Problem Set 5

Prateek Kumar

Problem 1. Understanding the power of a test.

n = 10							
simulation				null			
pval<.05	pvalnp<.05	bayesf>2	bayesf<-2	pval<.05	pvalnp<.05	bayesf>2	bayesf<-2
0.048	0.022	0.038	0	0.064	0.028	0.056	0

For n=10,

- we can see that about 5% of the cases are found to be significant in the standard t-test. In this case, it is almost identical to the number that are found to be significant in the null distribution (we'd expect 5% to occur just by chance if there were no difference.).
- The number we find in the binomial test is lower in both cases
- The number of Bayes factors > 2 is about the same as well.
- We see that although the Bayes factor test only finds a few cases that are significant, it is otherwise typically in an ambivalent state–it never finds support for the null hypothesis either.

n = 50							
simulation				null			
pval<.05	pvalnp<.05	bayesf>2	bayesf<-2	pval<.05	pvalnp<.05	bayesf>2	bayesf<-2
0.104	0.044	0.046	0	0.05	0.044	0.024	0

For n = 50,

- we can see that about 10% of the cases are found to be significant in the standard t-test. In this case, it is almost double to the number that are found to be significant in the null.
- The number we find in the binomial test is equal in both cases
- The number of Bayes factors > 2 is about half for null hypothesis than simulation.
- We see that although the Bayes factor test only finds a few cases that are significant, it is otherwise typically in an ambivalent state–it never finds support for the null hypothesis either.

n = 200							
simulation				null			
pval<.05	pvalnp<.05	bayesf>2	bayesf<-2	pval<.05	pvalnp<.05	bayesf>2	bayesf<-2
0.312	0.168	0.128	0	0.048	0.048	0.02	0

For n = 200,

- we can see that about 30% of the cases are found to be significant in the standard t-test. But for significant in the null distribution we still have 5%.
- The number we find in the binomial test is higher in simulation compared to null where it is close to 0.5
- The number of Bayes factors > 2 is more for simulation.
- We see that although the Bayes factor test only finds a few cases that are significant, it is otherwise typically in an ambivalent state–it never finds support for the null hypothesis either.

n = 1000							
simulation				null			
pval<.05	pvalnp<.05	bayesf>2	bayesf<-2	pval<.05	pvalnp<.05	bayesf>2	bayesf<-2
0.9	0.706	0.622	0	0.042	0.038	0.006	0

For n = 1000,

- we can see that about 90% of the cases are found to be significant in the standard t-test but for significant in the null distribution we still have 5%.
- The number we find in the binomial test is very high for simulation but low for null.
- The number of Bayes factors > 2 is high for simulation and very low(approx. to 0) for null.
- We see that although the Bayes factor test only finds a few cases that are significant, it is otherwise typically in an ambivalent state–it never finds support for the null hypothesis either.

n = 5000							
simulation				null			
pval<.05	pvalnp<.05	bayesf>2	bayesf<-2	pval<.05	pvalnp<.05	bayesf>2	bayesf<-2
1	1	1	0	0.056	0.052	0	0

For n = 5000,

- we can see that about 100% of the cases are found to be significant in the standard t-test but for significant in the null distribution we still have 5%.
- The number we find in the binomial test is 1 for simulation whereas it is 0.052 for null which is very low compared to simulation.
- The number of Bayes factors > 2 is 1 for simulation and 0 for null.
- We see that although the Bayes factor test only finds a few cases that are significant, it is otherwise typically in an ambivalent state–it never finds support for the null hypothesis either.

We can see that in general when the n value increases the 4 shown values above increase for simulation whereas it decreases for null hypothesis.

Problem 2: t-tests

```
> print(paste0("t = ", round(t_test, 4)))
[1] "t = 1.5608"
> print(paste0("p-value = ", round(p_value, 4)))
[1] "p-value = 0.0644"
```

```
> 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
```

Figure 1: Results with 245 in the vector

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.

```
> print(paste0("t = ", t_test1))
[1] "t = 4.53494050821014"
> 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
```

Figure 2: Results after removing 245

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

```
> # 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 = 108, p-value = 0.4628
alternative hypothesis: true location shift is greater than 0
```

Figure 3: Wilcox test

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 0.4628, 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

```
> print(t_values)
  [1] 5.916593e-02 5.769082e-02 3.007449e-01 2.120124e-03 1.093572e-03 8.299382e-02 3.199844e-01 5.115745e-03 9.657118e-02
 [10] 3.997159e-04 6.251694e-01 1.121243e-01 1.202202e-05 2.740665e-04 7.048909e-04 2.712425e-02 2.444945e-02 8.674926e-03
 [19] 1.250779e-01 7.234672e-05 2.786166e-02 8.898095e-03 1.917187e-03 1.143063e-01 3.197630e-01 7.424534e-04 2.787385e-03
```

Figure 4: Sample for t-test

```
> print(w_values)
  [1] 0.9756219 0.9898512 0.8128695 0.9979598 0.9998417 0.9155441 0.9052088 0.9972375 0.9356778 0.9998193 0.6184826 0.9563504
 [13] 0.9999871 0.9998351 0.9993752 0.9846550 0.9870915 0.9987290 0.9464446 0.9998512 0.9949305 0.9887612 0.9995858 0.8723226
 [25] 0.9266912 0.9984605 0.9975886 0.9963357 0.9983293 0.9713817 0.9999420 0.6918726 0.9658456 0.9946101 0.7426644 0.9865505
```

Figure 5: Sample for Wilcox test

```
> print(b_values)
  [1] -0.355881561 -0.335836557 -1.550141394  2.473866800  3.066022493 -0.622568001 -1.589355661  1.701175480 -0.739576243
 [10]  3.965350922 -1.948780040 -0.853482288  7.175324972  4.306454026  3.457011064  0.279218690  0.365919269  1.246213559
 [19] -0.935730659  5.519129539  0.256848961  1.221809201  2.562925046 -0.868049758 -1.588950826  3.408495601  2.232415073
```

Figure 6: Sample for Bayes factor t-test

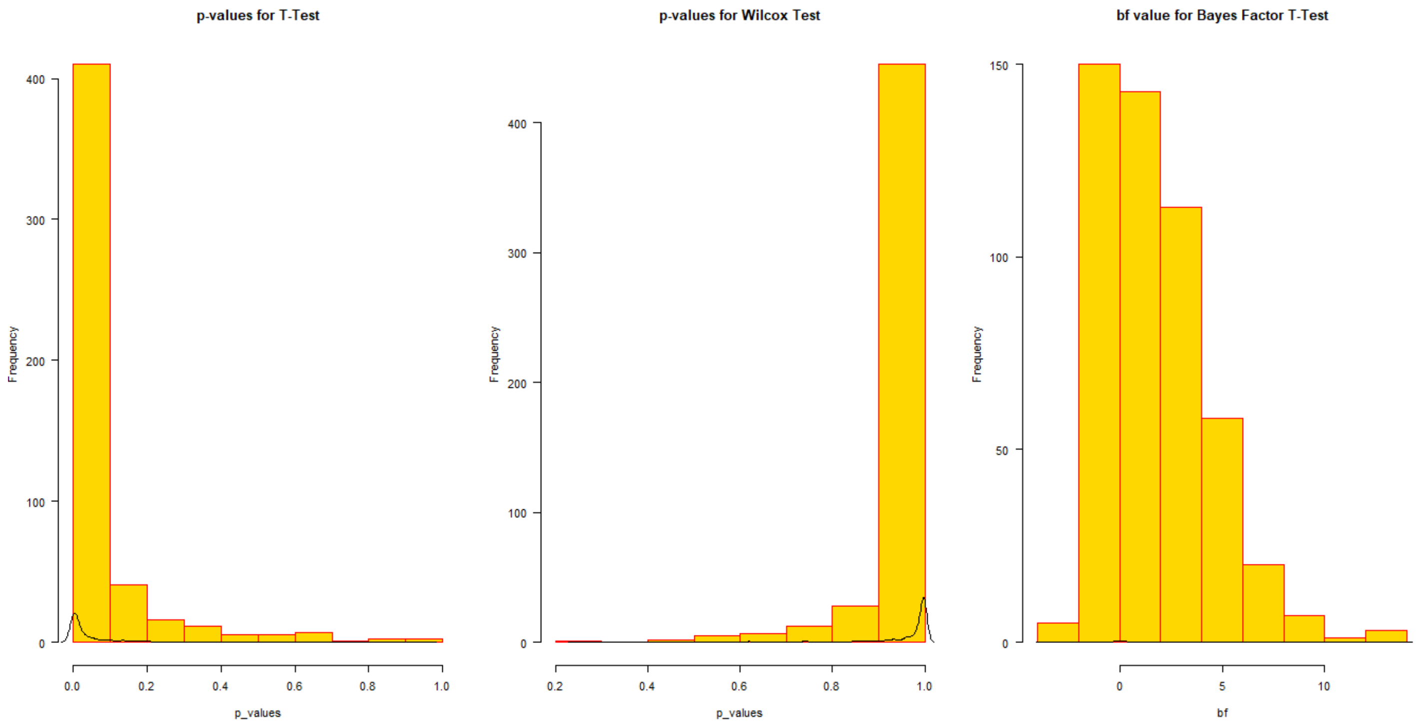

Figure 7: Histograms for all the 3 tests sampled 500 times

We can see in the above outputs that the results differ for the different tests.

Here I ran the 3 tests for 500 times and we can see a pattern for the p-values for the t-test. We see that the frequency of p-value = 0.1 is more than 400 out of 500 test results which means that the p-value is close to 0.1 approx. 95-99% times. And the count decreases substantially as the value approaches 1. Whereas this is the similar situation for Wilcox test but the result is the mirror image for what we obtained for t-test.

For the Bayes factor T-test I plotted the histogram for the bf value and we can see that the frequency of the bf value is highest at -1 and next at 1 and decreases substantially as it approaches towards the positive values.

Problem 5. Robustness to transforms

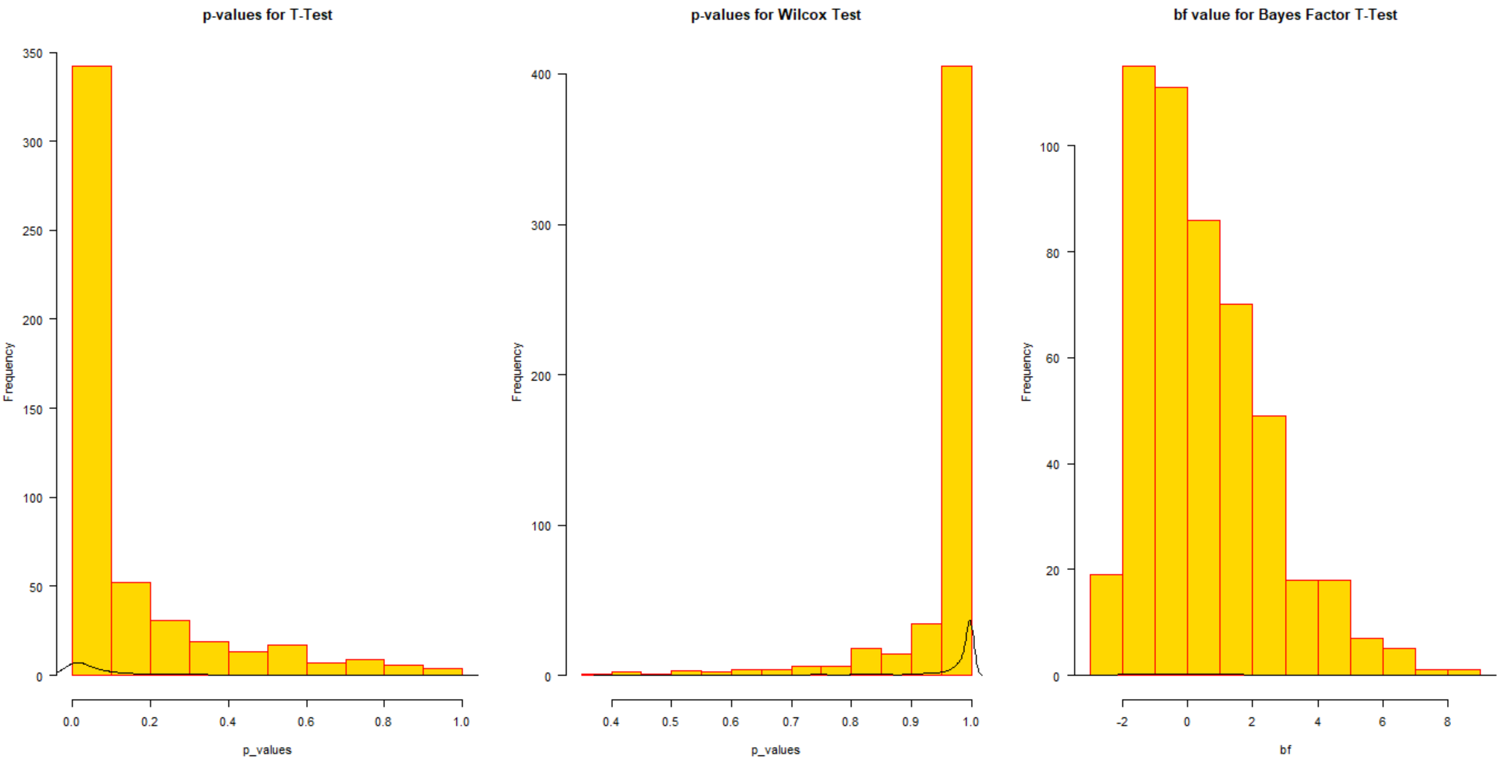

Figure 8: Histogram as Q4 but for exp() function

Here as well I ran the 3 tests for 500 times and we can see a similar pattern for the p-values for both the t-test and wilcox test as seen in Q4. We see that the frequency of p-value is still highest for both as was in Q4 but now the count has decreased to 340 for t-test and 410 for Wilcox test whereas earlier it was 400+ and 450+ respectively. For Wilcox Test the decrease in frequency for p-value is less as compared to that in t-test. And we can also see that the count decreases and increases substantially as the values reach 1 for t-test and Wilcox test respectively.

For the Bayes factor T-test I plotted the histogram for the bf value and we can see that the frequency of the bf value is now highest at -2 and next at -1 and decreases substantially as it approaches towards the positive values.

In Q4 we are using the rnorm() to create the vector so the vector will have mean = 0 so the values are evenly distributed across the value 0 for x and y(approx.) and when we apply the exp() on the values we get the positive results for them and the mean is between 1-3 so because of all the elements in the vector in exp() is positive we see such change in the values for the 3 tests.

Problem 6. Comparison to paired tests.

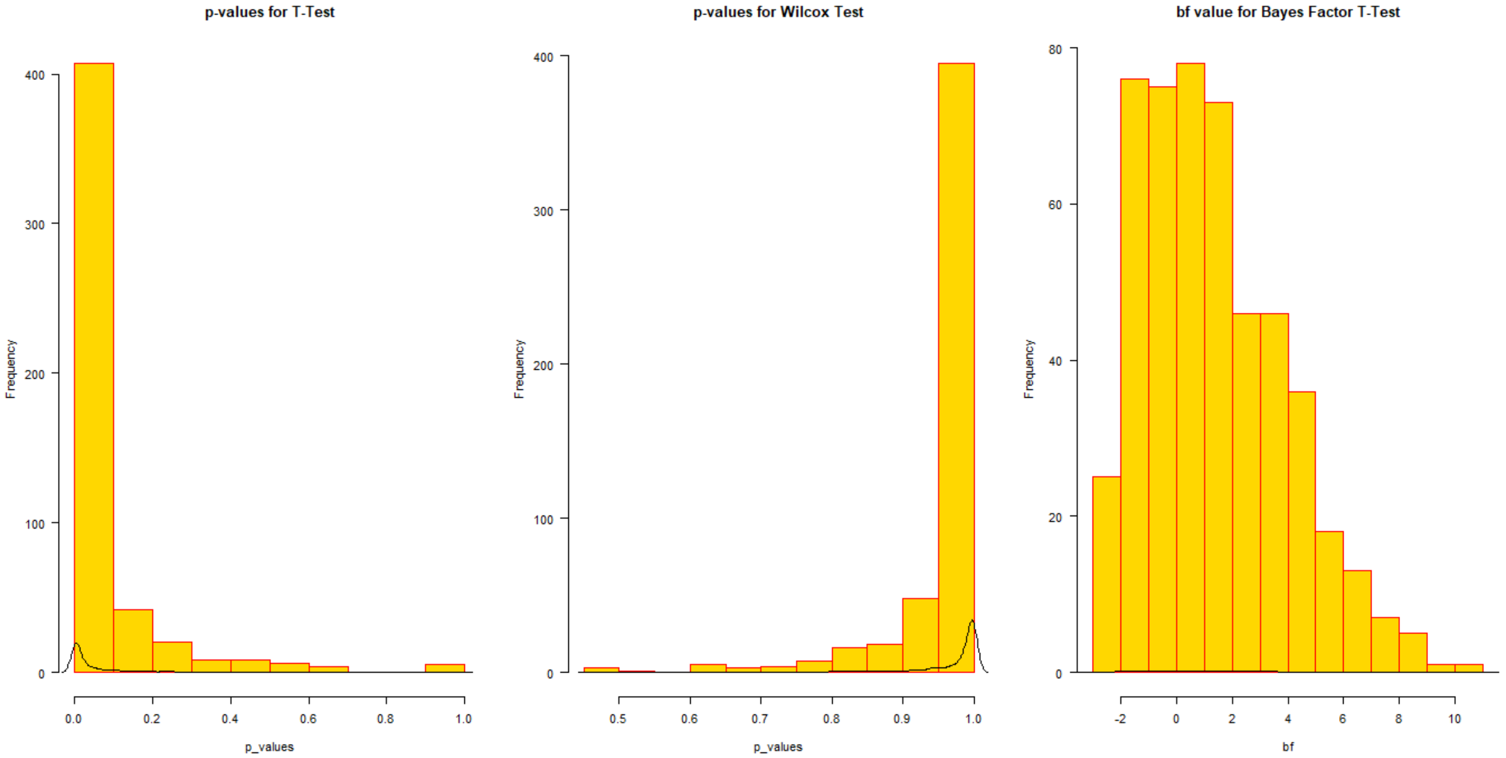

Figure 9: Histogram for all the 3 paired tests

Here I ran the 3 paired tests for 500 times.

For T-test we cannot see any change in the pattern for the paired and the independent test, but for wilcox test we can see that the histogram is a bit short at p-value = 1. Earlier the frequency was 450+ but now its equal to 400.

And for Bayes factor T-test also we can see that the paired tests result is much normally distributed than that of independent tests.

Problem 7: Correlations

```
> p
           p1        p2        p3        p4        p5
cor 0.8910841 0.8910841 0.8143037 0.8910841 0.7206512
> s
           s1        s2        s3        s4        s5
rho 0.8969577 0.8969577 0.8969577 0.8969577 0.7056466
```

Figure 10: p and s are the Pearson and Spearman correlations between x and z values respectively

```
> bf1<-ttestBF(x, z1)
> bf1@bayesFactor$bf
[1] 14.69223
> bf2<-ttestBF(x, z2)
t is large; approximation invoked.
> bf2@bayesFactor$bf
[1] 555.9153
> bf3<-ttestBF(x, z3)
> bf3@bayesFactor$bf
[1] 71.36502
> bf4<-ttestBF(x, z4)
t is large; approximation invoked.
> bf4@bayesFactor$bf
[1] 117.7799
> bf5<-ttestBF(x, z5)
> bf5@bayesFactor$bf
[1] 60.84914
```

Figure 11: Bayes factor for each correlation

Here we can see that among the Pearson correlation output the correlation between x and z3 and between x and z5 is less as compared to that of z1, z2 and z4. And in Spearman correlation we see that only the correlation between x and z5 is less whereas correlation between x and other z values is same.

For Pearson correlation we can see that all the values are greater than 0.5 so we have strong correlation with all the z values but correlation between x and z1, z2 & z4 with form a better straight line than that with z3 & z5.

For spearman correlation we can see that the strength between x and other z values except z5 is more.

Looking at the Bayes factor for each correlation we see that its highest for x and z2 and is lowest for x and z1.

APPENDIX

Problem 1. Understanding the power of a test.

```
set.seed(100)
##this function generates n data points extracts the p/value
##bayes factor of the one-sample two-sided test for each:
simdata1 <- function(n,mean=.1)
{
  data <- rnorm(n,mean=mean)

  c( t.test(data)$p.value,
    binom.test(sum(data>0),n)$p.value,
    exp(ttestBF(data)@bayesFactor$bf))
  ##exponentiate because bf is stored as a log and so 0 is unbiased.

}

runs <- 500
##this simulates 1000 experiments:
null <- data.frame(pval=rep(NA,runs),
                           pvalnp=rep(NA,runs),
                           bayesf=rep(NA,runs))

##this simulates 1000 experiments:
simulation <- data.frame(pval=rep(NA,runs),
                           pvalnp=rep(NA,runs),
                           bayesf=rep(NA,runs))

n<-10   ##this is how many samples are drawn in each experiment
```

```
for(i in 1:runs)
{
  simulation[i,] <- simdata1(n)
  null[i,] <- simdata1(n,0)
}
# Examine the number of significant tests when there is a true difference
mean(simulation$pval < .05) # significance for standard t-test
mean(simulation$pvalnp < .05) # number in binomial test
mean(simulation$bayesf > 2) # bayes factor greater than 2
mean(simulation$bayesf < (-2))

# Examine number of significant tests under the null hypothesis:
mean(null$pval < .05) # number found significant in null distribution
mean(null$pvalnp < .05)# number in binomial test
mean(null$bayesf > 2 )  #evidence for the alternative ## bayes factor greater than 2
mean(null$bayesf < (-2)) ##evidence for the null?
```

```
n<-50   ##this is how many samples are drawn in each experiment
for(i in 1:runs)
{
  simulation[i,] <- simdata1(n)
  null[i,] <- simdata1(n,0)
}
# Examine the number of significant tests when there is a true difference
mean(simulation$pval < .05) # significance for standard t-test
mean(simulation$pvalnp < .05) # number in binomial test
mean(simulation$bayesf > 2) # bayes factor greater than 2
mean(simulation$bayesf < (-2))

# Examine number of significant tests under the null hypothesis:
mean(null$pval < .05) # number found significant in null distribution
mean(null$pvalnp < .05)# number in binomial test
mean(null$bayesf > 2 )  #evidence for the alternative ## bayes factor greater than 2
mean(null$bayesf < (-2)) ##evidence for the null?
```

```
n<-200   ##this is how many samples are drawn in each experiment
for(i in 1:runs)
{
  simulation[i,] <- simdata1(n)
  null[i,] <- simdata1(n,0)
}
# Examine the number of significant tests when there is a true difference
mean(simulation$pval < .05) # significance for standard t-test
mean(simulation$pvalnp < .05) # number in binomial test
mean(simulation$bayesf > 2) # bayes factor greater than 2
mean(simulation$bayesf < (-2))

# Examine number of significant tests under the null hypothesis:
mean(null$pval < .05) # number found significant in null distribution
mean(null$pvalnp < .05)# number in binomial test
mean(null$bayesf > 2 )  #evidence for the alternative ## bayes factor greater than 2
mean(null$bayesf < (-2)) ##evidence for the null?
```

```
n<-1000   ##this is how many samples are drawn in each experiment
```

```
for(i in 1:runs)
{
  simulation[i,] <- simdata1(n)
  null[i,] <- simdata1(n,0)
}
# Examine the number of significant tests when there is a true difference
mean(simulation$pval < .05) # significance for standard t-test
mean(simulation$pvalnp < .05) # number in binomial test
mean(simulation$bayesf > 2) # bayes factor greater than 2
mean(simulation$bayesf < (-2))

# Examine number of significant tests under the null hypothesis:
mean(null$pval < .05) # number found significant in null distribution
mean(null$pvalnp < .05)# number in binomial test
mean(null$bayesf > 2 )  #evidence for the alternative ## bayes factor greater than 2
mean(null$bayesf < (-2)) ##evidence for the null?
```

```
n<-10000   ##this is how many samples are drawn in each experiment
for(i in 1:runs)
{
  simulation[i,] <- simdata1(n)
  null[i,] <- simdata1(n,0)
}
# Examine the number of significant tests when there is a true difference
mean(simulation$pval < .05) # significance for standard t-test
mean(simulation$pvalnp < .05) # number in binomial test
mean(simulation$bayesf > 2) # bayes factor greater than 2
mean(simulation$bayesf < (-2))

# Examine number of significant tests under the null hypothesis:
mean(null$pval < .05) # number found significant in null distribution
mean(null$pvalnp < .05)# number in binomial test
mean(null$bayesf > 2 )  #evidence for the alternative ## bayes factor greater than 2
mean(null$bayesf < (-2)) ##evidence for the null?
```

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)))

# 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)))
```

```r
# checking with t.test() function
t.test(data_q2, mu=100, alt="greater")

# 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))

#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))

t.test(data_q2B, mu=100, alt="greater")
```

Problem 3. Wilcox test

```r
x <- rnorm(20)
y <- sort(x)

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

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

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

```r
library(BayesFactor)

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)
}

print(t_values)
```

```r
print(w_values)

print(b_values)

# 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 5. Robustness to transforms

```r
library(BayesFactor)
```

```r
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.

```r
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
p1 <- cor.test(x,z1,method="pearson")
s1 <- cor.test(x,z1,method="spearman")

#x and z2
p2 <- cor.test(x,z2,method="pearson")
s2 <- cor.test(x,z2,method="spearman")

#x and z3
p3 <- cor.test(x,z3,method="pearson")
s3 <- cor.test(x,z3,method="spearman")
```

```
#x and z4
p4 <- cor.test(x,z4,method="pearson")
s4 <- cor.test(x,z4,method="spearman")

#x and z5
p5 <- cor.test(x,z5,method="pearson")
s5 <- cor.test(x,z5,method="spearman")

p <- cbind(p1=p1$estimate, p2=p2$estimate, p3=p3$estimate, p4=p4$estimate, p5=p5$estimate)
s <- cbind(s1=s1$estimate, s2=s2$estimate, s3=s3$estimate, s4=s4$estimate, s5=s5$estimate)

p
s

#Bayes factor
bf1<-ttestBF(x, z1)
bf1@bayesFactor$bf
bf2<-ttestBF(x, z2)
bf2@bayesFactor$bf
bf3<-ttestBF(x, z3)
bf3@bayesFactor$bf
bf4<-ttestBF(x, z4)
bf4@bayesFactor$bf
bf5<-ttestBF(x, z5)
bf5@bayesFactor$bf
```