

# Package ‘matie’

February 20, 2015

**Type** Package

**Title** Measuring Association and Testing Independence Efficiently

**Version** 1.2

**Date** 2013-05-17

**Author** Ben Murrell, Dan Murrell & Hugh Murrell.

**Maintainer** Hugh Murrell <hugh.murrell@gmail.com>

**Description**

Uses a ratio of weighted distributions to estimate association between variables in a data set.

**License** GPL-3

**Imports** dfoptim, mvtnorm, seriation, igraph, gplots, cba

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2013-11-04 17:51:26

## R topics documented:

matie-package . . . . .	2
agram . . . . .	3
amap . . . . .	4
baseballData . . . . .	5
fbvs . . . . .	6
fdg . . . . .	7
ma . . . . .	8
ma.nl . . . . .	10
ma.test . . . . .	11
pd . . . . .	13
rwt . . . . .	14
sbd . . . . .	14
shpd . . . . .	15
spa . . . . .	16
std . . . . .	18
tap . . . . .	19

---

matie-package

*Measuring Association and Testing Independence Efficiently*

---

## Description

A non-parametric measure of association between variables. The association score,  $A$ , ranges from 0 (when the variables are independent) to 1 (when they are perfectly associated).  $A$  is a kind of  $R^2$  estimate, and can be thought of as the proportion of variance in one variable explained by another (or explained by a number of other variables -  $A$  works for multivariate associations as well).

matie computes  $A$  by estimating a generalized  $R^2$ , which is computed from the ratio of the likelihood of an alternative model (allowing dependence between variables) over the likelihood of a null model (that forces the variables to be independent). See <http://arxiv.org/abs/1303.1828> for details.

The variables should be continuous.  $A$  is calculated from the ranks of the values, and ties are currently broken randomly. Ordinal data can be handled, but the properties of  $A$  have not been thoroughly investigated for non-continuous data with many ties.  $A$  is not currently implemented for category-valued data.

Version 1.2 provides underflow protection to fix a bug that appeared at high dimensions in earlier versions.

## Details

Package: matie  
Type: Package  
Version: 1.2  
Date: 2013-10-11  
License: GPL-3

## Author(s)

Ben Murrell, Dan Murrell, Hugh Murrell. Maintainer: Hugh Murrell <[hugh.murrell@gmail.com](mailto:hugh.murrell@gmail.com)>

## References

Discovering general multidimensional associations, <http://arxiv.org/abs/1303.1828>

## Examples

```
d <- sbd(NULL, n=400, Rsq=0.9)
aScore <- ma(d)
pValue <- ma.test(d, aScore)
```

---

agram	<i>Scattergram visualization of associations between all pairs of variables.</i>
-------	--

---

### Description

Uses the association array generated by `tap` to construct an array of association diagrams (pie charts and scatterplots).

### Usage

```
agram(dataSet, method, ranking, order, ...)
```

### Arguments

<code>dataSet</code>	the $n \times m$ data frame representing $n$ samples of the $m$ variables.
<code>method</code>	the method to use for linear association (included in the visualization) as performed by <code>cor</code> , one of "pearson", "spearman", "kendall". The default is "pearson".
<code>ranking</code>	a logical specifying whether or not to rank the data points before producing the scatter plots. The default is FALSE.
<code>order</code>	determines the variable ordering. one of FALSE (the default for no re-grouping) or "PC" (for principal components re-grouping) or "OLO" (for optimal leaf ordering)
<code>...</code>	passed on to graphics routines ...

### Details

Displays associations between all pairs of variables as a graphics array using pie-charts above the diagonal depicting the proportion of the total association that can be explained by a linear trend in blue, and the remaining non-linear association in red.

Below the diagonal, scatter plots are shown and colored according to proportional non-linearity (see `n13` returned by `ma.nl`).

### Value

Returns nothing.

### Note

The data set can be of any dimension.

### Author(s)

Ben Murrell, Dan Murrell & Hugh Murrell.

## References

This code was adapted from the `corrgram` package:

Author: Kevin Wright, Copyright: 2006 Kevin Wright, License: GPL2.

## See Also

[ma ma.nl tap](#)

## Examples

```
data(baseballData)
agram(baseballData,order="OLO",main="All pair associations for baseballData")
```

---

amap	<i>Heatmap and dendrogram from associations between all pairs of variables.</i>
------	---

---

## Description

Clusters variables in a dataframe using `dist = 1-A` and plots a heatmap with cluster dendrogram attached.

## Usage

```
amap(dataSet, palette, corAdjusted, method, title, ...)
```

## Arguments

<code>dataSet</code>	the $n \times m$ data frame representing $n$ samples of the $m$ variables.
<code>palette</code>	colour palette to use for the heatmap.
<code>corAdjusted</code>	a flag which if set causes bi-variate A scores to be multiplied by the sign of the same bi-variate correlation. This gives a direction to the association. defaults to FALSE.
<code>method</code>	the method to use for correlation association as performed by <code>cor</code> , one of "pearson", "spearman", "kendall". The default is "spearman".
<code>title</code>	a main title for the heatmap plot.
<code>...</code>	passed on to graphics routines ...

## Details

uses `heatmap.2` from the `gplots` package to display a heatmap and associated dendrogram with bi-variate scores provided by `matie`'s `A` value. If required these scores can be expanded into the range  $[-1, 1]$  through multiplication by the sign of the corresponding bi-variate correlation score.

**Value**

Returns nothing.

**Note**

The data set can be of any dimension.

**Author(s)**

Ben Murrell, Dan Murrell & Hugh Murrell.

**References**

This code was adapted from Moignard et al.

**See Also**

[ma tap](#)

**Examples**

```
data(baseballData)
amap(baseballData, corAdjusted=TRUE)
```

---

baseballData

*Baseball player performance and income dataset*

---

**Description**

This data set is a collection of performance statistics for a collection of baseball players including a Salary variable.

**Usage**

```
data(baseballData)
```

**Format**

A data frame consisting of 11 variables (columns) for 322 players (rows).

**Source**

CRAN: corrgram package

**References**

This dataset is a subset of the baseball dataset from the corrgram package.  
Author: Kevin Wright, Copyright: 2006 Kevin Wright, License: GPL2.

---

fbvs	<i>Forwards, backwards variable selection, picking variables to maximize explained variance.</i>
------	--

---

### Description

Attempts to find the best explanatory set of variables to explain a single variable in a data set. Iterates between adding the next best variable to the set and removing the variable (if any) whose exclusion maximizes the overall score.

### Usage

```
fbvs(dataSet, one, maxv, linear)
```

### Arguments

dataSet	the n x m data frame representing n observations of m variables.
one	a string specifying the name of one variable in the dataset, for which the best explanatory set is required. Defaults to the name of the last variable in the dataset.
maxv	an integer limiting the maximum number of variables in the explanatory set. Defaults to m-1.
linear	a boolean flag which causes fbvs to use a linear model to estimate R <sup>2</sup> instead of matie to estimate A when running the selection algorithm. Defaults to FALSE

### Details

Variable names are only added to the explanatory set if their inclusion results in an increase in the association measure.

### Value

Returns a list containing the following items:

one	the name of the one variable that requires the explanatory set
best	the best set of explanatory variables
Rsqr	an estimate for R <sup>2</sup> provided by the best set of explanatory variables

### Note

The data set can be of any dimension

### Author(s)

Ben Murrell, Dan Murrell & Hugh Murrell.

## References

Discovering general multidimensional associations, <http://arxiv.org/abs/1303.1828>

## See Also

[ma agram](#)

## Examples

```
# measure association for all pairs in a subrange of the baseball dataset
data(baseballData)
fbvs(baseballData,one="Salary")
fbvs(baseballData,one="Salary",linear=TRUE)

fbvs(baseballData,one="Salary",maxv=2)
fbvs(baseballData,one="Salary",maxv=2,linear=TRUE)
```

---

fdg	<i>Force directed graph visualization of associations between all pairs of variables.</i>
-----	---

---

## Description

Constructs and draws a force directed graph using an "all pairs" association matrix to drive attraction in the graph. This adjacency matrix can represent linear or non-linear associations.

## Usage

```
fdg(dataSet, dataName, method, cutoff, dim)
```

## Arguments

dataSet	a data frame
dataName	a string naming the dataSet. (set to NULL if unknown)
method	one of "A" for non-linear associations or "Rsq" for linear associations. Defaults to "A".
cutoff	a real number between 0 and 1, used to remove edges from the graph with association < cutoff. Defaults to 0.1
dim	graph layouts can be generated in 2 or 3 dimensions. note that rgl must be installed for viewing 3 dimensional layouts.

## Details

The igraph package is used to generate, layout and display the force directed graph. The layout employed is [layout.kamada.kawai](#).

**Value**

Returns nothing.

**Note**

All non numeric columns and non complete rows are stripped from the data set before the adjacency matrix is computed.

**Author(s)**

Ben Murrell, Dan Murrell & Hugh Murrell.

**References**

Discovering general multidimensional associations, <http://arxiv.org/abs/1303.1828>

**See Also**

[ma cor agram](#)

**Examples**

```
data(baseballData)
fdg(baseballData, dataName="baseballData", method="A", cutoff=0.35, dim=2)

# if you have rgl available then try dim=3
# fdg(baseballData, dataName="baseballData", method="Rsq", cutoff=0.15, dim=3)
```

---

ma

*Measure association*

---

**Description**

A non-parametric measure of association between variables. The association score  $A$  ranges from 0 (when the variables are independent) to 1 (when they are perfectly associated).  $A$  is a kind of  $R^2$  estimate, and can be thought of as the proportion of variance in one variable explained by another (or explained by a number of other variables -  $A$  works for multivariate associations as well).

**Usage**

```
ma(d, partition, ht, hp, hs, ufp)
```

**Arguments**

d	the $n \times m$ data frame containing $n$ observations of $m$ variables for which the maximal joint/marginal likelihood ratio score is required.
partition	a list of column indices specifying variable groupings. Defaults to <code>list(c(m), c(1:m-1))</code> where $m = \text{ncol}(d)$ which indicates explaining the last variable by means of all the other variables in the data set.
ht	tangent for the hyperbolic correction, default $ht = 43.6978644$ .
hp	power for the hyperbolic correction, default $hp = 0.8120818$ .
hs	scale for the hyperbolic correction, default $hs = 6.0049711$ .
ufp	for debugging purposes, default FALSE.

**Details**

An estimate of association (possibly nonlinear) is computed using a ratio of maximum likelihoods for the *marginal distribution* and maximum weighted likelihoods for the *joint distribution*.

Before the computation is carried out the data is ranked using the `rwt` function from the `matie` package. This estimate is usually conservative (ie low) and a small-samples hyperbolic correction is applied by adding an offset, `os`, to the joint likelihood given by:

$$os = \left(1 - \frac{1}{1+A \times ht}\right) \frac{n^{hp}}{hs}$$

before the likelihood ratio is re-computed.

As the dimension of the data set increases so does the under-estimation of  $A$  even with the hyperbolic correction.

**Value**

Returns a list of values ...

A	a score (including hyperbolic correction) estimating association for the data
rawA	the association score before hyperbolic correction
jointKW	the optimal kernel width for the joint distribution
altLL	the optimal weighted log likelihood for the alternate distribution
nullLL	the optimal log likelihood for the marginal distribution
marginalKW	the optimal kernel width for the marginal distribution
weight	the optimal weight used for the mixture
LRstat	the LR statistic, required for computing p values.
nRows	$n$ , the number of complete samples in the data set
mCols	$m$ , the number of variables in the data set
partition	user supplied partition for the variables in the data set
ufp	user supplied debugging flag

**Note**

The data set can be of any dimension.

**Author(s)**

Ben Murrell, Dan Murrell & Hugh Murrell.

**References**

Discovering general multidimensional associations, <http://arxiv.org/abs/1303.1828>

**See Also**

[rwt pd sbd shpd std](#)

**Examples**

```
# bivariate association
d <- shpd(n=1000,m=2,Rsq=0.9)
ma(d)$A
#
# multivariate association (the proportion of variance in "Salary"
# explained by "Hits" and "Years")
data(baseballData)
ma(baseballData,partition=list(11,c(2,7)))$A
```

---

ma.nl

*Calculates degree of nonlinearity for a particular association score.*

---

**Description**

This function estimates nonlinear proportion of variance between one variable and a group of variables after linear association between the variable and the group has been removed.

**Usage**

```
ma.nl(Y, X)
```

**Arguments**

Y                    A vector or a one column data frame.  
X                    a group of vectors or a data frame with the same number of samples as in Y

**Details**

A linear model,  $Y \sim X$ , is constructed and `ma` is used to compute  $R^2$  between Y and X.

**Value**

Returns a list of real numbers:

Rsq	linear association, the value of $R^2$ due to the linear model $Y \sim X$ .
A	total association (linear and nonlinear) between Y and the group X.
rA	the residual association (the association left in the residuals after the linear part has been regressed out of Y).
n11	$A - \text{Rsq}$ , the nonlinear part of the association.
n12	$(A - \text{Rsq}) / A$ , the nonlinear proportion of the association.
n13	$(A - \text{Rsq}) / (1 - \text{Rsq})$ , the proportion of total variance that is not explained by a linear model but is explained by A.

**Author(s)**

Ben Murrell, Dan Murrell & Hugh Murrell.

**References**

Discovering general multidimensional associations, <http://arxiv.org/abs/1303.1828>

**See Also**

[ma agram](#)

**Examples**

```
X1 = runif(1000)
X2 = runif(1000)
Y = sin(0.5*pi*X1) + sin(0.5*pi*X2) + rnorm(1000)*0.000001
ma.nl(Y,cbind(X1,X2))
#
# in the case of bivariate associations all these measures
# are symmetric apart from rA, the residual association
X = runif(1000)
Y = sin(0.5*pi*X) + rnorm(1000)*0.01
ma.nl(Y,X)$rA
ma.nl(X,Y)$rA
```

---

ma.test

*p-value for an association score.*

---

**Description**

This function computes the p-value for an association score using interpolation from a precomputed table (fast - only available for the bivariate case) or using a permutation test.

## Usage

```
ma.test(d, maStruct, permPartition, lookup, reps)
```

## Arguments

d	the data set
maStruct	the structure returned by a call to <a href="#">ma</a> . Must be provided. No default.
permPartition	a permutation partition for the data set depending on the statistical question being asked. Defaults to the partition returned by the call to <a href="#">ma</a> .
lookup	logical, if TRUE then a p-value is interpolated from a table of precomputed p-values. If FALSE then a monte carlo simulation is carried out. Table lookup is only accurate for associations on bivariate data with number of samples $\leq 400$ . By default, lookup is TRUE.
reps	no of repeats for the monte carlo simulation. defaults to 1000.

## Details

Computes a p value for an association from the likelihood ratio statistic computed by [ma](#). Uses interpolation by default for bivariate associations, and permutation testing for multivariate associations.

## Value

Returns a real number in the range [0,1].

## Author(s)

Ben Murrell, Dan Murrell & Hugh Murrell.

## References

Discovering general multidimensional associations, <http://arxiv.org/abs/1303.1828>

## See Also

[ma](#)

## Examples

```
d <- shpd(n=100,m=2, Rsq=0.2)
aScore <- ma(d)
ma.test(d, aScore)
# or set lookup to FALSE if you have computing time available
# ma.test(d, aScore, lookup=FALSE, reps=10000)
```

---

pd *Plot density (after an ma calculation).*

---

### Description

Optimal kernel widths output by [ma](#) are employed to recompute the weighted joint distribution for two variables in a data set, and a contour plot for this distribution is drawn.

### Usage

```
pd(d, iv=1, jv=2)
```

### Arguments

d	an n x m data frame with $m > 1$ .
iv	the column index of the independent variable
jv	the column index of the dependent variable

### Details

A data set of two variables is extracted from the user's data set and a full distribution is calculated using weighted marginal and joint likelihoods. The optimal kernel sizes and weighting are first computed via a call to [ma](#).

### Value

An n x n distribution of weighted likelihoods is returned.

### Note

The data set must contain at least 2 columns.

### Author(s)

Ben Murrell, Dan Murrell & Hugh Murrell.

### References

Discovering general multidimensional associations, <http://arxiv.org/abs/1303.1828>

### See Also

[ma](#)

**Examples**

```
f <- function(x,name="Sinusoidal",def="y = 1 + sin(x)"){
  return(1 + sin(x))
}
d <- sbd(f,min=-2*pi,max=2*pi,n=1000,Rsq=0.9)
ma(d)$A
dist <- pd(d)
```

rwt

*Rank with ties***Description**

This function replaces variable values in a data set with their ranks, using a random tie breaking strategy. It is called by [ma](#) and [pd](#).

sbd

*Generates sample bivariate data.***Description**

This function generates a sample bivariate data set.

**Usage**

```
sbd(func,min,max,n,Rsq)
```

**Arguments**

func	a user supplied function of one variable, $y = \text{func}(x)$ , near which data is generated.
min	min value for the domain of func
max	max value for the domain of func
n	number of sample points to generate
Rsq	coefficient of determination for the data set

**Details**

If func is NULL then a normal bivariate data set of n samples is generated with correlation coefficient  $\sqrt{\text{Rsq}}$ . If func is passed by the user then n sample points are scattered about  $y=\text{func}(x)$  with variance governed by the Rsq parameter

**Value**

Returns an  $n \times 2$  bivariate data set

**Note**

See examples below on how to set up user defined functions

**Author(s)**

Ben Murrell, Dan Murrell & Hugh Murrell.

**References**

Discovering general multidimensional associations, <http://arxiv.org/abs/1303.1828>

**See Also**

[ma](#)

**Examples**

```
f <- function(x,name="Sinusoidal",def="y = 1 + sin(x)"){
  return(1 + sin(x))
}
d <- sbd(f,min=-2*pi,max=2*pi,n=500, Rsq=0.9)
ma(d)$A
plot(d)
```

---

shpd

*Generates sample hyperplane data.*


---

**Description**

This function generates a sample hyperplane data set.

**Usage**

```
shpd(n,m=2, Rsq=0.7, Ri=0)
```

**Arguments**

n	number of sample points to generate
m	a hyperplane of dimension m-1 will be generated in m space
Rsq	the desired coefficient of determination for the hyperplane, indicates how far data points will vary from the plane.
Ri	correlation coefficient for "independent" variables within the hyperplane

**Details**

Scatters data around the hyperplane:  $V_m = \sum(V_1 \dots V_{m-1}) + \text{noise}$  where noise is adjusted so that the final dataset has coefficient of determination equal to Rsq. Variables  $V_1 \dots V_{m-1}$  are assumed to be independent but the user can make them dependent on each other by setting the  $V_i$  parameter away from zero.

**Value**

Returns a  $n \times m$  data set

**Author(s)**

Ben Murrell, Dan Murrell & Hugh Murrell.

**References**

Discovering general multidimensional associations, <http://arxiv.org/abs/1303.1828>

**See Also**

[ma](#)

**Examples**

```
d <- shpd(500,3,Rsq=0.8,Ri=0.01)
ma(d)$A
ma(d,partition=list(1,2))$A
ma(d,partition=list(3,1))$A
ma(d,partition=list(3,2))$A
ma(d,partition=list(3,c(1,2)))$A
cor(d)^2
# if m=3 and if you have rgl
# you can view the data set in 3D
# library("rgl")
# plot3d(d)
```

---

spa

*Semi-partial association (computes association while controlling for variables)*

---

**Description**

Computes the semi-partial association between a response variable and an explanatory variable, after controlling for a control variable.

**Usage**

```
spa(Y,X,C)
```

**Arguments**

Y            the response variable, a vector or column from a dataset  
X            the explanatory variable, a vector or column from a dataset  
C            the control variable, a vector or column from a dataset

**Details**

A semi-partial association (possibly nonlinear) is computed via:

```
ma(cbind(C,X,Y))$A - ma(cbind(C,Y))$A .
```

Inspired by the linear semi-partial correlation given by:

```
spcor.test(Y,X,C)
```

from the ppcor package.

**Value**

Returns a real number in the range [0,1].

**Note**

The parameters Y, X and C must be vectors of the same length.

**Author(s)**

Ben Murrell, Dan Murrell & Hugh Murrell.

**References**

Discovering general multidimensional associations, <http://arxiv.org/abs/1303.1828>

**See Also**

[ma](#)

**Examples**

```
# When the association between Y and X is entirely explained by C
C <- rnorm(1000)
X <- C+rnorm(1000)*0.5
Y <- C+rnorm(1000)*0.5
# See the relationship without controlling for C
ma(cbind(X,Y))$A
# See the relationship with C as a covariate (should be close to 0)
spa(Y,X,C)
# if you have ppcor then you can verify that
# the linear semi-partial correlation is similar,
# as these associations are all linear
# spcor.test(Y,X,C)$estimate^2
#
# When the association between Y and X is only partially explained by C
C <- rnorm(1000)
X <- C+rnorm(1000)*0.5
Y <- X+rnorm(1000)*0.5
# See the relationship without controlling for C
ma(cbind(X,Y))$A
# See the relationship with C as a covariate
# (should be lower than the uncontrolled one, but not as low as 0)
```

```

spa(Y,X,C)
# if you have ppcor then you can verify that
# the linear semi-partial correlation is similar,
# as these associations are all linear
# spcor.test(Y,X,C)$estimate^2
#
#
# if you have rgl you can plot the data
# library(rgl)
# plot3d(X,C,Y)

```

---

std

*Generates sample trivariate data set.*


---

### Description

This function generates a sample trivariate data set.

### Usage

```
std(func, xMin, xMax, yMin, yMax, n, Rsq)
```

### Arguments

func	a user supplied function of two variables, $z = \text{func}(x,y)$ , near which data is generated
xMin	min value for the x domain of func
xMax	max value for the x domain of func
yMin	min value for the y domain of func
yMax	max value for the y domain of func
n	number of sample points to generate
Rsq	coefficient of determination for the data set

### Details

If func is NULL then a normal trivariate data set of n samples is generated with correlation coefficients all set to  $\sqrt{\text{Rsq}}$ . If func is passed by the user then n sample points are scattered about  $z=\text{func}(x,y)$  with variance governed by the Rsq parameter.

### Value

Returns an  $n \times 3$  trivariate data set.

### Note

See examples below on how to set up user defined functions.

**Author(s)**

Ben Murrell, Dan Murrell & Hugh Murrell.

**References**

Discovering general multidimensional associations, <http://arxiv.org/abs/1303.1828>

**See Also**

[ma sbd](#)

**Examples**

```
f <- function(x,y,name="MexicanHat", def="z=(1-t^2)exp(-t^2div2), t^2=x^2+y^2"){
  t <- sqrt(x^2 + y^2)
  z <- (1.0 - t^2) * exp(- t * t / 2)
  return(z)
}
d <- std(f, xMin=-2, xMax=2, yMin=-2, yMax=2, n=500, Rsq=0.85)
ma(d)$A
# if you have rgl you can view the data set in 3D
# library("rgl")
# plot3d(d)
```

---

tap

*Test all pairs (computes association score for all pairs of variables).*

---

**Description**

Tries all pairs in an  $n \times m$  user supplied data set ( $n$  observations of  $m$  variables). Returns a square matrix of association measures.

**Usage**

```
tap(dataSet, one)
```

**Arguments**

dataSet	the $n \times m$ data frame representing $n$ observations of $m$ variables.
one	a string specifying the name of one variable in the dataset, for testing all pairs against one.

**Details**

All pairs of variables in a user supplied data set are measured for A association. Returns an  $m \times m$  data frame ( or  $m-1 \times m-1$  when one is set) storing A values for each pair of variables in the all pairs test.

**Value**

Returns a square data frame.

**Note**

The data set can be of any dimension.

**Author(s)**

Ben Murrell, Dan Murrell & Hugh Murrell.

**References**

Discovering general multidimensional associations, <http://arxiv.org/abs/1303.1828>

**See Also**

[ma](#)

**Examples**

```
# measure association for all pairs in a subrange of the baseball dataset
data(baseballData)
ap <- tap(baseballData)
```

# Index

## \*Topic **association**

agram, 3  
amap, 4  
fbvs, 6  
matie-package, 2  
spa, 16  
tap, 19

## \*Topic **baseballData**

agram, 3  
amap, 4

## \*Topic **coefficient of determination**

sbd, 14  
shpd, 15  
std, 18

## \*Topic **contour**

pd, 13

## \*Topic **correlation**

ma, 8  
spa, 16

## \*Topic **datasets**

baseballData, 5

## \*Topic **density**

matie-package, 2  
pd, 13

## \*Topic **distribution**

ma, 8  
matie-package, 2

## \*Topic **likelihood**

ma, 8  
matie-package, 2

## \*Topic **nonlinear**

ma, 8  
ma.nl, 10

## \*Topic **p value**

ma.test, 11

## \*Topic **semi-partial**

spa, 16

agram, 3, 7, 8, 11

amap, 4

baseballData, 5

cor, 8

fbvs, 6

fdg, 7

layout.kamada.kawai, 7

ma, 4, 5, 7, 8, 8, 11–17, 19, 20

ma.nl, 3, 4, 10

ma.test, 11

matie (matie-package), 2

matie-package, 2

pd, 10, 13, 14

rwt, 10, 14

sbd, 10, 14, 19

shpd, 10, 15

spa, 16

std, 10, 18

tap, 4, 5, 19