Package ‘remix’

February 20, 2015

Maintainer David Hajage <dhajage@gmail.com>
License GPL (>= 2)
Title Remix your data
Type Package
LazyLoad yes
Author David Hajage
Description remix provides remix, a quick and easy function for describing datasets. It can be view as a mix of cast (in package reshape) and summary.formula (in package Hmisc).
Version 2.1
Date 2010-03-26
Depends R (>= 2.13.0), plyr, survival
Imports Hmisc, ascii
Collate 'correlation.r' 'element.r' 'freq.r' 'funs.r' 'na.r' 'parse.r'
  'rbind.r' 'remix.r' 'summarize.by.r' 'summarize.r' 'survival.r'
  'table.r' 'tabular.r' 'test.r'
Repository CRAN
Date/Publication 2011-10-04 07:25:26
NeedsCompilation no

R topics documented:

  ascii.freq .......................... 2
  ascii.remix .......................... 3
  ascii.summarize ..................... 4
  ascii.summarize.by ................ 4
  ascii.survival ........................ 5
  ascii.tabular ........................ 6
demix .................................. 7
Ascii method for freq object (internal).

Usage

```r
## S3 method for class 'freq'
ascii(x, format = "nice", digits = 3,
      include.rownames = FALSE, include.colnames = TRUE, header = TRUE, lgroup = attr(x, "lgroup"), n.lgroup = attr(x, "n.lgroup"), ...)
```

Arguments

- `x`: a freq object
- `format`: see ?ascii in ascii package
- `digits`: see ?ascii in ascii package
- `include.rownames`: see ?ascii in ascii package
- `rownames`: see ?ascii in ascii package
- `include.colnames`: see ?ascii in ascii package
- `colnames`: see ?ascii in ascii package
- `header`: see ?ascii in ascii package
Ascii method for remix object.

Usage

```r
## S3 method for class 'remix'
ascii(x, caption.level = c("s", "e", "m"), format = "nice", digits = 2, ...)
```

Arguments

- `x`: a remix object
- `caption.level`: see ?ascii in ascii package
- `format`: see ?ascii in ascii package
- `digits`: see ?ascii in ascii package
- `...`: other arguments passed to `ascii` (all except `caption` which has no effect)

Author(s)

David Hajage
ascii.summarize

Ascii for summarize object.

Description

Ascii method for summarize object (internal).

Usage

## S3 method for class 'summarize'
ascii(x, format = "nice", digits =
5, include.rownames = TRUE, include.colnames = TRUE,
header = TRUE, ...)

Arguments

x

a summarize object

format

see ?ascii in ascii package
digits

see ?ascii in ascii package
include.rownames

see ?ascii in ascii package
include.colnames

see ?ascii in ascii package
header

see ?ascii in ascii package
...

other arguments passed to ascii

Author(s)

David Hajage

ascii.summarize.by

Ascii for summarize.by object.

Description

Ascii method for summarize.by object (internal).

Usage

## S3 method for class 'summarize.by'
ascii(x, format = "nice", digits =
5, include.rownames = FALSE, include.colnames = TRUE,
header = TRUE, lgroup = attr(x, "lgroup"), n.lgroup =
attr(x, "n.lgroup"), rgroup = attr(x, "rgroup"), n.rgroup
= attr(x, "n.rgroup"), rstyle = "d", ...)
Arguments

- `x`: a `summarize.by` object
- `format`: see `?ascii` in `ascii` package
- `digits`: see `?ascii` in `ascii` package
- `include.rownames`: see `?ascii` in `ascii` package
- `include.colnames`: see `?ascii` in `ascii` package
- `header`: see `?ascii` in `ascii` package
- `lgroup`: see `?ascii` in `ascii` package
- `n.lgroup`: see `?ascii` in `ascii` package
- `rgroup`: see `?ascii` in `ascii` package
- `n.rgroup`: see `?ascii` in `ascii` package
- `rstyle`: see `?ascii` in `ascii` package
- `...`: other arguments passed to `ascii`

Author(s)

David Hajage

ascii.survival  Ascii for survival object.

Description

Ascii method for survival object (internal).

Usage

```r
## S3 method for class 'survival'
asici(x, format = "nice", digits = 5,
      include.rownames = FALSE, include.colnames = TRUE, header = TRUE,
      rstyle = "d", caption = NULL, caption.level = NULL, ...)
```

Arguments

- `x`: a survival object
- `format`: see `?ascii` in `ascii` package
- `digits`: see `?ascii` in `ascii` package
- `include.rownames`: see `?ascii` in `ascii` package
include.colnames
    see ?ascii in ascii package
header
    see ?ascii in ascii package
rstyle
    see ?ascii in ascii package
caption
    see ?ascii in ascii package
caption.level
    see ?ascii in ascii package
...
other arguments passed to ascii

Author(s)
    David Hajage

---

Ascii for tabular object.

Description
Ascii method for tabular object (internal).

Usage
## S3 method for class 'tabular'
ascii(x, format = "nice", digits = 5,
    include.rownames = FALSE, include.colnames = TRUE, header
    = TRUE, rstyle = "d", caption = NULL, caption.level =
    NULL, ...)

Arguments
x            a tabular object
format       see ?ascii in ascii package
digits       see ?ascii in ascii package
include.rownames
    see ?ascii in ascii package
include.colnames
    see ?ascii in ascii package
header       see ?ascii in ascii package
rstyle       see ?ascii in ascii package
caption      see ?ascii in ascii package
caption.level
    see ?ascii in ascii package
...
other arguments passed to ascii

Author(s)
    David Hajage
Description

Transform a remix object into a (list of) data.frame(s).

Usage

demix(x)

Arguments

x  a remix object

Value

A list of data.frame.

Author(s)

David Hajage

See Also

remix

Examples

x <- remix(... ~ ., esoph, cum = TRUE)
demix(x)

Description

Display a test result

Usage

display.test(test, digits = 4, method = TRUE)
**Arguments**

- `test`  
- `digits` number of digits  
- `method` display method

**Value**

a character string

**Author(s)**

David Hajage

---

`is.freq`  
*Test if x is an freq object*

**Description**

Test if x is an freq object

**Usage**

`is.freq(x)`

**Arguments**

- `x` a freq object

---

`n`  
*Return the number of non NA observations*

**Description**

Return the number of non NA observations

**Usage**

`n(x, na.rm = FALSE)`

**Arguments**

- `x` a vector  
- `na.rm` not used

**Author(s)**

David Hajage
**na**

*Return the number of NA observations*

**Description**

Return the number of NA observations

**Usage**

```r
na(x, na.rm = FALSE)
```

**Arguments**

- `x`: a vector
- `na.rm`: not used

**Author(s)**

David Hajage

---

**print.freq**

*Print freq object.*

**Description**

Print freq object (internal).

**Usage**

```r
## S3 method for class 'freq'
print(x, type = "rest", lstyle = ",", ...)  
```

**Arguments**

- `x`: a freq object
- `type`: type of output (see ?ascii in ascii package)
- `lstyle`: see ?ascii in ascii package
- `...`: other arguments passed to ascii

**Author(s)**

David Hajage
print.remix 

Print a remix object

Description

Print remix object using ascii package

Usage

## S3 method for class `remix`
print(x, type = "rest", caption.level = 1:3, lstyle = "", tstyle = "", ...)

Arguments

x a remix object
type type of output. See \?ascii in ascii package
caption.level see \?ascii in ascii package
lstyle see \?ascii in ascii package
tstyle see \?ascii in ascii package
... other arguments passed to ascii (all except caption which has no effect)

Author(s)

David Hajage

print.tabular 

Print tabular object.

Description

Print tabular object (internal).

Usage

## S3 method for class `tabular`
print(x, type = "rest", lstyle = "", tstyle = "", ...)

Arguments

x a tabular object
type type of output (see \?ascii in ascii package)
lstyle see \?ascii in ascii package
tstyle see \?ascii in ascii package
... other arguments passed to ascii
Author(s)

David Hajage

Remix and describe.

Description

A quick and easy function for describing datasets.

Usage

remix(formula = cbind(...) ~ ., data = NULL, funs =
c(mean, sd, quantile, n, na), ..., cum = FALSE, margin =
0:2, addmargins = FALSE, useNA = c("no", "ifany",
"always"), propNA = TRUE, revert = FALSE, method =
c("pearson", "kendall", "spearman"), times = NULL,
followup = FALSE, test = FALSE, test.summarize =
test.summarize.auto, test_survival =
test_survival.logrank, test.tabular = test.tabular.auto,
show.test = display.test, plim = 4, show.method = TRUE,
label = FALSE)

Arguments

formula a formula (see Details).
data a data.frame.
funs functions for describing numeric variable. Can be c(fun1, fun2, fun3) or
c("fun1", "fun2", "fun3") or a list.
... further arguments (all passed to funs), for example na.rm = TRUE
cum should cumulated frequencies be reported?
margin index, or vector of indices to generate proportion in frequency tables (0: cell, 1: row, 2: col).
addmargins whether to add margins
useNA whether to include NA as a level (factor)
propNA whether to include NA in proportion calculation
revert whether to regroup factors or numeric variables when crossing factor with numeric variables
method a character string indicating which correlation coefficient is to be used. One of
"pearson", "kendall", or "spearman", can be abbreviated.
times vector of times (see ?summary.survival un package survival)
followup whether to display follow-up time
test whether to perform tests
test.summarize a function of two arguments (continuous variable and grouping variable) used to compare continuous variable, that return a list of two components: p.value and method (the test name). See test.summarize.auto, test.summarize.kruskal, test.summarize.oneway.equalvar, or test.summarize.unequalvar for example of such functions. Users can provide their own function.

test.survival a function of one argument (a formula) used to compare survival estimations, that returns the same components as created by test.summarize. See test.survival.logrank. Users can provide their own function.

test.tabular a function of three arguments (two categorical variables and a logical na) used to test association between two factors, that returns the same components as created by test.summarize. See test.tabular.auto and test.tabular.fisher. Users can provide their own function.

show.test a function used to display the test result. See display.test.

plim number of digits for the p value

show.method should display the test name?

label whether to display labels of variables (using label in package Hmisc)

Value

A remix object, basically a list with descriptive tables. It uses ascii package for printing output, and can be use with ascii function.

Note

The formula has the following format: \( x_1 + x_2 + \ldots y_1 + y_2 + \ldots \)

There are a couple of special variables: \ldots represents all other variables not used in the formula and . represents no variable, so you can do formula = var1 ~ .

If var1 is numeric, var1 ~ . produce a summary table using funs. If var1 is a factor, var1 ~ . produce a frequency table. If var1 is of class Surv, var1 ~ . produce a table with the estimates of survival at times. If var1 is numeric and var2 is numeric, var1 ~ var2 gives correlation. if var1 is numeric and var2 is a factor, var1 ~ var2 produce a summary table using funs according to the levels of var2. If var1 is a factor and var2 is a factor, var1 ~ var2 produce a contingency table. If var1 is of class Surv and var2 is a factor, var1 ~ var2 produce a table with the estimates of survival for each level of var2.

You can group several variables of the same type (numeric or factor) together with cbind(var1, var2, var3), they will be grouped in the same table. cbind(...) works (ie regroups all variables of the same type).

Author(s)

David Hajage, inspired by the design and the code of summary.formula (Hmisc package, FE Harrell) and cast (reshape package, H Wickham).

See Also

cast (reshape) and summary.formula (Hmisc).
Examples

```r
parwidth <- getOption("width")
options(width = 100)

library(remix)
remix(data = iris)
remix(cbind(...)(.., iris[, sapply(iris, is.numeric)], funs = c(median, mad, min, max))
remix(cbind(Sepal.Length, I(Sepal.Width^2)) ~ Species, iris, funs = quantile, probs = c(1/3, 2/3))
remix(Sepal.Length + Sepal.Width ~ Petal.Length + Petal.Width, iris)
remix(cbind(Sepal.Length, Sepal.Width) ~ cbind(Petal.Length, Petal.Width), iris)
remix(... ~ .., esoph, cum = TRUE)
remix(alcgp ~ tobgp, esoph, cum = TRUE)
remix(Surv(time, status) ~ x, data = aml, times = seq(0, 120, 12))

options(width = parwidth)
```

---

**survival.data.frame**

*Compute survival (data.frame input)*

**Description**

Compute survival (data.frame input)

**Usage**

```r
survival.data.frame(df, times = NULL, followup = FALSE, label = FALSE)
```

**Arguments**

- `df`
- `times`
- `followup`
- `label`

**Author(s)**

David Hajage
survival.data.frame.by

Compute survival according to a factor (data.frame input)

Description
Compute survival according to a factor (data.frame input)

Usage
survival.data.frame.by(df, by, times = NULL, followup = FALSE, test = FALSE, test.survival = test.survival.logrank, show.test = display.test, plim = 4, show.method = TRUE, label = FALSE)

Arguments
df
df
by
by
times	times
followup	followup
test
test
test.survival
test.survival
show.test
test.survival
plim
plim
show.method
show.method
label
label

Author(s)
David Hajage

test.summarize.auto
test for mean comparison

Description
Compute a oneway.test (with equal or unequal variance) or a kruskal.test as appropriate.

Usage
test.summarize.auto(x, g)
Arguments

    x         vector
    g         another vector

Value

    a list with two components: p.value and method

Author(s)

    David Hajage

---

test.summarize.kruskal

    test for mean comparison

---

Description

    Compute a kruskal.test.

Usage

    test.summarize.kruskal(x, g)

Arguments

    x         vector
    g         another vector

Value

    a list with two components: p.value and method

Author(s)

    David Hajage
test.summarize.oneway.equalvar

Description
Compute a one-way test with var.equal = TRUE

Usage
test.summarize.oneway.equalvar(x, g)

Arguments
x vector
g another vector

Value
a list with two components: p.value and method

Author(s)
David Hajage

test.summarize.oneway.unequalvar

Description
Compute a one-way test with var.equal = FALSE

Usage
test.summarize.oneway.unequalvar(x, g)

Arguments
x vector
g another vector

Value
a list with two components: p.value and method
**test.survival.logrank**

**Author(s)**
David Hajage

**Description**
Compute a logrank test

**Usage**
```
test.survival.logrank(formula)
```

**Arguments**
- `formula`: a formula

**Value**
a list with two components: p.value and method

**Author(s)**
David Hajage

---

**test.tabular.auto**

**Description**
Compute a chisq.test, a chisq.test with correction of continuity or a fisher test as appropriate

**Usage**
```
test.tabular.auto(x, y, na = FALSE)
```

**Arguments**
- `x`: vector
- `y`: another vector
- `na`: use NA?

**Value**
a list with two components: p.value and method
Author(s)

David Hajage

test.tabular.fisher test for contingency table

Description

Compute a fisher test

Usage

test.tabular.fisher(x, y, na = FALSE)

Arguments

x vector
y another vector
na use NA ?

Value

a list with two components: p.value and method

Author(s)

David Hajage
Index

*Topic **univar**
  ascii.freq, 2
  ascii.remix, 3
  ascii.summarize, 4
  ascii.summarize.by, 4
  ascii.survival, 5
  ascii.tabular, 6
  n, 8
  na, 9
  print.freq, 9
  print.remix, 10
  print.tabular, 10
  remix, 11

demix, 7
display.test, 7

**is.freq**, 8

n, 8
na, 9

print.freq, 9
print.remix, 10
print.tabular, 10
remix, 11

survival.data.frame, 13
survival.data.frame.by, 14

test.summarize.auto, 14
test.summarize.kruskal, 15
test.summarize.oneway.equalvar, 16

test.summarize.oneway.unequalvar, 16
test.survival.logrank, 17
test.tabular.auto, 17
test.tabular.fisher, 18