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).

```
library(reshape2) # for formatting and aggregation of data frames
library(ggplot2) # for creating graphs
library(dplyr) # for data manipulation and clean-up
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(plotly) # for creating interactive web graphics from ggplot2 graphs
```

```
##
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
##
##     last_plot

## The following object is masked from 'package:stats':
##
##     filter

## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(knitr) # required for generating PDF output
library(nortest)
```

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
inspection <- read.csv("C:/Users/nallagop/Desktop/Fall 2021/CS 567 Lab Studies/Assigments/Stats HW 1/ins
inspection
```

```
##    pbr checklist
## 1   20        28
## 2   19        20
## 3   13        20
## 4   17        28
## 5   21        16
## 6   14        16
## 7   19        23
## 8   17        24
## 9   24        20
## 10  23        22
## 11  18        26
## 12  24        29
## 13  23        22
## 14  20        23
## 15  15        16
## 16  22        26
## 17  27        23
## 18  20        22
## 19  16        18
## 20  16        20
## 21  19        16
## 22  23        24
## 23  26        25
## 24  19        23
## 25  20        17
## 26  19        24
## 27  20        20
## 28  14        16
## 29  24        15
## 30  20        21
```

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(inspection$pbr)
```

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

```r
median(inspection$pbr)
```

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

```r
mean(inspection$checklist)
```

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

```r
median(inspection$checklist)
```

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

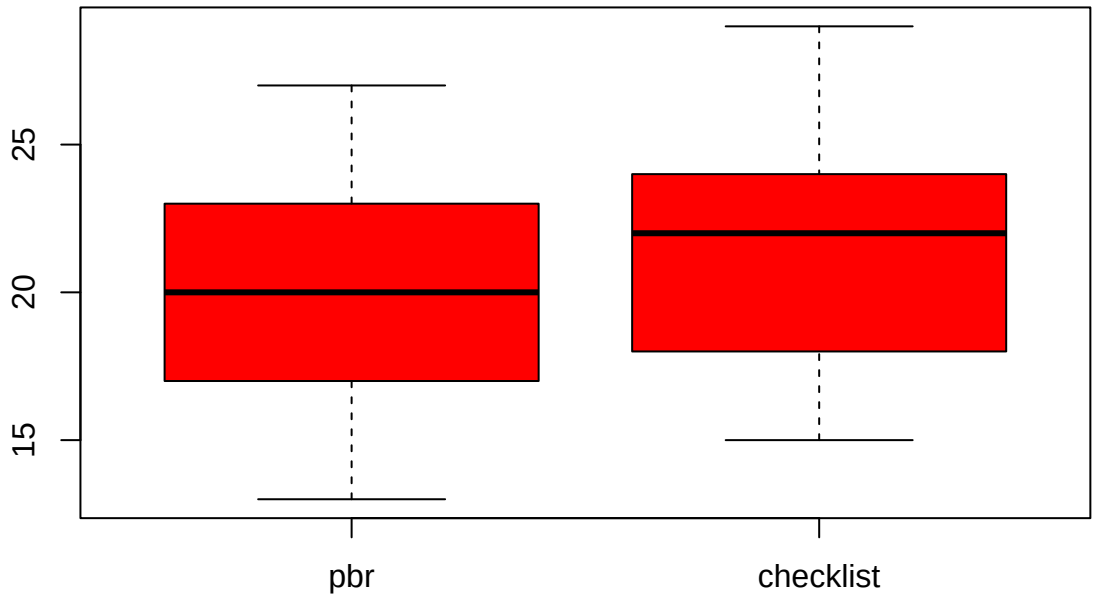
```r
summary(inspection$pbr)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   13.00   17.25   20.00   19.73   22.75   27.00
```

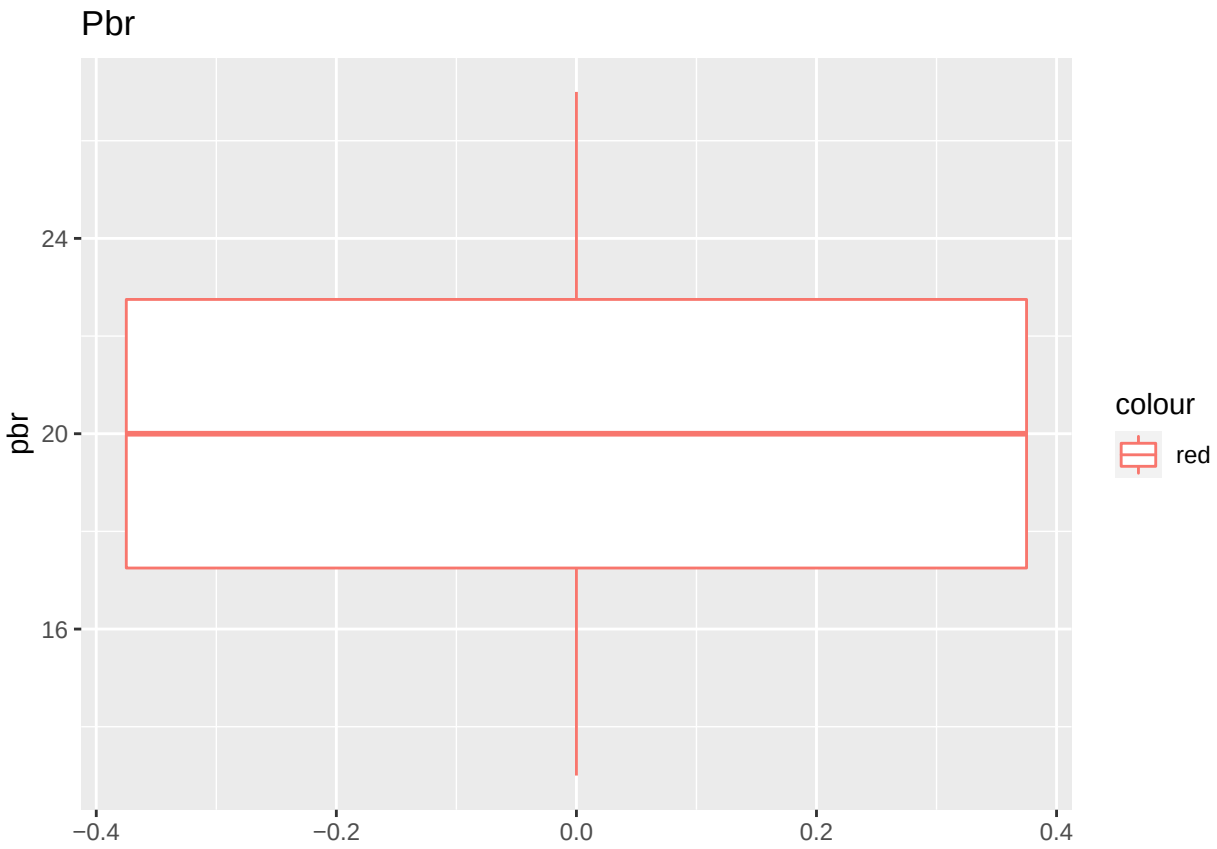
```r
summary(inspection$checklist)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   15.00   18.50   22.00   21.43   24.00   29.00
```

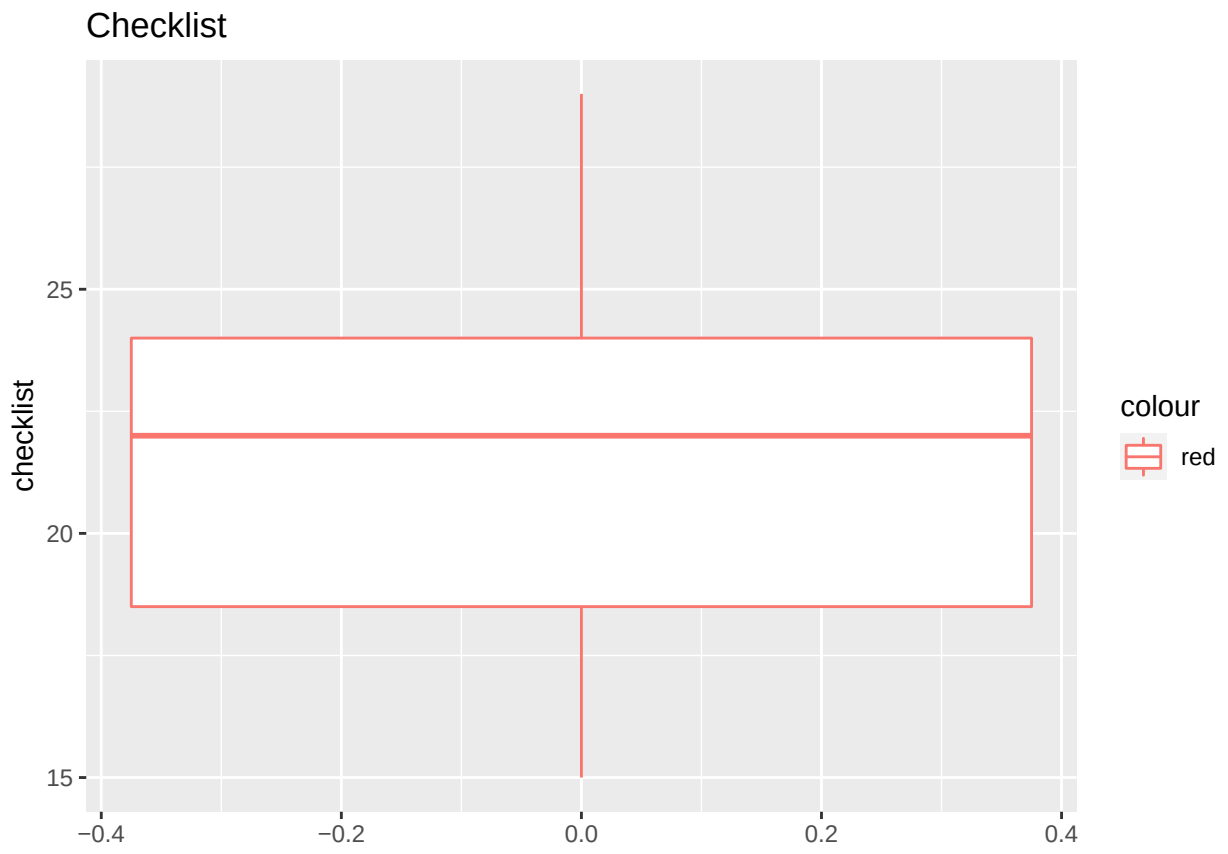
```r
#mean(inspection$pbr) is less than mean(inspection$checklist)
#median(inspection$pbr) is less than median(inspection$checklist)

boxplot(inspection, col = "red")
```




```
#boxplot - separately
ggplot(inspection, aes(y = pbr, col = "red")) + geom_boxplot() + labs(title = "Pbr", y = "pbr")
```




```
ggplot(inspection, aes(y = checklist, col = "red")) + geom_boxplot() + labs(title = "Checklist", y = "cl
```



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.

```
#code goes here

# Since the p-value is > 0.05 for both pbr and checklist we can use normality
#(are Normally distributed). Also, #Shapiro-wilk has more power when it comes to
#small samples (count <= 50), so it is better to use Shapiro-wilk than using
#Anderson-Darling. However, when it comes to larger values we can use Anderson-Darling.

shapiro.test(inspection$pbr)
```

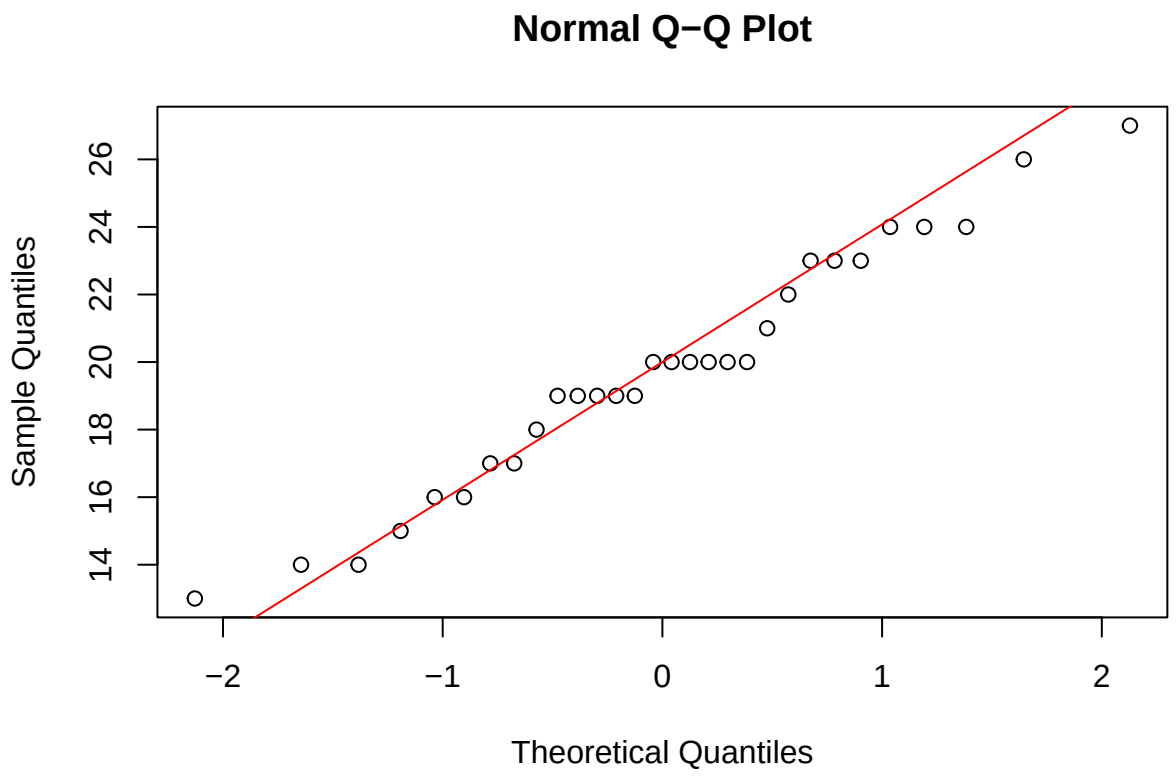
```
##
##  Shapiro-Wilk normality test
##
## data:  inspection$pbr
## W = 0.97343, p-value = 0.6365
```

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

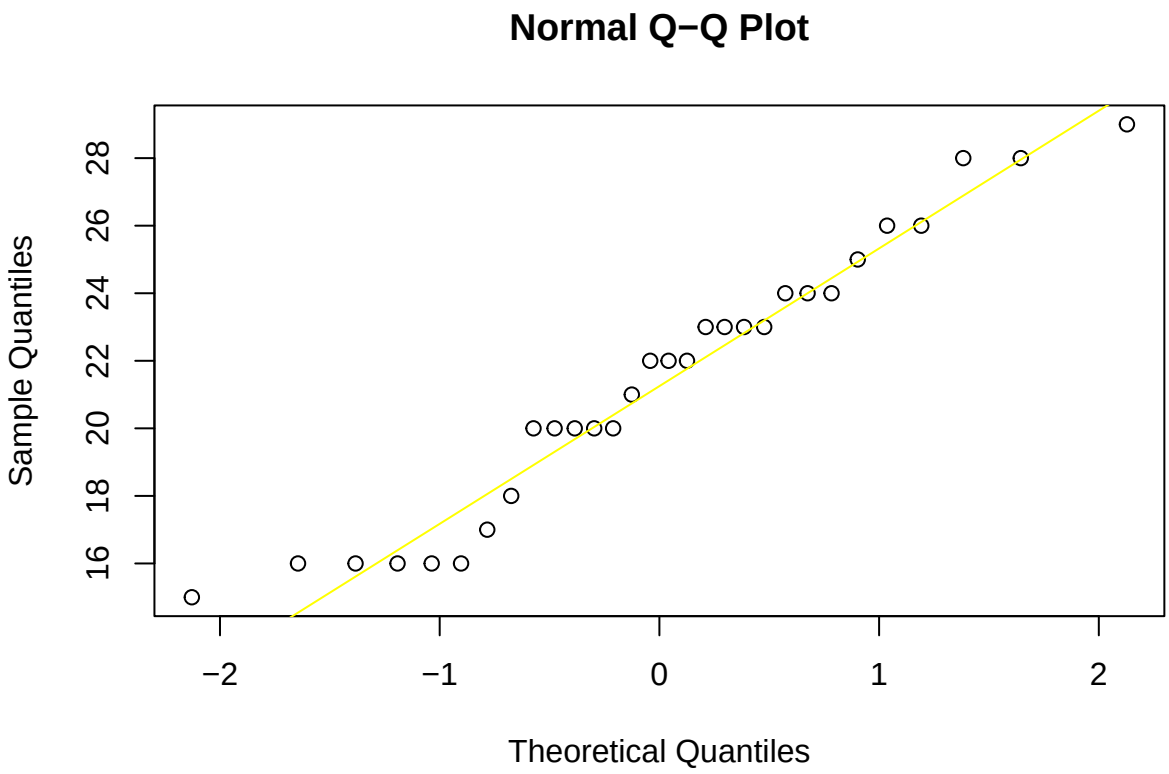
```
##
```

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

```
#From the output, the p-value > 0.05 implying that the distribution of the data
#are not significantly different from #normal distribution. In other words,
#we can assume the normality and they are statistically significant.

#In addition to this, the best approach is to combine visual inspection and
#statistical tests to ensure normality. #So, I've used QQ plot for visual
#inspection.

qqnorm(inspection$pbr)
qqline(inspection$pbr, col = "red")
```




```
qqnorm(inspection$checklist)
qqline(inspection$checklist, col = "yellow")
```




```
#if we want to use Anderson-Darling, this is how we can implement it.
ad.test(inspection$pbr)
```

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

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

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

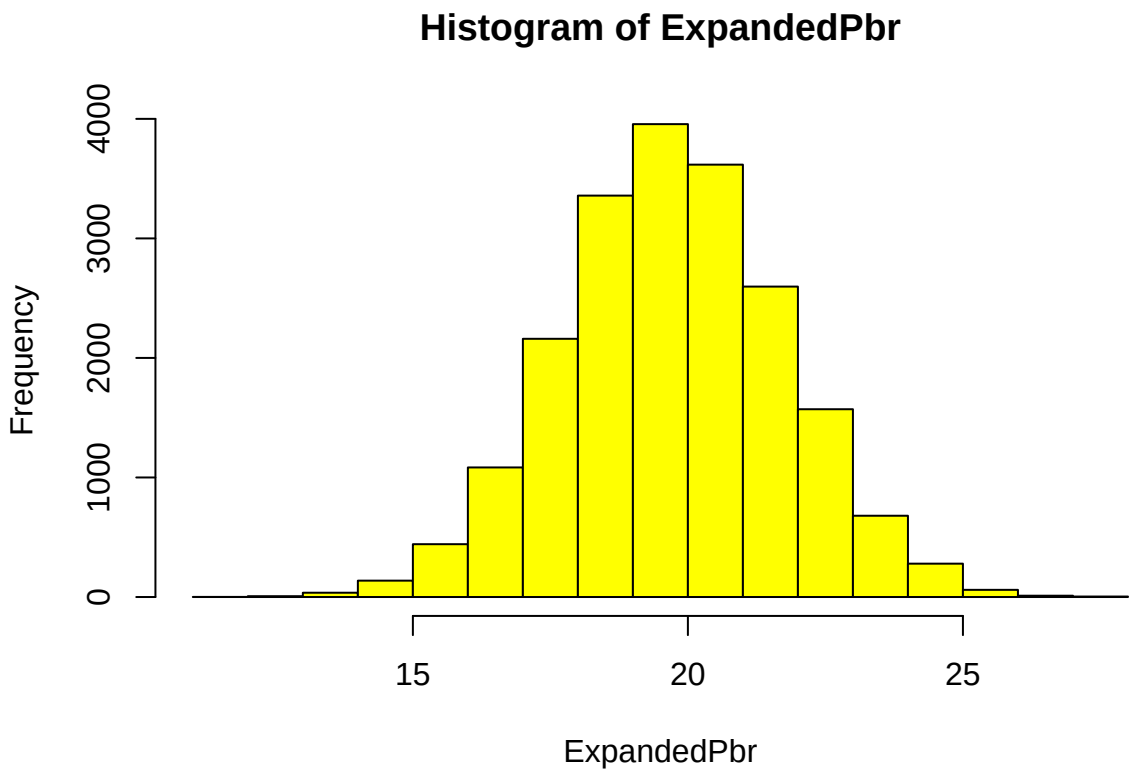
Bootstrapping

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.

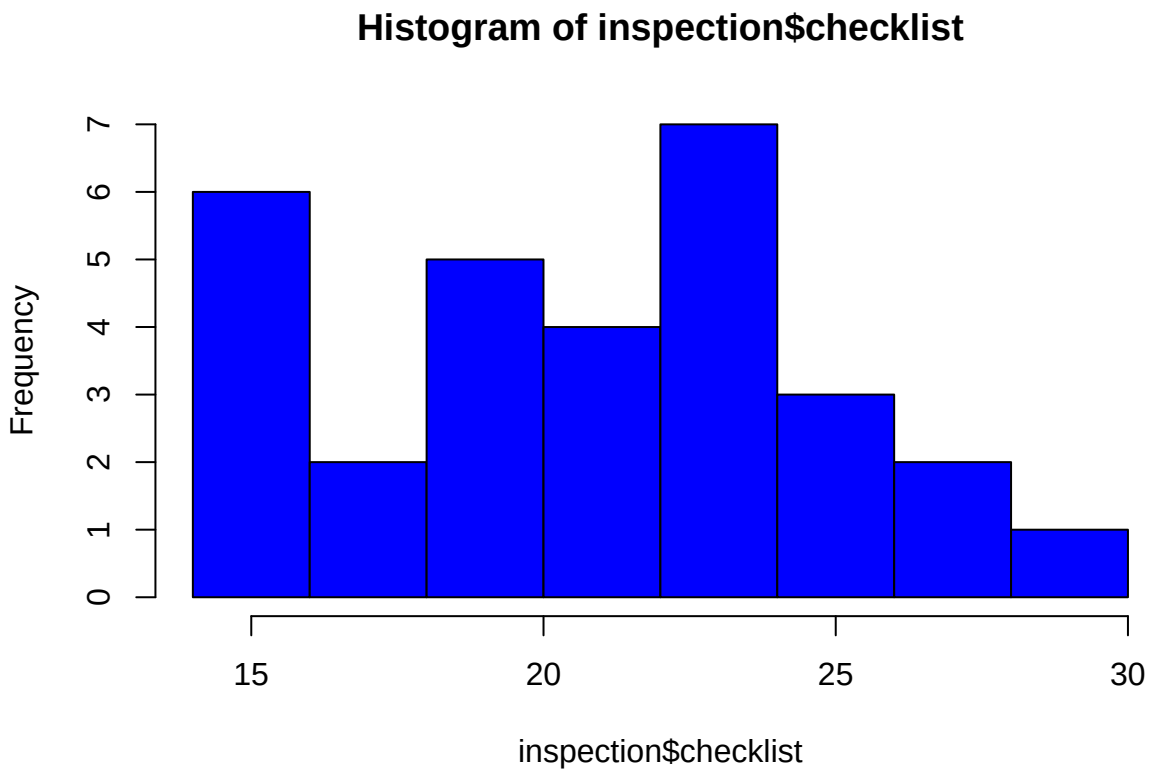
Bootstrap the data. Then compare and contrast the original dataset with the bootstrap (use descriptive statistics as before).

```
# Step 1: Randomly resample data points for each treatment 20000 times  (hint: you can use sample or re

#used set.seed(1) so that it will start from 1 and there will be random numbers
#for every execution. rnorm() is used #to randomly resample data points for each
#treatment 20000 times.
set.seed(1)

ExpandedPbr <- rnorm(20000, mean = mean(inspection$pbr), sd = 2)
ExpandedChecklist <-  rnorm(20000, mean = mean(inspection$checklist), sd = 2)
ExpandedInspection <- data.frame(pbr = ExpandedPbr, checklist = ExpandedChecklist)

# when I use sample(), I'm getting a histogram which is not a bell curve,
#however, the mean of original and expanded data remains the same for both pbr
#and checklist.
#ExpandedPbr <- sample((inspection$pbr),size = 40000, replace = TRUE)
#ExpandedChecklist <- sample((inspection$checklist),size = 40000, replace = TRUE)

# Step 2: Draw the histogram to compare the orignal with the bootstrap data for
#each treatment separately (hint: use #`hist`)
hist(inspection$pbr, col = "red")
```

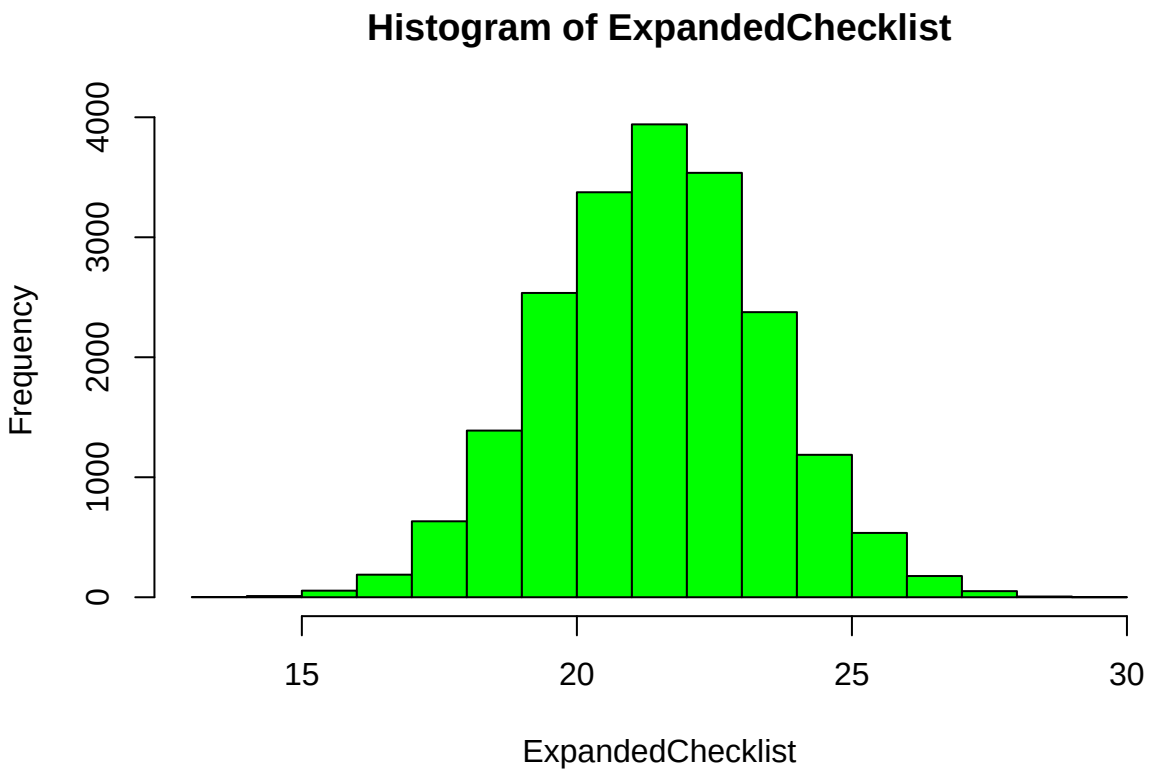
```
hist(ExpandedPbr, col = "yellow")
```




```
hist(inspection$checklist, col = "blue")
```




```
hist(ExpandedChecklist, col = "green")
```



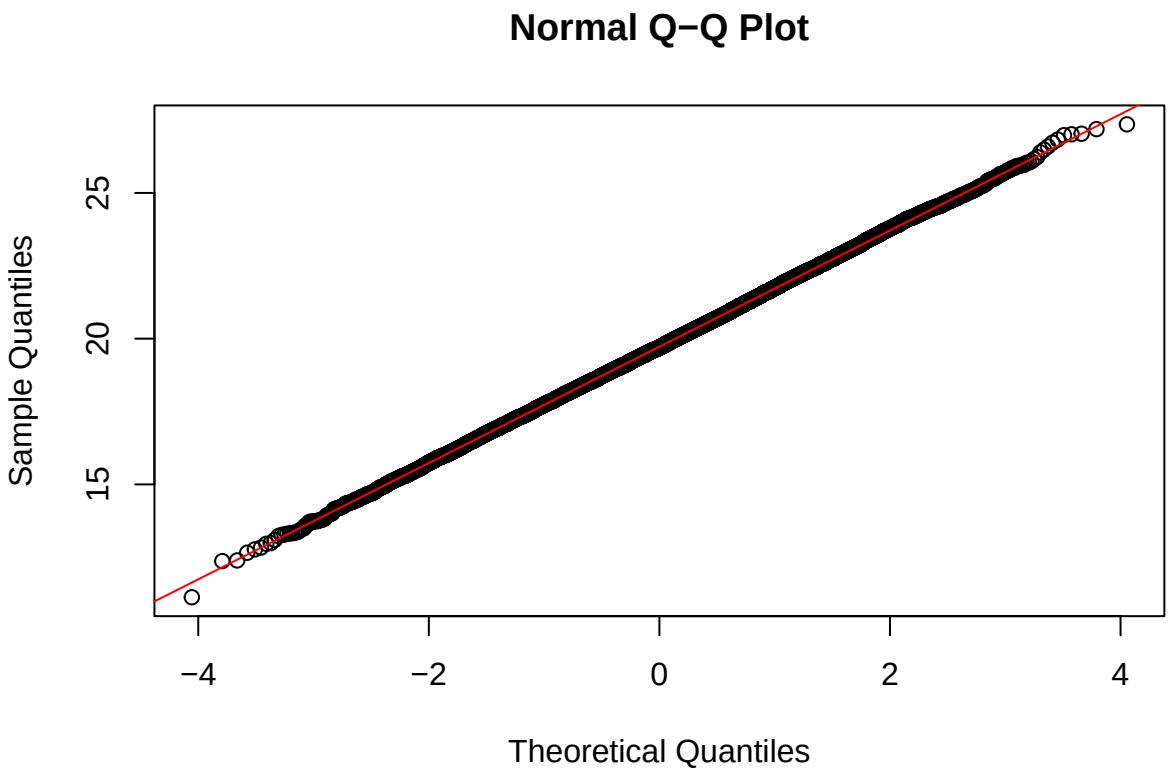

```
# Step 3: Check the normality of the bootstrapped data.
# For larger values it is always recommended to use Anderson-Darling to check
#the normality of the bootstrapped #data.
ad.test(ExpandedPbr)
```

```
##
##  Anderson-Darling normality test
##
## data:  ExpandedPbr
## A = 0.62077, p-value = 0.1061
```

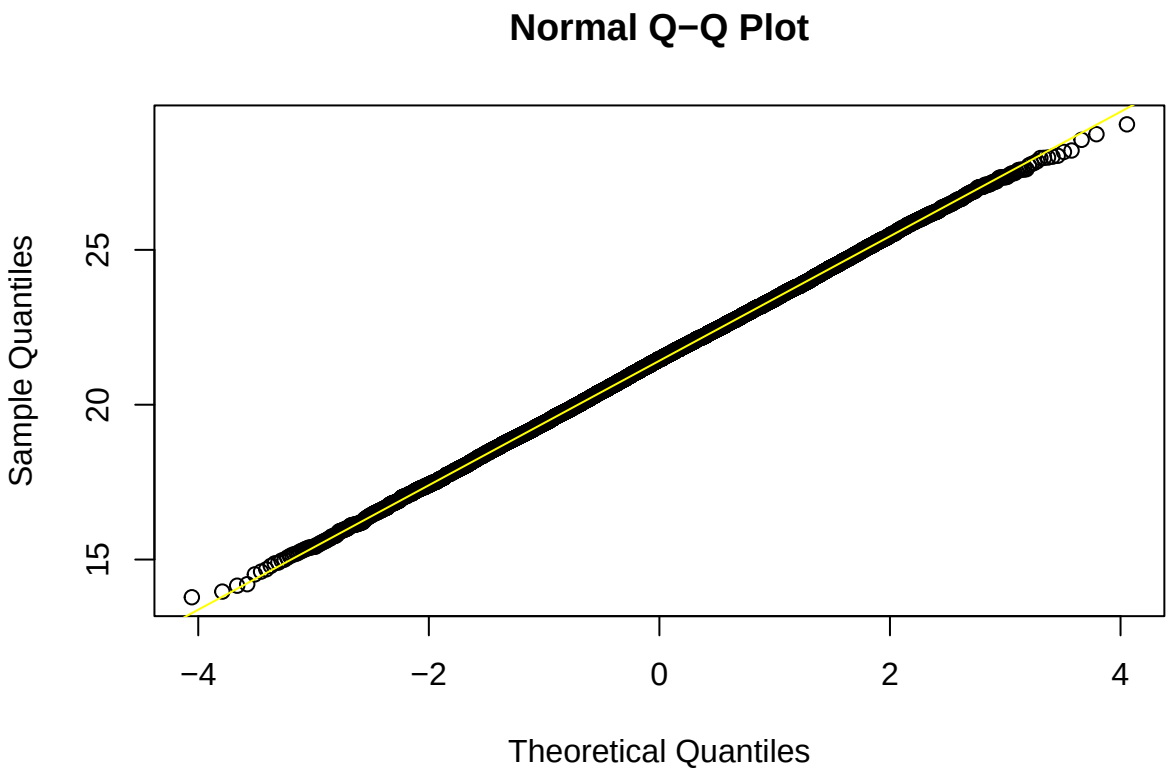
```
ad.test(ExpandedChecklist)
```

```
##
##  Anderson-Darling normality test
##
## data:  ExpandedChecklist
## A = 0.7903, p-value = 0.04049
```

```
qqnorm(ExpandedPbr)
qqline(ExpandedPbr, col = "red")
```




```
qqnorm(ExpandedChecklist)
qqline(ExpandedChecklist, col = "yellow")
```




```
# Step 4: Compare the descriptive statistics of original with the bootstrapped data.
mean(inspection$pbr)
```

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

```
mean(ExpandedPbr)
```

```
## [1] 19.72261
```

```
mean(inspection$checklist)
```

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

```
mean(ExpandedChecklist)
```

```
## [1] 21.43534
```

```
median(inspection$pbr)
```

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

```
median(ExpandedPbr)
```

```
## [1] 19.69821
```

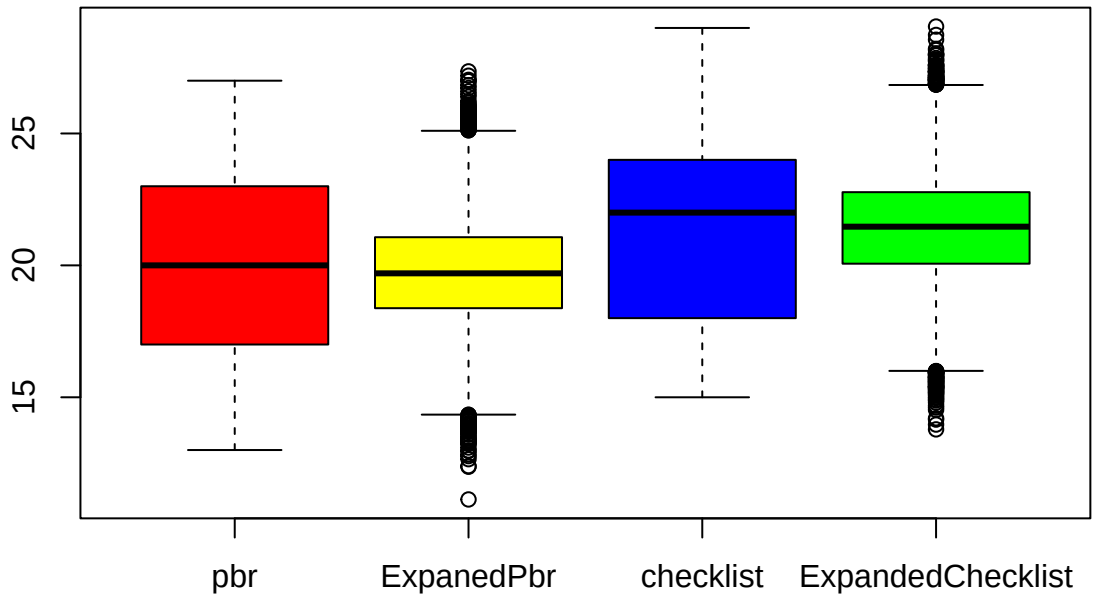
```
median(inspection$checklist)
```

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

```
median(ExpandedChecklist)
```

```
## [1] 21.46733
```

```
boxplot(inspection$pbr,ExpandedPbr,inspection$checklist, ExpandedChecklist, names = c("pbr","ExpanedPbr
```



Summary: # Mean: pbr = 19.73333 and Expanded pbr= 19.71235 which are nearly similar with slight variation # Median: pbr = 20 and Expanded pbr= 19.71263 which are nearly similar with slight variation # Mean: checklist = 21.43333 and Expanded checklist= 21.41639 which are nearly similar with slight variation # Median: checklist = 22 and Expanded checklist= 21.43127 which are nearly similar with slight variation # Before expanding the data the histograms of both pbr and checklist doesn't seem to look like normal distribution. #However, after replicating the samples we get histograms in a bell curve shape which sums our assumptions that the #data is normally distributed. # Boxplot: We can observe that in the boxplot of original pbr/checklist and expanded pbr/checklist (respectively), #the more relevant data/samples are much closer and concentrated near the mean and the outliers are marked down. #Hence, it gives a clear representation that the expanded data is normally distributed for both pbr and checklist.

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)


melted <- melt(inspection, measure.vars = c("pbr","checklist"))
row_num <- rep(c('1','2'),each = 30)
melted <- data.frame(row_num,melted)
names(melted) <- c("","treatment","time")
melted
```

```
##      treatment time
## 1  1       pbr   20
## 2  1       pbr   19
## 3  1       pbr   13
## 4  1       pbr   17
## 5  1       pbr   21
## 6  1       pbr   14
## 7  1       pbr   19
## 8  1       pbr   17
## 9  1       pbr   24
## 10 1       pbr   23
## 11 1       pbr   18
## 12 1       pbr   24
## 13 1       pbr   23
## 14 1       pbr   20
## 15 1       pbr   15
## 16 1       pbr   22
## 17 1       pbr   27
## 18 1       pbr   20
## 19 1       pbr   16
## 20 1       pbr   16
## 21 1       pbr   19
## 22 1       pbr   23
## 23 1       pbr   26
## 24 1       pbr   19
## 25 1       pbr   20
## 26 1       pbr   19
## 27 1       pbr   20
## 28 1       pbr   14
## 29 1       pbr   24
## 30 1       pbr   20
## 31 2 checklist   28
## 32 2 checklist   20
## 33 2 checklist   20
## 34 2 checklist   28
## 35 2 checklist   16
## 36 2 checklist   16
## 37 2 checklist   23
```

```
## 38 2 checklist   24
## 39 2 checklist   20
## 40 2 checklist   22
## 41 2 checklist   26
## 42 2 checklist   29
## 43 2 checklist   22
## 44 2 checklist   23
## 45 2 checklist   16
## 46 2 checklist   26
## 47 2 checklist   23
## 48 2 checklist   22
## 49 2 checklist   18
## 50 2 checklist   20
## 51 2 checklist   16
## 52 2 checklist   24
## 53 2 checklist   25
## 54 2 checklist   23
## 55 2 checklist   17
## 56 2 checklist   24
## 57 2 checklist   20
## 58 2 checklist   16
## 59 2 checklist   15
## 60 2 checklist   21
```

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.

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

```
# code goes here

t.test(melted$time ~ melted$treatment, alternative = "two.sided")
```

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

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

```r
# code goes here
#used less because PBR less time than checklist

t.test(melted$time ~ melted$treatment, alternative = "less", var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  melted$time by melted$treatment
## t = -1.7434, df = 58, p-value = 0.04328
## alternative hypothesis: true difference in means between group pbr and group checklist is less than 0
## 95 percent confidence interval:
##         -Inf -0.07004927
## sample estimates:
##       mean in group pbr mean in group checklist
##                19.73333                21.43333
```

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.

```r
# code goes here
t.test(melted$time~ melted$treatment, inspection = melted, paired = TRUE, alternative = "two.sided")
```

```
##
##  Paired t-test
##
## data:  melted$time by melted$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:
##  -3.34424486 -0.05575514
## 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.

```r
# code goes here for all 3 cases
library(rstatix)
```

```
##
## Attaching package: 'rstatix'

## The following object is masked from 'package:stats':
##
##     filter
```

```r
wilcox.test(melted$time ~ melted$treatment, alternative = "two.sided")
```

```
## Warning in wilcox.test.default(x = c(20L, 19L, 13L, 17L, 21L, 14L, 19L, : cannot
## compute exact p-value with ties

##
##  Wilcoxon rank sum test with continuity correction
##
## data:  melted$time by melted$treatment
## W = 337.5, p-value = 0.09584
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(melted$time ~ melted$treatment, alternative = "two.sided")
```

```
## Warning in wilcox.test.default(x = c(20L, 19L, 13L, 17L, 21L, 14L, 19L, : cannot
## compute exact p-value with ties

##
##  Wilcoxon rank sum test with continuity correction
##
## data:  melted$time by melted$treatment
## W = 337.5, p-value = 0.09584
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(melted$time ~ melted$treatment, inspection = melted, paired = TRUE, alternative = "two.sided")
```

```
## Warning in wilcox.test.default(x = c(20L, 19L, 13L, 17L, 21L, 14L, 19L, : cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(x = c(20L, 19L, 13L, 17L, 21L, 14L, 19L, : cannot
## compute exact p-value with zeroes

##
##  Wilcoxon signed rank test with continuity correction
##
## data:  melted$time by melted$treatment
## V = 123, p-value = 0.04143
## alternative hypothesis: true location shift is not equal to 0
```

```
```