Package ‘ClustImpute’

October 12, 2022

Type Package

Title K-Means Clustering with Build-in Missing Data Imputation

Version 0.2.4

Author Oliver Pfaffel

Maintainer Oliver Pfaffel <opfaffel@gmail.com>

Description
This k-means algorithm is able to cluster data with missing values and as a by-product completes the data set. The implementation can deal with missing values in multiple variables and is computationally efficient since it iteratively uses the current cluster assignment to define a plausible distribution for missing value imputation. Weights are used to shrink early random draws for missing values (i.e., draws based on the cluster assignments after few iterations) towards the global mean of each feature. This shrinkage slowly fades out after a fixed number of iterations to reflect the increasing credibility of cluster assignments. See the vignette for details.

License GPL-3

Encoding UTF-8

Imports ClusterR, copula, dplyr, magrittr, tidyr, ggplot2, rlang, knitr

Suggests ggExtra, rmarkdown, testthat (>= 2.1.0), Hmisc, tictoc, spelling, corrplot, covr

VignetteBuilder knitr

RoxygenNote 7.1.0

Language en-US

NeedsCompilation no

Repository CRAN

Date/Publication 2021-05-31 07:40:11 UTC

R topics documented:

check_replace_dups .................................................. 2
ClustImpute ............................................................ 2
default_wf .............................................................. 4
check_replace_dups  Check and replace duplicate (centroid) rows

Description

Internal function of ClustImpute: check new centroids for duplicate rows and replace with random draws in this case.

Usage

cHECK_REPLACE_DUPS(CENTROIDS, X, SEED)

Arguments

centroids  Matrix of centroids
X            Underlying data matrix (without missings)
seed        Seed used for random sampling

Value

Returns centroids where duplicate rows are replaced by random draws

ClustImpute  K-means clustering with build-in missing data imputation

Description

Clustering algorithm that produces a missing value imputation using on the go. The (local) imputation distribution is defined by the currently assigned cluster. The first draw is by random imputation.
Usage

ClustImpute(
    X,
    nr_cluster,
    nr_iter = 10,
    c_steps = 1,
    wf = default_wf,
    n_end = 10,
    seed_nr = 150519,
    assign_with_wf = TRUE,
    shrink_towards_global_mean = TRUE
)

Arguments

X                  Data frame with only numeric values or NAs
nr_cluster        Number of clusters
nr_iter           Iterations of procedure
c_steps           Number of clustering steps per iteration
wf                 Weight function. Linear up to n_end by default. Used to shrink X towards zero or the global mean (default). See shrink_towards_global_mean
n_end              Steps until convergence of weight function to 1
seed_nr           Number for set.seed()
assign_with_wf    Default is TRUE. If set to False, then the weight function is only applied in the centroid computation, but ignored in the cluster assignment.
shrink_towards_global_mean

Value

complete_data  Completed data without NAs
clusters        For each row of complete_data, the associated cluster
centroids       For each cluster, the coordinates of the centroids in tidy format
centroids_matrix For each cluster, the coordinates of the centroids in matrix format
imp_values_mean Mean of the imputed variables per draw
imp_values_sd   Standard deviation of the imputed variables per draw

Examples

# Random Dataset
set.seed(739)
n <- 750 # number of points
nr_other_vars <= 2
mat <- matrix(rnorm(nr_other_vars*n),n,nr_other_vars)
default_wf

K-means clustering with build-in missing data imputation

description

Default weight function. One minus the return value is multiplied with missing(=imputed) values. It starts with 1 and goes to 0 at n_end.

Usage

default_wf(n, n_end = 10)

Arguments

n current step

n_end steps until convergence of weight function to 0

Value

value between 0 and 1

Examples

x <- 0:20
plot(x, 1 - default_wf(x))
miss_sim

Simulation of missings

Description

Simulates missing at random using a normal copula to create correlations between the missing (type="MAR"). Missings appear in each column of the provided data frame with the same ratio.

Usage

miss_sim(dat, p = 0.2, type = "MAR", seed_nr = 123)

Arguments

dat Data frame with only numeric values
p Fraction of missings (for entire data frame)
type Type of missingness. Either MCAR (=missing completely at random) or MAR (=missing at random)
seed_nr Number for set.seed()

Value
data frame with only numeric values and NAs

Examples

data(cars)
cars_with_missings <- miss_sim(cars,p = .2,seed_nr = 4)
summary(cars_with_missings)

plot.kmeans_ClustImpute

Plot showing marginal distribution by cluster assignment

Description

Returns a plot with the marginal distributions by cluster and feature. The plot shows histograms or boxplots and, as a ggplot object, can be modified further.
Usage

```r
## S3 method for class 'kmeans_ClustImpute'
plot(
  x,
  type = "hist",
  vline = "centroids",
  hist_bins = 30,
  color_bins = "#56B4E9",
  color_vline = "#E69F00",
  size_vline = 2,
  ...
)
```

Arguments

- **x**: an object returned from ClustImpute
- **type**: either "hist" to plot a histogram or "box" for a boxplot
- **vline**: for "hist" a vertical line is plotted showing either the centroid value or the mean of all data points grouped by cluster and feature
- **hist_bins**: number of bins for histogram
- **color_bins**: color for the histogram bins
- **color_vline**: color for the vertical line
- **size_vline**: size of the vertical line
- **...**: currently unused

Value

Returns a ggplot object

---

`predict.kmeans_ClustImpute`

*Prediction method*

Description

Prediction method

Usage

```r
## S3 method for class 'kmeans_ClustImpute'
predict(object, newdata, ...)```
Arguments

object Object of class kmeans_ClustImpute
newdata Data frame

Value

integer value (cluster assignment)

Examples

# Random Dataset
set.seed(739)
n <- 750 # numer of points
nr_other_vars <- 2
mat <- matrix(rnorm(nr_other_vars*n),n,nr_other_vars)
me<-4 # mean
x <- c(rnorm(n/3,me/2,1),rnorm(2*n/3,-me/2,1))
y <- c(rnorm(n/3,0,1),rnorm(n/3,me,1),rnorm(n/3,-me,1))
dat <- cbind(mat,x,y)
dat<- as.data.frame(scale(dat)) # scaling

# Create NAs
dat_with_miss <- miss_sim(dat,p=.1,seed_nr=120)

res <- ClustImpute(dat_with_miss,nr_cluster=3)
predict(res,newdata=dat[1,])

print.kmeans_ClustImpute

Print method for ClustImpute

Description

Returns a plot with the marginal distributions by cluster and feature. The plot shows histograms or boxplots and, as a ggplot object, can be modified further.

Usage

## S3 method for class 'kmeans_ClustImpute'
print(x, ...)

Arguments

x an object returned from ClustImpute

... currently unused
**Value**

No return value (print function)

---

**var_reduction**

**Reduction of variance**

**Description**

Computes one minus the ratio of the sum of all within cluster variances by the overall variance

**Usage**

```
var_reduction(clusterObj)
```

**Arguments**

- `clusterObj` Object of class kmeans_ClustImpute

**Value**

integer value typically between 0 and 1

**Examples**

```r
# Random Dataset
set.seed(739)
n <- 750 # number of points
nr_other_vars <- 2
mat <- matrix(rnorm(nr_other_vars*n),n,nr_other_vars)
me<-4 # mean
x <- c(rnorm(n/3,me/2,1),rnorm(2*n/3,-me/2,1))
y <- c(rnorm(n/3,0,1),rnorm(n/3,me,1),rnorm(n/3,-me,1))
dat <- cbind(mat,x,y)
dat<- as.data.frame(scale(dat)) # scaling

# Create NAs
dat_with_miss <- miss_sim(dat,p=.1,seed_nr=120)

res <- ClustImpute(dat_with_miss,nr_cluster=3)
var_reduction(res)
```
Index

check_replace_dups, 2
ClustImpute, 2

default_wf, 4

miss_sim, 5

plot.kmeans_ClustImpute, 5
predict.kmeans_ClustImpute, 6
print.kmeans_ClustImpute, 7

var_reduction, 8