Package 'dtables'

November 1, 2016

Title Simplifying Descriptive Frequencies and Statistics

Version 0.2.0

Description Towards automation of descriptive frequencies and statistics tables.

Depends R (>= 3.2.2)

License GPL-3

LazyData true

URL https://github.com/gitronald/dtables

BugReports https://github.com/gitronald/dtables/issues

RoxygenNote 5.0.1

Imports psych

NeedsCompilation no

Author Ronald E. Robertson [cre, aut]

Maintainer Ronald E. Robertson <robertson.ron@husky.neu.edu>

Repository CRAN

Date/Publication 2016-11-01 16:30:00

R topics documented:

cor_test *Correlation tests*

Description

A wrapper for cor.test that returns data in a data.frame rather than a cumbersome list

Usage

```
cor_test(var1, var2, method = NULL, round = TRUE)
```

Arguments

var1	a vector to correlate with var2
var2	a vector to correlate with var1
method	the correlation method to use, either pearson, spearman, or kendall
round	logical, whether or not to round the results

Value

Returns a data.frame version of the standard htest output. Use intended for presentation of data because numbers are converted to character and then back to numeric, trimming each values length and costing you precision.

Examples

```
cor_test(sample(1:100, 100), sample(1:100, 100), method = "pearson")
```

create_list *Create a list of named lists*

Description

Use `create_list` to create a list of named lists, each of length `list.lengths`.

Usage

```
create_list(list.names, list.lengths)
```

Arguments

list.names	a string vector containing list names
list.lengths	a numeric value that determines the length of each `list.names` list

Value

Returns a list of named lists, each of length `list.lengths`.

Examples

```
# Generate lists, each of length 4
create_list(c("a", "b", "c"), 4)
```

`dfactor` *Demographic Factor Frequencies Tables*

Description

This function converts columns from a `data.frame` into a frequencies table formatted in standard presentation structure with percent symbols.

Usage

```
dfactor(data1, vars, neat = TRUE, sizesort = TRUE)
```

Arguments

`data1`	a `data.frame`
`vars`	a vector of one or more variable names in `data1`
`neat`	logical, `TRUE` returns a rounded table with percent symbols
`sizesort`	logical, `TRUE` returns table sorted by size

Value

Returns a demographic frequencies table in `data.frame` format.

Examples

```
# Single frequency table
dfactor(iris2, "Species")

# Multiple frequency tables
dfactor(iris2, c("Color", "Species"))
```

dft *Create a data.frame table (dft)*

Description

Create a descriptive frequencies table with descriptive statistics by group.

Usage

```
dft(data1, prop = TRUE, perc = TRUE, by = NULL, neat = TRUE,
  digits = 2)
```

Arguments

data1	a vector or data.frame column
prop	logical, if TRUE returns an additional proportion column
perc	logical, if TRUE returns an additional percentage column
by	numeric variable to return descriptive statistics for
neat	logical, if TRUE returns a tailored dataset
digits	integer, number of digits to round to

Value

a data.frame table with optional proportion, percentage, and descriptive statistics columns

Examples

```
dft(iris2$Species)
dft(iris2$Species, by = iris2$Sepal.Length)
```

dnum *Create standard descriptive statistics*

Description

Wrapper for psych::describe that adds the name of the dataset and the variable being examined to its output. Used in dtable for multiple variables.

Usage

```
dnum(data1, neat = TRUE, sizesort = FALSE)
```

Arguments

data1	a data.frame column or columns, or a list
neat	logical, returns rounded values if TRUE
sizesort	logical, returns sorted data by mean if TRUE

Value

Returns a data.frame with common descriptive statistics for a numeric variable, as defined in psych::describe, concatenated with the name of the dataset and the name of the variable.

See Also

See dtable

See describe for more details on the descriptive statistics returned

Examples

```
# Single variable
dnum(iris2$Sepal.Length)
dnum(iris2["Sepal.Length"])
dnum(iris2[, "Sepal.Length"])

# Multiple variables
dnum(iris2[, c("Sepal.Length", "Sepal.Width")])

# Will not save you from yourself (will create numeric data for factors):
dnum(iris2$Species)
```

---

dnumeric *Create standard descriptive statistics*

Description

Wrapper for psych::describe that adds the name of the dataset and the variable being examined to its output. Used in dtable for multiple variables.

Usage

```
dnumeric(data, vnames, neat = TRUE, sizesort = FALSE)
```

Arguments

data	a data.frame
vnames	a single variable name to examine with psych::describe
neat	logical, returns rounded values if TRUE
sizesort	logical, returns sorted data by mean if TRUE

Value

Returns a data.frame with common descriptive statistics for a numeric variable, as defined in psych::describe, concatenated with the name of the dataset and the name of the variable.

See Also

See dtable

See describe for more details on the descriptive statistics returned

Examples

```
# Single variable
dnumeric(iris2, "Sepal.Length")

# Use \link{\code{dtable}} for multiple variables
```

---

dtable *Generate descriptive frequencies and statistics tables*

Description

Simplifies the process and reduces the amount of code involved in generating descriptive frequencies and statistics tables by taking your entire dataset as input and generating the tables it predicts you would need given various variable aspects such as class.

Usage

```
dtable(data1, vars = NULL, frequencies = NULL, statistics = NULL,
  neat = TRUE, as.list = FALSE, sizesort = FALSE)
```

Arguments

data1	a data.frame or matrix
vars	select which columns of data1 to analyze
frequencies	select which columns to create frequencies tables for
statistics	select which columns to create statistics tables for
neat	logical, if TRUE returns rounded and formatted tables
as.list	logical, if TRUE it returns frequencies and statistics tables in a list format, split by variables
sizesort	logical, if TRUE returns data sorted by frequency and mean

Value

Returns descriptive frequencies and statistics tables for all variables in data1 by default. Predicts whether to create a frequencies table, statistics table, or both for each varible based on information gathered using dvariable.

Examples

```
# Analyze all data
dtable(iris2)

#Analyze two or more variables
dtable(iris2, vars = c("Color", "Sold", "LikelyToBuy"))

# Analyze a single variable
dtable(iris2, vars = "Color")

# Return raw output
dtable(iris2, neat = FALSE)

# Return list output
dtable(iris2, as.list = TRUE)

# Frequencies sorted by size
dtable(iris2, sizesort = TRUE)
```

---

dvariable *Data.frame of Variables and Classes*

Description

Helper function for `dtable`. Returns information about each variable's class, mode, type, and number of response levels in a `data.frame`, or `list` if `as.list = TRUE`.

Usage

```
dvariable(data1, vars = NULL, as.list = FALSE)
```

Arguments

`data1`	a `data.frame`
`vars`	specify one or more variable names with a character vector
`as.list`	logical, `TRUE` returns a list split by class

Value

Returns a data.frame or list with the variable names and their respective classes. Returns a data.frame by default.

See Also

`class` to examine method for extracting class.

`mode` to examine method for extracting mode.

`typeof` to examine method for extracting type.

Examples

```
# Describe all variables in iris2
dvariable(iris2)

# Describe one or more specific variables in iris2
dvariable(iris2, vars = "Species")

# Return variable and class data in list
dvariable(iris2, as.list = TRUE)
```

`factor_length` *Return factor level lengths for all variables in a data.frame*

Description

Return factor level lengths for all variables in a data.frame

Usage

```
factor_length(data1)
```

Arguments

`data1`	a `data.frame`

Value

a `data.frame` with variable names and factor level lengths

Examples

```
factor_length(iris2)
```

`iris2` *An Expansion of the Iris Dataset*

Description

An expansion of the well-known `iris` dataset. Extra variables from a range of classes, modes, types, and factor level lengths, spewed from the imagination of a formal consortium of wild monkeys, have been added to complicate the `iris` dataset and provide a stomping grounds to demonstrate the outrageous capabilities of the `dtables` package, and test new ones.

Usage

```
iris2
```

Format

An object of class `data.frame` with 150 rows and 12 columns.

Details

Sepal.Length Original iris variable

Sepal.Width Original iris variable

Petal.Length Original iris variable

Petal.Width Original iris variable

Species Original iris variable

Color A factor variable of class character; the color of the flower

Attractiveness A likert scale variable of class integer; the attractiveness of the flower

LikelyToBuy A likert scale variable of class integer with negative values; the likelihood of buying the flower

Sold A dichotomous variable of class logical; whether or not the flower was bought

Review A likert scale variable of class numeric; the customer's review of the flower

Approval A missing values variable of class logical; a spanner in the works

Date A date variable of class Date; the date the flower was bought

---

`write_object` *Write an object to a file*

Description

A wrapper for `write.table` that saves an object by name to a tab delimited (.tsv) file with options to specify directory and include the current date in the file name.

Usage

```
write_object(object, dir = NULL, date = TRUE, overwrite = FALSE,
  row.names = FALSE, col.names = TRUE)
```

Arguments

`object`	a data.frame, matrix, or vector to write out
`dir`	a character string specifying the directory (folder) in which save the object, defaults to current working directory.
`date`	logical, if `TRUE` appends the object name with todays date
`overwrite`	logical, if `TRUE` overwrites existing file of same name
`row.names`	logical, if `TRUE` adds rownames to output file
`col.names`	logical, if `TRUE` adds column names to output file

Value

Returns nothing. Saves "object_date.tsv" or "object.tsv" to current directory.

Examples

```
write_object(iris2)
```

Index