Package 'Conigrave'

November 6, 2018

Type Package

Title Flexible Tools for Multiple Imputation

Version 0.4.2

Date 2018-11-06

Author James Conigrave

Maintainer James Conigrave <james.conigrave@gmail.com>

Description Provides a set of tools that can be used across 'data.frame' and 'imputationList' objects.

License GPL-3

LazyData TRUE

Imports ggplot2 (>= 2.1.0), magrittr (>= 1.5), stringdist(>= 0.9.4.7), dplyr(>= 0.7.3), stringr(>= 1.3.1), ppcor(>= 1.1), mitools(>= 2.3), miceadds(>= 1.8-0)

RoxygenNote 6.1.0

NeedsCompilation no

Repository CRAN

Date/Publication 2018-11-06 05:30:02 UTC

R topics documented:

autoModel *autoModel*

Description

autoModel uses a genetic algorithm to optimize regression models for increased explained variance. Overly complicated models are penalized for adding additional regression terms in order to combat over-fitting.

Usage

```
autoModel(data, outcome, genepool = NULL, extinction = 30,
  children = 20, penalty = 0.03, samples = 5, include = c(),
  exclude = c(), set.seed = NULL)
```

Arguments

data	a data.frame or imputationList.
outcome	the colname of the dependent variable.
genepool	a vector. The genepool is the vector of variables names which shall be used to generate models. If not set, the genepool defaults to all variables in the supplied dataset other than the outcome variable.
extinction	a numeric. The algorithm will stop when no improvement has been made for this number of generations.
children	a numeric. The number of models to test in each generation.
penalty	a numeric. Model fitness will be reduced by this number for each regression coefficient. This results in a handicap for overly complicated models.
samples	a numeric. The number of sub-samples in which to test stability of r-squared.
include	a vector of colnames which must be included as predictors in each model.
exclude	a vector of colnames to be removed from the genepool.
set.seed	a numeric. If this argument is provided, the algorithm will use the given seed in order to present reproducible results.

Details

'autoModel' is a genetic algorithm which mutates regression models (predicting a specified outcome) in order to maximize r-squared (the explained variance).

The algorithm tests models at random. In each generation, it produces 'children' using the current best model as a seed. Each child of the previous winner will, on average, lose and gain a predictor. In each child, predictors have a smaller chance to gain or lose an interaction term. Over successive generations selecting seeds with larger r-squares causes a drift towards models which explain more variance.

Without intervention this algorithm generates very complicated models, e.g. 15 way interactions, in which all variance is explained. These overly-complicated models are almost certainly useless for explaining phenomenon outside of the training dataset. Generally, these models do no more than describe the exact configuration of the dataset in which they evolved. In order to deal with this situation, models are penalized for every predictor. This means that increased complexity will not be preferred unless it contributes substantially to the model's r-squared.

When the algorithm has failed to improve model fitness over many successive generations, it stops and returns the best model. It also presents the history of all previous winners. The algorithm tests the stability of each of these winners on multiple sub-samples (75% of rows with replacement). Stability is equal to 1, minus the standard deviation of the r-squares in each sub sample, divided by the r-square statistic of the model in question. Stability can range from 1 to negative values (if the standard deviation of sub-sample r-squares was larger than the model's r-squared).

Value

A list containing a tibble with all the best models the algorithm found, the summary results of the best model, and a plot tracking the algorithms' performance.

Examples

```
autoModel(mtcars,"mpg",set.seed = 2)
```

`check_names` *check_names*

Description

Takes in a vector of colnames, and a data.frame or imputationList. This function will trigger an error if names are not in the data object. In addition, this function will try to predict which names the user was trying to spell.

Usage

```
check_names(x, data)
```

Arguments

`x`	a vector of colnames.
`data`	a data.frame or imputationList.

Value

check_names will trigger an error if the supplied vector of names were not found in the supplied object. It will also predict which names the user was trying to spell.

ctrx *Correlatrix*

Description

Takes in a data.frame or imputationList, a vector of variable names and produces a correlation matrix with customizable significance stars.

Usage

```
ctrx(data, x = NULL, y = NULL, triangle = "both", round = 2,
  method = "pearson", n.matrix = F, abbreviate = 100,
  stars = c(0.05, 0.01, 0.001), partial = c(), describe = F,
  leading.zero = T, ...)
```

Arguments

data	a data.frame or imputationList.
x	a vector of variable names to correlate (optional).
y	a vector of column names for the creation of asymmetric correlation matrices.
triangle	a string containing one of "lower" "upper" or "both". Indicates if correlations are to be displayed above or below the diagonal. "Both" is selected by default.
round	a numeral indicating number of decimals.
method	a string containing one of "pearson","spearman" or "kendall".
n.matrix	logical. If TRUE, matrix of n returned.
abbreviate	a number indicating the maximum length of variable names.
stars	a numeric vector. For each numeral, a star will be assigned which indicates that the p-value for a given correlation was at, or smaller than, that level. The default is 0.05, 0.01 and 0.001.
partial	a vector of colnames. If supplied the function will output a matrix of partial correlations. All effects will be controlled for by the variables in this vector.
describe	a list of functions with names or a logical. If functions are supplied to describe, a new column will be appended to the final data.frame for each argument in the list. If TRUE is supplied, means and standard deviation is appended with na.rm = T.
leading.zero	a logical. If FALSE, leading zeros are removed.
...	the argument 'var.names' from previous versions has been deprecated, please use x instead.

Value

A data.frame containing a correlation matrix.

Examples

```
correlatrix(mtcars[,1:5])
library(magrittr)
mtcars %>%
ctrx(x = c("mpg","cyl","disp")
,y = c("wt","drat"),
round = 2,
stars = c(0.05),
describe = list("mean" = function(x) mean(x,na.rm=TRUE)))
```

`find_similar` *find_similar*

Description

Takes in two vectors of strings. This function will return strings in the second set, which are similar to those in the first.

Usage

```
find_similar(x, y, percent = 50)
```

Arguments

`x`	a vector of strings.
`y`	a vector of strings. Strings will be returned from y which are similar to those in x
`percent`	a numeric. Strings in y will be returned if they are at least this percent similar. Defaults to 50%.

Value

a vector containing strings from y, which are similar to those in x.

`get.numeric` *get.numeric*

Description

Takes in a data.frame or imputationList, removes non-numeric columns, and returns the object.

Usage

```
get.numeric(x)
```

Arguments

x	a data.frame or 'imputationList'

Value

Returns the original object with all non-numeric columns removed.

---

int.plot *Interaction plot*

Description

Calculates a standardized two way or three way interaction and plots using ggplot2.

Usage

```
int.plot(data, outcome, predictor, moderator, y.lim = c(-1, 1),
  x.lim = c(-1, 1), x.lab = "auto", y.lab = "auto", title = "auto",
  title.size = 15, SDs = 1, legend.name = "auto",
  colour = "ghostwhite", show.points = FALSE, save = F,
  path = getwd())
```

Arguments

data	an object of class 'data.frame' or 'imputationList'.
outcome	a string with the name of the outcome variable.
predictor	a string with the name of the predictor variable.
moderator	a vector of the names of up to two moderating variables.
y.lim	vector of numerals indicating y axis bounds.
x.lim	vector of numerals indicating x axis bounds.
x.lab	a string with the label of the x axis.
y.lab	a string with the label of the y axis.
title	a string containing title text.
title.size	a numeral containing the font size of the title.
SDs	a numeral indicating the standard deviations of the moderators.
legend.name	a character string indicating the title of the legend.
colour	a character string containing the colour of the data points.
show.points	logical to determine whether or not to include points.
save	logical as to whether or not to save the plot.
path	string containing path of where to save plot. Defaults to working directory.

Value

A ggplot

Examples

```
carsdata<-mtcars
int.plot(carsdata,"mpg","disp","cyl", y.lim = c(-2.5,2.5))
int.plot(carsdata,"mpg","disp", c("cyl","am"), y.lim = c(-5.0,2.0))
```

Index