Package ‘msos’

February 20, 2015

Type  Package
Title  Datasets and Functions used in Multivariate Statistics: Old
      School by John Marden
Version  1.0.1
Date  2014-01-22
Depends  R (>= 3.0.0), mclust, tree
Author  John Marden [aut, cph] and James Balamuta [cre, ctb, com]
Maintainer  James Balamuta <james.balamuta@gmail.com>
Description  Contains necessary Multivariate Analysis methods and Datasets used in STAT 571 at the University of Illinois at Urbana-Champaign
License  MIT + file LICENSE
NeedsCompilation  no
Repository  CRAN
Date/Publication  2014-01-23 15:12:57

R topics documented:

msos-package ................................................................. 2
births ........................................................................... 3
bothsidesmodel .............................................................. 4
bothsidesmodel.chisquare ............................................... 5
bothsidesmodel.df ......................................................... 6
bothsidesmodel.hotelling ............................................... 7
bothsidesmodel.lrt ....................................................... 8
bothsidesmodel.mle ....................................................... 9
bsm.fit ......................................................................... 10
bsm.simple ................................................................. 11
caffeine ........................................................................ 12
cars ............................................................................. 13
cereal ........................................................................... 14
crabs ........................................................................... 15
decathlon08 ................................................................. 16
**msos-package**

Datasets and Functions used in Multivariate Statistics - Old School by John Marden

**Description**
Contains necessary Multivariate Analysis methods and Datasets used in STAT 571. This package was built from source materials provided by Prof. John Marden

Index

<table>
<thead>
<tr>
<th>msos-package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>msos-package</td>
<td>Datasets and Functions used in Multivariate Statistics - Old School by John Marden</td>
</tr>
</tbody>
</table>
births

Details

Package: msos
Type: Package
Version: 1.0.1
Date: 2014-01-22
License: MIT

Author(s)

John Marden [aut, cph] and James Balamuta [cre, ctb, com]

Maintainer: James Balamuta <james.balamuta@gmail.com>

births Birthrates throughout the day in four Hospitals

Description

The data on average number of births for each hour of the day for four hospitals.

Usage

data(births)

Format

A double matrix with 24 observations on the following 4 variables.

Hospital1 Average number of births for each hour of the day within Hospital 1
Hospital2 Average number of births for each hour of the day within Hospital 2
Hospital3 Average number of births for each hour of the day within Hospital 3
Hospital4 Average number of births for each hour of the day within Hospital 4

Source

To be determined
bothsidesmodel  

Calculate the least squares estimates

Description

This function fits the model using least squares. It takes an optional pattern matrix \( P \) as in (6.51), which specifies which \( \beta_{ij} \)'s are zero. See Listing A.4 for the code.

Usage

\[
\text{bothsidesmodel}(x, y, z = \text{diag}(qq), \\
\text{pattern} = \text{matrix}(1, \text{nrow} = \text{p}, \text{ncol} = 1))
\]

Arguments

- **x**: An \( N \times P \) design matrix.
- **y**: The \( N \times Q \) matrix of observations.
- **z**: A \( Q \times L \) design matrix
- **pattern**: An optional \( N \times P \) matrix of 0's and 1's indicating which elements of \( \beta \) are allowed to be nonzero.

Value

A list with the following components:

- **Beta**: The least-squares estimate of \( \beta \).
- **SE**: The \( P \times L \) matrix with the \( ij \)th element being the standard error of \( \hat{\beta}_{ij} \).
- **T**: The \( P \times L \) matrix with the \( ij \)th element being the t-statistic based on \( \hat{\beta}_{ij} \).
- **Covbeta**: The estimated covariance matrix of the \( \hat{\beta}_{ij} \)'s.
- **df**: A \( p \)-dimensional vector of the degrees of freedom for the t-statistics, where the \( j \)th component contains the degrees of freedom for the \( j \)th column of \( \hat{\beta} \).
- **Sigmaz**: The \( Q \times Q \) matrix \( \hat{\Sigma}_z \).
- **Cx**: The \( Q \times Q \) residual sum of squares and crossproducts matrix.

See Also

bothsidesmodel.chisquare,bothsidesmodel.df,bothsidesmodel.hotelling,bothsidesmodel.lrt, and bothsidesmodel.mle.

Examples

\[
\#\text{Mouth Size Example from 6.4.1} \\
data(\text{mouths}) \\
x = \text{cbind}(1, \text{mouths}[,5]) \\
y = \text{mouths}[,1:4] \\
z = \text{cbind}(c(1,1,1,1),c(-3,-1,1,3),c(1,-1,-1,1),c(-1,3,-3,1)) \\
\text{bothsidesmodel}(x,y,z)
\]
bothsidesmodel.chisquare

Test subsets of $\beta$ are zero

Description

Tests the null hypothesis that an arbitrary subset of the $\beta_{ij}$’s is zero, based on the least squares estimates, using the $\chi^2$ test as in Section 7.1. The null and alternative are specified by pattern matrices $P_0$ and $P_A$, respectively. If the PA is omitted, then the alternative will be taken to be the unrestricted model.

Usage

bothsidesmodel.chisquare(x, y, z, pattern0, patternA = matrix(1, nrow = ncol(x), ncol = ncol(z)))

Arguments

- x: An $NxP$ design matrix.
- y: The $NxQ$ matrix of observations.
- z: A $QxL$ design matrix.
- pattern0: An $NxP$ matrix of 0’s and 1’s specifying the null hypothesis.
- patternA: An optional $NxP$ matrix of 0’s and 1’s specifying the alternative hypothesis.

Value

A list with the following components:

- Theta: The vector of estimated parameters of interest.
- Covtheta: The estimated covariance matrix of the estimated parameter vector.
- df: The degrees of freedom in the test.
- chisq: $T^2$ statistic in (7.4).
- pvalue: The p-value for the test.

See Also

bothsidesmodel, bothsidesmodel.df, bothsidesmodel.hotelling, bothsidesmodel.lrt, and bothsidesmodel.mle.

Examples

#TBA
bothsidesmodel.df  Obtain the degrees of freedom for a model.

Description

Determines the denominators needed to calculate an unbiased estimator of $\Sigma_R$.

Usage

bothsidesmodel.df(xx, n, pattern)

Arguments

- xx: Result of $(X^T \ast X)$, where T denotes transpose.
- n: Number of rows in observation matrix given
- pattern: An $N \times P$ matrix of 0's and 1's indicating which elements of $\beta$ are allowed to be nonzero.

Value

Returns a double matrix of size $N \times N$ containing the degrees of freedom for the test.

See Also

bothsidesmodel, bothsidesmodel.chisquare, bothsidesmodel.hotelling, bothsidesmodel.lrt, and bothsidesmodel.mle.

Examples

#Find the DF for a likelihood ratio test statistic.
x = cbind(1,c(-2,-1,0,1,2),c(2,-1,-2,-1,2),c(-1,2,0,-2,1),c(1,-4,6,-4,1))
# or x = cbind(1,poly(1:5,4))
data(skulls)
x = kronecker(x,rep(1,30))
y = skulls[,1:4]
z = diag(4)
pattern = rbind(c(1,1,1,1),1,0,0,0)
x = t(x)%%x
bothsidesmodel.df(xx,nrow(y),pattern)
bothsidesmodel.hotelling

Test blocks of $\beta$ are zero.

---

**Description**

Performs tests of the null hypothesis $H_0 : \beta^* = 0$, where $\beta^*$ is a block submatrix of $\beta$ as in Section 7.2.

**Usage**

bothsidesmodel.hotelling(x, y, z, rows, cols)

**Arguments**

- **x**: An $N \times P$ design matrix.
- **y**: The $N \times Q$ matrix of observations.
- **z**: A $Q \times L$ design matrix
- **rows**: The vector of rows to be tested.
- **cols**: The vector of columns to be tested.

**Value**

A list with the following components:

- **Hotelling**: A list with the components of the Lawley-Hotelling $T^2$ test (7.22):
  - $T^2$: The $T^2$ statistic (7.19).
  - $F$: The $F$ version (7.22) of the $T^2$ statistic.
  - $df$: The degrees of freedom for the $F$.
  - $pvalue$: The $p$-value of the $F$.

- **Wilks**: A list with the components of the Wilks $\Lambda$ test (7.37):
  - $\lambda$: The $\Lambda$ statistic (7.35).
  - $\chi^2$: The $\chi^2$ version (7.37) of the $\Lambda$ statistic, using Bartlett’s correction.
  - $df$: The degrees of freedom for the $\chi^2$.
  - $pvalue$: The $p$-value of the $\chi^2$.

**See Also**

bothsidesmodel, bothsidesmodel.chisquare, bothsidesmodel.df, bothsidesmodel.lrt, and bothsidesmodel.mle.
Examples

# Finds the Hotelling values for example 7.3.1
data(mouths)
x = cbind(1, mouths[,5])
y = mouths[,1:4]
z = cbind(c(1,1,1,1), c(-3,-1,1,3), c(1,-1,-1,1), c(-1,3,-3,1))
bothsidesmodel.lrt(x, y, z, 1:2, 3:4)

Description

Tests the null hypothesis that an arbitrary subset of the $\beta_{ij}$’s is zero, using the likelihood ratio test as in Section 9.4. The null and alternative are specified by pattern matrices $P_0$ and $P_A$, respectively. If the $P_A$ is omitted, then the alternative will be taken to be the unrestricted model.

Usage

bothsidesmodel.lrt(x, y, z, pattern0,
patternA = matrix(1, nrow = ncol(x), ncol = ncol(z)))

Arguments

x An $NxP$ design matrix.
y The $NxQ$ matrix of observations.
z A $QxL$ design matrix.
pattern0 An $NxP$ matrix of 0’s and 1’s specifying the null hypothesis.
patternA An optional $NxP$ matrix of 0’s and 1’s specifying the alternative hypothesis.

Value

A list with the following components:

chisq The likelihood ratio statistic in (9.44).
df The degrees of freedom in the test.
pvalue The p-value for the test.

See Also

bothsidesmodel.chisquare, bothsidesmodel.df, bothsidesmodel.hotelling, bothsidesmodel, and bothsidesmodel.mle.
Examples

data(caffeine)
x = cbind(rep(1,28),
    c(rep(-1,9),rep(0,10),rep(1,9)),
    c(rep(1,9),rep(-1,8,10),rep(1,9))
)
y = caffeine[-1]
z = cbind(c(1,1),c(1,-1))
pattern = cbind(c(rep(1,3)),1)
bsm = bothsidesmodel.lrt(x,y,z,pattern)

Description

This function fits the model using maximum likelihood. It takes an optional pattern matrix \( P \) as in (6.51), which specifies which \( \beta_{ij} \)'s are zero. See Listing A.6 for the code.

Usage

bothsidesmodel.mle(x, y, z = diag(qq),
    pattern = matrix(1,nrow = p,ncol = 1))

Arguments

\( x \) An \( NxP \) design matrix.
\( y \) The \( NxQ \) matrix of observations.
\( z \) A \( QxL \) design matrix.
\( \text{pattern} \) An optional \( NxP \) matrix of 0's and 1's indicating which elements of \( \beta \) are allowed to be nonzero.

Value

A list with the following components:

\( \text{Beta} \) The least-squares estimate of \( \beta \).
\( \text{SE} \) The \( PxL \) matrix with the \( ij \)th element being the standard error of \( \hat{\beta}_{ij} \).
\( \text{T} \) The \( PxL \) matrix with the \( ij \)th element being the t-statistic based on \( \hat{\beta}_{ij} \).
\( \text{Covbeta} \) The estimated covariance matrix of the \( \hat{\beta}_{ij} \)’s.
\( \text{df} \) A \( p \)-dimensional vector of the degrees of freedom for the t-statistics, where the \( j \)th component contains the degrees of freedom for the \( j \)th column of \( \hat{\beta} \).
\( \text{Sigmaz} \) The \( QxQ \) matrix \( \hat{\Sigma}_z \).
\( \text{Cx} \) The \( QxQ \) residual sum of squares and crossproducts matrix.
\( \text{ResidSS} \) The dimension of the model, counting the nonzero \( \beta_{ij} \)’s and components of \( \Sigma_z \).
Deviance  Mallow’s $C_p$ Statistic.
Dim        The dimension of the model, counting the nonzero $\beta_{ij}$’s and components of $\Sigma_z$
AICc       The corrected AIC criterion from (9.87) and (aic19)
BIC        The BIC criterion from (9.56).

See Also
bothsidesmodel.chisquare, bothsidesmodel.df, bothsidesmodel.hotelling, bothsidesmodel.lrt,
and bothsidesmodel.

Examples

data(mouths)
x = cbind(1, mouths[, 5])
y = mouths[, 1:4]
z = cbind(1, c(-3, -1, 1, 3), c(-1, 1, 1, -1), c(-1, 3, -1))
bothsidesmodel.mle(x, y, z, cbind(c(1, 1), 1, 0, 0))

bsm.fit  Helper function to determine $\beta$ estimates for MLE regression with patterning.

Description
Generates $\beta$ estimates for MLE using a conditioning approach with patterning support.

Usage
bsm.fit(x, y, z, pattern)

Arguments
x  An $N \times P + F$ design matrix, where $F$ is the number of columns conditioned on.
   This is equivalent to the multiplication of $xyzb$.

y  The $N \times Q - F$ matrix of observations, where $F$ is the number of columns conditioned on.
   This is equivalent to the multiplication of $Yz_a$.

z  A $Q - F \times L$ design matrix, where $F$ is the number of columns conditioned on.

pattern  An optional $N - F \times F$ matrix of 0’s and 1’s indicating which elements of $\beta$ are
           allowed to be nonzero.
bsm.simple

Value

A list with the following components:

- **Beta**  
  The least-squares estimate of $\beta$.

- **SE**  
  The $P + F \times L$ matrix with the $ij$th element being the standard error of $\hat{\beta}_{i,j}$.

- **T**  
  The $P + F \times L$ matrix with the $ij$th element being the t-statistic based on $\hat{\beta}_{i,j}$.

- **Covbeta**  
  The estimated covariance matrix of the $\hat{\beta}_{i,j}$'s.

- **df**  
  A $p$-dimensional vector of the degrees of freedom for the t-statistics, where the $j$th component contains the degrees of freedom for the $j$th column of $\hat{\beta}$.

- **Sigmaz**  
  The $Q - F \times Q - F$ matrix $\hat{\Sigma}_z$.

- **Cx**  
  The $Q \times Q$ residual sum of squares and crossproducts matrix.

See Also

- `bothsidesmodel.mle` and `bsm.simple`

Examples

```
#NA
```

bsm.simple  

*Helper function to determine $\beta$ estimates for MLE regression.*

Description

Generates $\beta$ estimates for MLE using a conditioning approach.

Usage

```
bsm.simple(x, y, z)
```

Arguments

- **x**  
  An $N \times P + F$ design matrix, where $F$ is the number of columns conditioned on. This is equivalent to the multiplication of $\text{xyzb}$.

- **y**  
  The $N \times Q - F$ matrix of observations, where $F$ is the number of columns conditioned on. This is equivalent to the multiplication of $\text{Yz_a}$.

- **z**  
  A $Q - F \times L$ design matrix, where $F$ is the number of columns conditioned on.

Details

The technique used to calculate the estimates is described in section 9.3.3.
Value

A list with the following components:

- **Beta**: The least-squares estimate of $\beta$.
- **SE**: The $P + FxL$ matrix with the $ij$th element being the standard error of $\hat{\beta}_{i,j}$.
- **T**: The $P + FxL$ matrix with the $ij$th element being the t-statistic based on $\hat{\beta}_{i,j}$.
- **Covbeta**: The estimated covariance matrix of the $\hat{\beta}_{i,j}$'s.
- **df**: A $p$-dimensional vector of the degrees of freedom for the t-statistics, where the $j$th component contains the degrees of freedom for the $j$th column of $\hat{\beta}$.
- **Sigmaz**: The $Q - FxQ - F$ matrix $\hat{\Sigma}_z$.
- **Cx**: The $QxQ$ residual sum of squares and crossproducts matrix.

See Also

- `bothsidesmodel.mle` and `bsm.fit`

Examples

```r
# Taken from section 9.3.3 to show equivalence to methods.
data(caffeine)
x = cbind(1,mouths[,5])
y = mouths[,1:4]
z = cbind(1,c(-3,-1,1,3),c(-1,1,1,-1),c(-3,-3,3,1))
yz = yz%*%solve(t(z))
yza = yz[,1:2]
xyz = cbind(x,yz[,3:4])
lm(yz ~ xyz - 1)
bsm.simple(xyz,yza,diag(2))
```

Description

Henson et al. [1996] conducted an experiment to see whether caffeine has a negative effect on short-term visual memory. High school students were randomly chosen: 9 from eighth grade, 10 from tenth grade, and 9 from twelfth grade. Each person was tested once after having caffeinated Coke, and once after having decaffeinated Coke. After each drink, the person was given ten seconds to try to memorize twenty small, common objects, then allowed a minute to write down as many as could be remembered. The main question of interest is whether people remembered more objects after the Coke without caffeine than after the Coke with caffeine.

Usage

```r
data(caffeine)
```
Format

A double matrix with 28 observations on the following 3 variables.

Grade Grade of the Student, which is either 8th, 10th, or 12th
With Number of items remembered after drinking Coke with Caffeine
Without Number of items remembered after drinking Coke without Caffeine

Source

Claire Henson, Claire Rogers, and Nadia Reynolds. Always Coca-Cola. Technical report, University Laboratory High School, Urbana, IL, 1996.

Description

The data set cars [Consumers’ Union, 1990] contains 111 models of automobile. The original data can be found in the S-Plus? [TIBCO Software Inc., 2009] data frame cu.dimensions. In cars, the variables have been normalized to have medians of 0 and median absolute deviations (MAD) of 1.4826 (the MAD for a N(0, 1)).

Usage

data(cars)

Format

A double matrix with 111 observations on the following 11 variables.

Length Overall length, in inches, as supplied by manufacturer
Wheelbase Length of wheelbase, in inches, as supplied by manufacturer.
Width Width of car, in inches, as supplied by manufacturer.
Height Height of car, in inches, as supplied by manufacturer
FrontHd Distance between the car’s head-liner and the head of a 5 ft. 9 in. front seat passenger, in inches, as measured by CU.
RearHd Distance between the car’s head-liner and the head of a 5 ft 9 in. rear seat passenger, in inches, as measured by CU.
FrtLegRoom Maximum front leg room, in inches, as measured by CU.
RearSeating Rear fore-and-aft seating room, in inches, as measured by CU.
FrtShld Front shoulder room, in inches, as measured by CU.
RearShld Rear shoulder room, in inches, as measured by CU
Luggage Luggage Area in Car

Source

Description

Chakrapani and Ehrenberg [1981] analyzed people's attitudes towards a variety of breakfast cereals. The data matrix cereal is 8 x 11, with rows corresponding to eight cereals, and columns corresponding to potential attributes about cereals. The original data consisted of the percentage of subjects who thought the given cereal possessed the given attribute. The present matrix has been doubly centered, so that the row means and columns means are all zero. (The original data can be found in the S-Plus [TIBCO Software Inc., 2009] data set cereal.attitude.)

Usage

data(cereal)

Format

A double matrix with 8 observations on the following 11 variables.

Return  A cereal one would come back to
Tasty  Tastes good
Popular  Popular with the entire family
Nourishing  Cereal is fulfilling
NaturalFlavor  Cereal lacks flavor additives
Affordable  Cereal is priced well for the content
GoodValue  Quantity for Price
Crispy  Stays crispy in milk
Fit  Keeps one fit
Fun  Fun for children

Source

Description

The crabs data frame has 200 rows and 8 columns, describing 5 morphological measurements on 50 crabs each of two colour forms and both sexes, of the species *Leptograpsus variegatus* collected at Fremantle, W. Australia.

Usage

crabs

Format

This data frame contains the following columns:

- **sp** species - "B" or "O" for blue or orange.
- **sex** as it says.
- **index** index 1:50 within each of the four groups.
- **FL** frontal lobe size (mm).
- **RW** rear width (mm).
- **CL** carapace length (mm).
- **CW** carapace width (mm).
- **BD** body depth (mm).

Source


MASS, R-Package

References

Decathlon Event Data from 2008 Olympics.

Description
The decathlon data set has scores on the top 24 men in the decathlon (a set of ten events) at the 2008 Olympics. The scores are the numbers of points each participant received in each event, plus each person’s total points.

Usage
data(decathlon08)

Format
A double matrix with 24 observations on the following 11 variables.

- \( x_{1 \text{meter}} \) Individual point score for 100 Meter event.
- \( \text{LongJump} \) Individual point score for Long Jump event.
- \( \text{ShotPut} \) Individual point score for Shot Put event.
- \( \text{HighJump} \) Individual point score for High Jump event.
- \( x_{4 \text{meter}} \) Individual point score for 400 Meter event.
- \( \text{Hurdles} \) Individual point score for 110 Hurdles event.
- \( \text{Discus} \) Individual point score for Discus event.
- \( \text{PoleVault} \) Individual point score for Pole Vault event.
- \( \text{Javelin} \) Individual point score for Javelin event.
- \( x_{15 \text{meter}} \) Individual point score for 1500 Meter event.
- \( \text{Total} \) Individual total point score for events participated in.

Source
NBC’s Olympic site
Decathlon Event Data from 2012 Olympics.

Description

The decathlon data set has scores on the top 26 men in the decathlon (a set of ten events) at the 2012 Olympics. The scores are the numbers of points each participant received in each event, plus each person's total points.

Usage

data(decathlon12)

Format

A double matrix with 26 observations on the following 11 variables.

- Meter100: Individual point score for 100 Meter event.
- ShotPut: Individual point score for Shot Put event.
- HighJump: Individual point score for High Jump event.
- Meter400: Individual point score for 400 Meter event.
- Hurdles110: Individual point score for 110 Hurdles event.
- Discus: Individual point score for Discus event.
- PoleVault: Individual point score for Pole Vault event.
- Javelin: Individual point score for Javelin event.
- Meter1500: Individual point score for 1500 Meter event.
- Total: Individual total point score for events participated in.

Source

NBC's Olympic site
### election

**Presidential Election Data**

**Description**

The data set election has the results of the first three US presidential races of the 2000’s (2000, 2004, 2008). The observations are the 50 states plus the District of Columbia, and the values are the \((D - R)/(D + R)\) for each state and each year, where D is the number of votes the Democrat received, and R is the number the Republican received.

**Usage**

`data(election)`

**Format**

A double matrix with 51 observations on the following 3 variables.

- **2000** Results for 51 States in Year 2000
- **2004** Results for 51 States in Year 2004
- **2008** Results for 51 States in Year 2008

**Source**

Calculated by Prof. John Marden, data source to be announced.

---

### exams

**Statistics Students' Scores on Exams**

**Description**

The exams matrix has data on 191 statistics students, giving their scores (out of 100) on the three midterm exams, and the final exam.

**Usage**

`data(exams)`

**Format**

A double matrix with 191 observations on the following 4 variables.

- **midterm1** Student score on the first midterm out of 100.
- **midterm2** Student score on the second midterm out of 100.
- **midterm3** Student score on the third midterm out of 100.
- **finalExam** Student score on the Final Exam out of 100.
Source
Data from one of Prof. John Marden's earlier classes

fillout Make a square matrix

Description
The function fillout takes a $Q(x(Q - L))$ matrix $Z$ and fills it out so that it is a square matrix $QxQ$.

Usage
fillout(z)

Arguments
$z$ A $Q(x(Q - L))$ matrix

Value
Returns a square matrix $QxQ$

See Also
tr, logdet

Examples
#Creates a 3x3 Matrix from 3x2 Data
fillout(cbind(c(1,2,3),c(4,5,6)))

grades Grades

Description
The data set contains grades of 107 students.

Usage
data(grades)
Format

A double matrix with 107 observations on the following 7 variables.

Gender  Sex (0=Male, 1=Female)
HW  Student Score on all Homework.
Labs  Student Score on all Labs.
InClass  Student Score on all In Class work.
Midterms  Student Score on all Midterms.
Final  Student Score on the Final.
Total  Student’s Total Score

Source

Data from one of Prof. John Marden’s earlier classes

Description

Sixteen dogs were treated with drugs to see the effects on their blood histamine levels. The dogs were split into four groups: Two groups received the drug morphine, and two received the drug trimethaphan, both given intravenously. For one group within each pair of drug groups, the dogs had their supply of histamine depleted before treatment, while the other group had histamine intact. (Measurements with the value "0.10" marked data that was missing and, were filled with that value arbitrarily.)

Usage

data(histamine)

Format

A double matrix with 16 observations on the following 4 variables.

Before  Histamine levels (in micrograms per milliliter of blood) before the inoculation.
After1  Histamine levels (in micrograms per milliliter of blood) one minute after inoculation.
After3  Histamine levels (in micrograms per milliliter of blood) three minute after inoculation.
After5  Histamine levels (in micrograms per milliliter of blood) five minutes after inoculation.

Source

**imax**  
*Obtain largest value index*

**Description**
Obtains the index of a vector that contains the largest value in the vector.

**Usage**

```r
imax(z)
```

**Arguments**

- `z` A vector of any length

**Value**
Returns the index of the largest value in a vector.

**Examples**

```r
#Iris example
x.iris = as.matrix(iris[,1:4])
#Gets group vector (1,...,1,2,...,2,3,...,3)
y.iris = rep(1:3,c(50,50,50))
yld.iris = lda(x.iris,y.iris)
disc = x.iris
disc = sweep(disc,2,yld.iris$c,'+')
yhat = apply(disc,1,imax)
```

---

**lda**  
*Linear Discrimination*

**Description**
Finds the coefficients $a_k$ and constants $c_k$ for Fisher’s linear discrimination function $d_k$ in (11.31) and (11.32).

**Usage**

```r
lda(x, y)
```

**Arguments**

- `x` The $N \times P$ data matrix.
- `y` The $N$-vector of group identities, assumed to be given by the numbers 1,...,$K$ for $K$ groups.
Value

A list with the following components:

a  A $P \times K$ matrix, where column $K$ contains the coefficients $a_k$ for (11.31). The final column is all zero.

c  The $K$-vector of constants $c_k$ for (11.31). The final value is zero.

See Also

sweep

Examples

#Iris example
x.iris = as.matrix(iris[,1:4])  
#Gets group vector (1,...,1,2,...,2,3,...,3)
y.iris = rep(1:3,c(50,50,50))  
ld.iris = lda(x.iris,y.iris)

---

leprosy  Leprosy Patients

Description

Dataset with leprosy patients found in Snedecor and Cochran [1989]. There were 30 patients, randomly allocated to three groups of 10. The first group received drug A, the second drug D, and the third group received a placebo. Each person had their bacterial count taken before and after receiving the treatment.

Usage

data(leprosy)

Format

A double matrix with 30 observations on the following 3 variables.

Before  Bacterial count taken before receiving the treatment.

After  Bacterial count taken after receiving the treatment.

Group  Group Coding: 0 = Drug A, 1 = Drug B, 2 = Placebo

Source

logdet  

*Log Determinant*

**Description**

Takes the log determinant of a square matrix. Log is that of base e sometimes referred to as ln().

**Usage**

```r
logdet(a)
```

**Arguments**

- `a` Square matrix ($Q \times Q$)

**Value**

Returns a single-value double.

**See Also**

- `tr`
- `fillout`

**Examples**

```r
# Identity Matrix of size 2
logdet(diag(c(2,2)))
```

---

mouths  

*Mouth Sizes*

**Description**

Measurements were made on the size of mouths of 27 children at four ages: 8, 10, 12, and 14. The measurement is the distance from the "center of the pituitary to the pteryomaxillary fissure" in millimeters. These data can be found in Potthoff and Roy [1964]. There are 11 girls (Sex=1) and 16 boys (Sex=0).

**Usage**

```r
data(mouths)
```
Format

A data frame with 27 observations on the following 5 variables.

- **Age8** Measurement on child’s month at age eight.
- **Age10** Measurement on child’s month at age ten.
- **Age12** Measurement on child’s month at age twelve.
- **Age14** Measurement on child’s month at age fourteen.
- **Sex** Sex Coding: Girl=1 and Boys=0

Source


negent

*Estimating negative entropy*

Description

Calculates the histogram-based estimate (A.2) of the negentropy,

\[
Negent(g) = \frac{1}{2} \left( 1 + \log(2\pi\sigma^2) \right) - \text{Entropy}(g)
\]

, for a vector of observations.

Usage

```r
negent(x, K = ceiling(log2(length(x)) + 1))
```

Arguments

- **x** The n-vector of observations.
- **K** The number of bins to use in the histogram.

Value

The value of the estimated negentropy.

See Also

`negent2D`, `negent3D`

Examples

```r
#NA
```
Maximizing negentropy for \( q = 2 \) dimensions

Description

Searches for the rotation that maximizes the estimated negentropy of the first column of the rotated data, for \( q = 2 \) dimensional data.

Usage

```
negent2D(y, m = 100)
```

Arguments

- `y`: The \( n \times 2 \) data matrix.
- `m`: The number of angles (between 0 and \( \pi \)) over which to search.

Value

A list with the following components:

- `vectors`: The \( 2 \times 2 \) orthogonal matrix \( G \) that optimizes the negentropy.
- `values`: Estimated negentropies for the two rotated variables. The largest is first.

See Also

`negent`, `negent3D`  

Examples

```
data(iris)
# Centers and scales the variables.
y = scale(as.matrix(iris[,1:2]))

# Obtains Negent Vectors for 2x2 matrix
gstar = negent2D(y, m=10)$vectors
```
negent3D

Maximizing negentropy for \( Q = 3 \) dimensions

Description

Searches for the rotation that maximizes the estimated negentropy of the first column of the rotated data, and of the second variable fixing the first, for \( q = 3 \) dimensional data. The routine uses a random start for the function optim using the simulated annealing option SANN, hence one may wish to increase the number of attempts by setting nstart to a integer larger than 1.

Usage

```r
negent3D(y, nstart = 1, m = 100, ...)
```

Arguments

- `y` The \( N \times 3 \) data matrix.
- `nstart` The number of times to randomly start the search routine.
- `m` The number of angles (between 0 and \( \pi \)) over which to search to find the second variables.
- `...` Further optional arguments to pass to the `optim` function to control the simulated annealing algorithm.

Value

A list with the following components:

- `vectors` The \( 3 \times 3 \) orthogonal matrix \( G \) that optimizes the negentropy.
- `values` Estimated negentropies for the three rotated variables, from largest to smallest.

Examples

```r
# Not run:
#Running this example will take approximately 30s.
#Centers and scales the variables.
y = scale(as.matrix(iris[,1:3]))

#Obtains Negent Vectors for 3x3 matrix
gstar = negent3D(y,nstart=100)$vectors

# End(Not run)
```
The Painter's Data of de Piles

Description

The subjective assessment, on a 0 to 20 integer scale, of 54 classical painters. The painters were assessed on four characteristics: composition, drawing, colour and expression. The data is due to the Eighteenth century art critic, de Piles.

Usage

painters

Format

The row names of the data frame are the painters. The components are:

Composition  Composition score.
Drawing      Drawing score.
Colour       Colour score.
Expression   Expression score.
School       The school to which a painter belongs, as indicated by a factor level code as follows: "A": Renaissance; "B": Mannerist; "C": Seicento; "D": Venetian; "E": Lombard; "F": Sixteenth Century; "G": Seventeenth Century; "H": French.

Source


MASS, R-Package

References

Description

Find the BIC and MLE from a set of observed eigenvalues for a specific pattern.

Usage

pcbic(eigenvals, n, pattern)

Arguments

eigenvals: The $Q$-vector of eigenvalues of the covariance matrix, in order from largest to smallest.
n: The degrees of freedom in the covariance matrix.
pattern: The pattern of equalities of the eigenvalues, given by the $K$-vector ($Q_1$, ..., $Q_K$) as in (13.8).

Value

A list with the following components:

lambdaHat: A $Q$-vector containing the MLE’s for the eigenvalues.
Deviance: The deviance of the model, as in (13.13).
Dimension: The dimension of the model, as in (13.12).
BIC: The value of the BIC for the model, as in (13.14).

See Also

pcbic.stepwise, pcbic.unite, and pcbic.subpatterns.

Examples

```r
# Build cars1
require("mclust")
mcars = Mclust(cars)
cars1 = cars[mcars$classification==1,]
xcars = scale(cars1)
eg = eigen(var(xcars))
pcbic(eg$values, 95, c(1, 1, 3, 3, 2, 1))
```
Description

Uses the stepwise procedure described in Section 13.1.4 to find a pattern for a set of observed eigenvalues with good BIC value.

Usage

pcbic.stepwise(eigenvals, n)

Arguments

eigenvals The $Q$-vector of eigenvalues of the covariance matrix, in order from largest to smallest.
n The degrees of freedom in the covariance matrix.

Value

A list with the following components:

Patterns A list of patterns, one for each value of length $K$.
BICs A vector of the BIC’s for the above patterns.
BestBIC The best (smallest) value among the BIC’s in BICs.
BestPattern The pattern with the best BIC.
lambdahat A $Q$-vector containing the MLE’s for the eigenvalues for the pattern with the best BIC.

See Also

pcbic, pcbic.unite, and pcbic.subpatterns.

Examples

#Build cars1
require("mclust")
mcars = Mclust(cars)
cars1 = cars[mcars$classification==1,]
xcars = scale(cars1)
eg = eigen(var(xcars))
pcbic.stepwise(eg$values,95)
pcbic.subpatterns  Obtain the best subpattern among the patterns.

Description
Obtains the best pattern and its BIC among the patterns obtainable by summing two consecutive terms in pattern0.

Usage
pcbic.subpatterns(eigenvals, n, pattern0)

Arguments
eigenvals  The $Q$-vector of eigenvalues of the covariance matrix, in order from largest to smallest.
n  The degrees of freedom in the covariance matrix.
pattern0  The pattern of equalities of the eigenvalues, given by the $K$-vector ($Q_1, \ldots, Q_K$) as in (13.8).

Value
pattern  A double matrix containing the pattern evaluated.
bic  A vector containing the BIC for the above pattern matrix.

See Also
pcbic, pcbic.stepwise, and pcbic.unite.

Examples
#NA

pcbic.unite  Obtain pattern

Description
Returns the pattern obtained by summing $q_i$ and $q_{i+1}$.

Usage
pcbic.unite(pattern, index1)
Arguments

- **pattern**: The pattern of equalities of the eigenvalues, given by the $K$-vector $(Q_1, \ldots, Q_K)$ as in (13.8).
- **index**: Index $i$ where $1 \leq i < K$

Value

Returns a vector containing a pattern.

See Also

- `pcbic`, `pcbic.stepwise`, and `pcbic.subpatterns`.

Examples

```r
#NA
```

<table>
<thead>
<tr>
<th>planets</th>
<th>Planets</th>
</tr>
</thead>
</table>

Description

Six astronomical variables are given on each of the historical nine planets (or eight planets, plus Pluto).

Usage

```r
data(planets)
```

Format

A double matrix with 9 observations on the following 6 variables.

- **Distance**: Average distance in millions of miles the planet is from the sun.
- **Day**: The length of the planet’s day in Earth days.
- **Year**: The length of year in Earth days.
- **Diameter**: The planet’s diameter in miles.
- **Temperature**: The planet’s temperature in degrees Fahrenheit.
- **Moons**: Number of moons.

Source

**predict_qda**

*Quadratic discrimination prediction*

**Description**

The function uses the output from the function `qda` (Section A.3.2) and a $P$-vector $X$, and calculates the predicted group for this $X$.

**Usage**

`predict_qda(qd, newx)`

**Arguments**

- `qd` The output from `qda`.
- `newx` A $P$-vector $X$ whose components match the variables used in the `qda` function.

**Value**

A $K$-vector of the discriminant values $d_k^Q(X)$ in (11.48) for the given $X$.

**See Also**

`qda`

**Examples**

```r
# Iris example
data(iris)
x.iris = as.matrix(iris[,1:4])
n = nrow(x.iris)
# Gets group vector (1,...,1,2,...,2,3,...,3)
y.iris = rep(1:3,c(50,50,50))
qd.iris = qda(x.iris,y.iris)
yhat.qd = NULL
for (i in 1:n) {
  yhat.qd = c(yhat.qd,imax(predict_qda(qd.iris,x.iris[i,])))
}
table(yhat.qd,y.iris)
```
Description

Data from Ware and Bowden [1977] taken at six four-hour intervals (labelled T1 to T6) over the course of a day for 10 individuals. The measurements are prostaglandin contents in their urine.

Usage

data(prostaglandin)

Format

A double matrix with 10 observations on the following 6 variables.

T1 First four-hour interval measurement of prostaglandin
T2 Second four-hour interval measurement of prostaglandin
T3 Third four-hour interval measurement of prostaglandin
T4 Fourth four-hour interval measurement of prostaglandin
T5 Fifth four-hour interval measurement of prostaglandin
T6 Sixth four-hour interval measurement of prostaglandin

Source


Description

The function returns the elements needed to calculate the quadratic discrimination in (11.48). Use the output from this function in predict.qda (Section A.3.2) to find the predicted groups.

Usage

qda(x, y)

Arguments

x The N x P data matrix.
y The N-vector of group identities, assumed to be given by the numbers 1,...,K for K groups.
Value

A list with the following components:

Mean       A $P \times K$ matrix, where column $K$ contains the coefficients $a_k$ for (11.31). The final column is all zero.
Sigma      A $K \times P \times P$ array, where the Sigma[$k$,] contains the sample covariance matrix for group $k$, $\Sigma_k$.
C          The $K$-vector of constants $c_k$ for (11.48).

See Also

predict_qda and lda

Examples

#Iris example
x.iris = as.matrix(iris[,1:4])
#Gets group vector (1,2,3,3,3,3)
y.iris = rep(1:3,c(50,50,50))
qd.iris = qda(x.iris,y.iris)

reverse.kronecker Reverses the matrices in a Kronecker product

Description

This function takes a matrix that is Kronecker product $A \otimes B$ (Definition 3.5), where $A$ is $P \times Q$ and $B$ is $N \times M$, and outputs the matrix $B \otimes A$.

Usage

reverse.kronecker(ab, p, qq)

Arguments

ab       The $(NP) \times (QM)$ matrix $A \otimes B$.
p       The number of rows of $A$.
qq       The number of columns of $A$.

Value

The $(NP) \times (QM)$ matrix $B \otimes A$.

See Also

kronecker
SAheart

Examples

( A = diag(1,3) )
( B = matrix(1:6, ncol = 2) )
( kron = kronecker(A,B) )
( reverse.kronecker(kron, 3,3))
( kron = kronecker(B,A) )

Description

A retrospective sample of males in a heart-disease high-risk region of the Western Cape, South Africa.

Usage

data(SAheart)

Format

A data frame with 462 observations on the following 10 variables.

- **sbp** systolic blood pressure
- **tobacco** cumulative tobacco (kg)
- **ldl** low density lipoprotein cholesterol
- **adiposity** a numeric vector
- **famhist** family history of heart disease, a factor with levels Absent Present
- **typea** type-A behavior
- **obesity** a numeric vector
- **alcohol** current alcohol consumption
- **age** age at onset
- **chd** response, coronary heart disease

Details

A retrospective sample of males in a heart-disease high-risk region of the Western Cape, South Africa. There are roughly two controls per case of CHD. Many of the CHD positive men have undergone blood pressure reduction treatment and other programs to reduce their risk factors after their CHD event. In some cases the measurements were made after these treatments. These data are taken from a larger dataset, described in Rousseauw et al, 1983, South African Medical Journal.

Source

ElemStatLearn, R-Package
Silhouettes for K-Means Clustering

Description

Find the silhouettes (12.9) for K-means clustering from the data and and the groups’ centers.

Usage

silhouette.km(x, centers)

Arguments

x The N x P data matrix.

centers The K x P matrix of centers (means) for the K Clusters, row k being the center for cluster K.

Details

This function is a bit different from the silhouette function in the cluster package, [Maechler et al., 2005].

Value

The n-vector of silhouettes, indexed by the observations’ indices.

Examples

#Uses sports data.
data(sportsranks)
#Obtain the K-means clustering for sports ranks.
kms = kmeans(sportsranks,centers=5,nstart=10)
#silhouettes
sil = silhouette.km(sportsranks,kms$centers)

Egyptian Skulls

Description

The data concern the sizes of Egyptian skulls over time, from Thomson and Randall-MacIver [1905]. There are 30 skulls from each of five time periods, so that n = 150 all together.

Usage

data(skulls)
**Format**

A double matrix with 150 observations on the following 5 variables.

- **MaximalBreadth**: Maximum length in millimeters
- **BasibregmaticHeight**: Basibregmatic Height in millimeters
- **BasialveolarLength**: Basialveolar Length in millimeters
- **NasalHeight**: Nasal Height in millimeters
- **TimePeriod**: Time groupings

**Source**


---

**softdrinks**

**Soft Drinks**

**Description**

A data set that contains 23 peoples’ ranking of 8 soft drinks: Coke, Pepsi, Sprite, 7-up, and their diet equivalents

**Usage**

data(softdrinks)

**Format**

A double matrix with 23 observations on the following 8 variables.

- **Coke**: Ranking given to Coke
- **Pepsi**: Ranking given to Pepsi
- **7up**: Ranking given to 7-up
- **Sprite**: Ranking given to Sprite
- **DietCoke**: Ranking given to Diet Coke
- **DietPepsi**: Ranking given to Diet Pepsi
- **Diet7up**: Ranking given to Diet 7-up
- **DietSprite**: Ranking given to Diet Sprite

**Source**

Data from one of Prof. John Marden’s earlier classes
sort_silhouette  

Sort the silhouettes by group

Description

Sorts the silhouettes, first by group, then by value, preparatory to plotting.

Usage

\[
\text{sort\_silhouette}(\text{sil}, \text{cluster})
\]

Arguments

- \text{sil}  
The \(n\)-vector of silhouette values.
- \text{cluster}  
The \(n\)-vector of cluster indices.

Value

The \(n\)-vector of sorted silhouettes.

Examples

\[
\begin{align*}
\text{#Uses sports data.} \\
\text{data(sportsranks)} \\
\text{#Obtain the K-means clustering for sports ranks.} \\
\text{kms = kmeans(sportsranks, centers=5, nstart=10)} \\
\text{#silhouettes} \\
\text{sil = silhouette(km, sportsranks, kms$centers)} \\
\text{ssil = sort\_silhouette(sil, kms$cluster)}
\end{align*}
\]

Spam

Description

In the Hewlett-Packard spam data, a set of \(n = 4601\) emails were classified according to whether they were spam, where "0" means not spam, "1" means spam. Fifty-seven explanatory variables based on the content of the emails were recorded, including various word and symbol frequencies. The emails were sent to George Forman (not the boxer) at Hewlett-Packard labs, hence emails with the words "George" or "hp" would likely indicate non-spam, while "credit" or "!" would suggest spam. The data were collected by Hopkins et al. [1999], and are in the data matrix Spam. (They are also in the R data frame spam from the ElemStatLearn package [Halvorsen, 2009], as well as at the UCI Machine Learning Repository [Frank and Asuncion, 2010].)

Usage

\[
\text{data}\left(\text{Spam}\right)
\]
Format

A double matrix with 4601 observations on the following 58 variables.

- WFmake Percentage of words in the e-mail that match make.
- WFaddress Percentage of words in the e-mail that match address.
- WFall Percentage of words in the e-mail that match all.
- WF3d Percentage of words in the e-mail that match 3d.
- WFfour Percentage of words in the e-mail that match our.
- WFOver Percentage of words in the e-mail that match over.
- WFremove Percentage of words in the e-mail that match remove.
- WFINTERNET Percentage of words in the e-mail that match internet.
- WForder Percentage of words in the e-mail that match order.
- WFMAIL Percentage of words in the e-mail that match mail.
- WFreceive Percentage of words in the e-mail that match receive.
- WFWILL Percentage of words in the e-mail that match will.
- WFPeople Percentage of words in the e-mail that match people.
- WFreport Percentage of words in the e-mail that match report.
- WFaddresses Percentage of words in the e-mail that match addresses.
- WFFREE Percentage of words in the e-mail that match free.
- Wfbusiness Percentage of words in the e-mail that match business.
- WFemail Percentage of words in the e-mail that match email.
- Wfyou Percentage of words in the e-mail that match you.
- WFCREDIT Percentage of words in the e-mail that match credit.
- Wfoy Percentage of words in the e-mail that match your.
- WFfont Percentage of words in the e-mail that match font.
- WF000 Percentage of words in the e-mail that match 000.
- WFmoney Percentage of words in the e-mail that match money.
- Wfhp Percentage of words in the e-mail that match hp.
- WFgeorge Percentage of words in the e-mail that match george.
- WF650 Percentage of words in the e-mail that match 650.
- WFlab Percentage of words in the e-mail that match lab.
- WFlabs Percentage of words in the e-mail that match labs.
- WFTELNET Percentage of words in the e-mail that match telnet.
- WF857 Percentage of words in the e-mail that match 857.
- WFDATA Percentage of words in the e-mail that match data.
- WF415 Percentage of words in the e-mail that match 415.
- WF85 Percentage of words in the e-mail that match 85.
- WFTechnology Percentage of words in the e-mail that match technology.
WM1999  Percentage of words in the e-mail that match 1999.
WMparts  Percentage of words in the e-mail that match parts.
WMpm     Percentage of words in the e-mail that match pm.
WMdirect Percentage of words in the e-mail that match direct.
WMcs     Percentage of words in the e-mail that match cs.
WMmeeting Percentage of words in the e-mail that match meeting.
WMoriginal Percentage of words in the e-mail that match original.
WMproject Percentage of words in the e-mail that match project.
WMre     Percentage of words in the e-mail that match re.
Wmedu    Percentage of words in the e-mail that match edu.
Wetable  Percentage of words in the e-mail that match table.
Wconference Percentage of words in the e-mail that match conference.
CFsemicolon Percentage of characters in the e-mail that match SEMICOLON
CFparen  Percentage of characters in the e-mail that match PARENTHESES
CFbracket Percentage of characters in the e-mail that match BRACKET
CFexclam Percentage of characters in the e-mail that match EXCLAMATION
CFdollar Percentage of characters in the e-mail that match DOLLAR
CFpound  Percentage of characters in the e-mail that match POUND
CRLaverage Average length of uninterrupted sequences of capital letters
CRLlongest Length of longest uninterrupted sequence of capital letters
CRLtotal Total number of capital letters in the e-mail
spam Denotes whether the e-mail was considered spam (1) or not (0), i.e. unsolicited commercial e-mail.

Source

| sportsranks | Sports ranking |

Description
Louis Roussos asked n = 130 people to rank seven sports, assigning #1 to the sport they most wish to participate in, and #7 to the one they least wish to participate in. The sports are baseball, football, basketball, tennis, cycling, swimming and jogging.

Usage
data(sportsranks)
Format

A double matrix with 130 observations on the following 7 variables.

Baseball  Baseball’s ranking out of seven sports.
Football   Football’s ranking out of seven sports.
Basketball Basketball’s ranking out of seven sports.
Tennis     Tennis’ ranking out of seven sports.
Cycling    Cycling’s ranking out of seven sports.
Swimming   Swimming’s ranking out of seven sports.
Jogging    Jogging’s ranking out of seven sports.

Source

Data from one of Prof. John Marden’s earlier classes

Description

A data set containing several demographic variables on the 50 United States, plus D.C.

Usage

data(states)

Format

A double matrix with 51 observations on the following 11 variables.

Population  In thousands
PctCities   The percentage of the population that lives in metropolitan areas
Doctors     Number per 100,000 people
SchoolEnroll The percentage enrollment in primary and secondary schools.
TeacherSalary The average salary of primary and secondary school teachers.
CollegeEnroll The percentage full-time enrollment at college
Crime       Violent crimes per 100,000 people
Prisoners   Number of people in prison per 10,000 people.
Poverty     Percentage of people below the poverty line.
Employment  Percentage of people employed
Income      Median household income
**Source**

**References**
http://www.census.gov/statab/www/ranks.html

---

**tr**

*Trace of a Matrix*

**Description**
Takes the traces of a matrix by extracting the diagonal entries and then summing over.

**Usage**
tr(x)

**Arguments**
 x Square matrix ($Q x Q$)

**Value**
Returns a single-value double.

**See Also**
logdet, fillout

**Examples**
#Identity Matrix of size 4, gives trace of 4.
tr(diag(4))
Index

*Topic **bothsidesmodel**
  - bothsidesmodel, 4
  - bothsidesmodel.chisquare, 5
  - bothsidesmodel.df, 6
  - bothsidesmodel.hotelling, 7
  - bothsidesmodel.lrt, 8
  - bothsidesmodel.mle, 9
  - bsm.fit, 10
  - bsm.simple, 11

*Topic **datasets**
  - births, 3
  - caffeine, 12
  - cars, 13
  - cereal, 14
  - crabs, 15
  - decathlon08, 16
  - decathlon12, 17
  - election, 18
  - exams, 18
  - grades, 19
  - histamine, 20
  - leprosy, 22
  - mouths, 23
  - painters, 27
  - planets, 31
  - prostaglandin, 33
  - SHeart, 35
  - skulls, 36
  - softdrinks, 37
  - Spam, 38
  - sportsranks, 40
  - states, 41

*Topic **kmeans**
  - silhouette.km, 36
  - sort.silhouette, 38

*Topic **lda**
  - lda, 21

*Topic **mle**
  - bsm.fit, 10
  - bsm.simple, 11

*Topic **negent**
  - negent, 24
  - negent2d, 25
  - negent3d, 26

*Topic **package**
  - msos-package, 2

*Topic **pcbic**
  - pcbic, 28
  - pcbic.stepwise, 29
  - pcbic.subpatterns, 30
  - pcbic.unite, 30

*Topic **qda**
  - predict_qda, 32
  - qda, 33

births, 3
bothsidesmodel, 4, 5–8, 10
bothsidesmodel.chisquare, 4, 5, 6–8, 10
bothsidesmodel.df, 4, 5, 6, 7, 8, 10
bothsidesmodel.hotelling, 4–6, 7, 8, 10
bothsidesmodel.lrt, 4–7, 8, 10
bothsidesmodel.mle, 4–8, 9, 11, 12
bsm.fit, 10, 12
bsm.simple, 11, 11

cafeine, 12

cars, 13

cereal, 14

crabs, 15

decathlon08, 16

decathlon12, 17

election, 18

exams, 18

fillout, 19, 23, 42

grades, 19
histamine, 20
imax, 21
kronecker, 34
lda, 21, 34
leprosy, 22
logdet, 19, 23, 42
mouths, 23
msos (msos-package), 2
msos-package, 2
negent, 24, 25
negent2D, 24, 25
negent3D, 24, 25, 26
optim, 26
painters, 27
pcbic, 28, 29–31
pcbic.stepwise, 28, 29, 30, 31
pcbic.subpatterns, 28, 29, 30, 31
pcbic.unite, 28–30, 30
planets, 31
predict_qda, 32, 34
prostaglandin, 33
qda, 32, 33
reverse.kronecker, 34
SAheart, 35
silhouette.km, 36
skulls, 36
softdrinks, 37
sort_silhouette, 38
Spam, 38
sportsranks, 40
states, 41
sweep, 22
tr, 19, 23, 42