Package 'ExploreModelMatrix'

October 14, 2021

Type Package

Title Graphical Exploration of Design Matrices

Version 1.4.0

Description Given a sample data table and a design formula, ExploreModelMatrix generates an interactive application for exploration of the resulting design matrix. This can be helpful for interpreting model coefficients and constructing appropriate contrasts in (generalized) linear models. Static visualizations can also be generated.

License MIT + file LICENSE

Encoding UTF-8

Imports shiny (>= 1.5.0), shinydashboard, DT, cowplot, utils, dplyr, magrittr, tidyr, ggplot2, stats, methods, rintrojs, scales, tibble, MASS, limma, S4Vectors, shinyjs

RoxygenNote 7.1.1

Suggests testthat (>= 2.1.0), knitr, rmarkdown, htmltools, BiocStyle

VignetteBuilder knitr

URL https://github.com/csoneson/ExploreModelMatrix

BugReports https://github.com/csoneson/ExploreModelMatrix/issues

biocViews ExperimentalDesign, Regression, DifferentialExpression

git_url https://git.bioconductor.org/packages/ExploreModelMatrix

git_branch RELEASE_3_13

git_last_commit 72ccfe4

git_last_commit_date 2021-05-19

Date/Publication 2021-10-14

Author Charlotte Soneson [aut, cre] (<https://orcid.org/0000-0003-3833-2169>), Federico Marini [aut] (<https://orcid.org/0000-0003-3252-7758>), Michael Love [aut] (<https://orcid.org/0000-0001-8401-0545>), Florian Geier [aut] (<https://orcid.org/0000-0002-9076-9264>), Michael Stadler [aut] (<https://orcid.org/0000-0002-2269-4934>)

Maintainer Charlotte Soneson <charlottesoneson@gmail.com>

R topics documented:

`ExploreModelMatrix` *Explore model matrix*

Description

Given a sample data table and a design formula, explore the resulting design matrix graphically in an interactive application.

Usage

```
ExploreModelMatrix(sampleData = NULL, designFormula = NULL)
```

Arguments

`sampleData`	(optional) A `data.frame` or `DataFrame` with sample information. If set to `NULL`, the user can upload the sample information from a tab-separated text file inside the app, or choose among a collection of example designs provided in the app.
`designFormula`	(optional) A `formula`. All components of the terms must be present as columns in `sampleData`. If set to `NULL`, the design formula can be specified after launching the app.

Value

A Shiny app object

Author(s)

Charlotte Soneson, Federico Marini, Michael I Love, Florian Geier, Michael B Stadler

Examples

```
app <- ExploreModelMatrix(
  sampleData = data.frame(genotype = rep(c("A", "B"), each = 4),
                          treatment = rep(c("treated", "untreated"), 4)),
  designFormula = ~genotype + treatment
)
if (interactive()) shiny::runApp(app)
```

`ExploreModelMatrix-pkg` *ExploreModelMatrix*

Description

ExploreModelMatrix is an R package for visualizing design matrices generated by the `model.matrix()` R function. Provided with a sample data table and a design formula, the `ExploreModelMatrix()` function launches a shiny app where the user can explore the fitted values (in terms of the model coefficients) for each combination of predictor values.

`VisualizeDesign` *Visualize design matrix*

Description

Given a sample table and a design formula, generate a collection of static plots for exploring the resulting design matrix graphically. This function is called internally by `ExploreModelMatrix()`, but can also be used directly if interactivity is not required.

Usage

```
VisualizeDesign(
  sampleData,
  designFormula,
  flipCoordFitted = FALSE,
  flipCoordCoocc = FALSE,
  textSizeFitted = 5,
  textSizeCoocc = 5,
  textSizeLabsFitted = 12,
  textSizeLabsCoocc = 12,
  lineWidthFitted = 25,
  addColorFitted = TRUE,
  colorPaletteFitted = scales::hue_pal(),
  dropCols = NULL,
  designMatrix = NULL
)
```

Arguments

`sampleData`	A `data.frame` of `DataFrame` with sample information.
`designFormula`	A `formula`. All components of the terms must be present as columns in `sampleData`.
`flipCoordFitted, flipCoordCoocc`	A `logical`, whether to flip the coordinate axes in the fitted values/co-occurrence plot, respectively.

`textSizeFitted, textSizeCoocc`
: A `numeric` scalar giving the text size in the fitted values/co-occurrence plot, respectively.

`textSizeLabsFitted, textSizeLabsCoocc`
: A `numeric` scalar giving the text size for the axis labels in the fitted values/co-occurrence plot, respectively.

`lineWidthFitted`
: A `numeric` scalar giving the maximal length of a row in the fitted values plot, before it is split and printed on multiple lines

`addColorFitted`
: A `logical` scalar indicating whether the terms in the fitted values plot should be shown in different colors.

`colorPaletteFitted`
: A `function` returning a color palette to use for coloring the model coefficients in the fitted values plot.

`dropCols`
: A character vector with columns to drop from the design matrix, or NULL if no columns should be dropped.

`designMatrix`
: A `numeric` matrix, which can be supplied as an alternative to `designFormula`. Rows must be in the same order as the rows in `sampleData`.

Value

A list with the following elements:

- `sampledata`: A `data.frame`, expanded from the input `sampleData`
- `plotlist`: A list of plots, displaying the fitted values for each combination of predictor values, in terms of the model coefficients.
- `designmatrix`: The design matrix, after removing any columns in `dropCols`
- `pseudoinverse`: The pseudoinverse of the design matrix
- `vifs`: A `data.frame` with calculated variance inflation factors
- `colors`: A vector with colors to use for different model coefficients
- `cooccurrenceplots`: A list of plots, displaying the co-occurrence pattern for the predictors (i.e., the number of observations for each combination of predictor values)
- `totnbrrows`: The total number of "rows" in the list of plots of fiitted values. Useful for deciding the required size of the plot canvas.

Author(s)

Charlotte Soneson

Examples

```
VisualizeDesign(
  sampleData = data.frame(genotype = rep(c("A", "B"), each = 4),
                          treatment = rep(c("treated", "untreated"), 4)),
  designFormula = ~genotype + treatment
)
```

Index