

Package ‘concurve’

May 15, 2019

Type Package

Date 2019-05-10

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

Version 1.0.6

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 ggplot2, metafor

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 2019-05-14 22:30:02 UTC

R topics documented:

genintervals	2
meanintervals	3
metaintervals	4
plotpint	6
plotsint	7
Index	9

genintervals	<i>Produce Consonance Intervals for Linear Models</i>
--------------	-------------------------------------------------------

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-value and s-value.

Usage

```
genintervals(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<-genintervals(rob, "GroupB", method="lm")
```

meanintervals *Produce 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-value and s-value.

Usage

```
meanintervals(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<-meanintervals(GroupA, GroupB, RandomData)
```

metaintervals *Produce 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-value and s-value.

Usage

```
metaintervals(x, measure = "norm", 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. If the measure is set to "log", 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 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<-escalr(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<-metaintervals(res)

```

plotpint

Plot the P-Value Function

Description

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

Usage

```
plotpint(x)
```

Arguments

`x` The dataframe produced by one of the interval functions in which the intervals are stored.

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<-genintervals(RandomModel, "GroupB")  
  
p<-plotpint(intervalsdf)  
p
```

plotsint

Plot the S-Value Function

Description

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

Usage

```
plotsint(x)
```

Arguments

`x` The dataframe produced by one of the interval functions in which the intervals are stored.

Value

Plot with intervals at every consonance level graphed with their corresponding 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<-genintervals(RandomModel, "GroupB")

s<-plotsint(intervalsdf)
s
```


Index

genintervals, [2](#)

meanintervals, [3](#)

metaintervals, [4](#)

plotpint, [6](#)

plotsint, [7](#)