Package 'EmpiricalCalibration'

August 16, 2016

Type Package

Title Routines for Performing Empirical Calibration of Observational Study Estimates

Version 1.1.1

Date 2016-08-16

Author Martijn Schuemie, Marc Suchard

Maintainer Martijn Schuemie <schuemie@ohdsi.org>

Description Routines for performing empirical calibration of observational study estimates. By using a set of negative control hypotheses we can estimate the empirical null distribution of a particular observational study setup. This empirical null distribution can be used to compute a calibrated p-value, which reflects the probability of observing an estimated effect size when the null hypothesis is true taking both random and systematic error into account.

VignetteBuilder knitr

Imports ggplot2 (>= 2.0.0), methods

Suggests knitr, rmarkdown

License Apache License 2.0

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

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

RoxygenNote 5.0.1

R topics documented:

calibrateConfidenceInterval 2
calibrateP 3
caseControl 4
cohortMethod 5
computeTraditionalP 6
EmpiricalCalibration 6
fitMcmcNull 7
fitNull 8
fitSystematicErrorModel 9
plotCalibration 9

plotCalibrationEffect 10
plotCiCalibration 11
plotForest 12
plotMcmcTrace 13
plotTrueAndObserved 13
sccs 14
simulateControls 15

Index **16**

`calibrateConfidenceInterval` *Calibrate confidence intervals*

Description

Calibrate confidence intervals

Usage

```
calibrateConfidenceInterval(logRr, seLogRr, model, ciWidth = 0.95)
```

Arguments

`logRr`	A numeric vector of effect estimates 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>) - log(<effect estimate>))/qnorm(0.025).
`model`	An object of type `systematicErrorModel` as created by the `fitSystematicErrorModel` function.
`ciWidth`	The width of the confidence interval. Typically this would be .95, for the 95 percent confidence interval.

Details

Compute calibrated confidence intervals based on a model of the systematic error.

Value

A data frame with calibrated confidence intervals and point estimates.

Examples

```
data <- simulateControls(n = 50 * 3, mean = 0.25, sd = 0.25, trueLogRr = log(c(1, 2, 4)))
model <- fitSystematicErrorModel(data$logRr, data$seLogRr, data$trueLogRr)
newData <- simulateControls(n = 15, mean = 0.25, sd = 0.25, trueLogRr = log(c(1, 2, 4)))
result <- calibrateConfidenceInterval(newData$logRr, newData$seLogRr, model)
result
```

`calibrateP` *Calibrate the p-value*

Description

`calibrateP` computes calibrated p-values using the fitted null distribution

Usage

```
calibrateP(null, logRr, seLogRr, ...)

## S3 method for class 'null'
calibrateP(null, logRr, seLogRr, ...)

## S3 method for class 'mcmcNull'
calibrateP(null, logRr, seLogRr, pValueOnly, ...)
```

Arguments

`null`	An object of class `null` created using the `fitNull` function or an object of class `mcmcNull` created using the `fitMcmcNull` function.
`logRr`	A numeric vector of one or more effect estimates 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>) - log(<effect estimate>))/qnorm(0.025)
`...`	Any additional parameters (currently none).
`pValueOnly`	If true, will return only the calibrated P-value itself, not the credible interval.

Details

This function computes a calibrated two-sided p-value as described in Schuemie et al (2014).

Value

The two-sided calibrated p-value.

Methods (by class)

- `null`: Computes the calibrated P-value using asymptotic assumptions.
- `mcmcNull`: Computes the calibrated P-value and 95 percent credibel interval using Markov Chain Monte Carlo (MCMC).

References

Schuemie MJ, Ryan PB, Dumouchel W, Suchard MA, Madigan D. Interpreting observational studies: why empirical calibration is needed to correct p-values. Statistics in Medicine 33(2):209-18,2014

Examples

```
data(sccs)
negatives <- sccs[sccs$groundTruth == 0, ]
null <- fitNull(negatives$logRr, negatives$seLogRr)
positive <- sccs[sccs$groundTruth == 1, ]
calibrateP(null, positive$logRr, positive$seLogRr)
```

`caseControl` *Odds ratios from a case-control design*

Description

Odds ratios from a case-control design

Usage

```
data(caseControl)
```

Format

A data frame with 47 rows and 4 variables:

drugName Name of the drug

groundTruth Whether the drug is a positive (1) or negative (0) control

logRr The log of the incidence rate ratio

seLogRr The standard error of the log of the incidence rate ratio

Details

A dataset containing the odds ratios (and standard errors) produced using a case-control design. The outcome is upper GI bleeding, the drug of interest (groundTruth = 1) is sertraline. Also included are 46 negative control drugs, for which we believe there to be no causal relation with upper GI bleeding. We used a database of medical records from general practices in the USA, the General Electric (GE) Centricity database, which contains data on 11.2 million subjects. We restricted on study period (start of 1990 through November 2003), age requirements (18 years or older), available time prior to event (180 days), number of controls per case (6), and risk definition window (30 days following the prescription). Controls were matched on age and sex. Cases of upper GI bleeding were identified on the basis of the occurrence of ICD-9 diagnosis codes in the problem list. These codes pertain to esophageal, gastric, duodenal, peptic, and gastrojejunal ulceration, perforation, and hemorrhage, as well as gastritis and non-specific gastrointestinal hemorrhage. For more information on this set see Schuemie et al (2014).

References

Schuemie MJ, Ryan PB, Dumouchel W, Suchard MA, Madigan D. Interpreting observational studies: why empirical calibration is needed to correct p-values. Statistics in Medicine 33(2):209-18,2014

`cohortMethod` *Relative risks from a new-user cohort design*

Description

Relative risks from a new-user cohort design

Usage

```
data(cohortMethod)
```

Format

A data frame with 31 rows and 4 variables:

drugName Name of the drug

groundTruth Whether the drug is a positive (1) or negative (0) control

logRr The log of the incidence rate ratio

seLogRr The standard error of the log of the incidence rate ratio

Details

A dataset containing the relative risks (and standard errors) produced using a new-user cohort design. The outcome is acute liver injury, the drug of interest (groundTruth = 1) is Isoniazid Also included are 30 negative control drugs, for which we believe there to be no causal relation with acute liver injury. We used the Thomson MarketScan Medicare Supplemental Beneficiaries database, which contains data on 4.6 million subjects. We selected two groups (cohorts): (1) all subjects exposed to isoniazid and (2) all subjects having the ailment for which isoniazid is indicated, in this case tuberculosis, and having received at least one drug that is not known to cause acute liver injury. We removed all subjects who belonged to both groups and subjects for which less than 180 days of observation time was available prior to their first exposure to the drug in question. Acute liver injury was identified on the basis of the occurrence of ICD-9-based diagnosis codes from inpatient and outpatient medical claims and was defined broadly on the basis of codes associated with hepatic dysfunction, as have been used in prior observational database studies. The time at risk was defined as the length of exposure + 30 days, and we determined whether subjects experienced an acute liver injury during their time at risk. Using propensity score stratification, the cohorts were divided over 20 strata, and an odds ratio over all strata was computed using a Mantel-Haenszel test. The propensity score was estimated using Bayesian logistic regression using all available drug, condition, and procedure covariates occurring in the 180 days prior to first exposure, in addition to age, sex, calendar year of first exposure, Charlson index, number of drugs, number of visit days, and number of procedures. For more information on this set see Schuemie et al (2014).

References

Schuemie MJ, Ryan PB, Dumouchel W, Suchard MA, Madigan D. Interpreting observational studies: why empirical calibration is needed to correct p-values. Statistics in Medicine 33(2):209-18,2014

computeTraditionalP *Compute the (traditional) p-value*

Description

computeTraditionalP computes the traditional two-sided p-value based on the log of the relative risk and the standerd error of the log of the relative risk.

Usage

```
computeTraditionalP(logRr, seLogRr)
```

Arguments

logRr	A numeric vector of one or more effect estimates 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>) - log(<effect estimate>))/qnorm(0.025)

Value

The two-sided (traditional) p-value.

Examples

```
data(sccs)
positive <- sccs[sccs$groundTruth == 1, ]
computeTraditionalP(positive$logRr, positive$seLogRr)
```

EmpiricalCalibration *EmpiricalCalibration*

Description

EmpiricalCalibration

`fitMcmcNull` *Fit the null distribution using MCMC*

Description

`fitNull` fits the null distribution to a set of negative controls using Markov Chain Monte Carlo (MCMC).

Usage

```
fitMcmcNull(logRr, seLogRr, iter = 10000)
```

Arguments

`logRr`	A numeric vector of effect estimates 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>) - log(<effect estimate>))/qnorm(0.025)
`iter`	Number of iterations of the MCMC.

Details

This is an experimental function for computing the 95 percent credible interval of a calibrated p-value using Markov-Chain Monte Carlo (MCMC).

Value

An object of type `mcmcNull` containing the mean and standard deviation (both on the log scale) of the null distribution, as well as the MCMC trace.

Examples

```
data(sccs)
negatives <- sccs[sccs$groundTruth == 0, ]
null <- fitMcmcNull(negatives$logRr, negatives$seLogRr)
null
plotMcmcTrace(null)
positive <- sccs[sccs$groundTruth == 1, ]
calibrateP(null, positive$logRr, positive$seLogRr)
```

`fitNull` *Fit the null distribution*

Description

`fitNull` fits the null distribution to a set of negative controls

Usage

```
fitNull(logRr, seLogRr)
```

Arguments

`logRr`	A numeric vector of effect estimates 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>) - log(<effect estimate>))/qnorm(0.025)

Details

This function fits a Gaussian function to the negative control estimates as described in Schuemie et al (2014).

Value

An object containing the parameters of the null distribution.

References

Schuemie MJ, Ryan PB, Dumouchel W, Suchard MA, Madigan D. Interpreting observational studies: why empirical calibration is needed to correct p-values. Statistics in Medicine 33(2):209-18,2014

Examples

```
data(sccs)
negatives <- sccs[sccs$groundTruth == 0, ]
null <- fitNull(negatives$logRr, negatives$seLogRr)
null
```

fitSystematicErrorModel *Fit a systematic error model*

Description

Fit a systematic error model

Usage

```
fitSystematicErrorModel(logRr, seLogRr, trueLogRr)
```

Arguments

logRr	A numeric vector of effect estimates 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>) - log(<effect estimate>))/qnorm(0.025).
trueLogRr	A vector of the true effect sizes.

Details

Fit a model of the systematic error as a function of true effect size. This model is an extention of the method for fitting the null distribution. The mean and standard deviations of the error distributions are assumed to be linear with respect to the true effect size, and each component is therefore represented by an intercept and a slope.

Value

An object of type `systematicErrorModel`.

Examples

```
controls <- simulateControls(n = 50 * 3, mean = 0.25, sd = 0.25, trueLogRr = log(c(1, 2, 4)))
model <- fitSystematicErrorModel(controls$logRr, controls$seLogRr, controls$trueLogRr)
model
```

plotCalibration *Create a calibration plot*

Description

`plotCalibration` creates a plot showing the calibration of our calibration procedure

Usage

```
plotCalibration(logRr, seLogRr, useMcmc = FALSE, legendPosition = "right",
  fileName = NULL)
```

Arguments

logRr	A numeric vector of effect estimates 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>) - log(<effect estimate>))/qnorm(0.025)
useMcmc	Use MCMC to estimate the calibrated P-value?
legendPosition	Where should the legend be positioned? ("none", "left", "right", "bottom", "top")
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 calibration plot showing the number of effects with p < alpha for every level of alpha. The empirical calibration is performed using a leave-one-out design: The p-value of an effect is computed by fitting a null using all other negative controls. Ideally, the calibration line should approximate the diagonal. The plot shows both theoretical (traditional) and empirically calibrated p-values.

Value

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

Examples

```
data(sccs)
negatives <- sccs[sccs$groundTruth == 0, ]
plotCalibration(negatives$logRr, negatives$seLogRr)
```

---

plotCalibrationEffect *Plot the effect of the calibration*

Description

plotCalibrationEffect creates a plot showing the effect of the calibration.

Usage

```
plotCalibrationEffect(logRrNegatives, seLogRrNegatives, logRrPositives,
  seLogRrPositives, null = NULL, xLabel = "Relative risk",
  showCis = FALSE, fileName = NULL)
```

Arguments

logRrNegatives	A numeric vector of effect estimates of the negative controls on the log scale.
seLogRrNegatives	The standard error of the log of the effect estimates of the negative controls.
logRrPositives	A numeric vector of effect estimates of the positive controls on the log scale.

seLogRrPositives	The standard error of the log of the effect estimates of the positive controls.
null	An object representing the fitted null distribution as created by the fitNull function. If not provided, a null will be fitted before plotting.
xLabel	The label on the x-axis: the name of the effect estimate.
showCis	Show 95 percent credible intervals for the calibrated p = 0.05 boundary.
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 with the effect estimate on the x-axis and the standard error on the y-axis. Negative controls are shown as blue dots, positive controls as yellow diamonds. The area below the dashed line indicated estimates with p < 0.05. The orange area indicates estimates with calibrated p < 0.05.

Value

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

Examples

```
data(sccs)
negatives <- sccs[sccs$groundTruth == 0, ]
positive <- sccs[sccs$groundTruth == 1, ]
plotCalibrationEffect(negatives$logRr, negatives$seLogRr, positive$logRr, positive$seLogRr)
```

---

plotCiCalibration *Create a confidence interval calibration plot*

Description

plotCalibration creates a plot showing the calibration of our confidence interval calibration procedure

Usage

```
plotCiCalibration(logRr, seLogRr, trueLogRr, strata = as.factor(trueLogRr),
  legendPosition = "top", fileName = NULL)
```

Arguments

logRr	A numeric vector of effect estimates 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>) - log(<effect estimate>))/qnorm(0.025).
trueLogRr	The true log relative risk.
strata	Variable used to stratify the plot. Set strata = NULL for no stratification.
legendPosition	Where should the legend be positioned? ("none", "left", "right", "bottom", "top").
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 calibration plot showing the fraction of effects within the confidence interval. The empirical calibration is performed using a leave-one-out design: The confidence interval of an effect is computed by fitting a null using all other controls. Ideally, the calibration line should approximate the diagonal. The plot shows the coverage for both theoretical (traditional) and empirically calibrated confidence intervals.

Value

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

Examples

```
data <- simulateControls(n = 50 * 3, mean = 0.25, sd = 0.25, trueLogRr = log(c(1, 2, 4)))
plotCiCalibration(data$logRr, data$seLogRr, data$trueLogRr)
```

---

plotForest *Create a forest plot*

Description

plotForest creates a forest plot of effect size estimates.

Usage

```
plotForest(logRr, seLogRr, names, xLabel = "Relative risk", fileName = NULL)
```

Arguments

logRr	A numeric vector of effect estimates 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>) - log(<effect estimate>))/qnorm(0.025)
names	A vector containing the names of the drugs or outcomes
xLabel	The label on the x-axis: the name of the effect estimate
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 (ratios). Estimates that are significantly different from 1 (alpha = 0.05) are marked in orange, others are marked in blue.

Value

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

Examples

```
data(sccs)
negatives <- sccs[sccs$groundTruth == 0, ]
plotForest(negatives$logRr, negatives$seLogRr, negatives$drugName)
```

---

plotMcmcTrace *Plot the MCMC trace*

Description

Plot the MCMC trace

Usage

```
plotMcmcTrace(mcmcNull, fileName = NULL)
```

Arguments

mcmcNull	An object of type mcmcNull as generated using the fitMcmcNull function.
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

Plot the trace of the MCMC for diagnostics purposes.

Examples

```
data(sccs)
negatives <- sccs[sccs$groundTruth == 0, ]
null <- fitMcmcNull(negatives$logRr, negatives$seLogRr)
plotMcmcTrace(null)
```

---

plotTrueAndObserved *Plot true and observed values*

Description

Plot true and observed values, for example from a simulation study.

Usage

```
plotTrueAndObserved(logRr, seLogRr, trueLogRr, xLabel = "Relative risk",
  fileName = NULL)
```

Arguments

logRr	A numeric vector of effect estimates 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>) - log(<effect estimate>))/qnorm(0.025).
trueLogRr	A vector of the true effect sizes.
xLabel	The label on the x-axis: the name of the effect estimate.
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 (ratios). Estimates that are significantly different from the true value (alpha = 0.05) are marked in orange, others are marked in blue.

Value

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

Examples

```
data <- simulateControls(n = 50 * 3, mean = 0.25, sd = 0.25, trueLogRr = log(c(1, 2, 4)))
plotTrueAndObserved(data$logRr, data$seLogRr, data$trueLogRr)
```

---

sccs *Incidence rate ratios from Self-Controlled Case Series*

Description

Incidence rate ratios from Self-Controlled Case Series

Usage

```
data(sccs)
```

Format

A data frame with 46 rows and 4 variables:

drugName Name of the drug

groundTruth Whether the drug is a positive (1) or negative (0) control

logRr The log of the incidence rate ratio

seLogRr The standard error of the log of the incidence rate ratio

Details

A dataset containing the incidence rate ratios (and standard errors) produced using a Self-Controlled Case Series (SCCS) design. The outcome is upper GI bleeding, the drug of interest (groundTruth = 1) is sertraline. Also included are 45 negative control drugs, for which we believe there to be no causal relation with upper GI bleeding. We used a database of medical records from general practices in the USA, the General Electric (GE) Centricity database, which contains data on 11.2 million subjects. We restricted on study period (start of 1990 through November 2003), age requirements (18 years or older), available time prior to event (180 days), and risk definition window (30 days following the prescription). Time 30 days prior to the first prescription was removed to account for possible contra-indications. Cases of upper GI bleeding were identified on the basis of the occurrence of ICD-9 diagnosis codes in the problem list. These codes pertain to esophageal, gastric, duodenal, peptic, and gastrojejunal ulceration, perforation, and hemorrhage, as well as gastritis and non-specific gastrointestinal hemorrhage. For more information on this set see Schuemie et al (2014).

References

Schuemie MJ, Ryan PB, Dumouchel W, Suchard MA, Madigan D. Interpreting observational studies: why empirical calibration is needed to correct p-values. Statistics in Medicine 33(2):209-18,2014

`simulateControls` *Simulate (negative) controls*

Description

Simulate (negative) controls

Usage

```
simulateControls(n = 50, mean = 0, sd = 0.1, seLogRr = runif(n, min =
  0.01, max = 0.2), trueLogRr = 0)
```

Arguments

`n`	Number of controls to simulate.
`mean`	The mean of the error distribution (on the log RR scale).
`sd`	The standard deviation of the error distribution (on the log RR scale).
`seLogRr`	The standard error of the log of the relative risk. This is recycled for the controls. The default is to sample these from a uniform distribution.
`trueLogRr`	The true relative risk (on the log scale) used to generate these controls. This is recycled for the controls.

Details

Generate point estimates given known true effect sizes and standard errors

Examples

```
data <- simulateControls(n = 50 * 3, mean = 0.25, sd = 0.25, trueLogRr = log(c(1, 2, 4)))
plotTrueAndObserved(data$logRr, data$seLogRr, data$trueLogRr)
```

Index

∗Topic **datasets**
caseControl, 4
cohortMethod, 5
sccs, 14

calibrateConfidenceInterval, 2
calibrateP, 3
caseControl, 4
cohortMethod, 5
computeTraditionalP, 6

EmpiricalCalibration, 6
EmpiricalCalibration-package *(*EmpiricalCalibration*)*, 6

fitMcmcNull, 7
fitNull, 8
fitSystematicErrorModel, *2*, 9

plotCalibration, 9
plotCalibrationEffect, 10
plotCiCalibration, 11
plotForest, 12
plotMcmcTrace, 13
plotTrueAndObserved, 13

sccs, 14
simulateControls, 15