

# 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 data set before proceeding).

```
knitr::opts_chunk$set(echo = TRUE, tidy.opts = list(width.cutoff = 60), tidy = TRUE)
```

```
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(modeest) # required for `mfv()` function
```

```
## Registered S3 method overwritten by 'rmutil':  
##   method      from  
##   print.response httr
```

```
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_data <- read.csv("inspection.csv")  
print(inspection_data)
```

```
##      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
```

```
mode(inspection_data)
```

```
## [1] "list"
```

```
typeof(inspection_data)
```

```
## [1] "list"
```

```
is.data.frame(inspection_data)
```

```
## [1] TRUE
```

## 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
pbr_mean <- mean(inspection_data$pbr)
pbr_median <- median(inspection_data$pbr)
cat("\npbr's mean is:", pbr_mean)
```

```
##
## pbr's mean is: 19.73333
```

```
cat("\npbr's median is:", pbr_median)
```

```
##
## pbr's median is: 20
```

```
checklist_mean <- mean(inspection_data$checklist)
checklist_median <- median(inspection_data$checklist)
cat("\nchecklist's mean is:", checklist_mean)
```

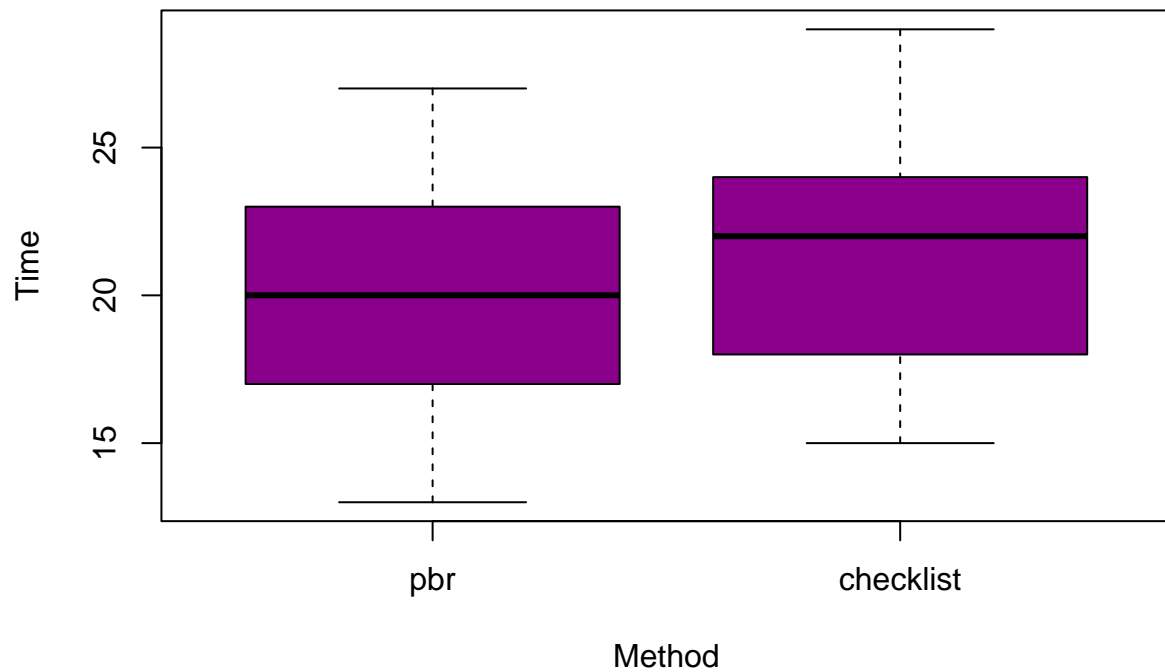
```
##
## checklist's mean is: 21.43333
```

```
cat("\nchecklist's median is:", checklist_median)
```

```
##
## checklist's median is: 22
```

```
boxplot(inspection_data, main = "Box-plot distributions", xlab = "Method",
        ylab = "Time", col = "darkmagenta")
```

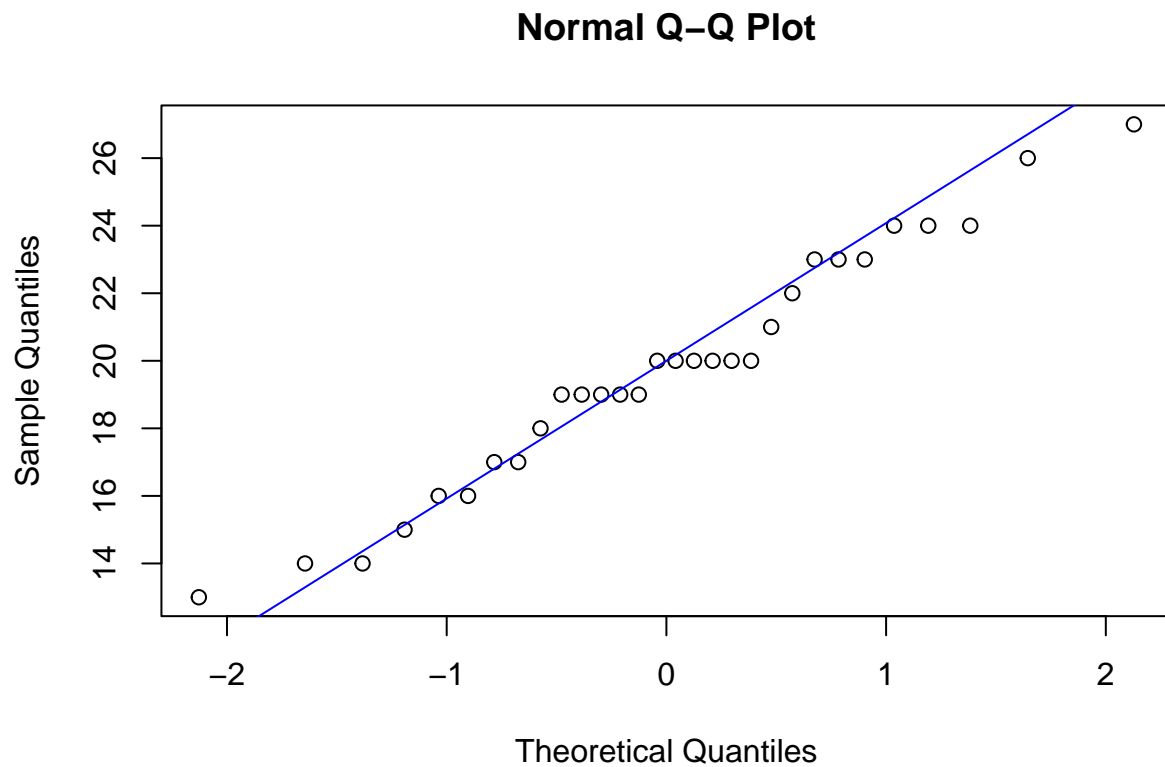
## Box-plot distributions



## 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
qqnorm(inspection_data$pbr)
qqline(inspection_data$pbr, col = "blue")
```



```
# Since the Shapiro-Wilk normality test is for a small size  
# test, and this sample is only 60, the Shapiro-Wilk  
# normality test is more appropriate than Anderson-Darling  
# test.
```

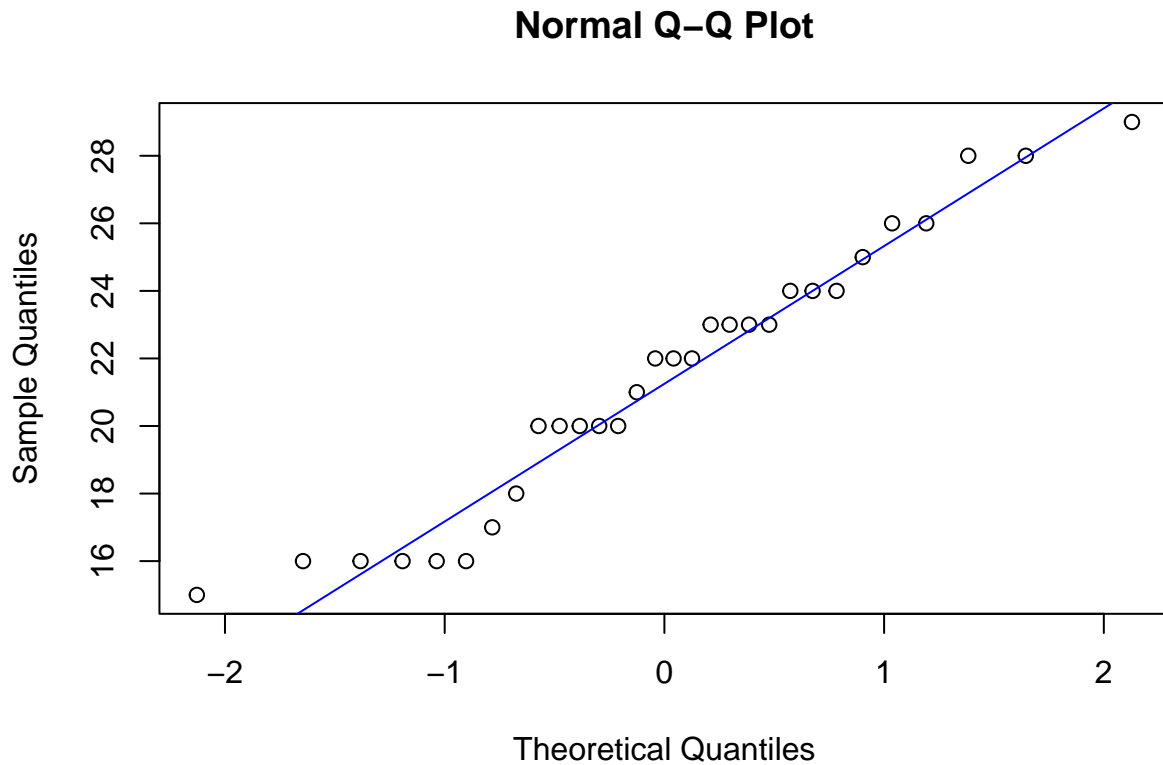
```
# Shapiro-Wilk normality test  
shapiro.test(inspection_data$pbr)
```

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

```
# Anderson-Darling normality test  
ad.test(inspection_data$pbr)
```

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

```
qqnorm(inspection_data$checklist)
qqline(inspection_data$checklist, col = "blue")
```



```
# Shapiro-Wilk normality test
shapiro.test(inspection_data$checklist)
```

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

```
# Anderson-Darling normality test
ad.test(inspection_data$checklist)
```

```
##
##  Anderson-Darling normality test
##
## data:  inspection_data$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 replicate)  
  
set.seed(1234)  
# expande_pbr <- replicate(n = 1, rnorm(20000, mean =  
# mean(inspection_data$pbr), sd=2), simplify = FALSE[[1]] )  
expande_pbr <- rnorm(20000, mean = mean(inspection_data$pbr),  
  sd = 2)  
expande_pbr <- round(expande_pbr, 0)  
cat("\nBootstrapping expande_pbr:\n")
```

```
##  
## Bootstrapping expande_pbr:
```

```
# expande_pbr  
  
expande_checklist <- rnorm(20000, mean = mean(inspection_data$checklist),  
  sd = 2)  
expande_checklist <- round(expande_checklist, 0)  
cat("\nBootstrapping expande_checklist:\n")
```

```
##  
## Bootstrapping expande_checklist:
```

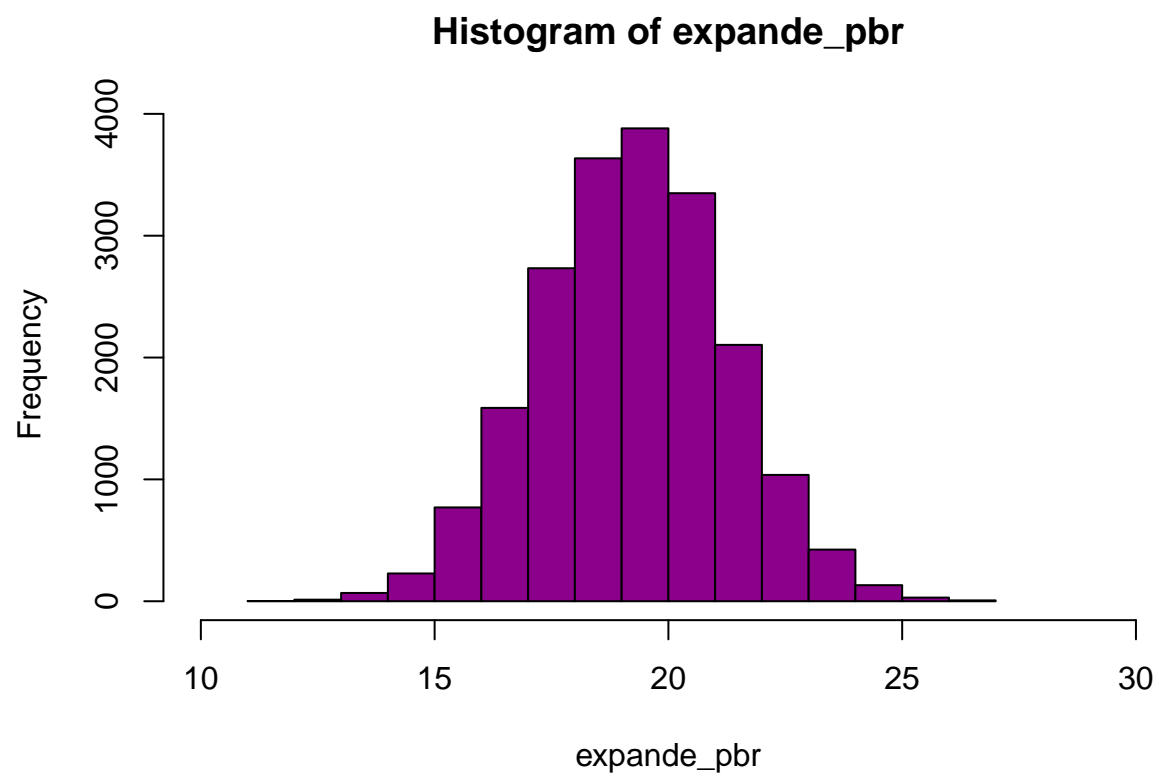
```
# expande_checklist  
  
# Step 2: Draw the histogram to compare the original with  
# the bootstrap data for each treatment separately (hint:  
# use `hist`)  
  
hist(inspection_data$pbr, xlim = c(10, 30), col = "darkmagenta")
```

**Histogram of inspection\_data\$pbr**

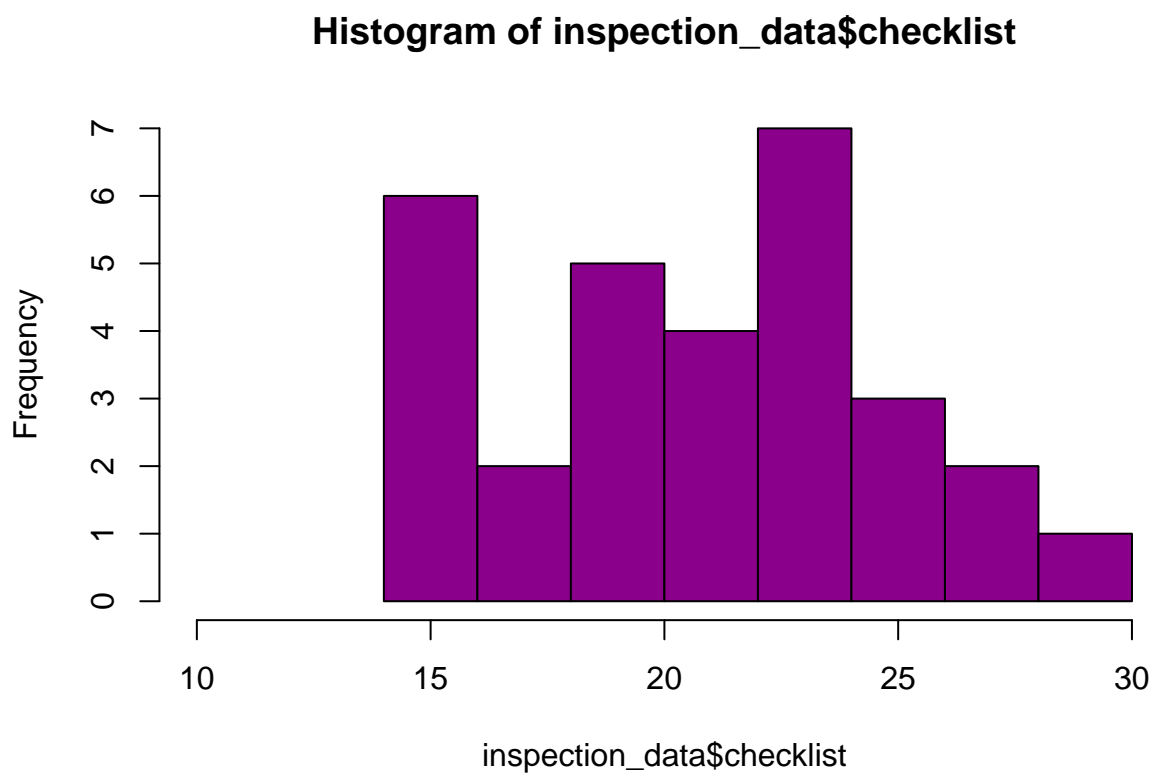


```
hist(expande_pbr, xlim = c(10, 30), col = "darkmagenta")
```



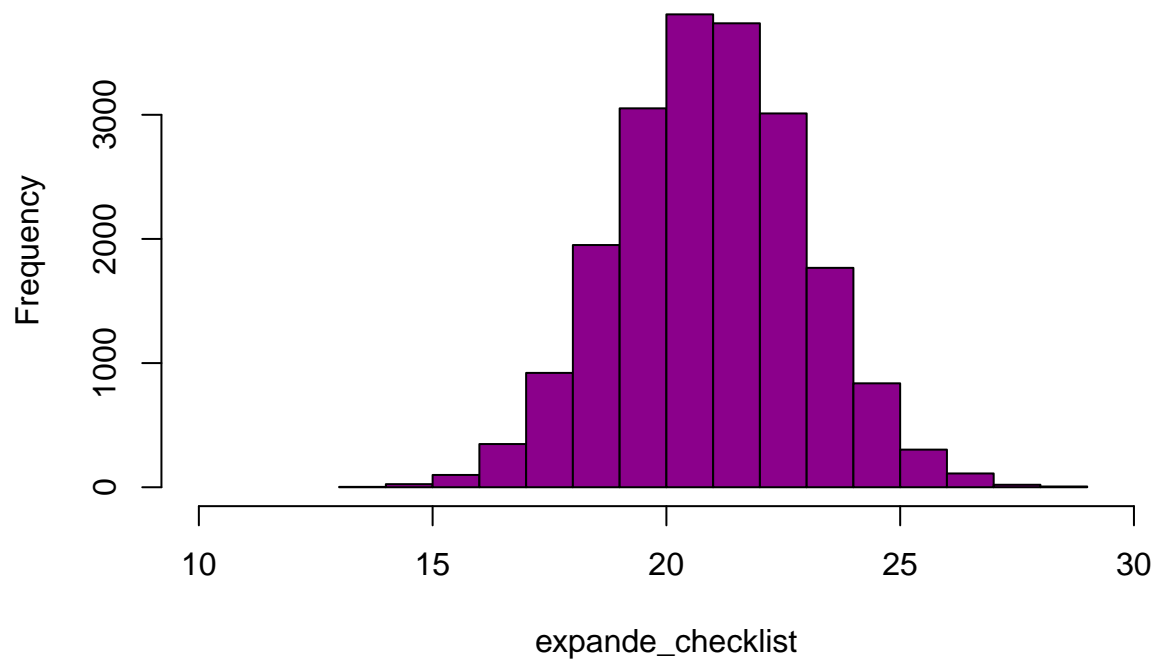


```
hist(inspection_data$checklist, xlim = c(10, 30), col = "darkmagenta")
```



```
hist(expande_checklist, xlim = c(10, 30), col = "darkmagenta")
```

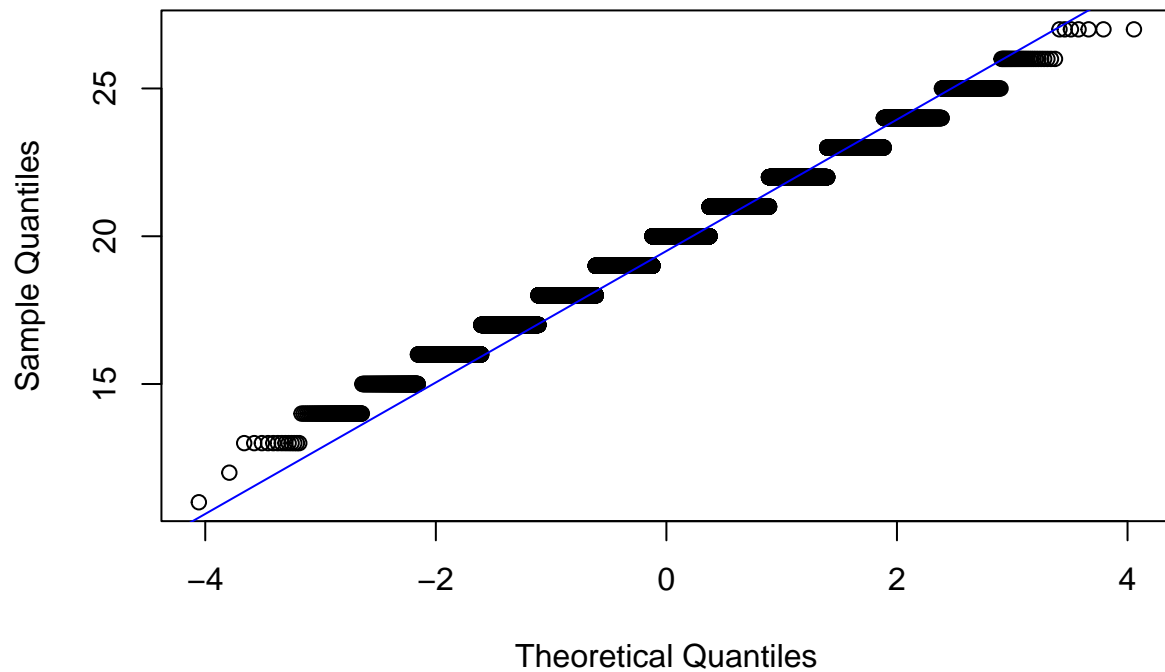
**Histogram of expande\_checklist**



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

```
qqnorm(expande_pbr)
qqline(expande_pbr, col = "blue")
```

## Normal Q-Q Plot

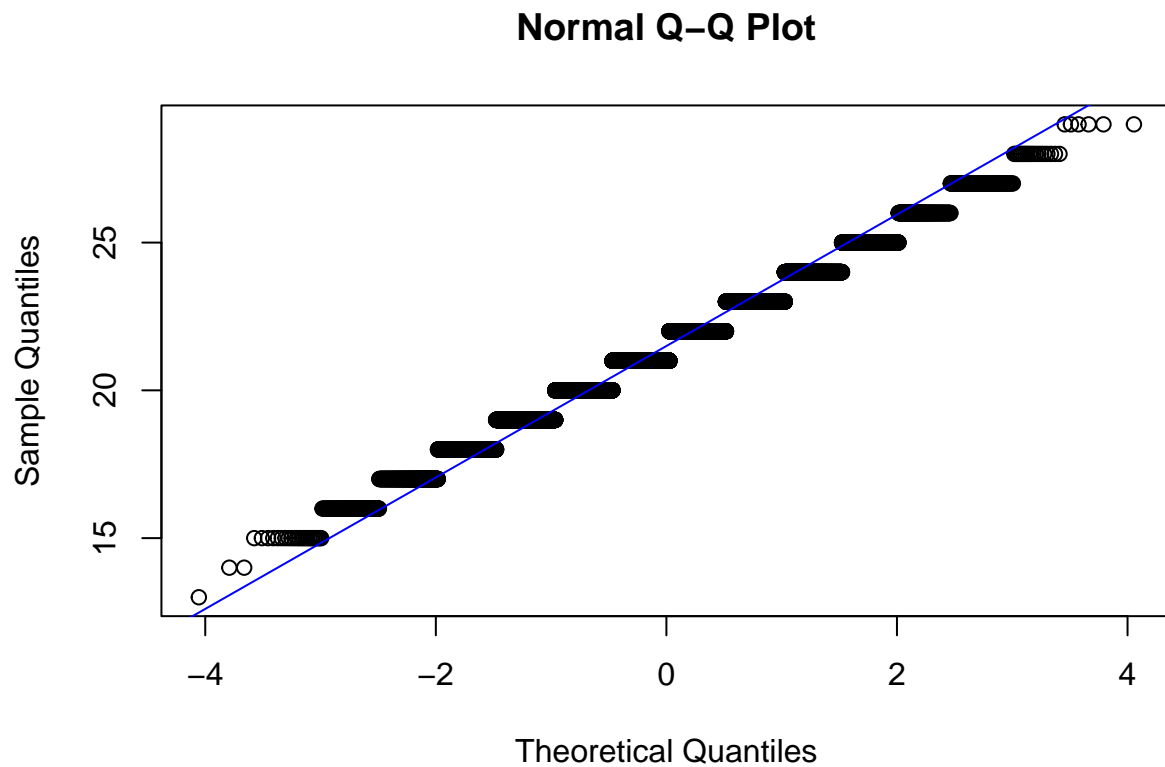


```
# Shapiro-Wilk normality test sample size must be between 3  
# and 5000 shapiro.test(expande_pbr)
```

```
# Anderson-Darling normality test  
ad.test(expande_pbr)
```

```
##  
## Anderson-Darling normality test  
##  
## data:  expande_pbr  
## A = 198.38, p-value < 2.2e-16
```

```
qqnorm(expande_checklist)  
qqline(expande_checklist, col = "blue")
```



```
# Shapiro-Wilk normality test sample size must be between 3  
# and 5000 shapiro.test(expande_checklist)
```

```
# Anderson-Darling normality test  
ad.test(expande_checklist)
```

```
##  
## Anderson-Darling normality test  
##  
## data:  expande_checklist  
## A = 196.09, p-value < 2.2e-16
```

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

```
cat("\npbr's mean is:", pbr_mean)
```

```
##  
## pbr's mean is: 19.73333
```

```
cat("\npbr's median is:", pbr_median)
```

```
##  
## pbr's median is: 20
```

```
cat("\nchecklist's mean is:", checklist_mean)
```

```
##  
## checklist's mean is: 21.43333
```

```
cat("\nchecklist's median is:", checklist_median)
```

```
##  
## checklist's median is: 22
```

```
cat("\ndescriptive statistics of expande_pbr:\n")
```

```
##  
## descriptive statistics of expande_pbr:
```

```
cat("\nexpande_pbr's mean is:")
```

```
##  
## expande_pbr's mean is:
```

```
mean(expande_pbr)
```

```
## [1] 19.73275
```

```
cat("\nexpande_pbr's median is:")
```

```
##  
## expande_pbr's median is:
```

```
median(expande_pbr)
```

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

```
summary(expande_pbr)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
##    11.00   18.00   20.00   19.73   21.00   27.00
```

```
cat("descriptive statistics of expande_checklist:\n")
```

```
## descriptive statistics of expande_checklist:
```

```
cat("\nexpande_checklist's mean is:")
```

```
##  
## expande_checklist's mean is:
```

```
mean(expande_checklist)
```

```
## [1] 21.45005
```

```
cat("\nexpande_checklist's median is:")
```

```
##
```

```
## expande_checklist's median is:
```

```
median(expande_checklist)
```

```
## [1] 21
```

```
summary(expande_checklist)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    13.00   20.00   21.00   21.45   23.00   29.00
```

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)
# melt(inspection_data, id.vars = '', measure.vars =
# c('pbr', 'checklist'))
```

```
tdata <- melt(inspection_data)
```

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

```
data <- data.frame(id = rep(c("1", "2"), each = 30), treatment = c(tdata$variable),
  time = c(tdata$value))
data
```

```
##      id 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
```

```
data2 <- melt(inspection_data, measure.vars = c("pbr", "checklist"))
names(data2) <- c("treatment", "time")
```



## 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(data$time ~ data$treatment, alternative = "two.sided",  
       var.equal = TRUE)
```

```
##
```

```
## Two Sample t-test
```

```
##
```

```
## data: data$time by data$treatment
```

```
## t = -1.7434, df = 58, p-value = 0.08656
```

```
## alternative hypothesis: true difference in means between group pbr and group checklist is not equal
```

```
## 95 percent confidence interval:
```

```
## -3.6518982 0.2518982
```

```
## 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.

```
# code goes here
```

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

```
##
```

```
## Two Sample t-test
```

```
##
```

```
## data: data$time by data$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
```

```
## 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.

```
# code goes here
```

```
t.test(inspection_data$pbr, inspection_data$checklist, paired = TRUE)
```

```
##
```

```
## Paired t-test
```

```
##
```

```
## data: inspection_data$pbr and inspection_data$checklist
## 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.

```
# code goes here for all 3 cases
```

```
# Wilcoxon tests
```

```
# two-tailed test
```

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

```
##
```

```
## Wilcoxon rank sum test with continuity correction
```

```
##
```

```
## data: data$time by data$treatment
```

```
## W = 337.5, p-value = 0.09584
```

```
## alternative hypothesis: true location shift is not equal to 0
```

```
# one-tailed test
```

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

```
##
```

```
## Wilcoxon rank sum test with continuity correction
```

```
##
```

```
## data: data$time by data$treatment
```

```
## W = 337.5, p-value = 0.04792
```

```
## alternative hypothesis: true location shift is less than 0
```

```
# paired (two-tailed) test
```

```
wilcox.test(inspection_data$pbr, inspection_data$checklist, paired = TRUE,
            exact = FALSE, var.equal = TRUE)
```

```
##
```

```
## Wilcoxon signed rank test with continuity correction
```

```
##
```

```
## data: inspection_data$pbr and inspection_data$checklist
```

```
## V = 123, p-value = 0.04143
```

```
## alternative hypothesis: true location shift is not equal to 0
```

```
““
```