれないため, 常にカットオフ値を得ることは不可能かもしれないことを注意する.

**Value**

クラス "profile.mle" のオブジェクト, "profile.mle-class" を見よ.

**メソッド**

signature(fitted = "ANY") 総称的関数: [profile](#) を見よ.

signature(fitted = "mle") "mle" オブジェクトの最適値の近くでのプロファイル尤度.

---

profile.mle-class	クラス "profile.mle"; "mle" オブジェクトのためのプロファイリング情報
-------------------	---

---

**Description**

尤度の各パラメータに沿った尤度プロファイル.

**クラスからのオブジェクト**

形式 `new("profile.mle", ...)` の呼び出しでオブジェクトを作ることができるが, 最もしばしば "mle" オブジェクトに対し `profile` を適用して得られる.

**スロット**

**profile:** リストオブジェクト. 各要求されたパラメータに対して一つのプロファイルのリスト. 各プロファイルはデータフレームで, `z` と呼ばれる最初の列は -2 倍の対数尤度比の符号付きルートで, 他は `par.vals.` が先頭についた名前のパラメータ.

**summary:** クラス "summary.mle" のオブジェクト. プロファイルされたオブジェクトの要約,

## メソッド

**confint** signature(object = "profile.mle") : パラメータに対する近似信頼区間の生成にプロファイルを使う.

**plot** signature(x = "profile.mle", y = "missing") : 各パラメータに対するプロファイルプロットする.

## See Also

[mle](#), [mle-class](#), [summary.mle-class](#)

show-methods

パッケージ *stats4* 中の関数 *show* のメソッド

## Description

クラス `mle` と `summary.mle` のオブジェクトを示す.

## Methods

signature(object = "mle") `mle` オブジェクトの簡単な要約をプリントする. 係数と呼び出しだけ.

signature(object = "summary.mle") 呼び出し, 係数の表と標準誤差, そして  $-2\log L$  を示す.

summary-methods

パッケージ *stats4* 中の関数 *summary* に対するメソッド

## Description

オブジェクトの要約

## メソッド

signature(object = "ANY") 総称的関数.

signature(object = "mle") クラス "summary.mle" のオブジェクトとして, 推定値, 漸近的 SE, そして  $-2\log L$  の値を持つ要約を生成する.



---

summary.mle-class      *mle* オブジェクトの要約

---

### Description

"mle" オブジェクトを取り出す。

#### クラスからのオブジェクト

オブジェクトは形式 `new("summary.mle", ...)` の形の呼び出しで作ることができるが、最も普通には "mle" オブジェクトに対して `summary` を起動することで作られる。これらは `show` によりプリントされることを意図した値を含む。

#### スロット

`call`: クラス "language" のオブジェクト。 "mle" オブジェクトを生成した呼び出し。

`coef`: クラス "matrix" のオブジェクト。 推定係数と標準誤差。

`m2logL`: クラス "numeric" のオブジェクト。 対数尤度のマイナスの 2 倍。

#### メソッド

`show signature(object = "summary.mle")`: object を見栄え良くプリントする。

`coef signature(object = "summary.mle")`: coef スロットの中身を取り出す。

### See Also

[summary](#), [mle](#), [mle-class](#)

---

update-methods      パッケージ *stats4* 中の *update* 関数に対するメソッド

---

### Description

"mle" オブジェクトを更新する。

### Usage

```
## S4 method for signature 'mle'  
update(object, ..., evaluate = TRUE)
```

### Arguments

`object`      既存の当てはめ。

`...`      呼び出しに対する追加引数, 又は変更される値の引数。 引数 `name` を取り去るには `name = NULL` とする。

`evaluate`      もし真なら新しい呼び出しを評価する, さもなければ呼び出しを返す。

## メソッド

signature(object = "ANY") 総称的関数: `update` を見よ.  
signature(object = "mle") 当てはめを更新.

## Examples

```
x <- 0:10
y <- c(26, 17, 13, 12, 20, 5, 9, 8, 5, 4, 8)
ll <- function(y, lambda = 15, xhalf = 6)
  -sum(stats::dpois(y, lambda = y/(1+x/xhalf), log = TRUE))
fit <- mle(ll)
## 記録された呼び出しが ..1 を含むことを注意, S4 選択適用に伴う問題
update(fit, fixed = list(xhalf = 3))
```

---

vcov-methods

パッケージ *stats4* 中の *vcov* 関数に対するメソッド

---

## Description

"mle" オブジェクトから近似分散共分散行列を取り出す.

## メソッド

signature(object = "ANY") 総称的関数: `vcov` を見よ.  
signature(object = "mle") 推定パラメータ(もしあれば)に対する推定分散共分散行列  
を取り出す.

# Index

- \*Topic **classes**
  - mle-class, 5
  - profile.mle-class, 7
  - summary.mle-class, 9
- \*Topic **methods**
  - coef-methods, 2
  - confint-methods, 2
  - logLik-methods, 3
  - plot-methods, 6
  - profile-methods, 6
  - show-methods, 8
  - summary-methods, 8
  - update-methods, 9
  - vcov-methods, 10
- \*Topic **models**
  - mle, 3
  - stats4-package, 1
- \*Topic **package**
  - stats4-package, 1
- AIC, 2
- BIC, 3
- coef, 2
- coef, ANY-method (coef-methods), 2
- coef, mle-method (coef-methods), 2
- coef, summary.mle-method (coef-methods), 2
- coef-methods, 2
- confint, 2
- confint, ANY-method (confint-methods), 2
- confint, mle-method (confint-methods), 2
- confint, profile.mle-method (confint-methods), 2
- confint-methods, 2
- integer, 5
- list, 5
- logLik, 2, 3
- logLik, ANY-method (logLik-methods), 3
- logLik, mle-method (logLik-methods), 3
- logLik-methods, 3
- mle, 2, 3, 5, 8, 9
- mle-class, 5
- nobs, mle-method (mle-class), 5
- optim, 3, 5
- plot, 6
- plot, ANY, ANY-method (plot-methods), 6
- plot, profile.mle, missing-method (plot-methods), 6
- plot-methods, 6
- profile, 7
- profile, ANY-method (profile-methods), 6
- profile, mle-method (profile-methods), 6
- profile-methods, 6
- profile.mle-class, 7
- S version 4, 2
- show, mle-method (show-methods), 8
- show, summary.mle-method (show-methods), 8
- show-methods, 8
- stats4 (stats4-package), 1
- stats4-package, 1
- summary, 9
- summary, ANY-method (summary-methods), 8
- summary, mle-method (summary-methods), 8
- summary-methods, 8
- summary.mle-class, 9
- update, 10
- update, ANY-method (update-methods), 9
- update, mle-method (update-methods), 9
- update-methods, 9
- vcov, 10
- vcov, ANY-method (vcov-methods), 10
- vcov, mle-method (vcov-methods), 10
- vcov-methods, 10