Package 'Bvalue'

October 12, 2022

Type Package

Title B-Value and Empirical Equivalence Bound

Version 1.0

Date 2020-01-08

Author Yi Zhao <zhaoyi1026@gmail.com>
Brian Caffo <bcaffo@gmail.com>
Joshua Ewen <ewen@kennedykrieger.org>

Maintainer Yi Zhao <zhaoyi1026@gmail.com>

Description Calculates B-value and empirical equivalence bound. B-value is defined as the maximum magnitude of a confidence interval; and the empirical equivalence bound is the minimum B-value at a certain level. A new two-stage procedure for hypothesis testing is proposed, where the first stage is conventional hypothesis testing and the second is an equivalence testing procedure using the introduced empirical equivalence bound. See Zhao et al. (2019) ``B-Value and Empirical Equivalence Bound: A New Procedure of Hypothesis Testing" <arXiv:1912.13084> for details.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2020-01-23 16:50:02 UTC

R topics documented:

Bvalue-package *B-Value and Empirical Equivalence Bound*

Description

Bvalue package calculates B-value and empirical equivalence bound. B-value is defined as the maximum magnitude of a confidence interval; and the empirical equivalence bound is the minimum B-value at a certain level. A new two-stage procedure for hypothesis testing is proposed, where the first stage is conventional hypothesis testing and the second is an equivalence testing procedure using the introduced empirical equivalence bound.

Author(s)

Yi Zhao, Indiana University, <zhaoyi1026@gmail.com>

Brian Caffo, Johns Hopkins University, <bcaffo@gmail.com>

Joshua Ewen, Kennedy Krieger Institute and Johns Hopkins University, <ewen@kennedykrieger.org>

Maintainer: Yi Zhao <zhaoyi1026@gmail.com>

References

Zhao et al. (2019) *"B-Value and Empirical Equivalence Bound: A New Procedure of Hypothesis Testing"* <arXiv:1912.13084>

EEB *The Empirical Equivalence Bound*

Description

This function calculates the Empirical Equivalence Bound (EEB) at given level.

Usage

```
EEB(beta, nu, delta = 0, S = 1, alpha = 0.05,
  type = c("marginal", "cond_NRej", "cond_Rej"),
  tol = 1e-04, max.itr = 5000)
```

Arguments

beta	a numeric between 0 and 1. This is the beta value in the EEB definition, see details.
nu	an integer, the degrees of freedom in the conventional t-test.
delta	a numeric value. Considering testing for difference of two population means, delta is the null value of the difference. Default is 0.
S	a numeric value. The standard error in the conventional t-test.

`alpha`	a numeric between 0 and 1. The Type I error rate aiming to control in the conventional t-test.
`type`	a character to specify the type of EEB to be calculated. `type = "marginal"` gives the marginal EEB; `type = "cond_NRej"` gives the EEB under the condition that one cannot reject the first-stage conventional t-test; `type = "cond_Rej"` gives the EEB under the condition that the first-stage conventional t-test is rejected.
`tol`	a numeric value of convergence tolerance.
`max.itr`	an integer, the maximum number of iterations.

Details

Consider a two-sample t-test setting with hypotheses

$$H_0 : \delta = 0 \quad \leftrightarrow \quad H_1 : \delta \neq 0,$$

where $\delta = \mu_1 - \mu_2$ is the difference of two population means. If the testing result is failure to reject the null, one cannot directly conclude equivalence of the two groups. In this case, an equivalence test is suggested by testing the hypotheses

$$H_3 : |\delta| \geq \Delta \quad \leftrightarrow \quad H_4 : |\delta| < \Delta,$$

where Δ is a pre-specified equivalence bound. A $100(1-2\alpha)\%$ confidence interval is formulated, denoted as $[L, U]$, to test for equivalence, where

$$L = \hat{\delta} - t_{\nu,1-\alpha} S, \quad U = \hat{\delta} + t_{\nu,1-\alpha} S,$$

$\hat{\delta}$ is the estimate of δ, $t_{\nu,1-\alpha}$ is the $100(1-\alpha)\%$ quantile of a t-distribution with degrees of freedom ν, and S is the standard error. We define the B-value as

$$B = \max\{|L|, |U|\},$$

and the Empirical Equivalence Bound (EEB) is defined as

$$\mathbf{EEB}_\alpha(\beta|C) = \inf_{b \in [0,\infty]} \{b : F_B(b|C, H_0) \geq \beta\},$$

where $\beta \in (0, 1)$ is a pre-specified level; C denotes the status of the hypothesis test, which takes value of empty set (`type = "marginal"`), cannot reject H_0 (`type = "cond_NRej"`), and reject H_0 (`type = "cond_Rej"`); and $F_B(\cdot|C, H_0)$ is the conditional cumulative distribution function of B-value.

Value

Gives the Empirical Equivalence Bound value.

Author(s)

Yi Zhao, Indiana University, <zhaoyi1026@gmail.com>

Brian Caffo, Johns Hopkins University, <bcaffo@gmail.com>

Joshua Ewen, Kennedy Krieger Institute and Johns Hopkins University, <ewen@kennedykrieger.org>

References

Zhao et al. (2019) *"B-Value and Empirical Equivalence Bound: A New Procedure of Hypothesis Testing"* <arXiv:1912.13084>

See Also

pB

Examples

```
########################################
# R Plant Growth Data

data("PlantGrowth")

PlantGrowth$group
comb.mat<-cbind(c(1,2),c(1,3))
comb.name<-paste0(levels(PlantGrowth$group)[2:3],"-",levels(PlantGrowth$group)[1])
colnames(comb.mat)<-comb.name

alpha<-0.05
# consider a series of beta values
beta.vec<-c(0.5,0.75,0.8,0.9,0.95,0.99)

# two-stage hypothesis testing
# Stage I: conventional two-sample t-test
# Stage II: based on Stage I result to calculate EEB
stat<-matrix(NA,ncol(comb.mat),10+length(beta.vec)*3)
colnames(stat)<-c("delta","LB0","UB0","LB","UB","S","nu","tv","statistic","pvalue",
                  paste0(rep(c("EEB","EEB_NRej","EEB_Rej"),
                    each=length(beta.vec)),"_beta",rep(beta.vec,3)))
rownames(stat)<-comb.name
for(kk in 1:ncol(comb.mat))
{
  x2<-PlantGrowth$weight[which(as.numeric(PlantGrowth$group)==comb.mat[1,kk])]
  x1<-PlantGrowth$weight[which(as.numeric(PlantGrowth$group)==comb.mat[2,kk])]

  n1<-length(x1)
  n2<-length(x2)

  S<-sqrt((sum((x1-mean(x1))^2)+sum((x2-mean(x2))^2))/(n1+n2-2))*sqrt(1/n1+1/n2)

  nu<-n1+n2-2

  tv<-qt(1-alpha,df=nu)
  tv2<-qt(1-alpha/2,df=nu)

  stat[kk,1]<-mean(x1)-mean(x2)         # delta estimate
  stat[kk,2]<-stat[kk,1]-tv2*S          # (1-alpha)% CI lower bound
  stat[kk,3]<-stat[kk,1]+tv2*S          # (1-alpha)% CI upper bound
  stat[kk,4]<-stat[kk,1]-tv*S           # (1-2alpha)% CI lower bound
  stat[kk,5]<-stat[kk,1]+tv*S           # (1-2alpha)% CI upper bound
```

```
  stat[kk,6]<-S                         # standard error
  stat[kk,7]<-nu                        # degrees of freedom in the first-stage t-test
  stat[kk,8]<-tv
  stat[kk,9]<-stat[kk,1]/S              # test-statistic in the first-stage t-test
  stat[kk,10]<-(1-pt(abs(stat[kk,9]),df=nu))*2    # p-value in the first-stage t-test

  # marginal EEB
  stat[kk,11:(11+length(beta.vec)-1)]<-
    apply(as.matrix(beta.vec),1,
      function(x){return(EEB(beta=x,nu=nu,delta=0,S=S,alpha=alpha,type="marginal"))})

  # conditional on not rejection
  if(stat[kk,2]*stat[kk,3]<0)
  {
    stat[kk,(11+length(beta.vec)):(11+length(beta.vec)*2-1)]<-
      apply(as.matrix(beta.vec),1,
        function(x){return(EEB(beta=x,nu=nu,delta=0,S=S,alpha=alpha,type="cond_NRej"))})
  }

  # conditional on rejection
  if(stat[kk,2]*stat[kk,3]>0)
  {
    stat[kk,(11+length(beta.vec)*2):(11+length(beta.vec)*3-1)]<-
      apply(as.matrix(beta.vec),1,
        function(x){return(EEB(beta=x,nu=nu,delta=0,S=S,alpha=alpha,type="cond_Rej"))})
  }
}
print(data.frame(t(stat)))

cc<-colors()[c(24,136,564,500,469,50,200,460,17,2,652,90,8,146,464,52,2)]
beta.lgd<-sapply(1:length(beta.vec),
                 function(i){as.expression(substitute(beta==x,
                  list(x=format(beta.vec[i],digit=2,nsmall=2))))})

# Boxplot of data
oldpar<-par(no.readonly=TRUE)
par(mar=c(5,5,1,1))
boxplot(weight~group,data=PlantGrowth,ylab="Dried weight of plants",col=cc[c(3,2,2)],
        pch=19,notch=TRUE,varwidth=TRUE,cex.lab=1.25,cex.axis=1.25)
par(oldpar)

# Comparing t-test CI and equivalence CI using the EEB
# trt1-ctrl
kk<-1
oldpar<-par(no.readonly=TRUE)
par(mar=c(4,5,3,3))
par(oma=c(2.5,0,0,0))
plot(range(c(stat[kk,c(1,2,3,4,5,11:ncol(stat))],-stat[kk,c(11:ncol(stat))]),na.rm=TRUE),
  c(0.5,length(beta.vec)+0.5),type="n",
  xlab="",ylab=expression(beta),yaxt="n",main=comb.name[kk],
  cex.lab=1.25,cex.axis=1.25,cex.main=1.25)
axis(2,at=1:length(beta.vec),labels=FALSE)
text(rep(par("usr")[1]-(par("usr")[2]-par("usr")[1])/100*2,length(beta.vec)),1:length(beta.vec),
```

```
  labels=format(beta.vec,nsmall=2,digits=2),srt=0,adj=c(1,0.5),xpd=TRUE,cex=1.25)
for(ll in 1:length(beta.vec))
{
  if(stat[kk,2]*stat[kk,3]<0)
  {
   lines(c(-stat[kk,(10+length(beta.vec))+ll],stat[kk,(10+length(beta.vec))+ll]),rep(ll,2),
      lty=1,lwd=3,col=cc[1])
    points(0,ll,pch=15,cex=1.5,col=cc[1])
    text(-stat[kk,(10+length(beta.vec))+ll],ll,labels="[",cex=1.5,col=cc[1])
    text(stat[kk,(10+length(beta.vec))+ll],ll,labels="]",cex=1.5,col=cc[1])
  }
  if(stat[kk,2]*stat[kk,3]>0)
  {
   lines(c(-stat[kk,(10+length(beta.vec)*2)+ll],stat[kk,(10+length(beta.vec)*2)+ll]),rep(ll,2),
      lty=1,lwd=3,col=cc[1])
    points(0,ll,pch=15,cex=1.5,col=cc[1])
    text(-stat[kk,(10+length(beta.vec)*2)+ll],ll,labels="[",cex=1.5,col=cc[1])
    text(stat[kk,(10+length(beta.vec)*2)+ll],ll,labels="]",cex=1.5,col=cc[1])
  }

  lines(stat[kk,c(4,5)],rep(ll,2),lty=1,lwd=3,col=cc[2])
  points(stat[kk,1],ll,pch=19,cex=1.5,col=cc[2])
  text(stat[kk,4],ll,labels="[",cex=1.5,col=cc[2])
  text(stat[kk,5],ll,labels="]",cex=1.5,col=cc[2])
}
par(fig=c(0,1,0,1),oma=c(0,0,0,0),mar=c(0,2,0,2),new=TRUE)
plot(0,0,type="n",bty="n",xaxt="n",yaxt="n")
legend("bottom",legend=c("confidence interval","equivalence interval"),xpd=TRUE,horiz=TRUE,
       inset=c(0,0),pch=c(19,15),col=cc[c(2,1)],lty=1,lwd=2,cex=1.5,bty="n")
par(oldpar)

# trt2-ctrl
kk<-2
oldpar<-par(no.readonly=TRUE)
par(mar=c(4,5,3,3))
par(oma=c(2.5,0,0,0))
plot(range(c(stat[kk,c(1,2,3,4,5,11:ncol(stat))],-stat[kk,c(11:ncol(stat))]),na.rm=TRUE),
  c(0.5,length(beta.vec)+0.5),type="n",
  xlab="",ylab=expression(beta),yaxt="n",main=comb.name[kk],
  cex.lab=1.25,cex.axis=1.25,cex.main=1.25)
axis(2,at=1:length(beta.vec),labels=FALSE)
text(rep(par("usr")[1]-(par("usr")[2]-par("usr")[1])/100*2,length(beta.vec)),1:length(beta.vec),
     labels=format(beta.vec,nsmall=2,digits=2),srt=0,adj=c(1,0.5),xpd=TRUE,cex=1.25)
for(ll in 1:length(beta.vec))
{
  if(stat[kk,2]*stat[kk,3]<0)
  {
   lines(c(-stat[kk,(10+length(beta.vec))+ll],stat[kk,(10+length(beta.vec))+ll]),rep(ll,2),
      lty=1,lwd=3,col=cc[1])
    points(0,ll,pch=15,cex=1.5,col=cc[1])
    text(-stat[kk,(10+length(beta.vec))+ll],ll,labels="[",cex=1.5,col=cc[1])
    text(stat[kk,(10+length(beta.vec))+ll],ll,labels="]",cex=1.5,col=cc[1])
  }
```

```
  if(stat[kk,2]*stat[kk,3]>0)
  {
   lines(c(-stat[kk,(10+length(beta.vec)*2)+ll],stat[kk,(10+length(beta.vec)*2)+ll]),rep(ll,2),
      lty=1,lwd=3,col=cc[1])
    points(0,ll,pch=15,cex=1.5,col=cc[1])
    text(-stat[kk,(10+length(beta.vec)*2)+ll],ll,labels="[",cex=1.5,col=cc[1])
    text(stat[kk,(10+length(beta.vec)*2)+ll],ll,labels="]",cex=1.5,col=cc[1])
  }

  lines(stat[kk,c(4,5)],rep(ll,2),lty=1,lwd=3,col=cc[2])
  points(stat[kk,1],ll,pch=19,cex=1.5,col=cc[2])
  text(stat[kk,4],ll,labels="[",cex=1.5,col=cc[2])
  text(stat[kk,5],ll,labels="]",cex=1.5,col=cc[2])
}
par(fig=c(0,1,0,1),oma=c(0,0,0,0),mar=c(0,2,0,2),new=TRUE)
plot(0,0,type="n",bty="n",xaxt="n",yaxt="n")
legend("bottom",legend=c("confidence interval","equivalence interval"),xpd=TRUE,horiz=TRUE,
       inset=c(0,0),pch=c(19,15),col=cc[c(2,1)],lty=1,lwd=2,cex=1.5,bty="n")
par(oldpar)
########################################
```

---

pB *The B-Value Distribution*

Description

This function gives the cumulative distribution function of the B-value.

Usage

```
pB(b, nu, delta = 0, S = 1, alpha = 0.05, type = c("marginal", "cond_NRej", "cond_Rej"))
```

Arguments

b	vector of quantiles
nu	an integer, the degrees of freedom in the conventional t-test.
delta	a numeric value. Considering testing for difference of two population means, delta is the null value of the difference. Default is 0.
S	a numeric value. The standard error in the conventional t-test.
alpha	a numeric between 0 and 1. The Type I error rate aiming to control in the conventional t-test.
type	a character to specify the type of EEB to be calculated. `type = "marginal"` gives the marginal EEB; `type = "cond_NRej"` gives the EEB under the condition that one cannot reject the first-stage conventional t-test; `type = "cond_Rej"` gives the EEB under the condition that the first-stage conventional t-test is rejected.

Details

Consider a two-sample t-test setting with hypotheses

$$H_0 : \delta = 0 \quad \leftrightarrow \quad H_1 : \delta \neq 0,$$

where $\delta = \mu_1 - \mu_2$ is the difference of two population means. If the testing result is failure to reject the null, one cannot directly conclude equivalence of the two groups. In this case, an equivalence test is suggested by testing the hypotheses

$$H_3 : |\delta| \geq \Delta \quad \leftrightarrow \quad H_4 : |\delta| < \Delta,$$

where Δ is a pre-specified equivalence bound. A $100(1-2\alpha)\%$ confidence interval is formulated, denoted as $[L, U]$, to test for equivalence, where

$$L = \hat{\delta} - t_{\nu,1-\alpha}S, \quad U = \hat{\delta} + t_{\nu,1-\alpha}S,$$

$\hat{\delta}$ is the estimate of δ, $t_{\nu,1-\alpha}$ is the $100(1-\alpha)\%$ quantile of a t-distribution with degrees of freedom ν, and S is the standard error. We define the B-value as

$$B = \max\{|L|, |U|\}.$$

The cumulative distribution function of the B-value is defined under three conditions: (1) the marginal distribution (`type = "marginal"`); (2) the conditional distribution given that one cannot reject H_0 in the conventional t-test (`type = "cond_NRej"`); and (3) the conditional distribution given that H_0 is rejected in the conventional t-test (`type = "cond_Rej"`).

Value

Gives the cumulative distribution function of the B-value.

Author(s)

Yi Zhao, Indiana University, <zhaoyi1026@gmail.com>

Brian Caffo, Johns Hopkins University, <bcaffo@gmail.com>

Joshua Ewen, Kennedy Krieger Institute and Johns Hopkins University, <ewen@kennedykrieger.org>

References

Zhao et al. (2019) *"B-Value and Empirical Equivalence Bound: A New Procedure of Hypothesis Testing"* <arXiv:1912.13084>

See Also

EEB

Examples

```
##############################################
# An Example: demonstration of marginal/conditional distribution of the B-value
alpha<-0.05

delta<-0
n1=n2=n<-10
S<-0.325
nu<-n1+n2-2

# compare three types of B-value distributions
oldpar<-par(no.readonly=TRUE)
par(mar=c(6,5,2,2))
plot(c(0,2),c(0,1),type="n",xlab=expression(b),ylab=expression(F[B](b*~"|"*~C,H[0])),
     cex.lab=1.25,cex.axis=1.25,cex.main=1.25)
abline(h=1,lty=1,lwd=2,col=8)
abline(h=0,lty=1,lwd=2,col=8)
curve(pB(x,nu=nu,delta=delta,S=S,alpha=alpha,type="marginal"),
      lty=1,lwd=3,col=1,n=1000,from=0,to=20,add=TRUE)
curve(pB(x,nu=nu,delta=delta,S=S,alpha=alpha,type="cond_NRej"),
      lty=2,lwd=3,col=2,n=1000,from=0,to=20,add=TRUE)
curve(pB(x,nu=nu,delta=delta,S=S,alpha=alpha,type="cond_Rej"),
      lty=3,lwd=3,col=4,n=1000,from=0,to=20,add=TRUE)
par(fig=c(0,1,0,1),oma=c(0,0,0,0),mar=c(0,2,0,2),new=TRUE)
plot(0,0,type="n",bty="n",xaxt="n",yaxt="n")
legend("bottom",legend=c("marginal","conditional (not reject)","conditional (reject)"),
      xpd=TRUE,horiz=TRUE,inset=c(0,0),col=c(1,2,4),lty=c(1,2,3),lwd=2,bty="n",cex=1.25)
par(oldpar)
##############################################
```

Index