Package 'refineR'

August 21, 2024

Type Package

Version 1.6.2

Date 2024-08-14

Title Reference Interval Estimation using Real-World Data

Author Tatjana Ammer [aut, cre],
Christopher M Rank [aut],
Andre Schuetzenmeister [aut]

Maintainer Tatjana Ammer <tatjana.ammer@roche.com>

Depends R (>= 3.2.0)

Imports stats, ash, future, future.apply, parallel, graphics, grDevices

Suggests knitr, rmarkdown

Description Indirect method for the estimation of reference intervals using Real-World Data ('RWD'). It takes routine measurements of diagnostic tests, containing pathological and non-pathological samples as input and uses sophisticated statistical methods to derive a model describing the distribution of the non-pathological samples. This distribution can then be used to derive reference intervals. Furthermore, the package offers functions for printing and plotting the results of the algorithm. See ?refineR for a more comprehensive description of the features. Version 1.0 of the algorithm is described in detail in 'Ammer et al. (2021)' <doi:10.1038/s41598-021-95301-2>. Additional guidance on the usage of the algorithm is given in 'Ammer et al. (2023)' <doi:10.1093/jalm/jfac101>.

License GPL (>= 3)

VignetteBuilder knitr, rmarkdown

NeedsCompilation no

LazyData true

Repository CRAN

Date/Publication 2024-08-21 21:40:02 UTC

Contents

`refineR-package` *refineR: Reference Interval Estimation using Real-World Data (RWD)*

Description

This package includes the implementation of the refineR algorithm (Ammer et al., 2021) which is an indirect method for the estimation of reference intervals using Real-World Data (RWD). It takes routine measurements of diagnostic tests, containing pathological and non-pathological samples as input and uses sophisticated statistical methods to derive a model describing the distribution of the non-pathological samples. This distribution can then be used to derive reference intervals. Main function of this package is `findRI` that takes an input data set and tries to find a model that best explains the non-pathological distribution. Furthermore, the package offers functions for printing `print.RWDRI` and plotting `plot.RWDRI` the results of the algorithm operating on S3-objects of class 'RWDRI'.

Details

Package:	refineR
Type:	Package
Version:	1.6.2
Date:	2024-08-14
License:	GPL (>=3)
LazyLoad:	yes

Author(s)

Tatjana Ammer <tatjana.ammer@roche.com>, Christopher M Rank <christopher.rank@roche.com>, Andre Schuetzenmeister <andre.schuetzenmeister@roche.com>

References

Ammer, T., Schuetzenmeister, A., Prokosch, HU., Rauh, M., Rank, C.M., Zierk, J. refineR: A Novel Algorithm for Reference Interval Estimation from Real-World Data. Sci Rep 11, 16023 (2021).

addGrid *Add a grid to an existing plot.*

Description

It is possible to use automatically determined grid lines (x=NULL, y=NULL) or specifying the number of cells x = 3, y = 4 as done by grid. Additionally, x- and y-locations of grid-lines can be specified, e.g. x = 1:10, y = seq(0,10,2).

Usage

```
addGrid(x = NULL, y = NULL, col = "lightgray", lwd = 1L, lty = 3L)
```

Arguments

x	(integer, numeric) single integer specifies number of cells, numeric vector specifies vertical grid-lines
y	(integer, numeric) single integer specifies number of cells, numeric vector specifies horizontal grid-lines
col	(character) color of grid-lines
lwd	(integer) line width of grid-lines
lty	(integer) line type of grid-lines

Author(s)

Andre Schuetzenmeister <andre.schuetzenmeister@roche.com>

`as.rgb` *Convert color-names or RGB-code to possibly semi-transparent RGB-code.*

Description

Function takes the name of a color and converts it into the rgb space. Parameter "alpha" allows to specify the transparency within [0,1], 0 meaning completey transparent and 1 meaning completey opaque. If an RGB-code is provided and alpha != 1, the RGB-code of the transparency adapted color will be returned.

Usage

```
as.rgb(col = "black", alpha = 1)
```

Arguments

`col` (character) name of the color to be converted/transformed into RGB-space (code). Only those colors can be used which are part of the set returned by function colors(). Defaults to "black".

`alpha` (numeric) value specifying the transparency to be used, 0 = completely transparent, 1 = opaque.

Value

RGB-code

Author(s)

Andre Schuetzenmeister <`andre.schuetzenmeister@roche.com`>

Examples

```
## Not run:
# convert character string representing a color to RGB-code using alpha-channel of .25 (75\
     as.rgb("red", alpha = .25)

# same thing now using the RGB-code of red (alpha=1, i.e. as.rgb("red"))
     as.rgb("#FF0000FF", alpha = .25)

## End(Not run)
```

ashDensity *Estimate density of distribution employing the R package "ash" using R-wrapper function.*

Description

Estimate density of distribution employing the R package "ash" using R-wrapper function.

Usage

```
ashDensity(x, ab, nbin, m, kopt = c(2, 1), normToAB = FALSE)
```

Arguments

`x` (numeric) vector of data points

`ab` (numeric) vector of lower and higher truncation limit of density estimation

`nbin` (integer) specifying the number of bins used for density estimation

`m` (integer) specifying the width of the smoothing kernel(s) used for density estimation

`kopt` (integer) vector specifying the smoothing kernel

`normToAB` (logical) specifying if the density is normed to the interval ab or to all data points in x

Value

(list) with density estimation (x values, y values, m and ab).

Author(s)

Christopher Rank <`christopher.rank@roche.com`>, Tatjana Ammer <`tatjana.ammer@roche.com`>

BoxCox *One-parameter Box-Cox transformation.*

Description

One-parameter Box-Cox transformation.

Usage

```
BoxCox(x, lambda)
```

Arguments

x (numeric) data to be transformed

lambda (numeric) Box-Cox transformation parameter

Value

(numeric) vector with Box-Cox transformation of x

Author(s)

Andre Schuetzenmeister <andre.schuetzenmeister@roche.com>

---

calculateCostHist *Calculate costs for a specific combinations of lambda, muVec and sigmaVec.*

Description

Calculate costs for a specific combinations of lambda, muVec and sigmaVec.

Usage

```
calculateCostHist(
  lambda,
  muVec,
  sigmaVec,
  HistData,
  alpha = 0.01,
  alphaMcb = 0.1,
  pNormLookup
)
```

Arguments

lambda (numeric) transformation parameter for inverse Box-Cox transformation

muVec (numeric) vector of mean values of non-pathological Gaussian distribution in transformed space

sigmaVec (numeric) vector of sd values of non-pathological Gaussian distribution in transformed space

HistData (list) with histogram data generated by function generateHistData

alpha (numeric) specifying the confidence region used for selection of histgram bins in cost calculation

alphaMcb (numeric) specifying the confidence level defining the maximal allowed counts below asymmetric confidence region

pNormLookup (list) with lookup table for pnormApprox function pnormApprox

Value

(numeric) vector with (lambda, mu, sigma, P, cost).

Author(s)

Tatjana Ammer <tatjana.ammer@roche.com>

---

defineSearchRegions *Helper function to define search regions for mu and sigma and to get the region around the main peak 'ab'*

Description

The function estimates the start search regions for mu and sigma for each lambda. Further it determines an appropriate region around the main peak 'ab' that is used for all lambdas.

Usage

```
defineSearchRegions(x, lambdaVec, roundingBase, abEst = NULL)
```

Arguments

x	(numeric) values specifying data points comprising pathological and non-pathological values
lambdaVec	(numeric) transformation parameter for inverse Box-Cox transformation
roundingBase	(numeric) describing the rounding base of the dataset
abEst	(numeric) vector with already estimated abSearchReg and abHist for second definition of search regions

Value

(list) with (abEst, search region for mu and sigma)

Author(s)

Tatjana Ammer <tatjana.ammer@roche.com>

`estimateAB` *Helper function to find region around the main peak of a distribution*

Description

Helper function to find region around the main peak of a distribution

Usage

```
estimateAB(x)
```

Arguments

`x`	(numeric) vector of data points

Value

(list) with two numeric vectors with lower and upper bound of region around the main peak used for 1) defining the search regions and 2) estimating the histogram with overlapping bins

Author(s)

Tatjana Ammer <`tatjana.ammer@roche.com`>

`findMainPeak` *Helper function to find the main peak of a distribution*

Description

The function uses a combination of the area under the curve between valleys and the peak height to detect the main peak.

Usage

```
findMainPeak(x, ab, mStart, withHeight = FALSE, prevPeak = NULL)
```

Arguments

`x`	(numeric) vector of data points
`ab`	(numeric) vector specifying the lower and higher truncation limit of density estimation
`mStart`	(integer) specifying the width of the smoothing kernel(s) used for density estimation
`withHeight`	(logical) specifying if only the area under the curve (FALSE) or a combination of AUC and peak height (TRUE) should be used to detect the main peak
`prevPeak`	(numeric) specifying the modEst of the previously estimated peak

Value

(list) with the two numeric values peakInd, modEst, and a density list

Author(s)

Tatjana Ammer <`tatjana.ammer@roche.com`>

`findPeaksAndValleys` *Find the index of the peaks and valleys of the density estimation.*

Description

Find the index of the peaks and valleys of the density estimation.

Usage

```
findPeaksAndValleys(Dens)
```

Arguments

`Dens`	(list) with density estimation (x values, y values)

Value

(list) specifying the index of the peaks and valleys of the density estimation.

Author(s)

Tatjana Ammer <`tatjana.ammer@roche.com`>

`findRI` *Function to estimate reference intervals for a single population*

Description

The function estimates the optimal parameters lambda, mu and sigma for a raw data set containing pathological and non-pathological values. The optimization is carried out via a multi-level grid search to minimize the cost function (negative log-likelihood with regularization) and to find a model that fits the distribution of the physiological values and thus separates pathological from non-pathological values.

Usage

```
findRI(
  Data = NULL,
  model = c("BoxCox", "modBoxCoxFast", "modBoxCox"),
  NBootstrap = 0,
  seed = 123,
  ...
)
```

Arguments

`Data`	(numeric) values specifying data points comprising pathological and non-pathological values
`model`	(character) specifying the applied model (can be either "BoxCox" (default), "modBoxCoxFast" or "modBoxCox"), option "modBoxCoxFast" and "modBoxCox" first runs the original optimization using the Box-Cox transformation, afterwards the modified Box-Cox transformation is utilized and an optimal shift is identified ('fast': only 1 iteration is carried out to find a shift)
`NBootstrap`	(integer) specifying the number of bootstrap repetitions
`seed`	(integer) specifying the seed used for bootstrapping
`...`	additional arguments to be passed to the method

Value

(object) of class "RWDRI" with parameters optimized

Author(s)

Tatjana Ammer <tatjana.ammer@roche.com>

Examples

```
# first example

resRI <- findRI(Data = testcase1)
print(resRI)
plot(resRI, showPathol = FALSE)

# second example
resRI <- findRI(Data = testcase2)
print(resRI, RIperc = c(0.025, 0.5, 0.975))
plot(resRI, showPathol = FALSE)

# third example, with bootstrapping
resRI <- findRI(Data = testcase3, NBootstrap = 30, seed = 123)
print(resRI)
getRI(resRI, RIperc = c(0.025, 0.5, 0.975), CIprop = 0.95, pointEst ="fullDataEst")
getRI(resRI, RIperc = c(0.025, 0.5, 0.975), CIprop = 0.95, pointEst ="medianBS")
plot(resRI)
```

```
# forth example, without values and pathological distribution in plot function
resRI <- findRI(Data = testcase4)
print(resRI)
plot(resRI, showValue = FALSE, showPathol =FALSE)

# fifth example, with bootstrapping
resRI <- findRI(Data = testcase5, NBootstrap = 30)
plot(resRI,  RIperc = c(0.025, 0.5, 0.975), showPathol = FALSE, showCI = TRUE)
```

findRoundingBase *Estimate rounding base of the input data.*

Description

Estimate rounding base of the input data.

Usage

```
findRoundingBase(x)
```

Arguments

x	(numeric) vector of data points

Value

(numeric) with estimated rounding base (e.g. 0.001 when rounded to 3 digits)

Author(s)

Christopher Rank <christopher.rank@roche.com>, Tatjana Ammer <tatjana.ammer@roche.com>

generateHistData *Generate list with histogram data.*

Description

Generate list with histogram data.

Usage

```
generateHistData(x, ab, roundingBase)
```

Arguments

`x` (numeric) vector of data points

`ab` (numeric) vector of lower and higher limit embedding appropriate region with the main peak

`roundingBase` (numeric) describing the rounding base of the dataset

Value

(list) with histogram data used in the calculation of cost.

Author(s)

Tatjana Ammer <`tatjana.ammer@roche.com`>

getRI *Method to calculate reference intervals (percentiles) for objects of class 'RWDRI'*

Description

Method to calculate reference intervals (percentiles) for objects of class 'RWDRI'

Usage

```
getRI(
  x,
  RIperc = c(0.025, 0.975),
  CIprop = 0.95,
  pointEst = c("fullDataEst", "medianBS"),
  Scale = c("original", "transformed", "zScore")
)
```

Arguments

`x` (object) of class 'RWDRI'

`RIperc` (numeric) value specifying the percentiles, which define the reference interval

`CIprop` (numeric) value specifying the central region for estimation of confidence intervals

`pointEst` (character) specifying the point estimate determination: (1) using the full dataset ("fullDataEst"), (2) calculating the median model from all bootstrap samples ("medianBS"), (2) works only if NBootstrap > 0

`Scale` (character) specifying if percentiles are calculated on the original scale ("or") or the transformed scale ("tr") or the z-Score scale ("z")

Value

(data.frame) with columns for percentile, point estimate and confidence intervals.

Author(s)

Christopher Rank <christopher.rank@roche.com>, Tatjana Ammer <tatjana.ammer@roche.com>

---

getSumForPArea *Helper function to calculate the amount of observed and estimated data points within specified regions around the peak.*

Description

The function helps to define the search region for P (fraction of non-pathological samples).

Usage

```
getSumForPArea(
  pLimitMin,
  pLimitMax,
  countsPred,
  HistData,
  lambda,
  mu,
  sigma,
  pCorr
)
```

Arguments

pLimitMin	(numeric) vector specifying the lower limits for the regions next to the peak
pLimitMax	(numeric) vector specifying the upper limits for the regions next to the peak
countsPred	(numeric) vector with the predicted counts
HistData	(list) with histogram data generated by function generateHistData
lambda	(numeric) transformation parameter for inverse Box-Cox transformation
mu	(numeric) parameter of the mean of non-pathological distribution
sigma	(numeric) parameter of the standard deviation of non-pathological distribution
pCorr	(numeric) correcting the cumulative probability of the truncated non-pathological distribution

Value

(list) with two numeric vectors specifying the amount of observed and estimated data points surrounding the peak

Author(s)

Tatjana Ammer <tatjana.ammer@roche.com>

---

invBoxCox *Inverse of the one-parameter Box-Cox transformation.*

Description

Inverse of the one-parameter Box-Cox transformation.

Usage

```
invBoxCox(x, lambda)
```

Arguments

x	(numeric) data to be transformed
lambda	(numeric) Box-Cox transformation parameter

Value

(numeric) vector with inverse Box-Cox transformation of x

Author(s)

Andre Schuetzenmeister <andre.schuetzenmeister@roche.com>

---

optimizeGrid *Helper function for grid search for mu and sigma.*

Description

Helper function for grid search for mu and sigma.

Usage

```
optimizeGrid(currentBestParam, paramUnique, iter, sigmLimit = TRUE)
```

Arguments

currentBestParam	(numeric) value specifying the current best value for this parameter
paramUnique	(numeric) vector of possible values for this parameter
iter	(integer) indicating the number of iteration, as in the first iteration the search region is larger than in the following iterations
sigmLimit	(logical) specifiying if parameter is sigma and thus minimum is 0

Value

(vector) specifying the new search region fo the parameter to be optimized

Author(s)

Tatjana Ammer <tatjana.ammer@roche.com>

---

plot.RWDRI *Standard plot method for objects of class ’RWDRI’*

Description

Standard plot method for objects of class ’RWDRI’

Usage

```
## S3 method for class 'RWDRI'
plot(
  x,
  Scale = c(”original”, ”transformed”, ”zScore”),
  RIperc = c(0.025, 0.975),
  Nhist = 60,
  showCI = TRUE,
  showPathol = FALSE,
  scalePathol = TRUE,
  showBSModels = FALSE,
  showValue = TRUE,
  CIprop = 0.95,
  pointEst = c(”fullDataEst”, ”medianBS”),
  xlim = NULL,
  ylim = NULL,
  xlab = NULL,
  ylab = NULL,
  title = NULL,
  ...
)
```

Arguments

`x`	(object) of class ’RWDRI’
`Scale`	(character) specifying if percentiles are calculated on the original scale ("or") or the transformed scale ("tr") or the z-Score scale ("z")
`RIperc`	(numeric) value specifying the percentiles, which define the reference interval (default c(0.025, 0.975))
`Nhist`	(integer) number of bins in the histogram (derived automatically if not set)
`showCI`	(logical) specifying if the confidence intervals are shown

`showPathol` (logical) specifying if the estimated pathological distribution shall be shown

`scalePathol` (logical) specifying if the estimated pathological distribution shall be weighted with the ration of pathol/non-pathol

`showBSModels` (logical) specifying if the estimated bootstrapping models shall be shown

`showValue` (logical) specifying if the exact value of the estimated reference intervals shall be shown above the plot

`CIprop` (numeric) value specifying the central region for estimation of confidence intervals

`pointEst` (character) specifying the point estimate determination: (1) using the full dataset ("fullDataEst"), (2) calculating the median model from the bootstrap samples ("medianBS"), (2) works only if NBootstrap > 0

`xlim` (numeric) vector specifying the limits in x-direction

`ylim` (numeric) vector specifying the limits in y-direction

`xlab` (character) specifying the x-axis label

`ylab` (character) specifying the y-axis label

`title` (character) specifying plot title

`...` additional arguments passed forward to other functions

Value

The applied plot limits in x-direction (xlim) are returned.

Author(s)

Christopher Rank <`christopher.rank@roche.com`>, Tatjana Ammer <`tatjana.ammer@roche.com`>

`pnormApprox` *Approximate calculation of CDF of normal distribution.*

Description

Approximate calculation of CDF of normal distribution.

Usage

```
pnormApprox(q, pNormVal, mean = 0, oneOverSd = 1, oneOverH = 10)
```

Arguments

`q` (numeric) vector of quantiles of data points

`pNormVal` (numeric) vector of lookup table for pNorm

`mean` (numeric) vector of mean values

`oneOverSd` (numeric) reciprocal vector of sd values

`oneOverH` (numeric) defining the precision of the approximation

Value

(numeric) vector of approximate CDFs of normal distribution.

Author(s)

Christopher Rank <christopher.rank@roche.com>

print.RWDRI *Standard print method for objects of class 'RWDRI'*

Description

Standard print method for objects of class 'RWDRI'

Usage

```
## S3 method for class 'RWDRI'
print(
  x,
  RIperc = c(0.025, 0.975),
  CIprop = 0.95,
  pointEst = c("fullDataEst", "medianBS"),
  ...
)
```

Arguments

x (object) of class 'RWDRI'

RIperc (numeric) value specifying the percentiles, which define the reference interval

CIprop (numeric) value specifying the central region for estimation of confidence intervals

pointEst (character) specifying the point estimate determination: (1) using the full dataset ("fullDataEst"), (2) calculating the median model from all bootstrap samples ("medianBS"), (2) works only if NBootstrap > 0

... additional arguments passed forward to other functions.

Value

No return value. Instead, a summary is printed.

Author(s)

Christopher Rank <christopher.rank@roche.com>

testcase1 *Simulated Testcase 1.*

Description

This dataset consists of N = 10,000 simulated measurements with 80% non-pathological and 20% pathological samples. Ground Truth for reference intervals (2.5% perc., 97.5% perc): [10.2, 29.8]

Usage

```
testcase1
```

Format

Numeric vector with data points.

testcase2 *Simulated Testcase 2.*

Description

This dataset consists of N = 50,000 simulated measurements with 60% non-pathological and 40% pathological samples. Ground Truth for reference intervals (2.5% perc., 97.5% perc): [59.8, 160]

Usage

```
testcase2
```

Format

Numeric vector with data points.

testcase3 *Simulated Testcase 3.*

Description

This dataset consists of N = 75,000 simulated measurements with 96% non-pathological and 4% pathological samples. Ground Truth for reference intervals (2.5% perc., 97.5% perc): [9.04, 13]

Usage

```
testcase3
```

Format

Numeric vector with data points.

`testcase4` *Simulated Testcase 4.*

Description

This dataset consists of N = 100,000 simulated measurements with 90% non-pathological and 10% pathological samples. Ground Truth for reference intervals (2.5% perc., 97.5% perc): [10, 50]

Usage

```
testcase4
```

Format

Numeric vector with data points.

`testcase5` *Simulated Testcase 5.*

Description

This dataset consists of N = 250,000 simulated measurements with 80% non-pathological and 20% pathological samples. Ground Truth for reference intervals (2.5% perc., 97.5% perc): [0.25, 4]

Usage

```
testcase5
```

Format

Numeric vector with data points.

testParam *Helper function to find optimal parameters lambda, mu and sigma.*

Description

Helper function to find optimal parameters lambda, mu and sigma.

Usage

```
testParam(
  lambdaVec,
  bestParam,
  Data,
  HistData,
  startValues,
  NIter,
  alpha = 0.01,
  alphaMcb = 0.1
)
```

Arguments

`lambdaVec`	(numeric) transformation parameter for inverse Box-Cox transformation
`bestParam`	(numeric) vector containing best guess for lambda, mu, sigma, P, cost
`Data`	(numeric) values specifying percentiles or data points comprising pathological and non-pathological values
`HistData`	(list) with histogram data
`startValues`	(list) with start search regions for mu and sigma
`NIter`	(integer) specifying the number of iterations for optimized grid-search
`alpha`	(numeric) specifying the confidence region used for selection of histogram bins in cost calculation
`alphaMcb`	(numeric) specifying the confidence level defining the maximal allowed counts below the asymmetric confidence region

Value

(numeric) vector with best parameters for lambda, mu, sigma, P, cost.

Author(s)

Tatjana Ammer <`tatjana.ammer@roche.com`>

Index