Package 'hipster'

June 29, 2020

Title Henri's Idiosyncratic Package for Somewhat Tedious Everyday Routines

Version 0.0.0.9010

Author Henri Kauhanen <henri@henr.in>

Maintainer Henri Kauhanen <henri@henr.in>

Description Routines I wish were included in R by default.

License GPL-3 | file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

R topics documented:

bootstrap *Bootstrap a Data Frame*

Description

Form a bootstrap sample from a data frame `data` by sampling `nrow(data)` rows of the data frame with replacement.

Usage

```
bootstrap(data)
```

Arguments

`data`	Data frame

Value

Data frame

cat_csv *Concatenate CSVs*

Description

Row-bind several CSV files into one dataframe.

Usage

```
cat_csv(folder, pattern = ".csv$")
```

Arguments

`folder`	Folder in which the CSV files live
`pattern`	Pattern which included files are to match

Value

A data frame

cat_csv.gz *Concatenate CSVs inside a .tar.gz*

Description

Row-bind several CSV files that live inside a tar.gz archive. It is assumed that these CSV files are contained within a directory called `foo` where `foo.tar.gz` is the filename of the archive; if this is not the case, unexpected behaviour will occur.

Usage

```
cat_csv.gz(gzfile, pattern, csv_folder = ".")
```

Arguments

`gzfile`	Archive
`pattern`	Pattern which the CSV files are to match
`csv_folder`	Folder within the archive in which the CSVs are found

Details

The `csv_folder` argument can be used to specify an optional directory within the `foo` directory that contains the CSV files.

Value

Data frame

closest_in_vector *Get the Element(s) of a Vector Closest to a Reference Value*

Description

Return the *k* elements of a vector closest (in terms of Euclidean distance) to a reference value, as well as those elements' indices in the vector.

Usage

```
closest_in_vector(x, vec, k = 1)
```

Arguments

`x`	Reference value
`vec`	Vector
`k`	How many closest neighbours to return

Value

A data frame with the following columns, sorted by increasing distance:

`index` Index of element (i.e. `value` is the same as `vec[index]`)

`value` Element

`dist` Euclidean distance of element to reference value

---

df2tex *Data Frame to TeX Tabular*

Description

Prints a data frame as a (La)TeX tabular.

Usage

```
df2tex(df, file, col.names = TRUE, row.names = TRUE, header = "",
  before = NULL, after = NULL)
```

Arguments

`df`	Data frame
`file`	File to write the table into
`col.names`	Should column names be printed?
`row.names`	Should row names be printed?
`header`	Optional tabular header. If not specified, a default header of the form `\tabular{r...r}` is used with as many columns as there are columns in the data frame.
`before`	Optional vector of lines to write after header and (possible) column names but before data. Mostly useful to add custom column names, or an `\hline`.
`after`	Optional vector of lines to write before `\end{tabular}`

---

factor2character *Factor to Character*

Description

Turns a factor into a plain character vector.

Usage

```
factor2character(x)
```

Arguments

x	Factor to be transformed

Value

Character vector, i.e. the factor without the levels.

---

factor2numeric *Factor to Numeric*

Description

Turns a factor into a numeric vector.

Usage

```
factor2numeric(x)
```

Arguments

x	Factor to be transformed

Value

Numeric vector

---

knockout_lm *Knock Out Outliers in a Linear Regression*

Description

Knock out outliers in a linear regression by recursively pruning those data points that contribute the most to the regression error, operationalized as the residual sum of squares at each knockout iteration.

Usage

```
knockout_lm(data, formula, id.var)
```

Arguments

data	Data frame
formula	Regression formula
id.var	Name of identifier variable (column)

Value

Data frame with the following columns:

`iteration` Knockout iteration

`knockee` Data pointed knocked out at this iteration

`RSS_reduction` Reduction in residual sum of squares resulting from the knockout

---

`logistic` *Logistic Function*

Description

Generalized logistic function.

Usage

```
logistic(t, s = 1, k = 0, U = 1, L = 0)
```

Arguments

`t`	Variable (usually time, hence `t`)
`s`	Slope
`k`	Intercept
`U`	Upper asymptote
`L`	Lower asymptote

Value

Value of the function at `t`, given the parameters.

`logseq` *Logarithmic Sequences*

Description

Generates a logarithmically spaced sequence.

Usage

```
logseq(from, to, length.out)
```

Arguments

from	Starting point of sequence
to	Endpoint of sequence
length.out	Desired length of sequence

Value

A numeric vector

---

prettyround *Pretty Print Rounded Figures*

Description

Pretty print a rounded figure so that all trailing digits are shown, including zeroes.

Usage

```
prettyround(x, digits)
```

Arguments

x	Number
digits	Number of digits

Value

Pretty-printed number. Note that this is of class character, not numeric.

---

refresh *Package Refresh*

Description

"Refresh" a package by unloading its namespace and immediately re-requiring that namespace. This is useful if an updated version of the package has been installed by another process on the machine in the meantime, to avoid having to restart the R session.

Usage

```
refresh(pkg)
```

Arguments

pkg	Package name (character vector)

`seed_seed` *Seed Seed*

Description

First, seeds the random number generator with `seed`. Then generates a sequence of `n` random integers between 1 and `.Machine$integer.max`. Finally, seeds the random number generator with the `n`th (i.e. last) number in this sequence. The point of this function is that it may be used to seed the RNG at the start of a job that is run on a cluster, so that parallel jobs are using – if not statistically independent random number streams – at least not the exact same stream.

Usage

```
seed_seed(n, seed, ...)
```

Arguments

`n`	Number (e.g. id of parallel job)
`seed`	Initial seed; this should be set so that results are replicable
`...`	Further parameters passed to `set.seed`

Value

The final seed, i.e. the `n`th number in the sequence

Index