

Package ‘concurve’

July 10, 2019

Type Package

Date 2019-07-10

Title Computes and Plots Consonance (Confidence) Intervals, P-Values, and S-Values to Form Consonance and Surprisal Functions

Version 2.0

Maintainer Zad R. Chow <zad@lesslikely.com>

Description Allows one to compute consonance (confidence) intervals for various statistical tests along with their corresponding P-values and S-values. The intervals can be plotted to create consonance and surprisal functions allowing one to see what effect sizes are compatible with the test model at various consonance levels rather than being limited to one interval estimate such as 95%. These methods are discussed by Poole C. (1987) <doi:10.2105/AJPH.77.2.195>, Schweder T, Hjort NL. (2002) <doi:10.1111/1467-9469.00285>, Singh K, Xie M, Strawderman WE. (2007) <arXiv:0708.0976>, Rothman KJ, Greenland S, Lash TL. (2008, ISBN:9781451190052), Amrhein V, Trafimow D, Greenland S. (2019) <doi:10.1080/00031305.2018.1543137>, and Greenland S. (2019) <doi:10.1080/00031305.2018.1529625>.

Imports parallel, ggplot2, metafor, dplyr, tibble, survival, survminer, scales

Suggests testthat, knitr, covr

License GPL-3 | file LICENSE

URL <https://data.lesslikely.com/concurve/>,
<https://github.com/Zadchow/concurve>, <https://lesslikely.com/>

BugReports <https://github.com/Zadchow/concurve/issues>

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

X-schema.org-keywords confidence, compatibility, consonance, surprisal, interval, function, curve

NeedsCompilation no

Author Zad R. Chow [aut, cre] (<<https://orcid.org/0000-0003-1545-8199>>),
Andrew D. Vigotsky [aut] (<<https://orcid.org/0000-0003-3166-0688>>)

Repository CRAN

Date/Publication 2019-07-10 07:40:03 UTC

R topics documented:

curve_corr	2
curve_gen	3
curve_mean	4
curve_meta	5
curve_rev	7
curve_surv	8
ggconcurve	9
plot_concurve	11

Index **13**

curve_corr	<i>Computes consonance intervals for correlations</i>
------------	---

Description

Computes consonance intervals to produce P- and S-value functions for correlational analyses using the cor.test function in base R and places the interval limits for each interval level into a data frame along with the corresponding p-values and s-values.

Usage

```
curve_corr(x, y, alternative, method, steps = 10000)
```

Arguments

x	A vector that contains the data for one of the variables that will be analyzed for correlational analysis.
y	A vector that contains the data for one of the variables that will be analyzed for correlational analysis.
alternative	Indicates the alternative hypothesis and must be one of "two.sided", "greater" or "less". You can specify just the initial letter. "greater" corresponds to positive association, "less" to negative association.
method	A character string indicating which correlation coefficient is to be used for the test. One of "pearson", "kendall", or "spearman", can be abbreviated.
steps	Indicates how many consonance intervals are to be calculated at various levels. For example, setting this to 100 will produce 100 consonance intervals from 0 to 100. Setting this to 10000 will produce more consonance levels. By default, it is set to 1000. Increasing the number substantially is not recommended as it will take longer to produce all the intervals and store them into a dataframe.

References

- Poole C. Beyond the confidence interval. *Am J Public Health*. 1987;77(2):195-199.
- Sullivan KM, Foster DA. Use of the confidence interval function. *Epidemiology*. 1990;1(1):39-42.
- Rothman KJ, Greenland S, Lash TL, Others. *Modern epidemiology*. 2008.

Examples

```
GroupA <- rnorm(50)
GroupB <- rnorm(50)

joe <- curve_corr(x = GroupA, y = GroupB,
  alternative = "two.sided", method = "pearson")

tibble::tibble(joe)
```

curve_gen

Computes consonance intervals for linear models

Description

Computes thousands of consonance (confidence) intervals for the chosen parameter in the selected model(ANOVA, ANCOVA, regression, logistic regression) and places the interval limits for each interval level into a data frame along with the corresponding p-values and s-values.

Usage

```
curve_gen(model, var, method = "default", replicates = 1000, steps = 10000)
```

Arguments

model	The statistical model of interest(ANOVA, regression, logistic regression) is to be indicated here.
var	The variable of interest from the model (coefficients, intercept) for which the intervals are to be produced.
method	Chooses the method to be used to calculate the consonance intervals. There are currently four methods: "default", "wald", "lm", and "boot". The "default" method uses the profile likelihood method to compute intervals and can be used for models created by the 'lm' function. The "wald" method is typically what most people are familiar with when computing intervals based on the calculated standard error. The "lm" method allows this function to be used for specific scenarios like logistic regression and the 'glm' function. The "boot" method allows for bootstrapping at certain levels.
replicates	Indicates how many bootstrap replicates are to be performed if bootstrapping is enabled as a method.

steps Indicates how many consonance intervals are to be calculated at various levels. For example, setting this to 100 will produce 100 compatibility intervals from 0 to 100. Setting this to 10000 will produce more consonance levels. By default, it is set to 1000. Increasing the number substantially is not recommended as it will take longer to produce all the intervals and store them into a dataframe.

References

Poole C. Beyond the confidence interval. *Am J Public Health*. 1987;77(2):195-199.
 Sullivan KM, Foster DA. Use of the confidence interval function. *Epidemiology*. 1990;1(1):39-42.
 Rothman KJ, Greenland S, Lash TL, Others. *Modern epidemiology*. 2008.

Examples

```
# Simulate random data

GroupA <- rnorm(50)
GroupB <- rnorm(50)

RandomData <- data.frame(GroupA, GroupB)

rob <- glm(GroupA ~ GroupB, data = RandomData)
bob <- curve_gen(rob, "GroupB", method = "lm")

tibble::tibble(bob)
```

curve_mean

Computes consonance intervals for mean differences

Description

Computes thousands of consonance (confidence) intervals for the chosen parameter in a statistical test that compares means and places the interval limits for each interval level into a data frame along with the corresponding p-values and s-values.

Usage

```
curve_mean(x, y, data, paired = F, method = "default",
  replicates = 1000, steps = 10000)
```

Arguments

x Variable that contains the data for the first group being compared.
y Variable that contains the data for the second group being compared.
data Data frame from which the variables are being extracted from.

paired	Indicates whether the statistical test is a paired difference test. By default, it is set to "F", which means the function will be an unpaired statistical test comparing two independent groups. Inserting "paired" will change the test to a paired difference test.
method	By default this is turned off (set to "default"), but allows for bootstrapping if "boot" is inserted into the function call.
replicates	Indicates how many bootstrap replicates are to be performed if bootstrapping is enabled as a method.
steps	Indicates how many consonance intervals are to be calculated at various levels. For example, setting this to 100 will produce 100 consonance intervals from 0 to 100. Setting this to 10000 will produce more consonance levels. By default, it is set to 1000. Increasing the number substantially is not recommended as it will take longer to produce all the intervals and store them into a dataframe.

References

- Poole C. Beyond the confidence interval. *Am J Public Health*. 1987;77(2):195-199.
- Sullivan KM, Foster DA. Use of the confidence interval function. *Epidemiology*. 1990;1(1):39-42.
- Rothman KJ, Greenland S, Lash TL, Others. *Modern epidemiology*. 2008.

Examples

```
# Simulate random data

GroupA <- runif(100, min = 0, max = 100)
GroupB <- runif(100, min = 0, max = 100)

RandomData <- data.frame(GroupA, GroupB)

bob <- curve_mean(GroupA, GroupB, RandomData)

tibble::tibble(bob)
```

curve_meta

Computes consonance intervals for meta-analysis data

Description

Computes thousands of consonance (confidence) intervals for the chosen parameter in the meta-analysis done by the metafor package and places the interval limits for each interval level into a data frame along with the corresponding p-values and s-values.

Usage

```
curve_meta(x, measure = "default", steps = 10000)
```

Arguments

x	Object where the meta-analysis parameters are stored, typically a list produced by 'metafor'
measure	Indicates whether the object has a log transformation or is normal/default. The default setting is "default". If the measure is set to "ratio", it will take logarithmically transformed values and convert them back to normal values in the dataframe. This is typically a setting used for binary outcomes such as risk ratios, hazard ratios, and odds ratios.
steps	Indicates how many consonance intervals are to be calculated at various levels. For example, setting this to 100 will produce 100 consonance intervals from 0 to 100. Setting this to 10000 will produce more consonance levels. By default, it is set to 1000. Increasing the number substantially is not recommended as it will take longer to produce all the intervals and store them into a dataframe.

References

- Viechtbauer W. Conducting meta-analyses in R with the metafor package. *J Stat Softw.* 2010;36(3). <https://www.jstatsoft.org/article/view/v036i03/v36i03.pdf>.
- Poole C. Beyond the confidence interval. *Am J Public Health.* 1987;77(2):195-199.
- Sullivan KM, Foster DA. Use of the confidence interval function. *Epidemiology.* 1990;1(1):39-42.
- Rothman KJ, Greenland S, Lash TL, Others. *Modern epidemiology.* 2008.

Examples

```
# Simulate random data for two groups in two studies

GroupAData <- runif(20, min = 0, max = 100)
GroupAMean <- round(mean(GroupAData), digits = 2)
GroupASD <- round(sd(GroupAData), digits = 2)

GroupBData <- runif(20, min = 0, max = 100)
GroupBMean <- round(mean(GroupBData), digits = 2)
GroupBSD <- round(sd(GroupBData), digits = 2)

GroupCData <- runif(20, min = 0, max = 100)
GroupCMean <- round(mean(GroupCData), digits = 2)
GroupCSD <- round(sd(GroupCData), digits = 2)

GroupDData <- runif(20, min = 0, max = 100)
GroupDMean <- round(mean(GroupDData), digits = 2)
GroupDSD <- round(sd(GroupDData), digits = 2)

# Combine the data

StudyName <- c("Study1", "Study2")
MeanTreatment <- c(GroupAMean, GroupCMean)
MeanControl <- c(GroupBMean, GroupDMean)
SDTreatment <- c(GroupASD, GroupCSD)
```

```

SDControl <- c(GroupBSD, GroupDSD)
NTreatment <- c(20, 20)
NControl <- c(20, 20)

metadf <- data.frame(
  StudyName, MeanTreatment, MeanControl,
  SDTreatment, SDControl, NTreatment, NControl
)

# Use metafor to calculate the standardized mean difference

library(metafor)

dat <- escalc(
  measure = "SMD", m1i = MeanTreatment, sd1i = SDTreatment, n1i = NTreatment,
  m2i = MeanControl, sd2i = SDControl, n2i = NControl, data = metadf
)

# Pool the data using a particular method. Here "FE" is the fixed-effects model

res <- rma(yi, vi, data = dat, slab = paste(StudyName, sep = ", "), method = "FE", digits = 2)

# Calculate the intervals using the metainterval function

metaf <- curve_meta(res)

tibble::tibble(metaf)

```

curve_rev	<i>Reverse engineer consonance and surprisal functions from confidence limits and point estimates</i>
-----------	---

Description

Using the confidence limits and point estimates from a dataset, one can use these estimates to compute thousands of consonance intervals and graph the intervals to form a consonance and surprisal function.

Usage

```
curve_rev(point, LL, UL, measure = "default")
```

Arguments

point	The point estimate from an analysis. Ex: 1.20
LL	The lower confidence limit from an analysis Ex: 1.0
UL	The upper confidence limit from an analysis Ex: 1.4
measure	The type of data being used. If they involve mean differences, then the "default" option should be used, which is also the default setting. If the data are ratios, then the "ratio" option should be used.

References

- Poole C. Beyond the confidence interval. *Am J Public Health*. 1987;77(2):195-199.
- Sullivan KM, Foster DA. Use of the confidence interval function. *Epidemiology*. 1990;1(1):39-42.
- Rothman KJ, Greenland S, Lash TL, Others. *Modern epidemiology*. 2008.

Examples

```
# From a real published study. Point estimate of the result was hazard ratio of 1.61 and
# lower bound of the interval is 0.997 while upper bound of the interval is 2.59.

df <- curve_rev(point = 1.61, LL = 0.997, UL = 2.59, measure = "ratio")

tibble::tibble(df)
```

curve_surv

Produce Consonance Intervals for Survival Data

Description

Computes thousands of consonance (confidence) intervals for the chosen parameter in the Cox model computed by the 'survival' package and places the interval limits for each interval level into a data frame along with the corresponding p-value and s-value.

Usage

```
curve_surv(data, x, steps = 10000)
```

Arguments

- | | |
|-------|---|
| data | Object where the Cox model is stored, typically a list produced by the 'survival' package. |
| x | Predictor of interest within the survival model for which the consonance intervals should be computed. |
| steps | Indicates how many consonance intervals are to be calculated at various levels. For example, setting this to 100 will produce 100 consonance intervals from 0 to 100. Setting this to 10000 will produce more consonance levels. By default, it is set to 1000. Increasing the number substantially is not recommended as it will take longer to produce all the intervals and store them into a dataframe. |

References

- Poole C. Beyond the confidence interval. *Am J Public Health*. 1987;77(2):195-199.
- Sullivan KM, Foster DA. Use of the confidence interval function. *Epidemiology*. 1990;1(1):39-42.
- Rothman KJ, Greenland S, Lash TL, Others. *Modern epidemiology*. 2008.

ggconcurve	<i>Plots the P-Value (Consonance) and S-value (Surprisal) Function via ggplot2</i>
------------	--

Description

Takes the dataframe produced by the interval functions and plots the p-values/s-values, consonance (confidence) levels, and the interval estimates to produce a p-value/s-value function using ggplot2 graphics.

Usage

```
ggconcurve(type, data, measure, nullvalue, position,
           title, subtitle, caption, xaxis, yaxis, color, fill)
```

Arguments

type	Choose whether to plot a "consonance" function or a "surprisal" function. The default option is set to "consonance". The type must be set in quotes, for example <code>ggconcurve(type = "surprisal")</code> or <code>ggconcurve(type = "consonance")</code> .
data	The dataframe produced by one of the interval functions in which the intervals are stored.
measure	Indicates whether the object has a log transformation or is normal/default. The default setting is "default". If the measure is set to "ratio", it will take logarithmically transformed values and convert them back to normal values in the dataframe. This is typically a setting used for binary outcomes and their measures such as risk ratios, hazard ratios, and odds ratios.
nullvalue	Indicates whether the null value for the measure should be plotted. By default, it is set to "absent", meaning it will not be plotted as a vertical line. Changing this to "present", will plot a vertical line at 0 when the measure is set to "default" and a vertical line at 1 when the measure is set to "ratio". For example, <code>ggconcurve(type = "consonance", data = df, measure = "ratio", nullvalue = "present")</code> . This feature is not yet available for surprisal functions.
position	Determines the orientation of the P-value (consonance) function. By default, it is set to "pyramid", meaning the p-value function will stand right side up, like a pyramid. However, it can also be inverted via the option "inverted". This will also change the sequence of the y-axes to match the orientation. This can be set as such, <code>ggconcurve(type = "consonance", data = df, position = "inverted")</code>
title	A custom title for the graph. By default, it is set to "Consonance Function". In order to set a title, it must be in quotes. For example, <code>ggconcurve(x, title = "Custom Title")</code> .
subtitle	A custom subtitle for the graph. By default, it is set to "The function contains consonance/confidence intervals at every level and the P-values." In order to set a subtitle, it must be in quotes. For example, <code>ggconcurve(x, subtitle = "Custom Subtitle")</code> .

caption	A custom caption for the graph. By default, it is set to "Produced with the concurve R package.". In order to set a caption, it must be in quotes. For example, <code>ggconcurve(data, caption = "Custom Caption")</code> .
xaxis	A custom x-axis title for the graph. By default, it is set to "Range of Values. In order to set a x-axis title, it must be in quotes. For example, <code>ggconcurve(data, xaxis = "Hazard Ratio")</code> .
yaxis	A custom y-axis title for the graph. By default, it is set to "Consonance Level". In order to set a y-axis title, it must be in quotes. For example, <code>ggconcurve(data, yaxis = "Confidence Level")</code> .
color	Item that allows the user to choose the color of the points and the ribbons in the graph. By default, it is set to <code>color = "#555555"</code> . The inputs must be in quotes. For example, <code>ggconcurve(data, color = "#333333")</code> .
fill	Item that allows the user to choose the color of the ribbons in the graph. By default, it is set to <code>fill = "#239a98"</code> . The inputs must be in quotes. For example, <code>ggconcurve(data, fill = "#333333")</code> .

Value

Plot with intervals at every consonance level graphed with their corresponding p-values and compatibility levels.

References

- Poole C. Beyond the confidence interval. *Am J Public Health*. 1987;77(2):195-199.
- Sullivan KM, Foster DA. Use of the confidence interval function. *Epidemiology*. 1990;1(1):39-42.
- Rothman KJ, Greenland S, Lash TL, Others. *Modern epidemiology*. 2008.

Examples

```
# Simulate random data

GroupA <- rnorm(50)
GroupB <- rnorm(50)

RandomData <- data.frame(GroupA, GroupB)
RandomModel <- lm(GroupA ~ GroupB, data = RandomData)

intervalsdf <- curve_gen(RandomModel, "GroupB")

p <- ggconcurve(type = "consonance", data = intervalsdf, nullvalue = "present")
p

r <- ggconcurve(type = "consonance", data = intervalsdf, position = "inverted")
r

s <- ggconcurve(type = "surprisal", data = intervalsdf)
s
```

plot_concurve	<i>Plots the P- (Consonance) and S-Value (Surprisal) Functions using base R graphics.</i>
---------------	---

Description

Takes the dataframe produced by the interval functions and plots the p-values, s-values, consonance (confidence) levels, and the interval estimates to produce p- and s-value functions using base R graphics.

Usage

```
plot_concurve(type, data, title, xlab, ylab1, ylab2)
```

Arguments

type	Choose whether to plot a "consonance" function or a "surprisal" function. The default option is set to "consonance". The type must be set in quotes, for example <code>plot_concurve(type = "surprisal")</code> or <code>plot_concurve(type = "consonance")</code> .
data	Dataframe where the results from a <code>curve_</code> function is stored.
title	The title for the graph. By default, it is set to "Consonance Function". In order to set a title, it must be in quotes. For example, <code>plot_concurve(type = "consonance", data = data, title = "Custom Title")</code> .
xlab	The label for the x-axis. By default, it is set to "Theta.". In order to set a label, it must be in quotes. For example, <code>plot_concurve(type = "consonance", data = data, xlab = "Custom Caption")</code> .
ylab1	A label for the y-axis on the left side of the graph. By default, it is set to "P-value." In order to set a custom y-axis label, it must be in quotes. For example, <code>plot_concurve(type = "consonance", data = data, ylab1= "Custom y-axis title")</code> .
ylab2	A label for the y-axis on the right side of the graph. By default, it is set to "Confidence Level. "" In order to set a custom y-axis label, it must be in quotes. For example, <code>plot_concurve(type = "consonance", data = data, ylab2= "Custom y-axis title")</code> .

Value

Plot with intervals at every consonance level graphed with their corresponding p- and s-values.

References

Amrhein V, Trafimow D, Greenland S. Inferential Statistics as Descriptive Statistics: There Is No Replication Crisis If We Don't Expect Replication. *Am Stat*; 2018.

Greenland S. Valid P-values behave exactly as they should: Some misleading criticisms of P-values and their resolution with S-values. *Am Stat*. 2018;18(136).

Greenland S. The unconditional information in P-values, and its refutational interpretation via S-values. 2018.

Shannon CE. A Mathematical Theory of Communication. Bell System Technical Journal. 1948;27(3):379-423. doi:10.1002/j.1538-7305.1948.tb01338.x

Poole C. Beyond the confidence interval. Am J Public Health. 1987;77(2):195-199.

Sullivan KM, Foster DA. Use of the confidence interval function. Epidemiology. 1990;1(1):39-42.

Rothman KJ, Greenland S, Lash TL, Others. Modern epidemiology. 2008.

Examples

```
# Simulate random data

GroupA <- rnorm(50)
GroupB <- rnorm(50)

RandomData <- data.frame(GroupA, GroupB)
RandomModel <- lm(GroupA ~ GroupB, data = RandomData)

intervalsdf <- curve_gen(RandomModel, "GroupB")

s <- plot_concurve(type = "consonance", data = intervalsdf)
s
```

Index

[curve_corr](#), 2
[curve_gen](#), 3
[curve_mean](#), 4
[curve_meta](#), 5
[curve_rev](#), 7
[curve_surv](#), 8

[ggconcurve](#), 9

[plot_concurve](#), 11