The malicious p-value

Matti Meyer

May 04, 2015

Contents

Abstract

Most of the empirical science uses the NHST to determine p-values for establishing their findings. In this document it will be proved that p-values are a very nonscientific and malicious tool.

The Data

The data frame i will use is avalaible here. This are the findings of a psychological experiment made by students of the university Koblenz-Landau. They were testing if there is a difference between the results of doing the Corsi-Test by varying conditions. There is the controll condition TS.A and a varied condition TS.B. We will do a t-test for condition TS.A and TS.B, but before that we will load the required packages and the data.

Because i'm using a operating system based on Linux, i need the *rio* package which is avaliable on github and can be installed with the *devtools* package. Windows user can load the data in an easier way, which i will show in a comment.

```r
library("devtools")

#To install rio use:
#install_github("leeper/rio")

library("rio")
```

```
Corsi.data <-import("https://raw.githubusercontent.com/MattiMeyer/Empra2/master/Daten%2Beingelesen-1.csv

##For Windows user
#Corsi.data <- read.csv("https://raw.githubusercontent.com/MattiMeyer/Empra2/master/Daten%2Beingelesen-

#We have to change some names of the data frame, we just need Person, TS.A and TS.B, ignore the rest
names(Corsi.data) <- c("Person","Bedingung","Alter","Geschlecht","Haendig","Studienfach","Semester","UBS
```

p-values and NHST

The function

To get a p-value the following function will help us. We will get a random samples of a count of persons that were participating by the experiment. Because the values are paired, we need to be sure, that a value in TS.A will get the right value from the same person in TS.B. Than it will test the two samples against each other to search for a significant result. This result is the p-value and this is the only thing the most scientist are interested in that's why the function will only produce this single value and nothing more. Please feel free to use the function for your own research!

```
p.value <- function(Pers,n,x,y){
  #Getting a sample of personen
  sample.per <- sample(Pers, n, replace=T)
  #Getting two samples of the two conditions
  data.x <- data.frame(x)
  sample.x<- data.x[c(sample.per),]

  data.y <- data.frame(y)
  sample.y <- data.y[c(sample.per),]

  #Making a t-test
  object.t <- t.test(sample.x,sample.y, paired = T,alternative = "greater")

  print("The p-value")
  object.t$p.value
}
```

Testing the function

We will test the function on condition A and B with 30 Persons, because this a good number for calculating.

```
p.value(Corsi.data$Person,30,Corsi.data$TS.A,Corsi.data$TS.B)
```

```
## [1] "The p-value"
```

```
## [1] 0.3707051
```

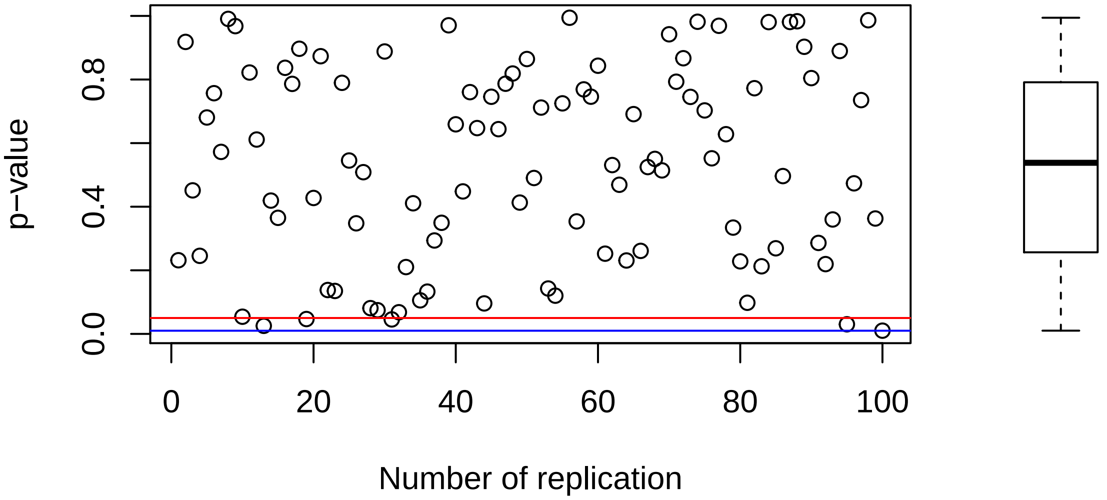
Because everytime this functions will be run there will be another p-value, so i cant't see what number you get. Maybe it was less than 0.05 and you can celebrate yourself for a significant result. Maybe you just wil revolutionize the use of the corsi test, because your results are significant. That's what scientist often do when they get a significant result.

Ok now we want more, much more! Let‘s see what will happen if we replicate this experiment more often, maybe 100 times. You maybe think “ok, just replicate, but the result will be the same. This change of the corsi-test is super un/effective because the result is un/significant”. Ok let’s see:

```
repl.p <- replicate(100, p.value(Corsi.data$Person,30,Corsi.data$TS.A,Corsi.data$TS.B))
```

To see the results we want to visualize the p-values in a scatter-plot. I marked the limit of getting a significant result of 0.05 with a red line and the 0.01 limit with a blue line:

```
par(fig=c(0,0.8,0,0.8), new=TRUE)
plot(repl.p, xlab = "Number of replication", ylab="p-value")
abline(h=0.05,col="red")
abline(h=0.01,col="blue")
par(fig=c(0.65,1,0,0.8),new=TRUE)
boxplot(repl.p, axes=FALSE)
```



I guess you can see in plot above that there are some results which are highly significant, some who are far away of it and some others are scratching the surface. You can now imagine that you just need a little bit luck and the right sample of persons to get highly significant result, maybe also less than 0.01! That could be a very frightening result, imagening that some researchers are just relying on the p-value. They just need to replicate their experiment so often till they get the p-value they want and publish it in the journals. Another conclusion is that a p-value doesn‘t show us what we want to see: Is there a big effect or a low effect.

Maybe you want to count the significant results:

```
signif <- which(repl.p < 0.05)
length(signif)
```

```
## [1] 5
```

Maybe you should do this some time again. Here are some counts of significance results:

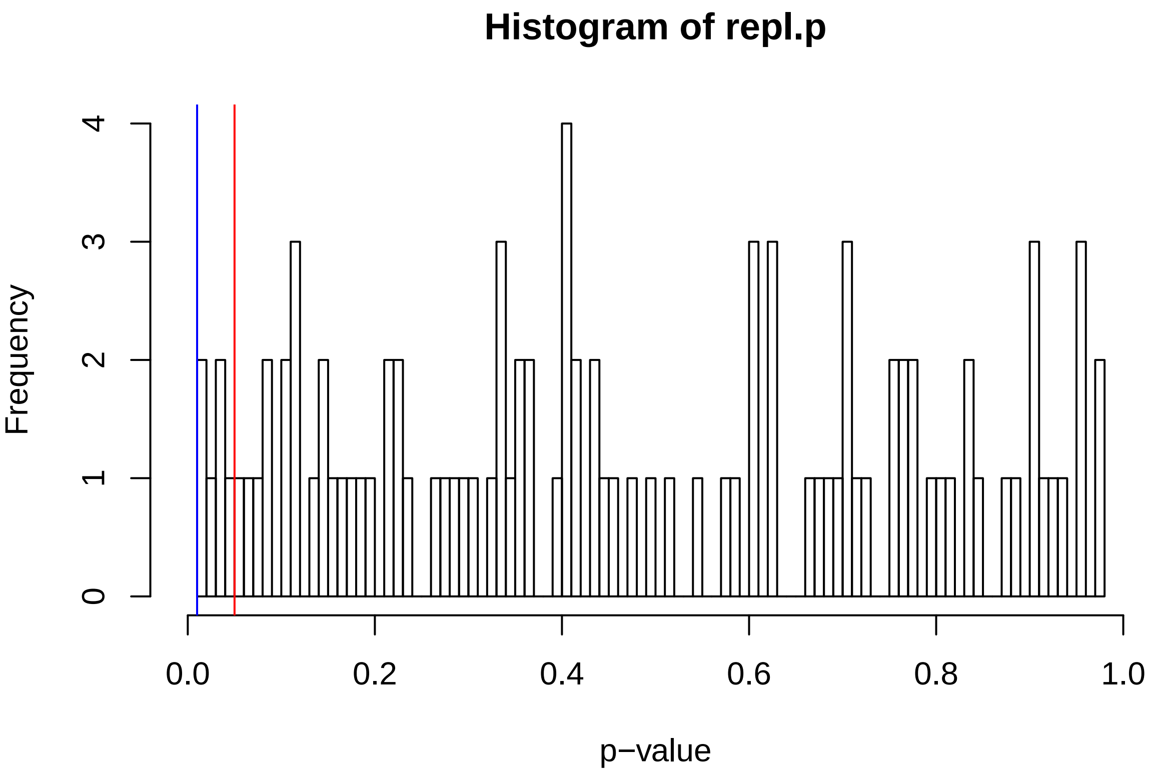
```
## [1] 7
```

```
## [1] 6
## [1] 6
```

I could do this the whole day, just watching surprised what p-value R will present to me!

Another way to visualise it is with a histogram, which will show us the same as above:

```
hist(repl.p,breaks = 100,xlab = "p-value",xlim =c(0,1)  )
abline(v=0.05,col="red")
abline(v=0.01, col="blue")
```



Getting over NHST

Effect Sizes

To getting over the NHST, we will use the effect sizes. In contrast to the NHST we don't ask now "Is there a significant difference?", we will ask "What for a difference is there? What effect can be expected?" To get the effect size we build a similiar function like before, including the *ci.sm* function from the *MBESS* package:

```
library(MBESS)
effect.size<- function(Pers,n,x,y){
  #Getting a sample of personen
  sample.per <- sample(Pers, n, replace=T)
  #Getting two samples of the two conditions
  data.x <- data.frame(x)
  sample.x<- data.x[c(sample.per),]

  data.y <- data.frame(y)
  sample.y <- data.y[c(sample.per),]

  #Making a t-test
    object.t <- t.test(sample.x,sample.y, paired = T,alternative = "greater")

  #Getting the standard deviation of the difference of   the variables
  sd.diff <- sd(sample.x - sample.y)

  #Getting the mean of differences
  mean.diff <- object.t$estimate

  #Using the ci.sm function for the effect sice and confidence intervall
  ci <- ci.sm(Mean=mean.diff,SD=sd.diff,N=n)

  #getting the effect sice
  effect <- data.frame(ci[2])
  effect[1,1]

}
```

Ok now we will use the function on our data frame to get one effect size.

```
effect <- effect.size(Pers = Corsi.data$Person,n = 30,x = Corsi.data$TS.A,y = Corsi.data$TS.B)
```

```
effect
```

```
## [1] 0.2685649
```

But that isn't enough for us, is it? And again we replicate the function 100 times.

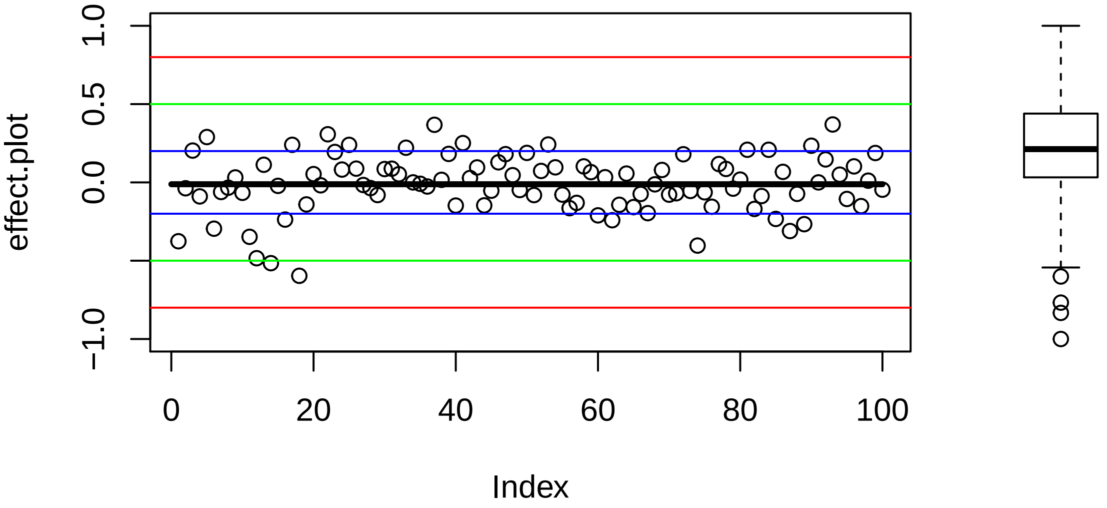
```
effect.plot <- replicate(n = 100, expr =effect.size(Pers = Corsi.data$Person,n = 30,x = Corsi.data$TS.A
```

```
summary(effect.plot)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.59640 -0.09356 -0.01063 -0.01180  0.09763  0.36990
```

To visualize this we will plot the effect! For a better understanding the limits of the effects are picured. The blue lines limit the 0.2 intervall (small effect), the green ones the 0.5 intervall (middle effect) and the red ones the 0.8 intervall (large effect). And also the mean of all effect sizes is in there as a black line.

```
par(fig=c(0,0.8,0,0.8), new=TRUE)
plot(effect.plot,ylim=c(-1,1))
arrows(x0 = 0,y0 =mean(effect.plot),x1 = 100,y1 =  mean(effect.plot),code = 0,col = "black",lwd = 3)

abline(h = 0.2 ,col="blue")
abline(h = -0.2 ,col="blue")

abline(h=0.5,col="green")
abline(h=-0.5,col="green")
abline(h=0.8,col="red")
abline(h=-0.8,col="red")
par(fig=c(0.65,1,0,0.8),new=TRUE)
boxplot(effect.plot, axes=F)
```



As we can see now the mean shows us that there is nearly now effect. But you may say that the points are also widely spreaded like the p-values. I would say that the difference here is about the understanding of the difference between p-values and effect sizes. When a p-value gives us the hint to accept our hypothesis or reject it, the effect size shows us how small or large the difference between groups is. We can assume that there will be effect, whatever you test. There must be one. But now we can interpretate if the effect is large or low or in which direction. And as we can see here, most of the the effect sizes settle between 0 till 0.2, which is the lowest limitation of Cohens limits. To prove that we will search for number of effect sizes under the border of 0.2:

```
under.low.effect <- effect.plot[(effect.plot < 0.2) & (effect.plot > -0.2)]
length(under.low.effect)
```

```
## [1] 74
```

And now we want to see the values which are above a small effect:

```
over.low.effect <- effect.plot[(effect.plot > 0.2) | (effect.plot < -0.2) & (effect.plot < 0.5)&(effect
length(over.low.effect)
```

```
## [1] 24
```

Ok so you can see there are some values over the size of a small effect. But are there some going far above the middle effects?

```
over.middel.effect <- effect.plot[(effect.plot > 0.5) | (effect.plot < -0.5)]
length(over.middel.effect)
```

```
## [1] 2
```

You will see that there are less effect sizes above the limit of middle effects!

Conclusion

I want to make clear that the big difference between p-values and effect sizes is laying in the interpretation. You can't interpretate p-values laying a little bit under .05, because in a replication the p-value could be far over it. It makes sense to interpretate p-values which are so small that you can replicate them. But with an effect size you can make an interpretation how big the effect in real is and if it is negative or positive. And as we see in sum the effect is nearly null.

Confidence intervals

Another way and maybe the best way to get over the NHST are confidence intervals. The problem now is that it is not so easy to visualise them as the p-values or effect sizes, but when you see them you will understand what i mean. So for the first i will try to explain how to plot hunderts of confidence intervals and if you know a better or easier way to do this i like to invite you to recommend me at my Github Page. If you are not interested how to plot the intervals just switch to the finished plot and the explanation. Also i want to make clear that i will use 95% confidence intervals here, because many researchers forget to pronounce this and that leads to some confusion.

First we will write a function again, in which we use again the *ci.sm* function of the *MBESS* package. This will give us the lower and upper limits of one confidence intervall and also the mean of it, which we used before as the effect size.

```
confidence<- function(Pers,n,x,y){
  #Getting a sample of personen
  sample.per <- sample(Pers, n, replace=T)
  #Getting two samples of the two conditions
  data.x <- data.frame(x)
  sample.x<- data.x[c(sample.per),]

  data.y <- data.frame(y)
  sample.y <- data.y[c(sample.per),]

  #Making a t-test
  object.t <- t.test(sample.x,sample.y, paired = T,alternative = "greater")

  #Getting the standard deviation of the difference of   the variables
  sd.diff <- sd(sample.x - sample.y)

  #Getting the mean of differences
  mean.diff <- object.t$estimate

  #Using the ci.sm function for the effect sice and confidence intervall
  ci <- as.data.frame(ci.sm(Mean=mean.diff,SD=sd.diff,N=n))


  #getting the limits of confidence intervals and mean
     lower<- ci[1,1]
     mean <- ci[1,2]
     upper <- ci[1,3]
    c(lower,mean,upper)
}
```

We should test our new function:

```
confidence(Pers = Corsi.data$Person,n = 30,x = Corsi.data$TS.A,y=Corsi.data$TS.B)
```

```
## [1] "The 0.95 confidence limits for the standardized mean are given as:"
```

```
## [1] -0.31380055  0.04460313  0.40224201
```

So we can see that the first number is for the lower limit, in the middle is the mean and the last number is the upper limit.

Next we will start to prepare to build the plot. It is important to understand that we will build two data frames with different forms, one will have the "wide" and the other the "long" format. But they will base on the same data than we used before. We will use the long format to build a plot and the wide format to build arrows to visualise the confidence intervals. But before we can do this we have to make sure that every limit and mean is connected to a number which stands for the number of the interval, because at the end we want to plot the limits and means so that they are connected in one interval. For this sake we have to build data frames with the numbers from one to hundred, because we will replicate again the function a hundred times. We will connect these numbers later with the long and wide data frames.

```
hund  <- seq(1:100)
hund2 <- seq(1:100)
hund3 <- seq(1:100)
two   <- as.data.frame( c(hund,hund2))
three <- as.data.frame(c(hund,hund2,hund3))
```

Ok after that comes the replication. Maybe you just expected it, because of the things you read before:

```
repli <- as.data.frame(replicate(confidence(Pers = Corsi.data$Person,n = 30,x = Corsi.data$TS.A,y=Corsi
```

Ok after we've done that we have to seperate the values that we want and to stack the data frames, because we need this format.

```
lower.limit <- stack(repli[1,])
mean        <- stack(repli[2,])
upper.limit <- stack(repli[3,])
```

Now we can build the long format data frame and connect it with the "three"" data frame.

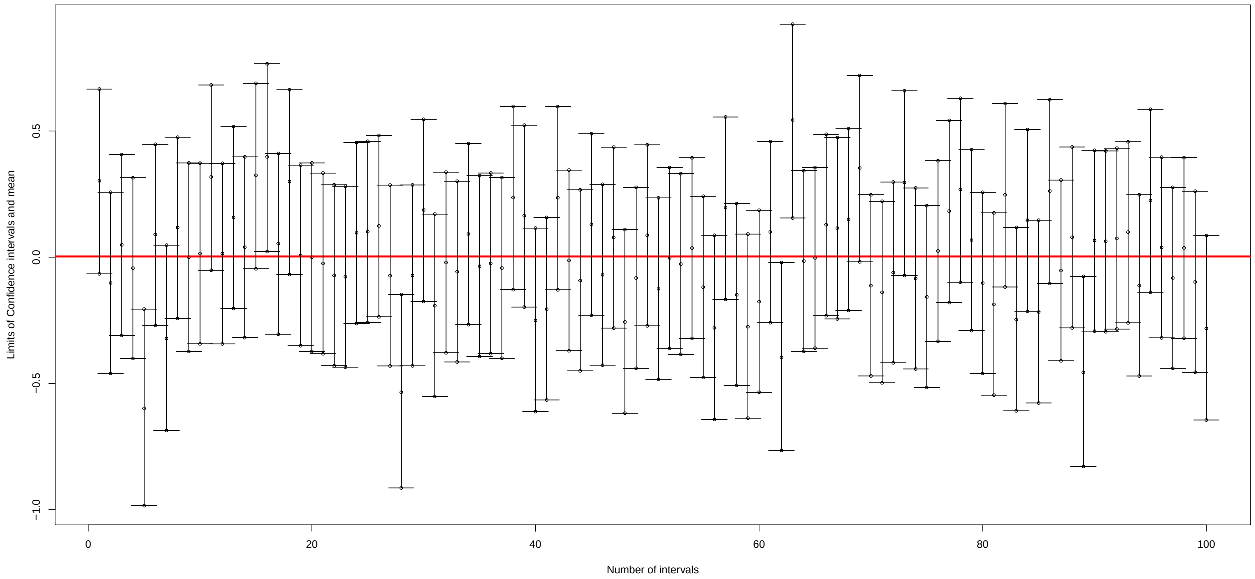
```
long.one      <- rbind(lower.limit,mean,upper.limit)
long.data <- cbind(long.one,three)
names(long.data) <- c("Limits/Means","V","number")
```

And now the wide data frame with the "two" data frame, because for the arrows we do not need the mean, just the limits of the intervals.

```
wide     <- cbind(lower.limit,upper.limit)
two.data   <- as.data.frame(two)
wide.data <- cbind(wide,two.data)

names(wide.data) <- c("lower","Id","Upper","Id","number")
```

The data frames are ready to be visualised. So we can plot the long data frame to get the points of the limits and means. Then we add a horizontal line to show the mean of the means. That is important because every confidence interval which does not include this line will get a significant p-value. After that we will build arrays from the upper to the lower limit of the intervals wtih the *for* function, because now we can compare the length of the intervals, which is something important as we will see later.

```
plot(y=long.data$`Limits/Means`,long.data$number,type = "p",xlab = "Number of intervals",ylab = "Limits

abline(h= mean(mean[,1]),lwd=3,col="red")

for( i in 1:nrow(wide.data) ){
  arrows(x0 = wide.data$number[i],
         x1 = wide.data$number[i],
         y0 = wide.data$lower[i],
         y1 = wide.data$Upper[i],
         code = 3,angle = 90,length = 0.2 )
}
```



Interesting, isn't it? The confidence intervals show us a picture of harmony. They all have permanent the same length and that ist something that we can use to interpretate the data. To show you some options for interpretating the CI I will cite Cumming and Finch (2005), because they give a really good overview and you can also get a deeper understanding of CI's if you are reading the articel. I also like to invite you for reading Geoff Cummings book "The New Statistics" (2012) in the following, because there you will find the explanation of all the things we've done here.

Here are possible interpretations of CI's from Cumming and Finch:

- researcher who routinely reports 95% CIs can expect over a lifetime that about 95% of those intervals will capture the parameters estimated
- Our CI is a range of plausible values for μ. Values outside the CI are relatively implausible
- We can be 95% confident that our CI includes μ
- The lower limit is a likely lower bound estimate of the parameter; the upper limit a likely upper bound
- consider interpretations of the lower and upper limits and compare these with interpretation of the mean
- Any value outside the CI, when considered as a null hypothesis, gives two-tailed p<.05. Any value inside the CI, when considered as a null hypothesis, gives p>.05. Considering our example CI (51.85, 72.15),

the null hypothesis 50 would give p<.05 because 50 is outside the interval, but the null hypothesis μ= 60 would give p >.05

- We can be 95% confident that our point estimate is no more than w from the true value of μ, so w is the largest error of estimation we are likely to make —although larger errors are possible— and w is an index of the precision of the study. Make a substantive interpretation of w.

I hope to show you the problems of p-values and maybe a better solution. If you have comments or critics, here is my Github Page.
Thanks very much!

Links

For a good understanding of the basics and getting even beyond:
http://www.statmethods.net/

To write your own functions use this interesting book:
http://shop.oreilly.com/product/0636920028574.do