

# Package ‘JWileymisc’

August 24, 2016

**Type** Package

**Title** Miscellaneous Utilities and Functions

**Version** 0.2.0

**Date** 2016-08-24

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

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

**Description** A collection of miscellaneous tools and functions,  
often just to save typing or redundancy, not implement new models.

**License** GPL (>= 3)

**Depends** R (>= 3.2.0), data.table (>= 1.9.6), ggplot2 (>= 0.9.0),  
cowplot, foreach, lavaan (>= 0.5-15)

**Imports** stats, utils, Hmisc, multcompView, devtools, graphics, mgcv,  
plyr, mice

**Suggests** scales, testthat

**RoxygenNote** 5.0.1

**NeedsCompilation** no

## R topics documented:

APAStyler . . . . .	2
APAStyler.lm . . . . .	2
APAStyler.mira . . . . .	3
APAStyler.SEMSummary . . . . .	3
cd . . . . .	4
compareIVs . . . . .	5
cor2cov . . . . .	5
corOK . . . . .	6
corplot . . . . .	7
eglttable . . . . .	9
empirical_pvalue . . . . .	10
formatHtest . . . . .	10
formatMedIQR . . . . .	11
formatPval . . . . .	11
lagk . . . . .	12
moments . . . . .	13
mvqq . . . . .	14

param_summary . . . . .	14
param_summary_format . . . . .	15
plot.SEMSummary . . . . .	16
plot.SEMSummary.list . . . . .	16
SEMSummary . . . . .	17
SEMSummary.fit . . . . .	18
star . . . . .	19
TukeyHSDgg . . . . .	20
updateInstall . . . . .	21
<b>Index</b>	<b>22</b>

---

APAStyler	<i>A generic function for pretty printing in (semi) APA Style</i>
-----------	---

---

**Description**

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

**Usage**

APAStyler(object, ...)

**Arguments**

- |        |  |
|--------|--|
| object | An object with a class matching one of the methods |
| ...    | Additional arguments passed on to methods.         |

---

APAStyler.lm	<i>A generic function for pretty printing in (semi) APA Style</i>
--------------	---

---

**Description**

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

**Usage**

```
## S3 method for class 'lm'
APAStyler(object, digits = 2, pdigits, file, ...)
```

**Arguments**

- |         |   |
|---------|---|
| 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. |
| ...     | Additional arguments passed on to write.table.                                  |

---

APAStyler.mira	<i>A generic function for pretty printing in (semi) APA Style</i>
----------------	---

---

**Description**

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

**Usage**

```
## S3 method for class 'mira'
APAStyler(object, lmobject, digits = 2, pdigits, file, ...)
```

**Arguments**

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.
...	Additional arguments passed on to write.table.

---

APAStyler.SEMSummary	<i>A generic function for pretty printing in (semi) APA Style</i>
----------------------	---

---

**Description**

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

**Usage**

```
## S3 method for class 'SEMSummary'
APAStyler(object, digits = 2, type = c("cov", "cor",
  "both"), stars = FALSE, file = ifelse(.Platform$OS.type == "windows",
  "clipboard", FALSE), sep = "\t", ...)
```

**Arguments**

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'.
stars	A logical value whether to include significance values as stars (*** p < .001, ** p < .01, * p < .05).
file	An optional argument indicating whether the output should be written to a file.
sep	Character what the separator for the table should be. Defaults to tabs.
...	Additional arguments passed on to write.table.

## Examples

```
m <- SEMSummary(~., data = mtcars)
APAStyler(m, type = "cor", stars = FALSE, file = FALSE)
APAStyler(m, type = "cov", stars = FALSE, file = FALSE)
APAStyler(m, type = "both", stars = FALSE, file = FALSE)
APAStyler(m, type = "cor", stars = TRUE, file = FALSE)
APAStyler(m, type = "cov", stars = TRUE, file = FALSE)
APAStyler(m, type = "both", stars = TRUE, file = FALSE)
```

---

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)
```

---

compareIVs	<i>Compares the effects of various independent variables on dependent variables</i>
------------	---

---

### Description

Utility to estimate the unadjusted, covariate adjusted, and multivariate adjusted unique contributions of one or more IVs on one or more DVs

### Usage

```
compareIVs(dv, type, iv, covariates = character(), data,
  multivariate = FALSE, ...)
```

### Arguments

dv	A character string or vector of the dependent variable(s)
type	A character string or vector indicating the type of dependent variable(s)
iv	A character string or vector giving the IV(s)
covariates	A character string or vector giving the covariate(s)
data	The data to be used for analysis
multivariate	A logical value whether to have models with all IVs simultaneously.
...	Additional arguments passed on to the internal function, <code>.runIt</code> .

### Value

A list with all the model results.

### Examples

```
test1 <- compareIVs(
  dv = c("mpg", "disp"),
  type = c("normal", "normal"),
  iv = c("hp", "qsec"),
  covariates = "am",
  data = mtcars, multivariate = TRUE)
test1$OverallSummary
rm(test1)
```

---

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**

**x** a square, symmetric matrix or object coercable to such (such as a data frame).

**maxiter** 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

x	The complete non missing matrix.
keep.indices	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, pvalues, plot = c("cor", "p", "coverage"), 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.
pvalues	An (optional) matrix with the same dimensions as x giving the p values for each correlation. To show, use plot = "p".
plot	A character string indicating what to show on top of the heatmap. Can be 'coverage', in which case bubble points show coverage; 'p', in which case p values are shown, or 'cor', in which case correlations are shown. Only has an effect if a coverage (or pvalue) matrix is passed also. Defaults to cor.
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.

<code>order</code>	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.
<code>control.grobs</code>	A list of additional <code>quote()</code> d options to customize the <code>ggplot2</code> output.
<code>...</code>	Additional arguments currently only passed to <code>hclust</code> and <code>corOK</code> .

## 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 `quote()`d `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)

# getting p values instead of coverage
plot(sdat, plot = "p")

# showing correlations instead of coverage
plot(sdat, plot = "cor")

# 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)
```



egltable

*EGL Table function makes nice tables***Description**

Give a dataset and a list of variables, or just the data in the vars. For best results, convert categorical variables into factors.

**Usage**

```
egltable(vars, g, data, strict = TRUE, parametric = TRUE,
  simChisq = FALSE, sims = 1e+06)
```

**Arguments**

vars	Either an index (numeric or character) of variables to access from the data argument, or the data to be described itself.
g	A variable used to group/separate the data prior to calculating descriptive statistics.
data	optional argument of the dataset containing the variables to be described.
strict	Logical, whether to strictly follow the type of each variable, or to assume categorical if the number of unique values is less than or equal to 3.
parametric	Logical whether to use parametric tests in the case of multiple groups to test for differences. Only applies to continuous variables. If TRUE, the default, uses one-way ANOVA, and a F test. If FALSE, uses the Kruskal-Wallis test.
simChisq	Logical whether to estimate p-values for chi-square test for categorical data when there are multiple groups, by simulation. Defaults to FALSE. Useful when there are small cells as will provide a more accurate test in extreme cases, similar to Fisher Exact Test but generalizing to large dimension of tables.
sims	Integer for the number of simulations to be used to estimate p-values for the chi-square tests for categorical variables when there are multiple groups.

**Value**

A data frame of the table.

**Examples**

```
egltable(iris)
egltable(colnames(iris)[1:4], "Species", iris)
egltable(iris, parametric = FALSE)
egltable(colnames(iris)[1:4], "Species", iris,
  parametric = FALSE)
egltable(colnames(iris)[1:4], "Species", iris,
  parametric = c(TRUE, TRUE, FALSE, FALSE))
egltable(colnames(iris)[1:4], "Species", iris,
  parametric = c(TRUE, TRUE, FALSE, FALSE), simChisq=TRUE)
```

---

empirical_pvalue	<i>Calculates an empirical p-value based on the data</i>
------------------	--

---

### Description

This function takes a vector of statistics and calculates the empirical p-value, that is, how many fall on the other side of zero. It calculates a two-tailed p-value.

### Usage

```
empirical_pvalue(x, na.rm = TRUE)
```

### Arguments

x	a data vector to operate on
na.rm	Logical whether to remove NA values. Defaults to TRUE

### Value

a named vector with the number of values falling at or below zero, above zero, and the empirical p-value.

### Author(s)

Joshua F. Wiley <josh@elkhartgroup.com>

### Examples

```
empirical_pvalue(rnorm(100))
```

---

formatHtest	<i>Function to format the results of a hypothesis test as text</i>
-------------	--

---

### Description

Function to format the results of a hypothesis test as text

### Usage

```
formatHtest(x, type = c("t", "F", "chisq", "kw", "mh"), ...)
```

### Arguments

x	A htest class object
type	The type of htest. Currently one of: “t”, “F”, “chisq”, “kw”, or “mh” for t-tests, F-tests, chi-square tests, kruskal-wallis tests, and Mantel-Haenszel tests, respectively.
...	Arguments passed on to p-value formatting

**Value**

A character string with results

**Examples**

```
formatHtest(t.test(extra ~ group, data = sleep), type = "t")
formatHtest(anova(aov(mpg ~ factor(cyl), data = mtcars)), type = "F")
formatHtest(chisq.test(c(A = 20, B = 15, C = 25)), type = "chisq")
formatHtest(kruskal.test(Ozone ~ Month, data = airquality))
formatHtest(mantelhaen.test(UCBAdmissions), type = "mh")
```

---

formatMedIQR

---

*Function to format the median and IQR of a variable*


---

**Description**

Function to format the median and IQR of a variable

**Usage**

```
formatMedIQR(x, d = 2, na.rm = TRUE)
```

**Arguments**

x	the data to have the median and IQR calculated
d	How many digits to display. Defaults to 2.
na.rm	Logical whether to remove missing values. Defaults to TRUE.

**Value**

A character string with results

**Examples**

```
formatMedIQR(mtcars$mpg)
```

---

formatPval

---

*Function to simplify formatting p-values for easy viewing / publication*


---

**Description**

Function to simplify formatting p-values for easy viewing / publication

**Usage**

```
formatPval(x, d = 3, sd, includeP = FALSE, includeSign = FALSE,
  dropLeadingZero = TRUE)
```

**Arguments**

x	p values to convert
d	number of digits
sd	number of scientific digits. Defaults to d if missing.
includeP	logical value whether to include the character “p” itself. Defaults to FALSE.
includeSign	logical value whether to include the character “=” or “<”. Defaults to FALSE and if includeP = TRUE it must be TRUE.
dropLeadingZero	logical value whether to drop leading zeros for p-values. Defaults to TRUE.

**Value**

A character string with stars

**Examples**

```
formatPval(c(.00052456, .000000124, .01035, .030489, .534946))
formatPval(c(.00052456, .000000124, .01035, .030489, .534946), 3, 3, FALSE, TRUE)
formatPval(c(.00052456, .000000124, .01035, .030489, .534946), 3, 3, TRUE, TRUE)
formatPval(c(.00052456, .000000124, .01035, .030489, .534946), 5)
formatPval(c(1, .15346, .085463, .05673, .04837, .015353462,
             .0089, .00164, .0006589, .0000000053326), 3, 5)
formatPval(c(1, .15346, .085463, .05673, .04837, .015353462,
             .0089, .00164, .0006589, .0000000053326), 3, 5, dropLeadingZero = FALSE)
```

---

lagk

---

*Create a lagged variable*


---

**Description**

Given a variable, create a k lagged version, optionally do it by a grouping factor, such as an ID.

**Usage**

```
lagk(x, k = 1, by)
```

**Arguments**

x	the variable to lag
k	the length to lag it
by	a variable to lag by. Must be sorted.

**Value**

a vector of the lagged values

**Examples**

```
lagk(1:4, 1)
```

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.
...	Additional arguments passed on to the <code>estimate.moments.EM</code> function in <b>lavaan</b> . Note this is not an exported function.

**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)
```

---

mvqq	<i>This is a simple plotting function designed to help examine multivariate normality using the Mahalanobis distance.</i>
------	---

---

### Description

This is a simple plotting function designed to help examine multivariate normality using the Mahalanobis distance.

### Usage

```
mvqq(dat, use = c("fiml", "pairwise.complete.obs", "complete.obs"))
```

### Arguments

dat	A data frame or matrix of multivariate data to be plotted
use	A character vector indicating how the moments (means and covariance matrix) should be estimated in the presence of missing data. The default is to use full information maximum likelihood based on functions in <b>lavaan</b> .

### See Also

[SEMSummary](#)

### Examples

```
mvqq(mtcars)
```

---

param_summary	<i>Calculates summaries for a parameter</i>
---------------	---

---

### Description

This function takes a vector of statistics and calculates several summaries: mean, median, 95 the empirical p-value, that is, how many fall on the other side of zero.

### Usage

```
param_summary(x, trans = function(x) x, ..., na.rm = TRUE)
```

### Arguments

x	a data vector to operate on
trans	A function to transform the data. Used for summaries, but not p-values. Defaults to the identity function.
na.rm	Logical whether to remove NA values. Defaults to TRUE
...	Additional arguments passed to <code>formatPval</code> to control p-value printing.

**Value**

A data frame of summary statistics

**Author(s)**

Joshua F. Wiley <josh@elkhartgroup.com>

**Examples**

```
param_summary(rnorm(100))
```

---

param_summary_format	<i>Format a data frame of summary statistics</i>
----------------------	--

---

**Description**

This functions nicely formats a data frame of parameter summary statistics and is designed to be used with the param\_summary() function.

**Usage**

```
param_summary_format(d, digits = getOption("digits"), pretty = FALSE)
```

**Arguments**

d	A data frame of the parameter summary statistics
digits	Number of digits to round to for printing
pretty	Logical value whether prettified values should be returned. Defaults to FALSE.

**Value**

A formatted data frame of summary statistics or a formatted vector (if pretty = TRUE).

**Author(s)**

Joshua F. Wiley <josh@elkhartgroup.com>

**Examples**

```
set.seed(1234)
xsum <- do.call(rbind, apply(matrix(rnorm(100*10), ncol = 10),
  2, param_summary))
rownames(xsum) <- letters[1:10]
param_summary_format(xsum)
param_summary_format(xsum, pretty = TRUE)

rm(xsum)
```

---

plot.SEMSummary	<i>Plots SEMSummary object</i>
-----------------	--------------------------------

---

**Description**

Plots SEMSummary object

**Usage**

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

**Arguments**

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

**See Also**

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

**Examples**

```
# default plot
plot(SEMSummary(~ ., data = mtcars))

# same as default
plot(SEMSummary(~ ., data = mtcars), plot = "coverage")

# shows p values
plot(SEMSummary(~ ., data = mtcars), plot = "p")

# shows correlations
plot(SEMSummary(~ ., data = mtcars), plot = "cor")
```

---

plot.SEMSummary.list	<i>Plots SEMSummary.list object</i>
----------------------	-------------------------------------

---

**Description**

Plots SEMSummary.list object

**Usage**

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



**Arguments**

<code>x</code>	An object of class <code>SEMSummary.list</code> .
<code>y</code>	Ignored
<code>which</code>	either a numeric vector based on the positions, or a character vector giving the names of the levels of the list to plot.
<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.

**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)
```

---

SEMSummary.fit

---

*Summary Statistics for a SEM Analysis*


---

**Description**

This is a low level fitting function, for SEMSummary.

**Usage**

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

**Arguments**

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

**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.

**See Also**

[SEMSummary](#)

---

star

*Function to simplify converting p-values to asterisks*

---

**Description**

Function to simplify converting p-values to asterisks

**Usage**

```
star(x, includeMarginal = FALSE)
```

**Arguments**

x	p values to convert to stars
includeMarginal	logical value whether to include a symbol for marginally significant >.05 but < .10 p-values. Defaults to FALSE.

**Value**

A character string with stars

**Examples**

```
star(c(.0005, .001, .005, .01, .02, .05, .08, .1, .5, 1))
```

---

TukeyHSDgg	<i>Tukey HSD Plot</i>
------------	-----------------------

---

**Description**

This calculates and displays means, confidence intervals as well as which groups are different based on Tukey's HSD. Inspired by <http://stackoverflow.com/questions/18771516/is-there-a-function-to-add-aov-post-hoc-testing-results-to-ggplot2-boxplot>

**Usage**

```
TukeyHSDgg(x, y, d, ci = 0.95, ordered = FALSE, ...)
```

**Arguments**

x	X
y	Y
d	D
ci	Confidence interval, defaults to .95
ordered	Logical, defaults to FALSE.
...	Additional arguments passed on.

**Value**

A ggplot graph object.

**Examples**

```
TukeyHSDgg("cyl", "mpg", mtcars)
TukeyHSDgg("Species", "Sepal.Length", iris, ci = .9)
```

---

updateInstall	<i>Update R and install my core package set</i>
---------------	---

---

**Description**

Update R and install my core package set

**Usage**

```
updateInstall(x, repo)
```

**Arguments**

x	A character vector of any additional packages to be installed
repo	The repository to be used. Defaults to <code>getOption("repos")</code>

**Value**

NULL, called for its side effect.

**Examples**

```
# updateInstall()
```

# Index

- \*Topic **hplot**
  - corplot, [7](#)
- \*Topic **misc**
  - formatHtest, [10](#)
  - formatMedIQR, [11](#)
  - formatPval, [11](#)
  - star, [19](#)
- \*Topic **multivariate**
  - moments, [13](#)
  - mvqq, [14](#)
  - SEMSummary, [17](#)
  - SEMSummary.fit, [18](#)
- \*Topic **plot**
  - TukeyHSDgg, [20](#)
- \*Topic **utilities**
  - cd, [4](#)
  - empirical\_pvalue, [10](#)
  - param\_summary, [14](#)
  - param\_summary\_format, [15](#)
- \*Topic **utils**
  - corOK, [6](#)
  - eglttable, [9](#)
  - lagk, [12](#)
  - updateInstall, [21](#)

APASTyler, [2](#), [18](#)  
APASTyler.lm, [2](#)  
APASTyler.mira, [3](#)  
APASTyler.SEMSummary, [3](#)

cd, [4](#)  
compareIVs, [5](#)  
cor2cov, [5](#)  
corOK, [6](#)  
corplot, [7](#), [16](#), [17](#)  
cov2cor, [6](#)

eglttable, [9](#)  
empirical\_pvalue, [10](#)

formatHtest, [10](#)  
formatMedIQR, [11](#)  
formatPval, [11](#)

lagk, [12](#)

moments, [13](#)  
mvqq, [14](#)

param\_summary, [14](#)  
param\_summary\_format, [15](#)  
plot.SEMSummary, [16](#)  
plot.SEMSummary.list, [16](#)

SEMSummary, [13](#), [14](#), [16](#), [17](#), [17](#), [19](#)  
SEMSummary.fit, [18](#)  
star, [19](#)

TukeyHSDgg, [20](#)

updateInstall, [21](#)