Package 'psytabs'

December 29, 2012

Type Package

Title Produce well-formatted tables for psychological research

Version 0.5

Date 2012-12-28

Author Johannes Beller, Soeren Kliem

Maintainer Johannes Beller <johannesbeller@gmail.com>

Depends psych, plyr, rtf, R2HTML, mokken, lavaan, semTools

Description Psytabs produces well-
formatted tables in .rtf or .html,which largely conform to ''psychological style''.

License GPL-2

Repository CRAN

Date/Publication 2012-12-29 07:30:22

R topics documented:

psytabs-package *Produce well-formatted tables for psychological research*

Description

Psytabs produces well-formatted tables in .rtf or .html, which largely conform to "psychological style".

Details

Package:	psytabs
Type:	Package
Version:	0.5
Date:	2012-12-28
License:	GPL-2

Author(s)

Johannes Beller, Soeren Kliem

Maintainer: Johannes Beller <johannesbeller@gmail.com>

addPercent *Add percent-signs to a table*

Description

Internal function. Should not be used.

Usage

```
addPercent(x)
```

Arguments

x	Table.

Value

A table with added percent-signs.

addPercentToCount *Add percent-values to a table.*

Description

Internal function. Should not be used.

Usage

```
addPercentToCount(x, y)
```

Arguments

x	Table containing absolute values.
y	Table containing values in percent.

Value

A table containing both absolute values and values in percent.

compareModels *Compare two cfa models fitted by lavaan.*

Description

Internal function. Should not be used.

The function is adapted from the lavaan package.

Usage

```
compareModels(fm0, fm1, scaled = FALSE, fm0.scaling = 1, fm1.scaling = 1)
```

Arguments

fm0	First model.
fm1	Second model.
scaled	Is chisq scaled, i.e. a Satorra-Bentler-scaled chi-squared statistic?
fm0.scaling	Scaling factor of first model.
fm1.scaling	Scaling factor of second model.

Value

Some model comparison statistics.

Examples

```
###
1+1
```

`corTable` *Correlation matrix table.*

Description

Produces a simple correlation matrix table.

Usage

```
corTable(data, sd = FALSE)
```

Arguments

`data`	data.frame containing the variables for which the correlation matrix should be calculated.
`sd`	logical value that toggles Whether the standard deviation should be displayed in the diagonal.

Value

A dataframe comprising the correlation matrix table.

Examples

```
data(myData)
(cor.tab <- corTable(myData[,1:4]))
corTable(myData[,1:4], sd=TRUE)
#saveTable(cor.tab, "corTab.rtf")
```

`demographicTable` *Demographic table.*

Description

Produces a table of the distribution of demographic characteristics.

Usage

```
demographicTable(hor_fact, ver_fact, count = TRUE, percent = TRUE, header = TRUE)
```

Arguments

hor_fact	factor constituting the columns of the table.
ver_fact	factor constituting the rows of the table.
count	logical value that toggles whether to include the absolute values in the table.
percent	logical value that toggles whether to include the values in percent in the table.
header	logical value that toggles whether to include a header for the row factor.

Value

A dataframe constituting the demographic table.

Examples

```
data(myData)

tab.1 <- demographicTable(myData$sex, myData$age_group7)
tab.1
tab.2 <- demographicTable(myData$sex, myData$age_group7, count=FALSE)
tab.2
tab.3 <- demographicTable(myData$sex, myData$age_group7, percent=FALSE)
tab.3

#saveTable(tab.1, "demographicTable.rtf")
```

---

measurementInvarianceTable

Measurement invariance table.

Description

Produces a table summarizing the results of a measurement invariance analysis as conducted by the respective function of the lavaan and semTools package.

Usage

```
measurementInvarianceTable(measurement.invariance)
```

Arguments

measurement.invariance	Results returned by the measurementInvariance function of the lavaan or semTools package.

Details

Please note that if the scaled chi-squared statistic is used a special chi-squared difference test is calculated, because the difference between two scaled chi-square statistics does not follow the chi-squared distribution. See also http://www.statmodel.com/chidiff.shtml.

Value

A measurement invariance table.

Examples

```
library(semTools)
#Example taken from the semTools package
HW.model <- ' visual =~ x1 + x2 + x3
              textual =~ x4 + x5 + x6
              speed =~ x7 + x8 + x9 '

mi.result <- measurementInvariance(HW.model, data=HolzingerSwineford1939, group="school")
tab.1 <- measurementInvarianceTable(mi.result)
tab.1

mi.strict.result <- measurementInvariance(HW.model, data=HolzingerSwineford1939, strict=TRUE, group="school")
tab.2 <- measurementInvarianceTable(mi.strict.result)
tab.2

mi.robust.result <- measurementInvariance(HW.model, data=HolzingerSwineford1939, estimator="MLM", group="schoo
tab.3 <- measurementInvarianceTable(mi.robust.result)
tab.3

mi.robstrict.result <- measurementInvariance(HW.model, data=HolzingerSwineford1939, estimator="MLM", strict=TR
tab.4 <- measurementInvarianceTable(mi.robstrict.result)
tab.4

#saveTable(tab.2, "measurementInvarianceTable.rtf")
```

---

mokkenTable *Mokken table.*

Description

Produces a table summarizing the results of mokken analyses.

Usage

```
mokkenTable(data)
```

Arguments

data	data.frame containing the variables (i.e. items) for which the table should be created.

Value

A mokken table.

Examples

```
data(myData)
mokkenTable(myData[,1:4])
#saveTable(mokkenTable(myData[,1:4]), "mokkentable.rtf")
```

---

myData *Simulated data.*

Description

A simulated data frame made with help of the psych package.

Usage

```
data(myData)
```

Format

A data frame with 500 observations on the following 7 variables.

`item1` a numeric vector

`item2` a numeric vector

`item3` a numeric vector

`item4` a numeric vector

`sex` a factor with levels `Female` `Male`

`age_group7` a factor with levels `< 21` `> 60` `21-30` `31-40` `41-50` `51-60`

`employment` a factor with levels `Employed` `Not employed`

Examples

```
data(myData)
str(myData)
summary(myData)
```

norms *Establish norms.*

Description

Internal function. Should not be used.

Usage

```
norms(sumscores, from, to, statistics = "PR")
```

Arguments

sumscores	The sumscore vector for which norms should be created.
from	numeric value. Lowest possible sumscore as a numeric value.
to	numeric value. Highest possible sumscore as a numeric value.
statistics	character vector that toggles which norm statistics are included in the norm table. Currently Percent ranks "PR", z-Statistic "z", Z-Statistic "Z", IQ-Stiatistc "IQ", T-Statistic "T" and Stanine "Stanine" are supported.

Value

Norms.

Examples

```
###
1+1
```

normTable *Norm table.*

Description

Produces a table of norms.

Usage

```
normTable(sumscores, from, to, statistics = "PR", group = NA, as.list = FALSE)
```

Arguments

sumscores	The sumscore vector for which norms should be created.
from	numeric value. Lowest possible sumscore as a numeric value.
to	numeric value. Highest possible sumscore as a numeric value.
statistics	character vector that toggles which norm statistics are included in the norm table. Currently Percent ranks "PR", z-Statistic "z", Z-Statistic "Z", IQ-Stiatistc "IQ", T-Statistic "T" and Stanine "Stanine" are supported.
group	List of subgroups by which the norms should be created.
as.list	logical vector that toggles whether the norm table should rather be returned as a (vertical) list than as a table. This option is useful if you have a lot of subgroups and/or norm statistics and the resulting table would not fit on the page horizontally. Note that currently when you set as.list = TRUE the resulting list cannot be saved by the saveTable function.

Details

The different norm statistics (besides the percent ranks) are created by transforming the z-Statistic. This means that at first the sumscores are z-Transformed and then the other statistics are calculated based on the z-Statistic.

One could also choose to first "normalize" the z-Statistic, i.e. calculate the percent ranks and then assign the z-values to the sumscores according to the percent ranks of the standard normal distribution. This option is—albeit it is sometimes called the scientific standard—debatle because if the population does not follow the normal curve then the normalisation will lead to an artificial spreading or shortening of scores. This leads to the scores having only an ordinal scale quality even when the scores in the population distribution had originally metric scale quality. Thus this option should be used with caution and only on a strong theoretical basis. Therefore this option is currently not implemented.

Value

A dataframe constituting the norm table.

Examples

```
data(myData)
myData.sumscore <- rowSums(myData[,c("item1", "item2", "item3", "item4")])

tab.1 <- normTable(myData.sumscore, from = 0, to = 12, statistics=c("PR"))
tab.1
tab.2 <- normTable(myData.sumscore, from = 0, to = 12, statistics=c("PR", "T", "Stanine"))
tab.2
tab.3 <- normTable(myData.sumscore, from = 0, to = 12, statistics=c("PR"), group=myData$sex)
tab.3
tab.4 <- normTable(myData.sumscore, from = 0, to = 12, statistics=c("PR", "T"), group=myData$employment)
tab.4
tab.5 <- normTable(myData.sumscore, from = 0, to = 12, statistics=c("T"), group=list(myData$sex, myData$employ
tab.5
list.5 <- normTable(myData.sumscore, from = 0, to = 12, statistics=c("PR", "T", "Z", "z"), group=list(myData$s
```

```
list.5
#saveTable(tab.2, "normTable.rtf")
```

---

saveTable *Save a table.*

Description

Saves a table either in rtf-format (recommended) or in HTML-format.

Usage

```
saveTable(x, file, HTML = FALSE)
```

Arguments

x	data.frame as generated by the different xTable functions in this package (for example by the mokkenTable function).
file	string; filename the table should be saved as.
HTML	logical value that toggles whether the file format should be HTML.

Details

Currently the text in the .rtf-Format is set to an 8-point size to accommodate the sometimes large tables.

Value

None

Examples

```
data(myData)
test <- demographicTable(myData$sex, myData$age_group7, percent=FALSE)
test

#saveTable(test, "test.rtf")
```

Index