

Package ‘semutils’

April 22, 2013

Type Package

Title Utilities to facilitate structural equation modeling

Version 0.1.1

Date 2013-3-01

Author Joshua Wiley <jwiley.psych@gmail.com>

Maintainer Joshua Wiley <jwiley.psych@gmail.com>

Description A collection of tools to facilitate exploring
data in preparation for fitting structural equation models and to aid in fitting them.

License GPL (>= 3)

Depends R (>= 2.15), lavaan (>= 0.4-14), methods, ggplot2 (>= 0.9.0)

Imports MplusAutomation (>= 0.5-4), boot, reshape2, graphics

Suggests scales

Collate ‘utils.R’ ‘descriptives.R’ ‘styler.R’ ‘plotting.R’

R topics documented:

APASTyler	2
cd	2
cor2cov	3
corOK	4
corplot	5
moments	6
plot.SEMSummary	7
SEMSummary	8
Index	10

APASTyler

A generic function for pretty printing in (semi) APA Style

Description

A generic function for pretty printing in (semi) APA Style

A generic function for pretty printing in (semi) APA Style

A generic function for pretty printing in (semi) APA Style

A generic function for pretty printing in (semi) APA Style

Arguments

object	An object with a class matching one of the methods
object	lm object
digits	The number of digits to round results to. Defaults to 2.
pdigits	The number of digits to use for p values. Defaults to digits + 1 if missing.
file	An optional argument indicating whether the output should be written to a file.
object	mira object
lmobject	an lm object the degrees of freedom of which can be used for conservative F tests
digits	The number of digits to round results to. Defaults to 2.
pdigits	The number of digits to use for p values. Defaults to digits + 1 if missing.
file	An optional argument indicating whether the output should be written to a file.
object	SEMSummary object
digits	The number of digits to round results to. Defaults to 2.
type	A character vector giving what to print. Defaults to 'cov', the covariances. Other options are 'cor' and 'both'.
file	An optional argument indicating whether the output should be written to a file.

cd

Change directory

Description

The function takes a path and changes the current working directory to the path. If the directory specified in the path does not currently exist, it will be created.

Usage

```
cd(base, pre, num)
```

Arguments

base	a character string with the base path to the directory. This is required.
pre	an optional character string with the prefix to add to the base path. Non character strings will be coerced to character class.
num	an optional character string, prefixed by pre. Non character strings will be coerced to character class.

Details

The function has been designed to be platform independent, although it has had limited testing. Path creation is done using `file.path`, the existence of the directory is checked using `file.exists` and the directory created with `dir.create`. Only the first argument, is required. The other optional arguments are handy when one wants to create many similar directories with a common base.

Value

NULL, changes the current working directory

Examples

```
## Not run:
# an example just using the base
cd("~/testdir")

# an example using the optional arguments
base <- "~/testdir"
pre <- "test_"

cd(base, pre, 1)
cd(base, pre, 2)

## End(Not run)
```

cor2cov	<i>Convert a correlation matrix and standard deviations to a covariance matrix</i>
---------	--

Description

This is a simple function designed to convert a correlation matrix (standardized covariance matrix) back to a covariance matrix. It is the opposite of `cov2cor`.

Usage

```
cor2cov(V, sigma)
```

Arguments

V	an n x n correlation matrix. Should be numeric, square, and symmetric.
sigma	an n length vector of the standard deviations. The length of the vector must match the number of columns in the correlation matrix.

Value

an n x n covariance matrix

See Also

[cov2cor](#)

Examples

```
# using a built in dataset
cor2cov(cor(longley), sapply(longley, sd))

# should match the above covariance matrix
cov(longley)
all.equal(cov(longley), cor2cov(cor(longley), sapply(longley, sd)))
```

corOK

Return a non-missing correlation matrix

Description

Given a square, symmetric matrix (such as a correlation matrix) this function tries to drop the fewest possible number of variables to return a (square, symmetric) matrix with no missing cells.

Usage

```
corOK(x, maxiter = 100)
```

Arguments

<code>x</code>	a square, symmetric matrix or object coercable to such (such as a data frame).
<code>maxiter</code>	a number indicating the maximum number of iterations, currently as a sanity check. See details.

Details

The assumption that `x` is square and symmetric comes because it is assumed that the number of missing cells for a given column are identical to that of the corresponding row. `corOK` finds the column with the most missing values, and drops that (and its corresponding row), and continues on in like manner until the matrix has no missing values. Although this was intended for a correlation matrix, it could be used on other types of matrices. Note that because `corOK` uses an iterative method, it can be slow when many columns/rows need to be removed. For the intended use (correlation matrices) there probably should not be many missing. As a sanity check and to prevent tediously long computations, the maximum number of iterations can be set.

Value

A list with two elements

<code>x</code>	The complete non missing matrix.
<code>keep.indices</code>	A vector of the columns and rows from the original matrix to be kept (i.e., that are nonmissing).

Examples

```
cormat <- cor(iris[, -5])
# set missing
cormat[cbind(c(1,2), c(2,1))] <- NA

# print
cormat

# return complete
corOK(cormat)

# using maximum iterations
corOK(cormat, maxiter=0)

# clean up
rm(cormat)
```

corplot

Heatmap of a Correlation Matrix

Description

This function creates a heatmap of a correlation matrix using **ggplot2**.

Usage

```
corplot(x, coverage, points = TRUE, digits = 2,
        order = c("cluster", "asis"), ...,
        control.grobs = list())
```

Arguments

x	A correlation matrix or some other square symmetric matrix.
coverage	An (optional) matrix with the same dimensions as x giving the proportion of data present. Particularly useful when the correlation matrix is a pairwise present.
points	Logical whether or not to include points (bubble plots) of the coverage. Only has an effect if a coverage matrix is passed also. Defaults to TRUE.
digits	The number of digits to round to when printing the correlations on the heatmap. Text is suppressed when a coverage matrix is passed and points = TRUE.
order	A character string indicating how to order the resulting plot. Defaults to 'cluster' which uses hierarchical clustering to sensibly order the variables. The other option is 'asis' in which case the matrix is plotted in the order it is passed.
...	Additional arguments currently only passed to hclust and corOK.
control.grobs	A list of additional quote()d options to customize the ggplot2 output.

Details

The actual plot is created using `ggplot2` and `geom_tile`. In addition to creating the plot, the variables are ordered based on a hierarchical clustering of the correlation matrix. Specifically, $1 - x$ is used as the distance matrix. If coverage is passed, will also add a bubble plot with the area proportional to the proportion of data present for any given cell. Defaults for `ggplot2` are set, but it is possible to use a named list of quoted `ggplot` calls to override all defaults. This is not expected for typical use. Particularly `main`, `points`, and `text` as these rely on internal variable names; however, `labels`, the gradient color, and area scaling can be adjusted more safely.

Value

Primarily called for the side effect of creating a plot. However, the `ggplot2` plot object is returned, so it can be saved, replotted, edited, etc.

Examples

```
# example plotting the correlation matrix from the
# mtcars dataset
corplot(cor(mtcars))

dat <- as.matrix(iris[, 1:4])

# randomly set 25% of the data to missing
set.seed(10)
dat[sample(length(dat), length(dat) * .25)] <- NA

# create a summary of the data (including coverage matrix)
sdat <- SEMSummary(~ ., data = dat)
# using the plot method for SEMSummary (which basically just calls corplot)
plot(sdat)

# use the control.grobs argument to adjust the coverage scaling
# to go from 0 to 1 rather than the range of coverage
corplot(x = sdat$sSigma, coverage = sdat$coverage,
  control.grobs = list(area = quote(scale_size_area(limits = c(0, 1))))
)

# also works with plot() on a SEMSummary
plot(x = sdat, control.grobs = list(area = quote(scale_size_area(limits = c(0, 1)))))

rm(dat, sdat)
```

moments

Estimate the first and second moments

Description

This function relies on the **lavaan** package to use the Expectation Maximization (EM) algorithm to estimate the first and second moments (means and [co]variances) when there is missing data.

Usage

```
moments(data)
```

Arguments

data A data frame or an object coercable to a data frame. The means and covariances of all variables are estimated.

Value

A list containing the estimates from the EM algorithm.

mu A named vector of the means.

sigma The covariance matrix.

See Also

[SEMSummary](#)

Examples

```
# sample data
Xmiss <- as.matrix(iris[, -5])
# make 25% missing completely at random
set.seed(10)
Xmiss[sample(length(Xmiss), length(Xmiss) * .25)] <- NA
Xmiss <- as.data.frame(Xmiss)

# true means and covariance
colMeans(iris[, -5])
# covariance with n - 1 divisor
cov(iris[, -5])

# means and covariance matrix using list wise deletion
colMeans(na.omit(Xmiss))
cov(na.omit(Xmiss))

# means and covariance matrix using EM
moments(Xmiss)
# clean up
rm(Xmiss)
```

plot.SEMSummary

Plots SEMSummary object

Description

Plots SEMSummary object

Usage

```
## S3 method for class 'SEMSummary'
plot(x, y, ...)
```

Arguments

<code>x</code>	An object of class <code>SEMSummary</code> .
<code>y</code>	Ignored
<code>...</code>	Additional arguments passed on to the real workhorse, <code>corplot</code> .

See Also

[corplot](#), [SEMSummary](#)

SEMSummary

Summary Statistics for a SEM Analysis

Description

This function is designed to calculate the descriptive statistics and summaries that are often reported on raw data when the main analyses use structural equation modelling.

Usage

```
SEMSummary(formula, data,
            use = c("fiml", "pairwise.complete.obs", "complete.obs"))
```

Arguments

<code>formula</code>	A formula of the variables to be used in the analysis. See the ‘details’ section for more information.
<code>data</code>	A data frame, matrix, or list containing the variables used in the formula. This is a required argument.
<code>use</code>	A character vector of how to handle missing data. Defaults to “fiml”.

Details

This function calculates a variety of relevant statistics on the raw data used in a SEM analysis. Because it is meant for SEM style data, for now it expects all variables to be numeric. In the future I may try to expand it to handle factor variables somehow.

Both the formula and data arguments are required. The formula should be the right hand side only. The most common way to use it would be with variable names separated by ‘+’. For convenience, a ‘.’ is expanded to mean “all variables in the data set”. For a large number of variables or when whole datasets are being analyzed, this can be considerably easier to write. Also it facilitates column indexing by simply passing a subset of the data (e.g., `data[, 1:10]`) and using the ‘.’ expansion to analyze the first 10 columns. The examples section demonstrate this use.

Also noteworthy is that `SEMSummary` is not really meant to be used on its own. It is the computational workhorse, but it is meant to be used with a styling or printing method to produce simple output. `APASTyler` has methods for `SEMSummary` output.

There are several new ways to handle missing data now including listwise deletion, pairwise deletion, and using the EM algorithm, the default.

Value

A list with S3 class “SEMSummary”

names	A character vector containing the variable names.
n	An integer vector of the length of each variable used (this includes available and missing data).
nmissing	An integer vector of the number of missing values in each variable.
mu	A vector of the arithmetic means of each variable (on complete data).
stdev	A numeric vector of the standard deviations of each variable (on complete data).
Sigma	The numeric covariance matrix for all variables.
sSigma	The numeric correlation matrix for all variables.
coverage	A numeric matrix giving the percentage (technically decimal) of information available for each pairwise covariance/correlation.
pvalue	The two-sided p values for the correlation matrix. Pairwise present N used to calculate degrees of freedom.
call	A call object giving the call that created the model.

See Also

[APAStyler](#)

Examples

```
## Example using the built in iris dataset
s <- SEMSummary(~ Sepal.Length + Sepal.Width + Petal.Length, data = iris)
s # show output ... not very nice

## Prettier output from SEMSummary
APAStyler(s)

#### Subset the dataset and use the . expansion ####

## summary for all variables in mtcars data set
## with 11 variables, this could be a pain to write out
SEMSummary(~ ., data = mtcars)

## . expansion is also useful when we know column positions
## but not necessarily names
SEMSummary(~ ., data = mtcars[, c(1, 2, 3, 9, 10, 11)])

## clean up
rm(s)
```

Index

*Topic **hplot**

corplot, [5](#)

*Topic **multivariate**

moments, [6](#)

SEMSummary, [8](#)

*Topic **utilities**

cd, [2](#)

*Topic **utils**

corOK, [4](#)

APASTyler, [2](#), [9](#)

APASTyler,lm-method (APASTyler), [2](#)

APASTyler,mira-method (APASTyler), [2](#)

APASTyler,SEMSummary-method
(APASTyler), [2](#)

cd, [2](#)

cor2cov, [3](#)

corOK, [4](#)

corplot, [5](#), [8](#)

cov2cor, [4](#)

moments, [6](#)

plot.SEMSummary, [7](#)

SEMSummary, [7](#), [8](#), [8](#)