R documentation

of 'clsPartialNA.Rd'

August 26, 2017

---

clsPartialNA — *Compute/display tuple frequency counts, and optionally account for NA values in parallel.*

Description

The function tupleFreqs is the workhorse function in the package, calculating frequency counts to be used in the graphs. It determines the set of unique tuples, and computes a count for each. The function clsPartialNA does the same, but in parallel. The k most- or least-frequent tuples will be reported, with the latter specified via negative k. Optionally, tuples with NA values will count less, but weigh toward everything that has existing numbers in common with it.

Usage

```
tupleFreqs(dataset,k=5,NAexp=1.0,countNAs=FALSE,saveCounts=TRUE,
   minFreq=NULL)
clsPartialNA(cls=NULL, dataset, k=5, NAexp=1, countNAs=FALSE)
discparcoord(data, k=5, grpcategory=NULL, permute=FALSE,
    interactive = TRUE, save=FALSE, name="Parcoords", labelsOff=TRUE,
   NAexp=1.0,countNAs=FALSE, accentuate=NULL, accval=100, inParallel=FALSE,
    cls=NULL, differentiate=FALSE, saveCounts=TRUE, minFreq=NULL)
```

Arguments

data	The data, in data frame or matrix form.
k	The number of lines to display in the parallel coordinates plot.
grpcategory	Grouping column/variable.
permute	If TRUE, randomly permute the columns before plotting.
interactive	If TRUE, use interactive plotting, allowing for interactively readjusting column order and scrubbing/brushing.

save	If this is TRUE and interactive mode is on, saved plot will be available from the browser.
name	The name for the plot.
labelsOff	If TRUE, labels are off. This only comes into effect when interactive=FALSE.
NAexp	Scale for NA counts.
countNAs	If TRUE, count NA values.
accentuate	Character expression specifying the property to accentuate.
accval	The value to accentuate. By default, this is 100.
inParallel	If TRUE, calculate tuple frequencies in parallel.
differentiate	If TRUE, randomize coloring to differentiate overlapping lines.
saveCounts	If TRUE, save the tuple counts to the file tupleCounts.
minFreq	The smallest frequency to be displayed.
dataset	The dataset to process, a data frame or data.table.
k	The number of tuples to return. These will be the k most frequent tuples, unless k is negative, in which case the least-frequent tuples will be returned. The latter is useful for hunting for outliers.
NAexp	Scale NA signifiance. This is experimental and may be extremely slow.
countNAs	Whether or not you want to count NA values.
cls	Cluster to be used if inParallel is TRUE. If inParallel is TRUE and cls is not supplied, it will use the sensed number of cores on the calling machine by default.

Details

The data will be converted into a data.table if it is not already in that form. For this and other reasons, it is advantageous to have the data in that form to begin with, say by using fread to read the data.

Optionally, tuples that partially match a full tuple pattern except for NA values will add a partial count to the frequency count for the full pattern. If for instance the data consist of 8-tuples and a row in the data matches a given 8-tuple pattern in 7 of 8 components, this row would add a count of 7/8 to the frequency for that pattern. To reduce this weight, use a value greater than 1.0 for NAexp. If that value is 2, for example, the 7/8 increment will be 7/8 squared.

Value

The functions tupleFreqs and clsPartialNA return an object of class c('pna','data.frame'), with each row consisting of a tuple and its count. In addition the object will have attributes k and minFreq.

The function discparcoord returns an object of class c('plotly','htmlwidget'). Printing the object causes display of the graph.

Author(s)

Norm Matloff <matloff@cs.ucdavis.edu>, Vincent Yang <vinyang@ucdavis.edu>, and Harrison Nguyen <hhnguy@ucdavis.edu>

Examples

```
data(Titanic_Passengers)
# Find frequencies in parallel
discparcoord(Titanic_Passengers, inParallel=TRUE)
data(hrdata)
input1 = list("name" = "average_montly_hours",
              "partitions" = 3, "labels" = c("low", "med", "high"))
input = list(input1)
# this will discretize the data by partitioning average monthly
# hours into 3 parts called low, med, and high
hrdata = discretize(hrdata, input)
# account for NA values and plot with parallel coordinates
discparcoord(hrdata)

# same as above, but with scrambled columns
discparcoord(hrdata, permute=TRUE)

# same as above, but show top k values
discparcoord(hrdata, k=8)

# same as above, but group according to profession
discparcoord(hrdata, grpcategory="sales")
```

Index