

Package ‘estmeansd’

March 26, 2019

Type Package

Title Estimating the Sample Mean and Standard Deviation from Commonly Reported Quantiles in Meta-Analysis

Version 0.2.0

Maintainer Sean McGrath <sean.mcgrath@mail.mcgill.ca>

Description Implements the methods of McGrath et al. (2019) <arXiv:1903.10498> for estimating the sample mean and standard deviation from commonly reported quantiles in meta-analysis. These methods can be applied to studies that report the sample median, sample size, and one or both of (i) the sample minimum and maximum values and (ii) the first and third quartiles.

Imports graphics, grDevices, metaBLUE, stats

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

URL <https://github.com/stmcg/estmeansd>

BugReports <https://github.com/stmcg/estmeansd/issues>

NeedsCompilation no

Author Sean McGrath [aut, cre] (<<https://orcid.org/0000-0002-7281-3516>>),
XiaoFei Zhao [aut],
Russell Steele [aut],
Andrea Benedetti [aut] (<<https://orcid.org/0000-0002-8314-9497>>)

Repository CRAN

Date/Publication 2019-03-26 05:30:06 UTC

R topics documented:

bc.mean.sd	2
plot.qe.fit	4
print.bc.mean.sd	6

print.qe.fit	7
print.qe.mean.sd	8
qe.fit	9
qe.mean.sd	11
summary.qe.mean.sd	12

Index 14

bc.mean.sd	<i>Box-Cox method for estimating the sample mean and standard deviation</i>
------------	---

Description

This function applies the Box-Cox (BC) method to estimate the sample mean and standard deviation from a study that presents one of the following sets of summary statistics:

- S1: median, minimum and maximum values, and sample size
- S2: median, first and third quartiles, and sample size
- S3: median, minimum and maximum values, first and third quartiles, and sample size

Usage

```
bc.mean.sd(min.val, q1.val, med.val, q3.val, max.val, n,
           preserve.tail = FALSE, avoid.mc = FALSE)
```

Arguments

min.val	numeric value giving the sample minimum.
q1.val	numeric value giving the sample first quartile.
med.val	numeric value giving the sample median.
q3.val	numeric value giving the sample third quartile.
max.val	numeric value giving the sample maximum.
n	numeric value giving the sample size.
preserve.tail	logical scalar indicating whether to preserve or remove (if applicable) the negative-domain left support (and the corresponding right support to maintain the symmetry of the underlying normal distribution) of the Box-Cox cumulative distribution function. The classical Box-Cox transformation only takes positive numbers as input, so this parameter has a default value of FALSE. It is not possible to avoid Monte Carlo simulation when this parameter is set to TRUE. When this parameter is set to TRUE, the data-modeling distribution corresponding to the inverse Box-Cox transformation of the underlying normal distribution can have a value of infinity for its mean and/or variance. In this case, the average of the corresponding mean and/or variance produced by this function does not converge.
avoid.mc	logical scalar indicating whether to avoid Monte Carlo simulation (if possible) when performing the inverse Box-Cox transformation (the default is FALSE). See 'Details'.

Details

The BC method incorporates the Box-Cox power transformation into the sample mean estimators of Luo et al. (2016) and the sample standard deviation estimators of Wan et al. (2014) so that their assumption of normality is more tenable. The BC method consists of the following steps, outlined below.

First, an optimal value of the power parameter λ is found so that the distribution of the Box-Cox transformed data is approximately normal. Then, the methods of Luo et al. and Wan et al. are applied to estimate the mean and standard deviation of the distribution of the transformed data. Finally, the inverse transformation is applied to estimate the sample mean and standard deviation of the original, untransformed data.

To perform the inverse transformation, either numerical integration or Monte Carlo simulation can be applied, which is controlled by the `avoid.mc` argument. When the estimated mean of the Box-Cox transformed data is negative or close to zero (i.e., below 0.01), numerical integration often does not converge. Therefore, Monte Carlo simulation is automatically used in this case.

Value

A object of class `bc.mean.sd`. The object is a list with the following components:

<code>est.mean</code>	Estimated sample mean.
<code>est.sd</code>	Estimated sample standard deviation.
<code>location</code>	Estimated mean of the Box-Cox transformed data.
<code>scale</code>	Estimated standard deviation of the Box-Cox transformed data.
<code>shape</code>	Estimated transformation parameter λ .
<code>bc.norm.rvs</code>	The random variables generated by the Box-Cox (or, equivalently, power-normal) distribution during the Monte Carlo simulation. If Monte Carlo simulation is not used, a value of NA is given.

The results are printed with the `print.bc.mean.sd` function.

References

- McGrath S., Zhao X., Steele R., Thombs B.D., Benedetti A., and the DEPRESSion Screening Data (DEPRESSD) Collaboration. (2019). Estimating the sample mean and standard deviation from commonly reported quantiles in meta-analysis. *ArXiv e-prints*. <https://arxiv.org/abs/1903.10498>.
- Box G.E.P., and D.R. Cox. (1964). An analysis of transformations. *Journal of the Royal Statistical Society Series B*. **26**(2):211-52.
- Luo D., Wan X., Liu J., and Tong T. (2016). Optimally estimating the sample mean from the sample size, median, mid-range, and/or mid-quartile range. *Statistical Methods in Medical Research*. **27**(6):1785-805
- Wan X., Wang W., Liu J., and Tong T. (2014). Estimating the sample mean and standard deviation from the sample size, median, range and/or interquartile range. *BMC Medical Research Methodology*. **14**:135.

Examples

```
## Generate S2 summary data
set.seed(1)
n <- 100
x <- stats::rlnorm(n, 2.5, 1)
quants <- stats::quantile(x, probs = c(0.25, 0.5, 0.75))
obs.mean <- mean(x)
obs.sd <- stats::sd(x)

## Estimate the sample mean and standard deviation using the BC method
bc.mean.sd(q1.val = quants[1], med.val = quants[2], q3.val = quants[3],
           n = n)
```

plot.qe.fit

Plot method for objects of class "qe.fit"

Description

This function plots the cumulative distribution functions of the fitted distributions along with the summary data reported by the study (i.e., the S1, S2, or S3 data).

Usage

```
## S3 method for class 'qe.fit'
plot(x, distributions = c("normal", "log-normal",
  "gamma", "weibull"), points = TRUE, limits, col, legend = TRUE,
     xlab = "x", ylab = "F(x)", ylim = c(0, 1), cex.points = 0.85,
     pch.points = 16, length.out = 10000, ...)
```

Arguments

x	object of class "qe.fit".
distributions	character vector specifying the names of distributions to be plotted. The options are: "normal", "log-normal", "gamma", "weibull", and "beta". By default, the normal, log-normal, gamma, and Weibull distributions are plotted. If one of the specified distributions was not successfully fit (e.g., <code>qe.fit</code> failed to converge for the given distribution), the distribution will not be included in the plot.
points	logical scalar indicating whether to plot the observed summary data (i.e., the sample quantiles reported by the study). The default is TRUE.
limits	numeric vector of length 2 specifying the bounds of the interval in which to evaluate of the cumulative distribution functions. See 'Details'.
col	character vector specifying the colors to use for the distributions. The kth element in this vector corresponds to the color to be used for the kth element in the distribution argument.
legend	logical scalar indicating whether to plot a legend. The default is TRUE.

xlab	x-axis label.
ylab	y-axis label.
ylim	y-axis limits.
cex.points	The magnification to be used for the plotted observed summary data (i.e., for the points argument).
pch.points	either an integer specifying a symbol or a single character to be used as the default in plotting the observed summary data (i.e., for the points argument). See points for possible values and their interpretation.
length.out	numeric scalar specifying the number of points to be used for evaluating each of the cumulative distribution functions.
...	other graphical parameters (see par).

Details

Users may need to use the `limits` argument so that a sensible interval is used for plotting. By default, the limits of the interval are the sample minimum and maximum values in scenarios S1 and S3, which is often adequate. In scenario S2, the limits are based on the quantiles of the distribution with the best fit (i.e., the fitted distribution obtaining the smallest distance between observed and distribution quantiles). If the normal distribution is the best fit, the limits of the interval are the $1/n$ th quantile and $1-1/n$ th quantile of the fitted normal distribution. If any of the other distributions are selected, the $1/n$ th quantile and 0.90 quantile are used as the limits of the interval. Depending on the skewness of the data, users may need to adjust the upper limit.

See Also

[qe.fit](#)

Examples

```
## Example 1
## Generate S3 summary data
set.seed(1)
n <- 100
x <- stats::rlnorm(n, 2.5, 1)
quants <- stats::quantile(x, probs = c(0, 0.25, 0.5, 0.75, 1))

## Fit distributions
res <- qe.fit(min.val = quants[1], q1.val = quants[2], med.val = quants[3],
             q3.val = quants[4], max.val = quants[5], n = n)
plot(res)

## Example 2
res <- qe.fit(q1.val = 1, med.val = 2, q3.val = 3, n = 100)
plot(res, limits = c(0, 5))
```

print.bc.mean.sd *Print method for objects of class "bc.mean.sd"*

Description

Print method for objects of class "bc.mean.sd".

Usage

```
## S3 method for class 'bc.mean.sd'  
print(x, ...)
```

Arguments

x	object of class "bc.mean.sd".
...	other arguments.

Value

No value is returned.

See Also

[bc.mean.sd](#)

Examples

```
## Generate S2 summary data  
set.seed(1)  
n <- 100  
x <- stats::rlnorm(n, 2.5, 1)  
quants <- stats::quantile(x, probs = c(0.25, 0.5, 0.75))  
obs.mean <- mean(x)  
obs.sd <- stats::sd(x)  
  
## Estimate the sample mean and standard deviation using the BC method  
res <- bc.mean.sd(q1.val = quants[1], med.val = quants[2],  
                  q3.val = quants[3], n = n)  
print(res)
```

print.qe.fit	<i>Print method for objects of class "qe.fit"</i>
--------------	---

Description

Print method for objects of class "qe.fit".

Usage

```
## S3 method for class 'qe.fit'  
print(x, ...)
```

Arguments

x	object of class "qe.fit".
...	other arguments.

Value

No value is returned.

See Also

[qe.fit](#)

Examples

```
## Generate S2 summary data  
set.seed(1)  
n <- 100  
x <- stats::rlnorm(n, 2.5, 1)  
quants <- stats::quantile(x, probs = c(0.25, 0.5, 0.75))  
  
## Fit distributions  
res <- qe.fit(q1.val = quants[1], med.val = quants[2], q3.val = quants[3],  
             n = n)  
print(res)
```

print.qe.mean.sd	<i>Print method for objects of class "qe.mean.sd"</i>
------------------	---

Description

Print method for objects of class "qe.mean.sd".

Usage

```
## S3 method for class 'qe.mean.sd'  
print(x, ...)
```

Arguments

x	object of class "qe.mean.sd".
...	other arguments.

Value

No value is returned.

See Also

[qe.mean.sd](#)

Examples

```
## Generate S2 summary data  
set.seed(1)  
n <- 100  
x <- stats::rlnorm(n, 2.5, 1)  
quants <- stats::quantile(x, probs = c(0.25, 0.5, 0.75))  
obs.mean <- mean(x)  
obs.sd <- stats::sd(x)  
  
## Estimate the sample mean and standard deviation using the BC method  
res <- qe.mean.sd(q1.val = quants[1], med.val = quants[2],  
  q3.val = quants[3], n = n)  
print(res)
```


Description

This function fits several parametric families of distributions from summary data in the following forms:

- S1: median, minimum and maximum values, and sample size
- S2: median, first and third quartiles, and sample size
- S3: median, minimum and maximum values, first and third quartiles, and sample size

Usage

```
qe.fit(min.val, q1.val, med.val, q3.val, max.val, n,
       two.sample.default = FALSE, qe.fit.control = list())
```

Arguments

min.val	numeric value giving the sample minimum.
q1.val	numeric value giving the sample first quartile.
med.val	numeric value giving the sample median.
q3.val	numeric value giving the sample third quartile.
max.val	numeric value giving the sample maximum.
n	numeric value giving the sample size.
two.sample.default	logical scalar. If set to TRUE, the candidate distributions, initial values, and box constraints are set to that of McGrath et al. (2018). If set to FALSE, the candidate distributions, initial values, and box constraints are set to that of McGrath et al. (2019). The default is FALSE.
qe.fit.control	optional list of control parameters for the minimization algorithm.
norm.mu.start	numeric value giving the starting value for the μ parameter of the normal distribution.
norm.sigma.start	numeric value giving the starting value for the σ parameter of the normal distribution.
lnorm.mu.start	numeric value giving the starting value for the μ parameter of the log-normal distribution.
lnorm.sigma.start	numeric value giving the starting value for the σ parameter of the log-normal distribution.
gamma.shape.start	numeric value giving the starting value for the shape parameter of the gamma distribution.
gamma.rate.start	numeric value giving the starting value for the rate parameter of the gamma distribution.
weibull.shape.start	numeric value giving the starting value for the shape parameter of the Weibull distribution.
weibull.scale.start	numeric value giving the starting value for the scale parameter of the Weibull distribution.
beta.shape1.start	numeric value giving the starting value for the shape1 (i.e., α) parameter of the beta distribution.
beta.shape2.start	numeric value giving the starting value for the shape2 (i.e., β) parameter of the beta distribution.
norm.mu.bounds	vector giving the bounds on the μ parameter of the normal distribution.
norm.sigma.bounds	vector giving the bounds on the σ parameter of the normal distribution.
lnorm.mu.bounds	vector giving the bounds on the μ parameter of the the log-normal distribution.

lnorm.sigma.bounds	vector giving the bounds on the σ parameter of the log-normal distribution.
gamma.shape.bounds	vector giving the bounds on the shape parameter of the gamma distribution.
gamma.rate.bounds	vector giving the bounds on the rate parameter of the gamma distribution.
weibull.shape.bounds	vector giving the bounds on the shape parameter of the Weibull distribution.
weibull.scale.bounds	vector giving the bounds on the scale parameter of the Weibull distribution.
beta.shape1.bounds	vector giving the bounds on the shape1 (i.e., α) parameter of the beta distribution.
beta.shape2.bounds	vector giving the bounds on the shape2 (i.e., β) parameter of the beta distribution.

Details

Distributions are fit by minimizing the distance between observed and distribution quantiles in the L2-norm. The limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS-M) algorithm implemented in the `optim` function is used for minimization.

Two different conventions may be used for setting the candidate distributions, parameter starting values, and parameter constraints, which is controlled by the `two.sample.default` argument. If the convention of McGrath et al. (2018) is used, the candidate distributions are the normal, log-normal, gamma, and Weibull distributions. If the convention of McGrath et al. (2019) is used, the beta distribution is also included. In either case, if a negative value is provided (e.g., for the minimum value or the first quartile value), only the normal distribution is fit.

Value

A object of class `qe.fit`. The object is a list with the following components:

<code>norm.par</code>	Estimated parameters of the normal distribution.
<code>lnorm.par</code>	Estimated parameters of the log-normal distribution.
<code>gamma.par</code>	Estimated parameters of the gamma distribution.
<code>weibull.par</code>	Estimated parameters of the Weibull distribution.
<code>beta.par</code>	Estimated parameters of the beta distribution.
<code>values</code>	Values of the objective functions evaluated at the estimated parameters of each candidate distribution.
<code>...</code>	Other elements.

The results are printed with the `print.qe.fit` function. The results can be visualized by using the `plot.qe.fit` function.

References

- McGrath S., Sohn H., Steele R., and Benedetti A. (2018). Two-sample aggregate data meta-analysis of medians. *ArXiv e-prints*. <https://arxiv.org/abs/1809.01278>.
- McGrath S., Zhao X., Steele R., Thombs B.D., Benedetti A., and the DEPRESSion Screening Data (DEPRESSD) Collaboration. (2019). Estimating the sample mean and standard deviation from commonly reported quantiles in meta-analysis. *ArXiv e-prints*. <https://arxiv.org/abs/1903.10498>.

Examples

```
## Generate S2 summary data
set.seed(1)
n <- 100
x <- stats::rlnorm(n, 2.5, 1)
quants <- stats::quantile(x, probs = c(0.25, 0.5, 0.75))

## Fit distributions
qe.fit(q1.val = quants[1], med.val = quants[2], q3.val = quants[3], n = n)
```

qe.mean.sd	<i>Quantile estimation method for estimating the sample mean and standard deviation</i>
------------	---

Description

This function applies the quantile estimation (QE) method to estimate the sample mean and standard deviation from a study that presents one of the following sets of summary statistics:

- S1: median, minimum and maximum values, and sample size
- S2: median, first and third quartiles, and sample size
- S3: median, minimum and maximum values, first and third quartiles, and sample size

Usage

```
qe.mean.sd(min.val, q1.val, med.val, q3.val, max.val, n,
           qe.fit.control = list())
```

Arguments

min.val	numeric value giving the sample minimum.
q1.val	numeric value giving the sample first quartile.
med.val	numeric value giving the sample median.
q3.val	numeric value giving the sample third quartile.
max.val	numeric value giving the sample maximum.
n	numeric value giving the sample size.
qe.fit.control	optional list of control parameters for qe.fit .

Details

In brief, the QE method fits candidate distribution(s) by minimizing the distance between observed and distribution quantiles. See [qe.fit](#) for further details concerning the distribution fitting step. If multiple candidate distributions are fit, the distribution with the best fit (i.e., the fitted distribution obtaining the smallest distance between observed and distribution quantiles) is selected as the underlying outcome distribution. The mean and standard deviation of the selected distribution are used to estimate the sample mean and standard deviation, respectively

Value

A list with the following components:

est.mean	Estimated sample mean.
est.sd	Estimated sample standard deviation.
selected.dist	Selected outcome distribution.
values	Values of the objective functions evaluated at the estimated parameters of each candidate distribution.
...	Other elements.

References

McGrath S., Zhao X., Steele R., Thombs B.D., Benedetti A., and the DEPRESSion Screening Data (DEPRESSD) Collaboration. (2019). Estimating the sample mean and standard deviation from commonly reported quantiles in meta-analysis. *ArXiv e-prints*. <https://arxiv.org/abs/1903.10498>.

Examples

```
## Generate S2 summary data
set.seed(1)
n <- 100
x <- stats::rlnorm(n, 2.5, 1)
quants <- stats::quantile(x, probs = c(0.25, 0.5, 0.75))
obs.mean <- mean(x)
obs.sd <- stats::sd(x)

## Estimate the sample mean and standard deviation using the QE method
qe.mean.sd(q1.val = quants[1], med.val = quants[2], q3.val = quants[3],
           n = n)
```

summary.qe.mean.sd *Summary method for objects of class "qe.mean.sd"*

Description

Summary method for objects of class "qe.mean.sd".

Usage

```
## S3 method for class 'qe.mean.sd'
summary(object, digits = 5, ...)
```

Arguments

object	object of class "qe.mean.sd".
digits	integer specifying the number of decimal places.
...	other arguments.

Value

A 5×3 matrix with columns for the estimated sample mean, estimated standard deviation, and sum of squares (of the objective function used in [qe.fit](#)) under each candidate distribution.

See Also

[qe.mean.sd](#)

Examples

```
## Generate S2 summary data
set.seed(1)
n <- 100
x <- stats::rlnorm(n, 2.5, 1)
quants <- stats::quantile(x, probs = c(0.25, 0.5, 0.75))
obs.mean <- mean(x)
obs.sd <- stats::sd(x)

## Estimate the sample mean and standard deviation using the BC method
res <- qe.mean.sd(q1.val = quants[1], med.val = quants[2],
  q3.val = quants[3], n = n)
summary(res)
```

Index

bc.mean.sd, [2](#), [6](#)

optim, [10](#)

par, [5](#)

plot.qe.fit, [4](#), [10](#)

points, [5](#)

print.bc.mean.sd, [3](#), [6](#)

print.qe.fit, [7](#), [10](#)

print.qe.mean.sd, [8](#)

qe.fit, [4](#), [5](#), [7](#), [9](#), [11](#), [13](#)

qe.mean.sd, [8](#), [11](#), [13](#)

summary.qe.mean.sd, [12](#)