

Package ‘furniture’

October 21, 2016

Type Package

Title Furniture for Health, Behavioral, and Social Scientists

Version 1.2.0

Date 2016-10-11

Author Tyson Barrett [aut, cre],
Emily Brignone [aut]

Maintainer Tyson Barrett <t.barrett@aggiemail.usu.edu>

Description Package contains two main functions (i.e., two pieces of furniture):
table1() which produces a well-formatted table of descriptives common as Table 1
in research articles; and washer() which is helpful in cleaning up your data.
Should be useful for quantitative research in applied fields (e.g., health,
behavioral, and social sciences). It also contains a simple operator to
help with data exploration.

Depends R (>= 2.10)

Imports knitr

Suggests dplyr, rmarkdown, testthat, MASS

LazyData true

VignetteBuilder knitr

Encoding UTF-8

License GPL-2

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-10-21 10:32:42

R topics documented:

| | |
|---------------------|---|
| furniture | 2 |
| nhanes10 | 3 |
| table1 | 4 |

| | |
|-------------------|---|
| table1_ | 5 |
| washer | 6 |
| %xt% | 7 |

| | |
|--------------|----------|
| Index | 8 |
|--------------|----------|

| | |
|-----------|------------------|
| furniture | <i>furniture</i> |
|-----------|------------------|

Description

The furniture package offers simple functions (i.e. pieces of furniture) and an operator that are aimed at helping applied researchers explore and communicate their data as well as clean their data in a tidy way. The package follows similar semantics to the "tidyverse" packages. It contains two main tools (along with an operator):

- `table1` provides a well-formatted descriptive table often seen as table 1 in academic journals,
- `washer` provides a simple way to clean up data where there are placeholder values, and
- `%xt%` is an operator that takes two factor variables and creates a cross tabulation and tests for significance via a chi-square test.

Details

Table 1 is the main function in furniture. It is useful in both data exploration and data communication. With minimal cleaning, the outputted table can be put into an academic, peer reviewed journal manuscript. As such, it is very useful in exploring your data when you have a stratifying variable. For example, if you are exploring whether the means of several demographic and behavioral characteristics are related to a health condition, the health condition (i.e. "yes" or "no"; "low", "mid", or "high"; or a list of conditions) as the stratifying variable. With little code, you can test for associations and check means or counts by the stratifying variable. See the vignette for more information.

Examples

```
## Not run:

library(furniture)

## Table 1
data %>%
  table1(var1, var2, var3,
         splitby = ~groupvar,
         test = TRUE)

## Washer
x = washer(x, 7, 8, 9)
x = washer(x, is.na, value=0)

## Crosstabs Operator
```

```
f1 %xt% f2
```

```
## End(Not run)
```

nhanes10

NHANES 2009-2010

Description

A dataset containing information on health, healthcare, and demographics of adolescents aged 12 - 19 in the United States from 2009 to 2010. This is a cleaned dataset which is only a subset of the 2009-2010 data release of the National Health and Nutrition Examination Survey (NHANES).

Usage

```
nhanes10
```

Format

A data frame with 4727 rows and 10 variables:

seqn individual ID

year year of data release (2006, 2008, 2010)

gender gender of the individual (factor with "male" or "female")

raceth race/ethnicity of the individual (factor with "MexAm", "OtherHisp", "White", "Black", "Other")

age age of individual (numeric)

income income as a ratio of income-to-poverty line

fmsize the size of the individual's family (up to 6)

limits whether the individual has physical limitations (factor 1 = Limitations, 0 = No Limitations)

medcond whether the individual has at least one medical condition [i.e. asthma, psoriasis, overweight, celiac, and/or trouble seeing] (factor 1 = Yes, 0 = No)

hinsur whether the individual has health insurance (factor 1 = insured, 0 = not insured)

Source

http://www.cdc.gov/nchs/nhanes/search/nhanes09_10.aspx

table1

*Table 1 for Simple and Stratified Descriptive Statistics***Description**

Produces a descriptive table, stratified by an optional categorical variable, providing means/frequencies and standard deviations/percentages. It is well-formatted for easy transition to academic article or report. Can be used within the piping framework [see `library(magrittr)`].

Usage

```
table1(.data, ..., splitby = NULL, splitby_labels = NULL, test = FALSE,
       test_type = "default", piping = FALSE, rounding = 3, var_names = NULL,
       format_output = "pvalues", output_type = "text", NAkeep = FALSE,
       m_label = "Missing", booktabs = TRUE, caption = NULL, align = NULL)
```

Arguments

| | |
|-----------------------------|---|
| <code>.data</code> | the data.frame that is to be summarized |
| <code>...</code> | variables in the data set that are to be summarized; unquoted names separated by commas (e.g. age, gender, race) or indices. If indices, it needs to be a single vector (e.g. <code>c(1:5, 8, 9:20)</code>) instead of <code>1:5, 8, 9:20</code>). As it is currently, it CANNOT handle both indices and unquoted names simultaneously. |
| <code>splitby</code> | the categorical variable to stratify by in formula form (e.g., <code>splitby = ~gender</code>); not too surprisingly, it requires that the number of levels be > 0 |
| <code>splitby_labels</code> | allows for custom labels of the splitby levels; must match the number of levels of the splitby variable |
| <code>test</code> | logical; if set to TRUE then the appropriate bivariate tests of significance are performed if splitby has more than 1 level |
| <code>test_type</code> | has two options: "default" performs the default tests of significance only; "or" also give unadjusted odds ratios as well based on logistic regression (only use if splitby has 2 levels) |
| <code>piping</code> | if TRUE then the table is printed and the original data is passed on. It is very useful in piping situations where one wants the table but wants it to be part of a larger pipe. |
| <code>rounding</code> | the number of digits after the decimal; default is 3 |
| <code>var_names</code> | custom variable names to be printed in the table |
| <code>format_output</code> | has three options: 1) "full" provides the table with the type of test, test statistic, and the p-value for each variable; 2) "pvalues" provides the table with the p-values; and 3) "stars" provides the table with stars indicating significance |
| <code>output_type</code> | default is "text"; the other options are all format options in the <code>kable()</code> function in <code>knitr</code> (e.g., latex, html, markdown, pandoc) |
| <code>NAkeep</code> | when sset to TRUE it also shows how many missing values are in the data for each categorical variable being summarized |

| | |
|----------|--|
| m_label | when NAkeep = TRUE this provides a label for the missing values in the table |
| booktabs | when output_type != "text"; option is passed to knitr::kable |
| caption | when output_type != "text"; option is passed to knitr::kable |
| align | when output_type != "text"; option is passed to knitr::kable |

Value

A table with the number of observations, means/frequencies and standard deviations/percentages is returned. The object is a table1 class object with a print method. Can be printed in LaTeX form.

Examples

```
## Ficticious Data ##
library(furniture)
library(dplyr)

x <- runif(1000)
y <- rnorm(1000)
z <- factor(sample(c(0,1), 1000, replace=TRUE))
a <- factor(sample(c(1,2), 1000, replace=TRUE))
df <- data.frame(x, y, z, a)

## Simple
table1(df, x, y, z, a)

## Stratified
## both below are the same
table1(df, x, y, z,
       splitby = ~ a)
table1(df, x, y, z,
       splitby = "a")

## With Piping
df %>%
  table1(x, y, z,
        splitby = ~a,
        piping = TRUE) %>%
  summarise(count = n())

## Adjust variables within function
table1(df, ifelse(x > 0, 1, 0), z,
       var_names = c("Dich X", "Z"))
```

Description

For internal use in table1().

Usage

```
table1_(..., d_, .cl = NULL)
```

Arguments

... the variables
d_ the data.frame
.cl the original functon call

Value

A data.frame

washer

Wash Your Data

Description

Washes the data by replacing values with either NA's or other values set by the user. Useful for replacing values such as 777's or 999's that represent missing values in survey research. Can also perform many useful functions on factors (e.g., removing a level, replacing a level, etc.)

Usage

```
washer(x, ..., value = NA)
```

Arguments

x the variable to have values adjusted
... the values in the variable that are to be replaced by either NA's or the value set by the user. Can be a function (or multiple functions) to specify values to change (e.g., is.nan(), is.na()).
value (optional) if specified, the values in ... will be replaced by this value (must be a single value)

Value

A table with the number of observations, means/frequencies and standard deviations/percentages is returned. The object is a table1 class object with a print method. Can be printed in LaTeX form.

Examples

```
x = c(1:20, NA, NaN)
washer(x, 9, 10)
washer(x, 9, 10, value=0)
washer(x, 1:10)
washer(x, is.na, is.nan, value=0)
washer(x, is.na, is.nan, 1:3, value=0)
```

%xt%

Simple Crosstabs Operator

Description

This operator takes two variables and computes a simple cross tab.

Usage

```
lhs %xt% rhs
```

Arguments

| | |
|-----|---|
| lhs | the left hand side of the operator, a vector |
| rhs | the right hand side of the operator, a vector |

Examples

```
b = c(1,0,0,1,1,0,1,1,1,0)
x = c(1,2,3,2,3,3,1,0,0,0)
y = rnorm(10)
z = c("Yes", "No", "Yes", "No", "No", "Yes", "No", "No", "Yes", "No")
data = data.frame(b, x, y, z)

factor(data$x) %xt% factor(data$b)
```

Index

*Topic **datasets**

nhanes10, [3](#)

%xt%, [7](#)

furniture, [2](#)

furniture-package (furniture), [2](#)

nhanes10, [3](#)

table1, [4](#)

table1_, [5](#)

washer, [6](#)