PS5

Prateek Kumar

14 October 2018

Problem 1. Understanding the power of a test.

Problem 2: t-tests

data\_q2 <- c(101,103,99,92,110,105,103,102,104,106,101,  
 101,101,102,103,101,99,104,105,102,102,103,  
 245,103,107,101,103,108,104,101,101,102)  
  
# calculating the mean, sd, and se   
mu <- mean(data\_q2)  
sd <- sd(data\_q2)  
se <- sd/sqrt(length(data\_q2))  
  
# calculate t-test with mean - 100  
t\_test <- (mu-100)/se  
print(paste0("t = ", round(t\_test, 4)))

## [1] "t = 1.5608"

# calculate onetail p-value, from t-test and degree of freedom  
p\_value <- 1 - pt(t\_test, length(data\_q2)-1)  
print(paste0("p-value = ", round(p\_value, 4)))

## [1] "p-value = 0.0644"

# checking with t.test() function  
t.test(data\_q2, mu=100, alt="greater")

##   
## One Sample t-test  
##   
## data: data\_q2  
## t = 1.5608, df = 31, p-value = 0.06436  
## alternative hypothesis: true mean is greater than 100  
## 95 percent confidence interval:  
## 99.39604 Inf  
## sample estimates:  
## mean of x   
## 107

Here we can see that the t and p-value we got through hands-on is same as that we got through the function t.test(). Also the average(mean) is reliably greater than 100.

# removing 245 from the vector  
data\_q2B <- c(101,103,99,92,110,105,103,102,104,106,101,  
 101,101,102,103,101,99,104,105,102,102,103,  
 103,107,101,103,108,104,101,101,102)  
  
#estimate its mean, sd, and se  
mu1 <- mean(data\_q2B)  
sd1 <- sd(data\_q2B)  
se1 <- sd1/sqrt(length(data\_q2B))  
  
#calculate t-test with mean - 100  
t\_test1 <- (mu1-100)/se1  
print(paste0("t = ", t\_test1))

## [1] "t = 4.53494050821014"

#calculate onetail p-value, from t-test and degree of freedom  
p\_value1 <- 1 - pt(t\_test1, length(data\_q2B)-1)  
print(paste0("p-value = ", p\_value1))

## [1] "p-value = 4.31557207791755e-05"

t.test(data\_q2B, mu=100, alt="greater")

##   
## One Sample t-test  
##   
## data: data\_q2B  
## t = 4.5349, df = 30, p-value = 4.316e-05  
## alternative hypothesis: true mean is greater than 100  
## 95 percent confidence interval:  
## 101.5946 Inf  
## sample estimates:  
## mean of x   
## 102.5484

Here as well we can see that the t and p-value we got through hands-on is same as that we got through the function t.test(). Also the average(mean) is reliably greater than 100.

The difference in both the cases is that in the vector including 245 has p-value greater than 0.05 whereas when we exclude that from our vector we get the p-value approx to 0. So in the first case we fail to reject the null hypothesis whereas in second case since the p-value < 0.05 we reject the null hypothesis so we can conclude that 245 is the strongest evidence for this kind of behaviour.

Problem 3. Wilcox test

x <- rnorm(20)  
y <- sort(x)  
  
# Independent Wilcox Test  
wilcox.test(x, y, paired = F, exact = F, alternative = "g") # equivalent to Mann-Whitney Test

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: x and y  
## W = 200, p-value = 0.5054  
## alternative hypothesis: true location shift is greater than 0

# Paired Wilcox Test  
wilcox.test(x, y, paired = T, exact = F, alternative = "g")

##   
## Wilcoxon signed rank test with continuity correction  
##   
## data: x and y  
## V = 98, p-value = 0.6103  
## alternative hypothesis: true location shift is greater than 0

In the independent samples Wilcox Test we get the W value as 200 as we can see in the above output. The first Wilcox Test i.e. the independent samples Wilcox test is equivalent to Mann-Whitney Test where we get the p-value = 0.5054 and in the second Wilcox Test we get the p-value as 0.4435, we can see that in the paired Wilcox test the p-value is less this might be because of the correlation between the values of x and y because we are calculating it pairwise and so paired Wilcox Test cannot compute the exact p-value which we get in the independent samples Wilcox Test.

Problem 4. Comparing t-test, wilcox, and Bayes factor t-test

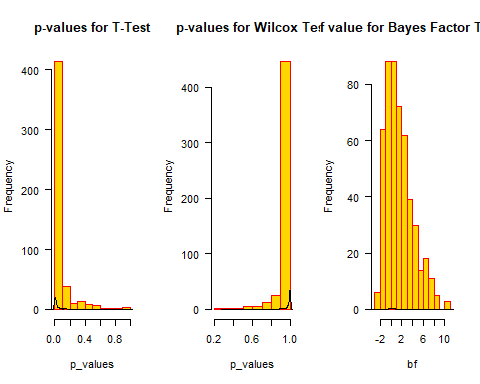
library(BayesFactor)

## Loading required package: coda

## Loading required package: Matrix

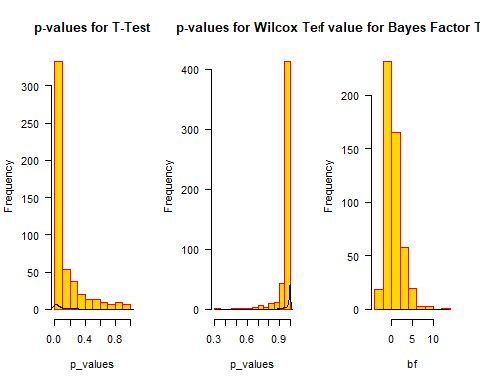
## \*\*\*\*\*\*\*\*\*\*\*\*  
## Welcome to BayesFactor 0.9.12-4.2. If you have questions, please contact Richard Morey (richarddmorey@gmail.com).  
##   
## Type BFManual() to open the manual.  
## \*\*\*\*\*\*\*\*\*\*\*\*

n <- 500  
t\_values <- c()  
w\_values <- c()  
b\_values <- c()  
  
# running for 500 times  
for(i in 1:n)  
{   
 x <- rnorm(150)  
 y <- rnorm(150) + .3  
   
 t <- t.test(x, y)  
 t\_values<- append(t\_values,t$p.value)  
 w <- wilcox.test(x, y, paired = F, exact = F, alternative = "g")  
 w\_values<- append(w\_values,w$p.value)  
 b <- ttestBF(x, y)  
 b\_values<- append(b\_values,b@bayesFactor$bf)  
}  
  
# Basic histogram  
par(mfrow=c(1,3))  
hist(t\_values, main = 'p-values for T-Test', xlab = 'p\_values', border = 'red', col = 'gold', las=1)  
lines(density(t\_values))  
hist(w\_values, main = 'p-values for Wilcox Test', xlab = 'p\_values', border = 'red', col = 'gold', las=1)  
lines(density(w\_values))  
hist(b\_values, main = 'bf value for Bayes Factor T-Test', xlab = 'bf', border = 'red', col = 'gold', las=1)  
lines(density(b\_values))

 #high value then strong evidence for alter hyp // for BF

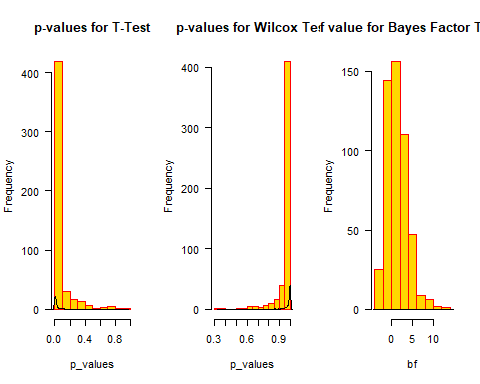
Problem 5. Robustness to transforms

library(BayesFactor)  
  
n <- 500  
t\_values <- c()  
w\_values <- c()  
b\_values <- c()  
  
# running for 500 times  
for(i in 1:n)  
{   
 x <- exp(rnorm(150))  
 y <- exp(rnorm(150) + .3)  
   
 t <- t.test(x, y)  
 t\_values<- append(t\_values,t$p.value)  
 w <- wilcox.test(x, y, paired = F, exact = F, alternative = "g")  
 w\_values<- append(w\_values,w$p.value)  
 b <- ttestBF(x, y)  
 b\_values<- append(b\_values,b@bayesFactor$bf)  
}  
  
# Basic histogram  
par(mfrow=c(1,3))  
hist(t\_values, main = 'p-values for T-Test', xlab = 'p\_values', border = 'red', col = 'gold', las=1)  
lines(density(t\_values))  
hist(w\_values, main = 'p-values for Wilcox Test', xlab = 'p\_values', border = 'red', col = 'gold', las=1)  
lines(density(w\_values))  
hist(b\_values, main = 'bf value for Bayes Factor T-Test', xlab = 'bf', border = 'red', col = 'gold', las=1)  
lines(density(b\_values))



Problem 6. Comparison to paired tests.

n <- 500  
t\_values <- c()  
w\_values <- c()  
b\_values <- c()  
  
# running for 500 times  
for(i in 1:n)  
{   
 x <- rnorm(150)  
 y <- rnorm(150) + .3  
   
 t <- t.test(x,y,paired=T)  
 t\_values<- append(t\_values,t$p.value)  
 w <- wilcox.test(x, y, paired = T, exact = FALSE, alternative = "g")  
 w\_values<- append(w\_values,w$p.value)  
 b <- ttestBF(x,y,paired=T)  
 b\_values<- append(b\_values,b@bayesFactor$bf)  
}  
  
# Basic histogram  
par(mfrow=c(1,3))  
hist(t\_values, main = 'p-values for T-Test', xlab = 'p\_values', border = 'red', col = 'gold', las=1)  
lines(density(t\_values))  
hist(w\_values, main = 'p-values for Wilcox Test', xlab = 'p\_values', border = 'red', col = 'gold', las=1)  
lines(density(w\_values))  
hist(b\_values, main = 'bf value for Bayes Factor T-Test', xlab = 'bf', border = 'red', col = 'gold', las=1)  
lines(density(b\_values))



Problem 7: Correlations

x <- runif(100)  
y <- runif(100)  
  
z <- x + .5\*y  
z1 <- z  
z2 <- z+10  
z3 <- log(z)  
z4 <- z\*10  
z5 <- z + runif(100)  
  
#the pearson and spearman correlations  
#x and z1  
cor.test(x,z1,method="pearson")

##   
## Pearson's product-moment correlation  
##   
## data: x and z1  
## t = 19.475, df = 98, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.8425558 0.9257658  
## sample estimates:  
## cor   
## 0.8914457

cor.test(x,z1,method="spearman")

##   
## Spearman's rank correlation rho  
##   
## data: x and z1  
## S = 16938, p-value < 2.2e-16  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho   
## 0.8983618

#x and z2  
cor.test(x,z2,method="pearson")

##   
## Pearson's product-moment correlation  
##   
## data: x and z2  
## t = 19.475, df = 98, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.8425558 0.9257658  
## sample estimates:  
## cor   
## 0.8914457

cor.test(x,z2,method="spearman")

##   
## Spearman's rank correlation rho  
##   
## data: x and z2  
## S = 16938, p-value < 2.2e-16  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho   
## 0.8983618

#x and z3  
cor.test(x,z3,method="pearson")

##   
## Pearson's product-moment correlation  
##   
## data: x and z3  
## t = 14.259, df = 98, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.7452550 0.8764406  
## sample estimates:  
## cor   
## 0.8214308

cor.test(x,z3,method="spearman")

##   
## Spearman's rank correlation rho  
##   
## data: x and z3  
## S = 16938, p-value < 2.2e-16  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho   
## 0.8983618

#x and z4  
cor.test(x,z4,method="pearson")

##   
## Pearson's product-moment correlation  
##   
## data: x and z4  
## t = 19.475, df = 98, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.8425558 0.9257658  
## sample estimates:  
## cor   
## 0.8914457

cor.test(x,z4,method="spearman")

##   
## Spearman's rank correlation rho  
##   
## data: x and z4  
## S = 16938, p-value < 2.2e-16  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho   
## 0.8983618

#x and z5  
cor.test(x,z5,method="pearson")

##   
## Pearson's product-moment correlation  
##   
## data: x and z5  
## t = 8.9318, df = 98, p-value = 2.514e-14  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.5452027 0.7655707  
## sample estimates:  
## cor   
## 0.6698845

cor.test(x,z5,method="spearman")

##   
## Spearman's rank correlation rho  
##   
## data: x and z5  
## S = 57360, p-value < 2.2e-16  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho   
## 0.6558056

#Bayes factor  
ttestBF(x, z1)

## Bayes factor analysis  
## --------------  
## [1] Alt., r=0.707 : 2747465 ±0%  
##   
## Against denominator:  
## Null, mu1-mu2 = 0   
## ---  
## Bayes factor type: BFindepSample, JZS

ttestBF(x, z2)

## t is large; approximation invoked.

## Bayes factor analysis  
## --------------  
## [1] Alt., r=0.707 : 2.331051e+241 ±0%  
##   
## Against denominator:  
## Null, mu1-mu2 = 0   
## ---  
## Bayes factor type: BFindepSample, JZS

ttestBF(x, z3)

## Bayes factor analysis  
## --------------  
## [1] Alt., r=0.707 : 4.456844e+29 ±0%  
##   
## Against denominator:  
## Null, mu1-mu2 = 0   
## ---  
## Bayes factor type: BFindepSample, JZS

ttestBF(x, z4)

## t is large; approximation invoked.

## Bayes factor analysis  
## --------------  
## [1] Alt., r=0.707 : 5.072941e+54 ±0%  
##   
## Against denominator:  
## Null, mu1-mu2 = 0   
## ---  
## Bayes factor type: BFindepSample, JZS

ttestBF(x, z5)

## Bayes factor analysis  
## --------------  
## [1] Alt., r=0.707 : 8.991646e+30 ±0%  
##   
## Against denominator:  
## Null, mu1-mu2 = 0   
## ---  
## Bayes factor type: BFindepSample, JZS