

# Stats Homework 1

Scenario: As a researcher you are interested in understanding how two methods of inspecting code work. One method uses a checklist, and the other is a method called perspective-based reading (PBR). We have provide simulated data for an experiment comparing these inspections methods (Note: Be sure to download a local copy of the dataset before proceeding).

```
require(ggplot2)
install.packages("reshape2")

## Installing package into 'C:/Users/delyar/Documents/R/win-library/4.1'
## (as 'lib' is unspecified)

## package 'reshape2' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
## C:\Users\delyar\AppData\Local\Temp\RtmpADMA45\downloaded_packages
```

```
install.packages('nortest')
```

```
## Error in install.packages : Updating loaded packages
```

## Getting help

To get help in R about a function, for example `boxplot`, type `?boxplot` in the command line.

## Loading the data

For this part, load the inspection data (“inspection.csv”) file located in the assignment folder with this file.

```
# code goes here
my_dataset <- read.csv("C:\\Users\\delyar\\Desktop\\CS 567\\Stats HW 1\\inspection.csv")
my_dataset
```

## Plotting

You would like to know the descriptive statistics of the two inspection methods. Compare the samples via their mean, median, and box-plot distributions.

```
# code goes here  
mean(my_dataset$pbr)
```

```
## [1] 19.73333
```

```
mean(my_dataset$checklist)
```

```
## [1] 21.43333
```

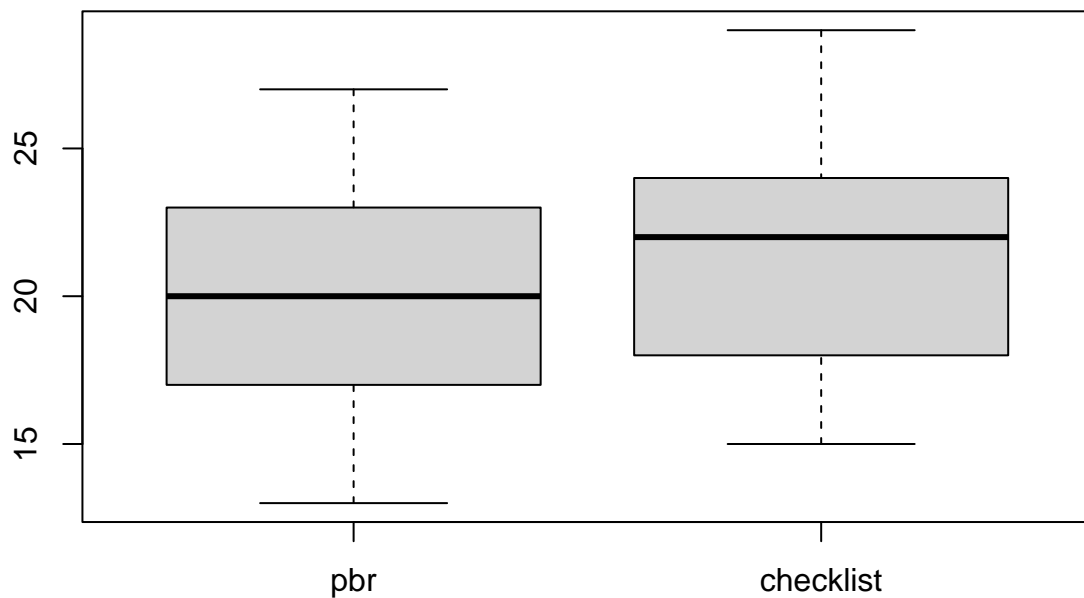
```
median(my_dataset$pbr)
```

```
## [1] 20
```

```
median(my_dataset$checklist)
```

```
## [1] 22
```

```
boxplot(my_dataset)
```



## Normality

You want to see if your data is normally distributed. Hint: You can use Shapiro-Wilk or Anderson-Darling. Justify which is more appropriate.

```
#Shapiro: The Prob < W value listed in the output is the p-value. If the chosen alpha level is 0.05 and  
shapiro.test(my_dataset$pbr)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: my_dataset$pbr  
## W = 0.97343, p-value = 0.6365
```

```
shapiro.test(my_dataset$checklist)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: my_dataset$checklist  
## W = 0.95113, p-value = 0.1812
```

```
# here we have p values of 0.63 and 0.18 respectively for pbr and checklist. both of them are bigger than 0.05  
#if you have smaller sample size, Shapiro test is more preferred.
```

```
library(nortest)  
ad.test(my_dataset$pbr)
```

```
##  
## Anderson-Darling normality test  
##  
## data: my_dataset$pbr  
## A = 0.34042, p-value = 0.4734
```

```
ad.test(my_dataset$checklist)
```

```
##  
## Anderson-Darling normality test  
##  
## data: my_dataset$checklist  
## A = 0.46139, p-value = 0.2415
```

```
#also for the ad test we can see that the p values are 0.47 and 0.24 that are bigger than 0.05 so we can conclude
```

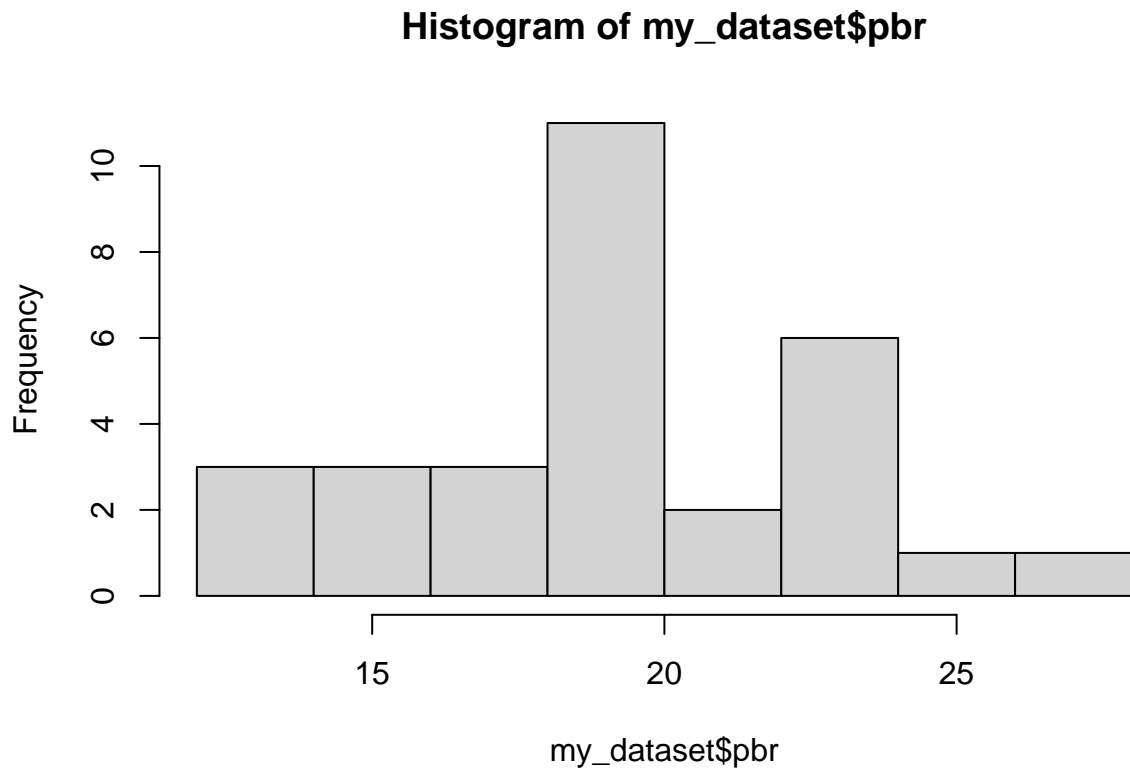
## Bootstrapping

You would like to do “bootstrap” your data to make sure that data parameters are robust. Bootstrapping is a statistical method for estimating the sampling distribution by sampling with replacement from the original

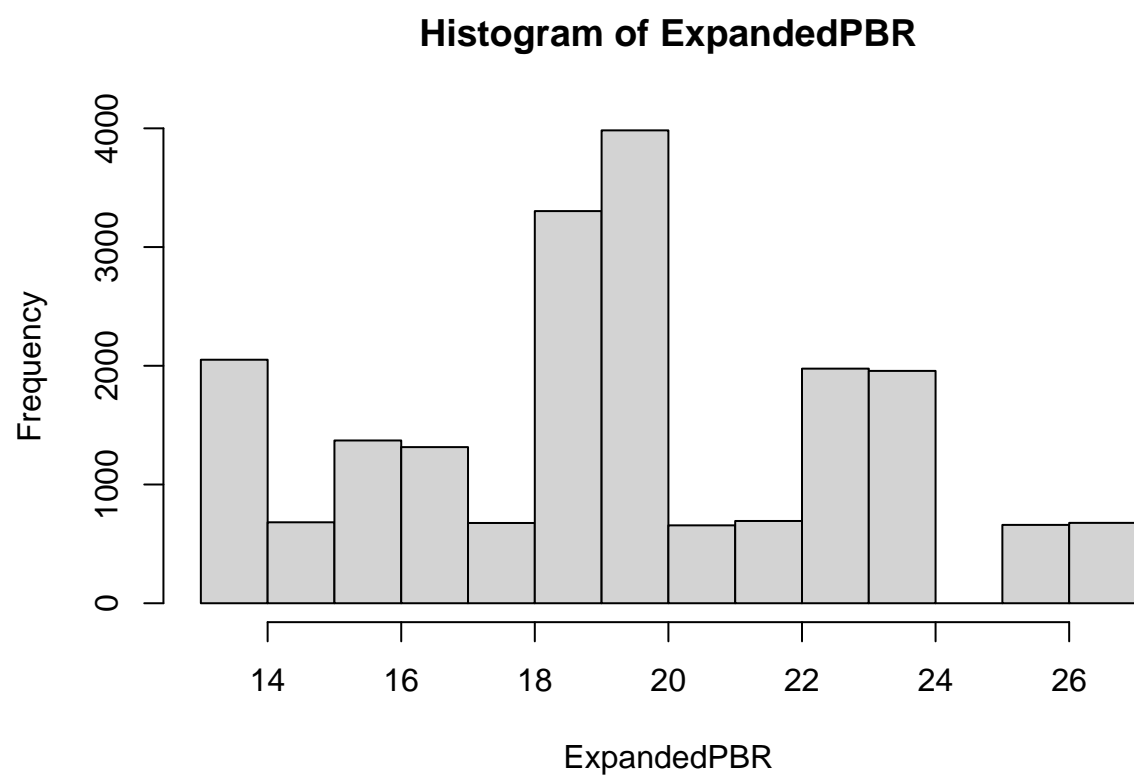
sample. Note: You will need to do this to expand your “term project data” to include enough data for analysis.

Bootstrap the data. Then compare and contrast the original dataset with the bootstrap (use descriptive statistics as before).

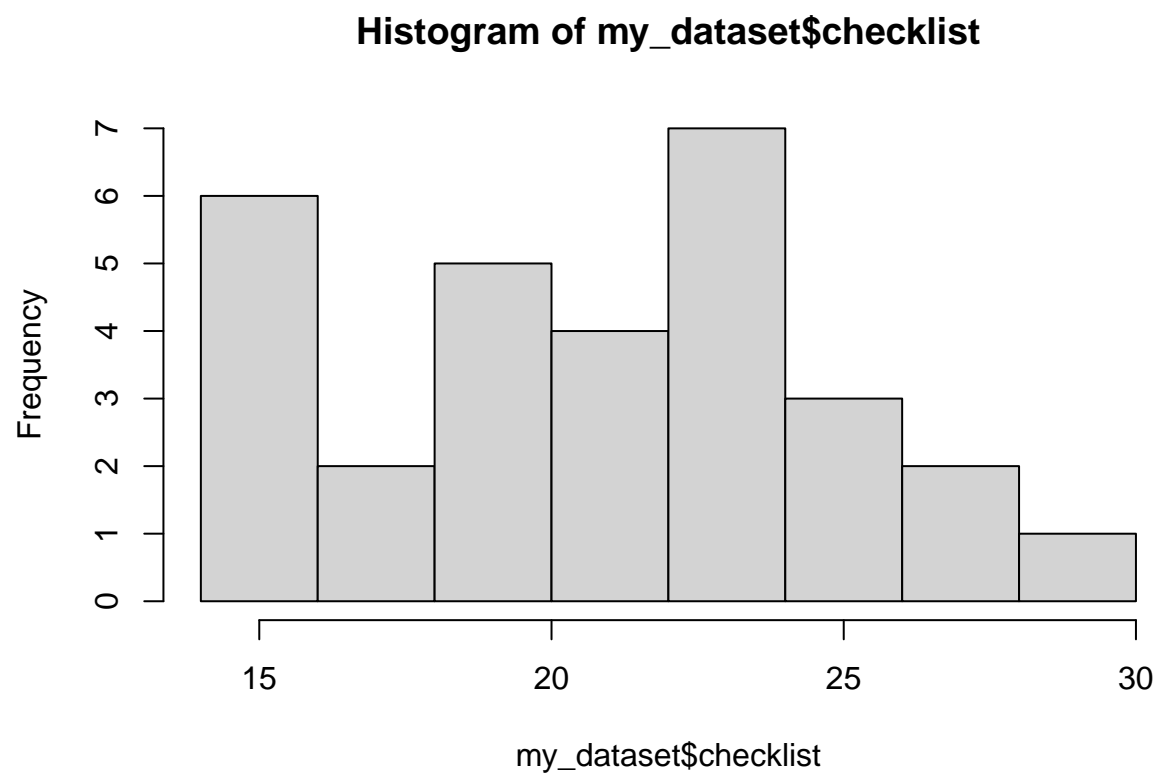
```
# Step 1: Randomly resample data points for each treatment 20000 times (hint: you can use sample or re  
  
ExpandedPBR <- sample(my_dataset$pbr, 20000, replace = TRUE , prob = NULL)  
ExpandedChecklist <- sample(my_dataset$checklist, 20000, replace = TRUE , prob = NULL)  
  
# Step 2: Draw the histogram to compare the original with the bootstrap data for each treatment separate  
  
hist(my_dataset$pbr)
```



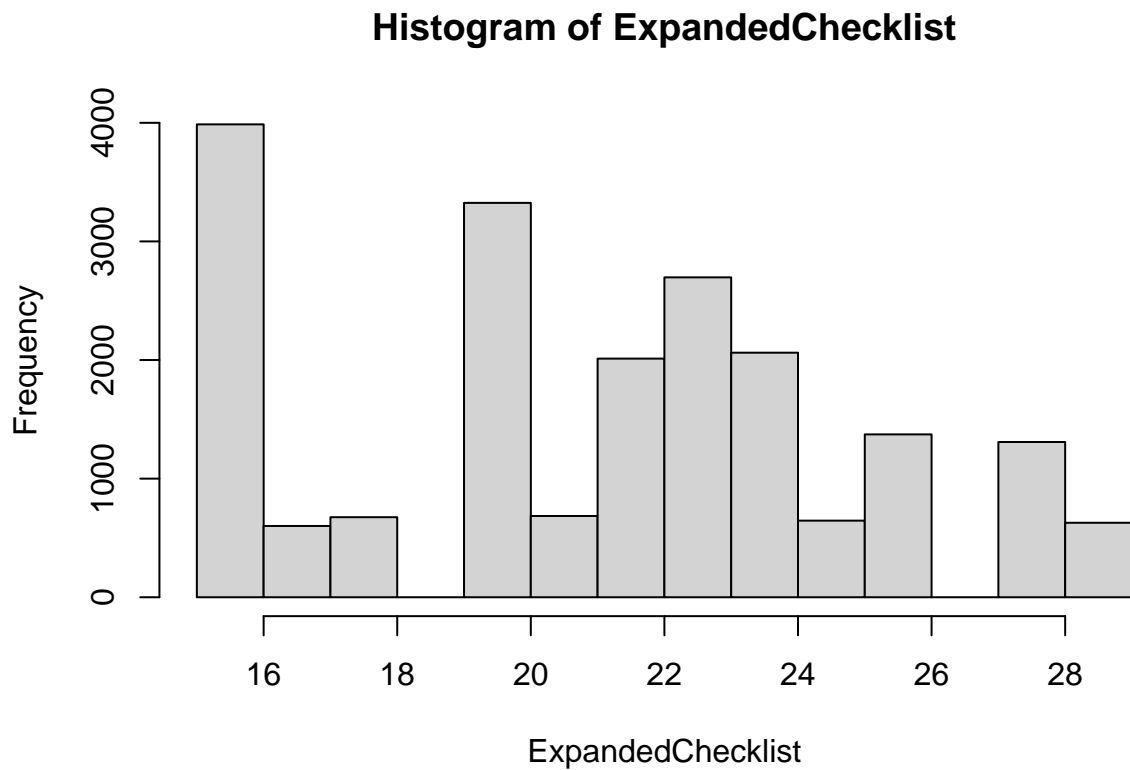
```
hist(ExpandedPBR)
```



```
hist(my_dataset$checklist)
```



```
hist(ExpandedChecklist)
```



*# Step 3: Check the normality of the bootstrapped data.*

*# from the visual method of the histograms we can tell that they are not sample but we can use AD test*

```
ad.test(ExpandedPBR)
```

```
##
## Anderson-Darling normality test
##
## data: ExpandedPBR
## A = 233.27, p-value < 2.2e-16
```

```
ad.test(ExpandedChecklist)
```

```
##
## Anderson-Darling normality test
##
## data: ExpandedChecklist
## A = 324.83, p-value < 2.2e-16
```

*#from the AD test we have p-value < 2.2e-16 --> the null hypothesis that the data are normally distributed is rejected*

*# Step 4: Compare the descriptive statistics of original with the bootstrapped data.*

```
mean(my_dataset$pbr)
```

```
## [1] 19.73333
```

```
mean(my_dataset$checklist)
```

```
## [1] 21.43333
```

```
mean(ExpandedPBR)
```

```
## [1] 19.7024
```

```
mean(ExpandedChecklist)
```

```
## [1] 21.44345
```

```
median(my_dataset$pbr)
```

```
## [1] 20
```

```
median(my_dataset$checklist)
```

```
## [1] 22
```

```
median(ExpandedPBR)
```

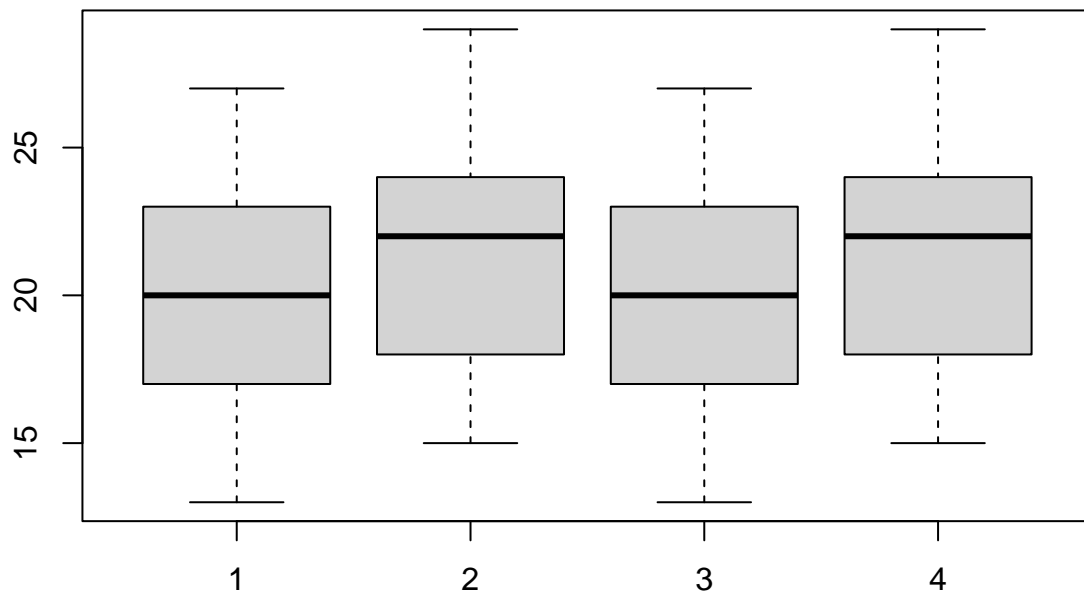
```
## [1] 20
```

```
median(ExpandedChecklist)
```

```
## [1] 22
```

```
boxplot(my_dataset$pbr, my_dataset$checklist, ExpandedPBR, ExpandedChecklist)
```





In the rest of the HW, we will use the original dataset.

*#dataFormatting* To run statistics you need your data needs to be 'reshaped' to look like this: "","treatment","time" "1","pbr",20 ..... "2","checklist",19

*#code goes here (hint: use melt or reshape)*

`library("reshape2")`

*#one way:*

`melted <- melt(my_dataset)`

`## No id variables; using all as measure variables`

*#another way (I prefer this cause it has names for the variables and the values)*

`Compare <- data.frame(treatment=rep(c("pbr", "checklist"), each=30), time=c(my_dataset$pbr, my_dataset$checklist))`  
`Compare`

## T-tests

Now you would like to statistically compare the mean time used for two inspection methods. Test and report for significance at 0.05.

- Perform a two-tailed t-test (assume the variances are equal).

```
# code goes here
t.test(Compare$time ~ Compare$treatment, alternative = "two.sided", var.equal = FALSE)
```

```
##
## Welch Two Sample t-test
##
## data: Compare$time by Compare$treatment
## t = 1.7434, df = 57.299, p-value = 0.08663
## alternative hypothesis: true difference in means between group checklist and group pbr is not equal to 0
## 95 percent confidence interval:
## -0.2524068 3.6524068
## sample estimates:
## mean in group checklist      mean in group pbr
##           21.43333           19.73333
```

```
#if the variances are equal we can also have it like this:
t.test(my_dataset$pbr ,my_dataset$checklist, alternative = "two.sided", var.equal=TRUE)
```

```
##
## Two Sample t-test
##
## data: my_dataset$pbr and my_dataset$checklist
## t = -1.7434, df = 58, p-value = 0.08656
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -3.6518982 0.2518982
## sample estimates:
## mean of x mean of y
## 19.73333 21.43333
```

```
#we have p-value = 0.08656 --> higher than 0.05 --> not statistically significant
```

- b) Perform a one-tailed t-test (assume PBR takes less time than checklist, variances are equal) and check if results are statistically significant.

```
# code goes here
t.test(Compare$time ~ Compare$treatment, alternative = "less", var.equal = TRUE)
```

```
##
## Two Sample t-test
##
## data: Compare$time by Compare$treatment
## t = 1.7434, df = 58, p-value = 0.9567
## alternative hypothesis: true difference in means between group checklist and group pbr is less than 0
## 95 percent confidence interval:
## -Inf 3.329951
## sample estimates:
## mean in group checklist      mean in group pbr
##           21.43333           19.73333
```

*#p-value = 0.9567 --> higher than 0.05 --> not statistically significant*

- c) Assume that in the study subjects were paired together by experience level and comparisons are done within pairs, and use a paired (two-tailed) t-test to check if the results are statistically significant.

*# code goes here*

```
t.test(Compare$time ~ Compare$treatment, alternative = "two.sided", paired=TRUE, var.equal=TRUE)

##
## Paired t-test
##
## data: Compare$time by Compare$treatment
## t = 2.1146, df = 29, p-value = 0.04318
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.05575514 3.34424486
## sample estimates:
## mean of the differences
##                1.7
```

- d) Re-do parts a,b,c using non-parametric tests instead (Wilcoxon tests, also known as Mann-Whitney) and compare the p-values to what you originally obtained.

*# code goes here for all 3 cases*

```
wilcox.test(Compare$time ~ Compare$treatment, mu=0, alt="two.sided", conf.int=T, conf.level=0.95, paired = F)
```

```
##
## Wilcoxon rank sum test with continuity correction
##
## data: Compare$time by Compare$treatment
## W = 562.5, p-value = 0.09584
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -1.605095e-05 3.999992e+00
## sample estimates:
## difference in location
##                1.999932
```

```
wilcox.test(Compare$time ~ Compare$treatment, mu=0, alt="less", conf.int=T, conf.level=0.95, paired = F)
```

```
##
## Wilcoxon rank sum test with continuity correction
##
## data: Compare$time by Compare$treatment
## W = 562.5, p-value = 0.9535
## alternative hypothesis: true location shift is less than 0
## 95 percent confidence interval:
##      -Inf 3.000043
## sample estimates:
## difference in location
##                1.999932
```

```
wilcox.test(Compare$time ~ Compare$treatment, mu=0, alt="two.sided", conf.int=T, conf.level=0.95, paired=TRUE)
```

```
##
## Wilcoxon signed rank test with continuity correction
##
## data: Compare$time by Compare$treatment
## V = 312, p-value = 0.04143
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
## 5.130501e-05 3.500051e+00
## sample estimates:
## (pseudo)median
## 1.999937
```

*#another way:*

```
wilcox.test(Compare$time ~ Compare$treatment, alternative = "two.sided", var.equal=TRUE, var.equal=TRUE)
```

```
##
## Wilcoxon rank sum test with continuity correction
##
## data: Compare$time by Compare$treatment
## W = 562.5, p-value = 0.09584
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(Compare$time ~ Compare$treatment, alternative="less", var.equal=TRUE, exact = FALSE)
```

```
##
## Wilcoxon rank sum test with continuity correction
##
## data: Compare$time by Compare$treatment
## W = 562.5, p-value = 0.9535
## alternative hypothesis: true location shift is less than 0
```

```
wilcox.test(Compare$time ~ Compare$treatment, alternative = "two.sided", paired=TRUE, exact = FALSE, var.equal=TRUE)
```

```
##
## Wilcoxon signed rank test with continuity correction
##
## data: Compare$time by Compare$treatment
## V = 312, p-value = 0.04143
## alternative hypothesis: true location shift is not equal to 0
```

*#only for part c the p-value is less than 0.05 -->If a p-value reported from a t test is less than 0.05*

““