Package 'EvidenceSynthesis'

February 26, 2018

Type Package

Title An R package for Combining Evidence from Multiple Sources

Version 0.0.2

Date 2018-02-26

Author Martijn Schuemie, Marc Suchard

Maintainer Martijn Schuemie <schuemie@ohdsi.org>

Description Routines for combining evidence and diagnostics across multiple sources, such as multiple data sites in a distributed study. This includes functions for performing meta-analysis and forest plots.

Imports ggplot2 (>= 2.0.0),
gridExtra,
meta,
EmpiricalCalibration

License Apache License 2.0

URL https://github.com/OHDSI/EvidenceSynthesis

BugReports https://github.com/OHDSI/EvidenceSynthesis/issues

RoxygenNote 6.0.1

R topics documented:

EvidenceSynthesis *EvidenceSynthesis*

Description

EvidenceSynthesis

1

plotCovariateBalances   *Plot covariate balances*

Description

plotCovariateBalances plots the covariate balance before and after matching for multiple data sources.

Usage

```
plotCovariateBalances(balances, labels, threshold = 0,
  beforeLabel = "Before matching", afterLabel = "After matching",
  fileName = NULL)
```

Arguments

balances	A list of covariate balance objects as created using the [computeCovariateBalance](#) function in the CohortMethod package. Each balance object is expected to be a data.frame with at least these two columns: beforeMatchingStdDiff and afterMatchingStdDiff.
labels	A vector containing the labels for the various sources.
threshold	Show a threshold value for the standardized difference.
beforeLabel	Label for before matching / stratification / trimming.
afterLabel	Label for after matching / stratification / trimming.
fileName	Name of the file where the plot should be saved, for example 'plot.png'. See the function ggsave in the ggplot2 package for supported file formats.

Details

Creates a plot showing the covariate balance before and after matching. Balance distributions are displayed as box plots combined with scatterplots.

Value

A Ggplot object. Use the ggsave function to save to file.

Examples

```
# Some example data:
balance1 <- data.frame(beforeMatchingStdDiff = rnorm(1000, 0.1, 0.1),
                       afterMatchingStdDiff = rnorm(1000, 0.0, 0.01))
balance2 <- data.frame(beforeMatchingStdDiff = rnorm(1000, 0.2, 0.1),
                       afterMatchingStdDiff = rnorm(1000, 0.0, 0.05))
balance3 <- data.frame(beforeMatchingStdDiff = rnorm(1000, 0.0, 0.1),
                       afterMatchingStdDiff = rnorm(1000, 0.0, 0.03))
plotCovariateBalances(balances = list(balance1, balance2, balance3),
                      labels = c("Site A", "Site B", "Site C"))
```

---

plotEmpiricalNulls *Plot empirical null distributions*

Description

`plotEmpiricalNulls` plots the empirical null distribution for multiple data sources.

Usage

```
plotEmpiricalNulls(logRr, seLogRr, labels, xLabel = "Relative risk",
  limits = c(0.1, 10), showCis = TRUE, fileName = NULL)
```

Arguments

logRr	A numeric vector of effect estimates for the negative controls on the log scale.
seLogRr	The standard error of the log of the effect estimates. Hint: often the standard error = (log(<lower bound 95 percent confidence interval>) - l og(<effect estimate>))/qnorm(0.025).
labels	A vector containing the labels for the various sources. Should be of equal length as logRr and seLogRr.
xLabel	The label on the x-axis: the name of the effect estimate.
limits	The limits of the effect size axis.
showCis	Show the 95 percent confidence intervals on the null distribution and distribution parameter estimates?
fileName	Name of the file where the plot should be saved, for example 'plot.png'. See the function ggsave in the ggplot2 package for supported file formats.

Details

Creates a plot showing the empirical null distributions. Distributions are shown as mean plus minus one standard deviation, as well as a distribution plot.

Value

A Ggplot object. Use the `ggsave` function to save to file.

Examples

```
# Some example data:
site1 <- EmpiricalCalibration::simulateControls(n = 50, mean = 0, sd = 0.1, trueLogRr = 0)
site1$label <- "Site 1"
site2 <- EmpiricalCalibration::simulateControls(n = 50, mean = 0.1, sd = 0.2, trueLogRr = 0)
site2$label <- "Site 2"
site3 <- EmpiricalCalibration::simulateControls(n = 50, mean = 0.15, sd = 0.25, trueLogRr = 0)
site3$label <- "Site 3"
sites <- rbind(site1, site2, site3)

plotEmpiricalNulls(logRr = sites$logRr,
                   seLogRr = sites$seLogRr,
                   labels = sites$label)
```

---

plotMetaAnalysisForest

Perform a meta-analysis and create a forest plot

Description

plotMetaAnalysisForest performs a meta-analysis and creates a forest plot of effect size estimates.

Usage

```
plotMetaAnalysisForest(logRr, logLb95Ci, logUb95Ci, labels,
  xLabel = "Relative risk", limits = c(0.1, 10), hakn = TRUE,
  fileName = NULL)
```

Arguments

logRr	A numeric vector of effect estimates on the log scale.
logLb95Ci	The lower bound of the 95 percent confidence interval on the log scale.
logUb95Ci	The upper bound of the 95 percent confidence interval on the log scale.
labels	A vector containing the labels for the various estimates.
xLabel	The label on the x-axis: the name of the effect estimate.
limits	The limits of the effect size axis.
hakn	A logical indicating whether method by Hartung and Knapp should be used to adjust test statistics and confidence intervals.
fileName	Name of the file where the plot should be saved, for example 'plot.png'. See the function ggsave in the ggplot2 package for supported file formats.

Details

Creates a forest plot of effect size estimates, and includes a meta-analysis estimate using a random effects model. The DerSimonian-Laird estimate (1986) is used.

Value

A Ggplot object. Use the ggsave function to save to file.

References

DerSimonian R, Laird N (1986), Meta-analysis in clinical trials. Controlled Clinical Trials, 7, 177-188.

Examples

```
plotMetaAnalysisForest(logRr = c(0, 0.2, -0.2, 0, 0.2, -0.2),
                       logLb95Ci = c(-0.2, -0.2, -0.6, -0.2, -0.2, -0.6),
                       logUb95Ci = c(0.2, 0.6, 0.2, 0.2, 0.6, 0.2),
                  labels = c("Site A", "Site B", "Site C", "Site D", "Site E", "Site F"))
```

---

plotPreparedPs *Plot the propensity score distribution*

Description

Plot the propensity score distribution

Usage

```
plotPreparedPs(preparedPsPlots, labels, treatmentLabel = "Target",
  comparatorLabel = "Comparator", fileName = NULL)
```

Arguments

preparedPsPlots

> A list of prepared propensity score data as created by the [preparePsPlot](#) function.

labels   A vector containing the labels for the various sources.

treatmentLabel A label to us for the treated cohort.
comparatorLabel

> A label to us for the comparator cohort.

fileName   Name of the file where the plot should be saved, for example 'plot.png'. See the
                function ggsave in the ggplot2 package for supported file formats.

Value

A ggplot object. Use the [ggsave](#) function to save to file in a different format.

Examples

```
treatment <- rep(0:1, each = 100)
propensityScore <- c(rnorm(100, mean = 0.4, sd = 0.25), rnorm(100, mean = 0.6, sd = 0.25))
data <- data.frame(treatment = treatment, propensityScore = propensityScore)
data <- data[data$propensityScore > 0 & data$propensityScore < 1, ]
preparedPlot <- preparePsPlot(data)
# Just reusing the same data three times for demonstration purposes:
preparedPsPlots <- list(preparedPlot, preparedPlot, preparedPlot)
labels <- c("Data site A", "Data site B", "Data site C")
```

---

preparePsPlot *Prepare to plot the propensity score distribution*

Description

preparePsPlot prepares to plot the propensity (or preference) score distribution. It computes the
distribution, so the output does not contain person-level data.

Usage

```
preparePsPlot(data, unfilteredData = NULL, scale = "preference")
```

Arguments

data   A data frame with at least the two columns described below

unfilteredData   To be used when computing preference scores on data from which subjects have
                 already been removed, e.g. through trimming and/or matching. This data frame
                 should have the same structure as data.

scale   The scale of the graph. Two scales are supported: `scale = 'propensity'` or
                    `scale = 'preference'`. The preference score scale is defined by Walker et al
                    (2013).

Details

The data frame should have a least the following two columns:

treatment	(integer)	Column indicating whether the person is in the treated (1) or comparator (0) group
propensityScore	(numeric)	Propensity score

Value

A data frame describing the propensity score (or preference score) distribution at 100 equally-
spaced points.

References

Walker AM, Patrick AR, Lauer MS, Hornbrook MC, Marin MG, Platt R, Roger VL, Stang P, and
Schneeweiss S. (2013) A tool for assessing the feasibility of comparative effectiveness research,
Comparative Effective Research, 3, 11-20

Examples

```
treatment <- rep(0:1, each = 100)
propensityScore <- c(rnorm(100, mean = 0.4, sd = 0.25), rnorm(100, mean = 0.6, sd = 0.25))
data <- data.frame(treatment = treatment, propensityScore = propensityScore)
data <- data[data$propensityScore > 0 & data$propensityScore < 1, ]
preparedPlot <- preparePsPlot(data)
```

Index