Package ‘TabularManifest’

March 20, 2014

Title Tabular Manifest

Description Assists the manipulation and exploration of wide datasets with tabular configuration files

Date 2014-03-20

Version 0.1-15

License LGPL

LazyData TRUE

VignetteBuilder knitr

Maintainer Will Beasley <wibeasley@hotmail.com>

URL http://melinae.com/

Depends R(>= 3.0.0),stats

Imports ggplot2,grid,mgcv,plyr,scales

Suggests datasets,devtools,knitr,RColorBrewer,testit,testthat

Roxygen list(wrap = FALSE)

R topics documented:

`TabularManifest-package` *Tabular Manifest*

Description

Our consulting company, Melinae, frequently assists clients with large datasets consisting of many variables of varying quality. Before we can develop sophisticated statistical models to provide our client with insight and a competitive advantage, we first learn the characteristics of their existing datasets. This package provides tools that assist our initial exploration of real-world datasets. Although these tools are not a substitute of thoughtful inspection in our later analyses, these make the exploration more time efficient. These tools allow us to more quickly start developing innovative solutions and delivering results.

The idea behind this package is that *configuring* metadata is quicker and more robust than *coding* the same repetitive code. [We need to write more as the package takes shape.]

Thanks, the Melinae Analytics Team

Note

Our company has benefited from many tools developed by the community, and we'd like to contribute back. Suggestions, criticisms, and code contributions are welcome. If any developer is interested in trying a direction that suits them better, we'll be happy to explain the package's internals and help you fork your own version. We have some starting material described in the `./documentation_for_developers/` directory. The repository is currently hosted at our GitHub server.

If your organization is interested in the consulting services of Melinae, please contact Steve Soloway at [what material belongs here]?

[Is there anything else someone would like to include?]

#TODO: this line needs to be adapated when we move to GitHub. For those interested in use the development version of `TabularManifest`, run

```
devtools::install_git(repo="Melinae/TabularManifest")
```

Author(s)

William Howard Beasley

Chad Scherrer

Steve Soloway

Maintainer: Will Beasley <wibeasley@hotmail.com>

References

[Do any article or book references make sense? Maybe reproducible research?]

Examples

```
CreateManifestExploreUnivariate(datasets::InsectSprays, writeToDisk=FALSE)

if( require(grDevices) ) {
  HistogramContinuous(dsObserved=beaver1, variableName="temp", binWidth=.1)
  HistogramDiscrete(dsObserved=infert, variableName="age")
}
```

`CalculateBins` *Internal function for creating default bins for dataset variables.*

Description

An internal function (ie, that's not currently exposed/exported outside the package) for creating default bins for dataset variables.

Usage

```
CalculateBins(dsObserved, binCountSuggestion = 30L)
```

Arguments

`dsObserved`	The `data.frame` with columns to calculate bins.
`binCountSuggestion`	An `integer` or `numeric` value for the suggested number of bins for each variable.

Value

Returns a `list`, with two elements. Each element is an array with as many values as columns in `dsObserved`.

1. `binWidth` The variable name (in `dsObserved`).
2. `binStart` The variable's `class`. (eg, numeric, Date, factor)

Examples

```
#TabularManifest:::CalculateBins(dsObserved=datasets::freeny)
#TabularManifest:::CalculateBins(dsObserved=datasets::InsectSprays)
```

CalculateRoundingDigits *Internal function for calculating rounding digits for dataset variables*

Description

An internal function (ie, that's not currently exposed/exported outside the package) for creating default bins for dataset variables.

Usage

```
CalculateRoundingDigits(dsObserved)
```

Arguments

`dsObserved`	The `data.frame` with columns to calculate bins.

Value

Returns a `numeric` vector, indicating how many rounding digits *might* be appropriate. Each element is an array with as many values as columns in `dsObserved`.

Examples

```
TabularManifest:::CalculateRoundingDigits(dsObserved=freeny)
TabularManifest:::CalculateRoundingDigits(dsObserved=InsectSprays)
TabularManifest:::CalculateRoundingDigits(dsObserved=beaver1)
```

ConstructGraph *Construct a graph or list of graphs*

Description

Construct a graph or list of graphs, whose characteristics are determined by a configuration file.

Usage

```
ConstructGraphUnivariate(variableName, dsMetadata, dsObserved)
```

Arguments

`variableName`	The name of the single variable to graph.
`dsMetadata`	The `data.frame` containing the metadata. See `CreateManifestExploreUnivariate`.
`dsObserved`	The `data.frame` containing the data to be graphed.

Examples

```
#dsObserved <- beaver1
dsObserved <- InsectSprays
dsManifest <- TabularManifest::CreateManifestExploreUnivariate(dsObserved, writeToDisk=FALSE)

ConstructGraphUnivariate(variableName="count", dsManifest, InsectSprays)

ConstructGraphListUnivariate(dsManifest=dsManifest, dsObserved=dsObserved)
```

`CreateManifestExploreUnivariate`

Create a manifest for exploratoring univariate patterns.

Description

This function creates a meta-dataset (from the `data.frame` passed as a parameter) and optionally saves the meta-dataset as a CSV. The meta-dataset specifies how the variables should be plotted.

Usage

```
CreateManifestExploreUnivariate(
    dsObserved,
    writeToDisk = TRUE,
    pathOut = getwd(),
    overwriteFile = FALSE,
    defaultClassGraph = c(
      numeric = "HistogramContinuous",
      integer = "HistogramContinuous",
      factor = "HistogramDiscrete",
      character = "HistogramDiscrete",
      notMatched = "histogram_generic"
    ),
    defaultFormat = c(
      numeric = "scales::comma",
      notMatched = "scales::comma"
    ),
    binCountSuggestion = 30L
)
```

Arguments

`dsObserved` The `data.frame` to create metadata for.

`writeToDisk` Indicates if the meta-dataset should be saved as a CSV.

`pathOut` The file path to save the meta-dataset.

`overwriteFile` Indicates if the CSV of the meta-dataset should be overwritten if a file already exists at the location.

`defaultFormat` A `character` array indicating which formatting function should be displayed on the axis of the univariate graph.

`defaultClassGraph`
: A `character` array indicating which graph should be used with variables of a certain class.

`binCountSuggestion`
: An `integer` value of the number of roughly the number bins desired for a histogram.

Value

Returns a `data.frame` where each row in the metadata represents a column in `dsObserved`. The metadata contains the following columns

1. `variableName` The variable name (in `dsObserved`). `character`.
2. `remark` A blank field that allows theuser to enter notes in the CSV for later reference.
3. `class` The variable's `class` (eg, numeric, Date, factor). `character`.
4. `shouldGraph` A boolean value indicating if the variable should be graphed. `logical`.
5. `graphFunction` The name of the function used to graph the variable. `character`.
6. `xLabelFormat` The name of the function used to format the x-axis. `character`.
7. `binWidth` The uniform width of the bins. `numeric`.
8. `binStart` The location of the left boundary of the first bin. `numeric`.

Examples

```
CreateManifestExploreUnivariate(datasets::InsectSprays, writeToDisk=FALSE)

#Careful, the first column is a `ts` class.
CreateManifestExploreUnivariate(datasets::freeny, writeToDisk=FALSE)
```

`HistogramContinuous` *Generate a Histogram for a* `numeric` *or* `integer` *variable.*

Description

Generate a histogram for a `numeric` or `integer` variable. This graph is intended to quickly provide the researcher with a quick, yet thorough representation of the continuous variable. The additional annotations may not be desired for publication-quality plots.

Usage

```
HistogramContinuous(dsObserved, variableName, binWidth = NULL,
  mainTitle = variableName, xTitle = paste0(variableName, " (each bin is ",
  scales::comma(binWidth), " units wide)"), yTitle = "Frequency",
  roundedDigits = 0L)
```

Arguments

dsObserved The `data.frame` with the variable to graph.

variableName The name of the variable to graph. `character`.

binWidth The width of the histogram bins. If NULL, the `ggplot2` default is used. `numeric`.

mainTitle The desired title on top of the graph. Defaults to `variableName`. If no title is desired, pass a value of `NULL`. `character`.

xTitle The desired title on the *x*-axis. Defaults to the `variableName` and the `binWidth`. If no axis title is desired, pass a value of `NULL`. `character`.

yTitle The desired title on the *y*-axis. Defaults to "Frequency". If no axis title is desired, pass a value of `NULL`. `character`.

roundedDigits The number of decimals to show for the mean and median annotations. `character`.

Value

Returns a histogram as a `ggplot2` object.

Examples

```
library(datasets)
#Don't run graphs on a headless machine without any the basic graphics packages installed.
if( require(grDevices) ) {
  HistogramContinuous(dsObserved=beaver1, variableName="temp", binWidth=.1)
}
```

`HistogramDiscrete` *Generate a Histogram for a* `character` *or* `factor` *variable.*

Description

Generate a histogram for a `character` or `factor` variable. This graph is intended to quickly provide the researcher with a quick, yet thorough representation of the continuous variable. The additional annotations may not be desired for publication-quality plots.

Usage

```
HistogramDiscrete(dsObserved, variableName, levelsToExclude = character(0),
  mainTitle = variableName, xTitle = NULL,
  yTitle = "Number of Included Records", textSizePercentage = 6,
  binWidth = 1L)
```

Arguments

dsObserved The `data.frame` with the variable to graph.

variableName The name of the variable to graph. `character`.

levelsToExclude An array of of the levels to be excluded from the histogram. Pass an empty variable (*ie*, `character(0)`) if all levels are desired; this is the default. `character`.

mainTitle The desired title on top of the graph. Defaults to `variableName`. If no title is desired, pass a value of `NULL`. `character`.

`xTitle` The desired title on the *x*-axis. Defaults to the number of included records. If no axis title is desired, pass a value of `NULL`. `character`.

`yTitle` The desired title on the *y*-axis. Defaults to “Frequency”. If no axis title is desired, pass a value of `NULL`. `character`.

`textSizePercentage`
The size of the percentage values on top of the bars. `character`.

`binWidth` (This parameter is included for compatibility with other graphing functions. It should always be `1` for discrete and boolean variables.)

Value

Returns a histogram as a `ggplot2` object.

Examples

```
library(datasets)
#Don't run graphs on a headless machine without any the basic graphics packages installed.
if( require(grDevices) ) {
  HistogramDiscrete(dsObserved=infert, variableName="education")
  HistogramDiscrete(dsObserved=infert, variableName="age")
}
```

ScatterModelContinuousXBinaryYLogit

Internal function for examining a logit performance

Description

Internal function for examining a logit performance

Usage

```
ScatterModelContinuousXBinaryYLogit(ds_plot, x_name, y_name = "y",
  yhat_name = "yhat", residual_name = "residual", alpha_point = 0.05,
  alpha_se_band = 0.15, xLabelFormat = scales::comma,
  color_smooth_observed = "#1b9e77", color_smooth_predicted = "#d95f02",
  color_smooth_residual = "#7570b3", vertical_limits = c(-0.05, 1.05),
  jitter_observed = ggplot2::position_jitter(w = 0, h = 0.2),
  jitter_predicted = ggplot2::position_jitter(w = 0, h = 0),
  seed_value = NA_real_)
```

Arguments

`ds_plot` The `data.frame` of observed and predicted values to plot.

`x_name` The name of the predictor `character`.

`y_name` The name of the observed response `character`.

`yhat_name` The name of the predicted response `character`.

`residual_name` The name of the model residual. `character`.

`alpha_point` The transparency of each plotted point. A `numeric` value from 0 to 1.

`alpha_se_band` The transparency of the standard error bands. A `numeric` value from 0 to 1.

`xLabelFormat` The name of the function used to format the *x*-axis. `character`.

`color_smooth_observed`
The plotted color of the observed values' GAM trend. `character`.

`color_smooth_predicted`
The plotted color of the predicted's GAM trend. `character`.

`color_smooth_residual`
The plotted color of the residual's GAM trend. `character`.

`vertical_limits`
The plotted limits of the response variable. A two-element `numeric` array.

`jitter_observed`
A function dictating how the observed values are jittered.

`jitter_predicted`
A function dictating how the predicted values are jittered.

`seed_value` The value of the RNG seed, which affects jittering. No seed is set if a value of `NA` is passed. `numeric`.

ScatterModelDiscreteXBinaryYLogit

Internal function for examining a logit performance

Description

Internal function for examining a logit performance

Usage

```
ScatterModelDiscreteXBinaryYLogit(dsPlot, xName, yName = "y",
  yHatName = "yhat", residualName = "residual", alphaPoint = 0.05,
  alphaSEBand = 0.15, xLabelFormat = scales::comma,
  colorSmoothObserved = "#1b9e77", colorSmoothPredicted = "#d95f02",
  colorSmoothResidual = "#7570b3", colorGroupCount = "tomato",
  verticalLimits = c(-0.05, 1.05),
  jitterObserved = ggplot2::position_jitter(w = 0.35, h = 0.2),
  jitterPredicted = ggplot2::position_jitter(w = 0.35, h = 0),
  seedValue = NA_real_)
```

Arguments

`dsPlot` The `data.frame` of observed and predicted values to plot.

`xName` The name of the predictor `character`.

`yName` The name of the observed response `character`.

`yHatName` The name of the predicted response `character`.

`residualName` The name of the model residual. `character`.

`alphaPoint` The transparency of each plotted point. A `numeric` value from 0 to 1.

`alphaSEBand` The transparency of the standard error bands. A `numeric` value from 0 to 1.

`xLabelFormat` The name of the function used to format the *x*-axis. `character`.

`colorSmoothObserved`
The plotted color of the observed values' GAM trend. `character`.

`colorSmoothPredicted`
The plotted color of the predicted's GAM trend. `character`.

`colorSmoothResidual`
The plotted color of the residual's GAM trend. `character`.

`colorGroupCount`
The color indicating how many cases belong to each level. `character`.

`verticalLimits` The plotted limits of the response variable. A two-element `numeric` array.

`jitterObserved` A function dictating how the observed values are jittered.

`jitterPredicted`
A function dictating how the predicted values are jittered.

`seedValue` The value of the RNG seed, which affects jittering. No seed is set if a value of `NA` is passed. `numeric`.

Index