Assignment 1

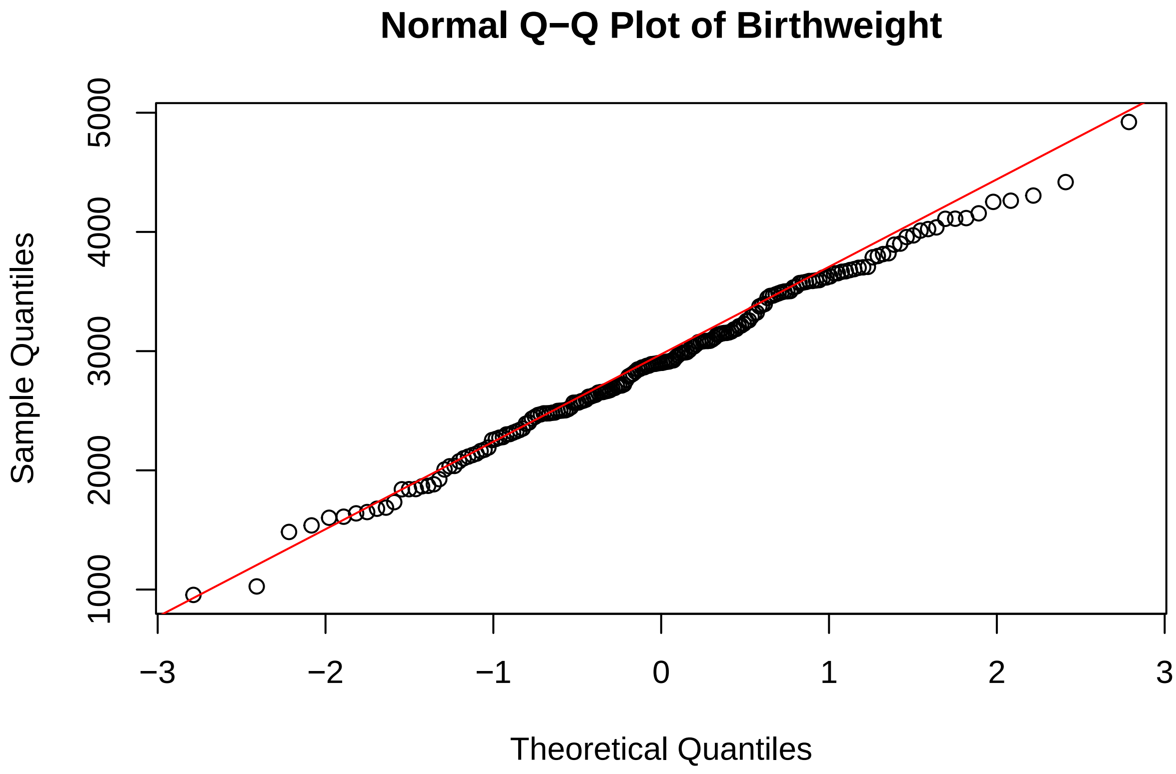
Group 61, Ikrame Zirar, Mohammed Majeed,Sergio Alejandro Gutierrez Maury

23 February 2023

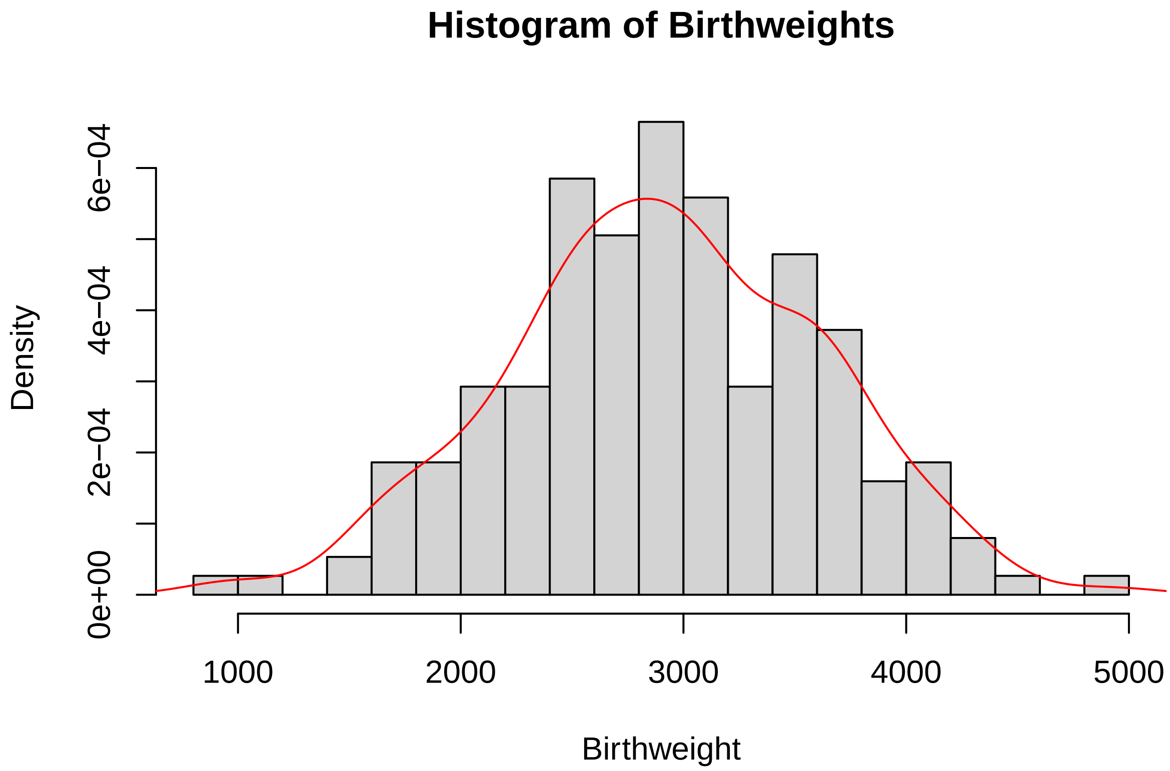
Exercise 1

A) Check for normality with QQ plot,histogram, and boxplot

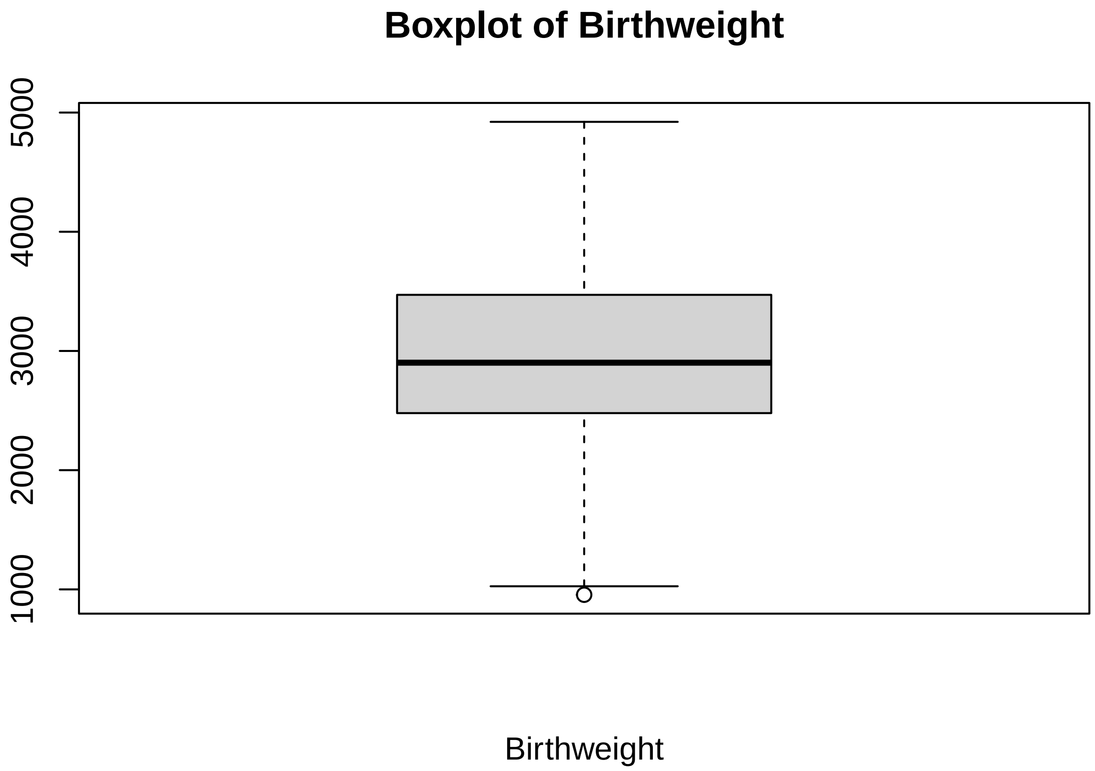
```
qqnorm(data, main = "Normal Q-Q Plot of Birthweight")
qqline(data,col=" red ")
```



```
# Check for normality with histogram
hist(x = data,breaks=15,main = "Histogram of Birthweights",
     xlab = " Birthweight ",freq = FALSE)
lines(x= density(x= data), col=" red ")
```



```
# Check for normality with boxplot
boxplot (data , main =" Boxplot of Birthweight ",
         xlab =" Birthweight ")#horizontal = TRUE)
```



calculate Shapiro-Wilk normality test

```
shapiro.test(data)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data
## W = 0.99595, p-value = 0.8995
```

Upon observation, the density displays a bell-shaped curve that closely resembles a normal distribution. The QQ-plot also appears to follow a straight line, indicating normality of the data. The boxplot exhibits some asymmetry, but with few outliers, it can still be assumed that the data is normally distributed. It is worth noting, however, that the histogram, QQ-plot, and boxplot show some variability, which may be attributed to the sample size. A larger sample size would likely result in less variability in these graphs. Additionally, the Shapiro-Wilk test for normality yields a non-significant result with a p-value of 0.90, suggesting that the null hypothesis cannot be rejected, and that the sample follows a normal distribution. Considering these findings, we can assume that the birth weights adhere to a normal distribution.

```
# construct Confidence Interval
mu <- mean(data)
stnd <-sd(data)
CI <- 0.96
alpha <- 1-CI
# Calculate the margin of error for a 95% confidence interval
```

```r
z <- qnorm(1 - alpha/2) # 98th percentile of standard normal distribution
# Calculate the margin of error
me <- z * stnd / sqrt(length(data))
# Calculate the confidence interval
lower_ci = mu - me
upper_ci = mu + me
# Print the confidence interval
cat("Confidence Interval: [", lower_ci, ", ", upper_ci, "]", sep = "")
```

```
## Confidence Interval: [2808.817, 3017.768]
```

```r
# construct a bounded 96%-CI for mu(mean)
for (sample_size in 1:1000) {
  lower_bound = mu - z*stnd/sqrt(sample_size)
  upper_bound = mu + z*stnd/sqrt(sample_size)
  CI_length <- upper_bound - lower_bound
  if (CI_length <= 100) {
    break
  }
}
cat("sample_size",sample_size)
```

```
## sample_size 821
```

In order to achieve a confidence interval with a maximum length of 100, a minimum sample size of 821 is required.

```r
# Compute a bootstrap 96%-CI for mu and compare it to the above CI.
library(boot)
B <- 1000 # Choose number of bootstrap resamples
boot_data <- boot(data, statistic = function(data, i) mean(data[i]), R = B)
boot_ci <- boot.ci(boot_data, type = "perc", conf = 0.96)
lower_bound_boot_ci <- boot_ci$percent[[4]]
upper_bound_boot_ci <- boot_ci$percent[[5]]
cat("boot_CI Confidence Interval: [",
    lower_bound_boot_ci, ", ",
    upper_bound_boot_ci, "]", sep = "")
```

```
## boot_CI Confidence Interval: [2805.59, 3011.871]
```

Since the sample is assumed to follow a normal distribution, the bootstrapped confidence interval is expected to be similar to the confidence interval calculated prior. This is because the sample mean is a consistent estimator of the population mean, and the bootstrapping method is based on re-sampling the data with replacement from the original sample.

B)

```r
# Verify this claim by using a one side t-test
t.test(data, mu = 2800, alternative = "greater")
```

```
##
##  One Sample t-test
##
## data:  data
## t = 2.2271, df = 187, p-value = 0.01357
## alternative hypothesis: true mean is greater than 2800
## 95 percent confidence interval:
##  2829.202      Inf
## sample estimates:
## mean of x
##  2913.293
```

performed a one-sample t-test with the null hypothesis(H_0) that the mean birthweight is equal to 2800 grams ($\mu = 2800$) and the alternative hypothesis(H_1) that the mean birthweight is bigger than 2800 grams.The p-value is less than the confidence level of 0.05, which does give evidence against the null hypothesis. Therefore, we reject the null hypothesis and conclude that the mean birthweight is greater than 2800 grams. The confidence interval (CI) in the R-output for this test represents a range of values within which the true population mean birthweight is likely to fall with a certain degree of confidence. For a one-tailed test, we only obtain either a lower bound or upper bound, which we call a one-sided confidence interval. In our case, we only obtained a lower bound CI [2829.202 Inf].

```r
# sign test
binom.test(sum(data > 2800), n=length(data), p = 0.5, alternative = "greater")
```

```
##
##  Exact binomial test
##
## data:  sum(data > 2800) and length(data)
## number of successes = 107, number of trials = 188, p-value = 0.03399
## alternative hypothesis: true probability of success is greater than 0.5
## 95 percent confidence interval:
##  0.5065781 1.0000000
## sample estimates:
## probability of success
##              0.5691489
```

The sign test p-value is less than the confidence level of 0.05, which gives enough evidence against the null hypothesis. Therefore, we reject the null hypothesis and conclude that the mean birthweight is bigger than 2800 grams.

C)

the powers of the t-test and sing

```r
B = 1000; n = 50
psign = numeric(B) # will contain P-values of sign test
pttest = numeric(B) # will contain P-values of T test

for(i in 1:B) {
  x = sample(data, n)
  pttest[i] = t.test(x, mu=2800, alt='greater')[[3]] # extract P-value
  psign[i] = binom.test(sum(x>2800), n, alt='greater')[[3]] # extract P-value
}
power_ttest = sum(pttest<0.05)/B
```

```r
power_sign = sum(psign<0.05)/B
cat("the powers of the t-test: ", power_ttest, ",
    the powers of the sing test: ",
    power_sign, sep = "")
```

```
## the powers of the t-test: 0.266,
##     the powers of the sing test: 0.154
```

One way to compute the power of the tests is through simulation. The estimated power of the one-sided t-test is 0.27, meaning that if the true population mean is actually 2913.293 (as estimated from the data), there is a 27% chance that the t-test will correctly reject the null hypothesis of a mean of 2800. The estimated power of the sign test is 0.16, meaning that there is a 16% chance that the sign test will correctly reject the null hypothesis of a median of 2800. These power estimates suggest that the t-test may be more powerful than the sign test for detecting a difference in the mean above 2800.

D) Recover the whole confidence interval and its confidence level.

```r
# Set the number of samples to take and the lower probability
n_samples = 1000
p_left = 0.25
# Create an empty vector to store the sample probabilities
sample_probabilities = numeric(n_samples)
# Take n_samples samples of size n_samples from the birthweight vector
# Calculate the proportion of samples that have a weight less than 2600 grams
# Store the sample proportion in the sample_probabilities vector
for(i in 1:n_samples){
  sample = sample(data, n_samples, replace = TRUE)
  sample_probabilities[i] = sum(sample < 2600)/n_samples
}
# Calculate the standard deviation of the sample proportions
sample_sd = sd(sample_probabilities)
# Calculate the mean of the sample proportions
p_hat = mean(sample_probabilities)
# margin of error
me = p_hat - p_left
# Calculate the upper confidence interval
p_right = p_hat + me
# Calculate the z-score
z = me / sqrt((p_hat*(1-p_hat))/length(data))
# Calculate the level of significance
alpha = (1 - pnorm(z))*2
# Calculate the confidence level
confidence_level = 1 - alpha

cat("Confidence Interval: [", p_left,
    ", ", p_right, "]",
    "confidence_level: ",
    confidence_level,
    sep = "")
```

```
## Confidence Interval: [0.25, 0.410448]confidence_level: 0.9806601
```

sample proportion denoted as $\hat{p}$, we can calculate the right side of the confidence interval using the margin of error. As a result, we obtain the confidence interval [0.25, 0.4] with a confidence level of around 98%.

E)

```
male_means = c()
female_means = c()

for (i in 1:1000) {
  # Select 34 males and 28 females with birthweight < 2600 g
  male_2600 = sample(data< 2600, 34)
  female_2600 = sample(data[data < 2600], 28)
   # Select 61 males and 65 females with birthweight >= 2600 g
  male_others = sample(data[data >= 2600], 61)
  female_others = sample(data[data >= 2600], 65)
   # Combine the selected males and females with birthweight < 2600 g
  males = c(male_2600, male_others)
  females = c(female_2600, female_others)
  # Calculate the mean birthweight for males and females
  male_means = c(male_means, mean(males))
  female_means = c(female_means, mean(females))
}
# Calculate the mean of the sample means for males and females
mean(male_means)
```

```
## [1] 2110.416
```

```
mean(female_means)
```

```
## [1] 2945.381
```

```
# Perform a two-sample t-test assuming unequal variances
t.test(male_means, female_means)
```

```
##
##  Welch Two Sample t-test
##
## data:  male_means and female_means
## t = -612.6, df = 1931.3, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -837.6380 -832.2918
## sample estimates:
## mean of x mean of y
##  2110.416  2945.381
```

A Two-sample t-test was conducted to examine whether the mean weight differs for male and female babies. The null hypothesis (H_0) assumed there is no different, while the alternative hypothesis (H_1) assumed that the mean weight is different of the weight for male and female babies. The obtained p-value was lower than the conventional significance level of 0.05, indicating strong evidence against the null hypothesis. As a result, the null hypothesis was rejected, and it was concluded that the mean weight is not different for male and female babies.

Exercise 2

A)

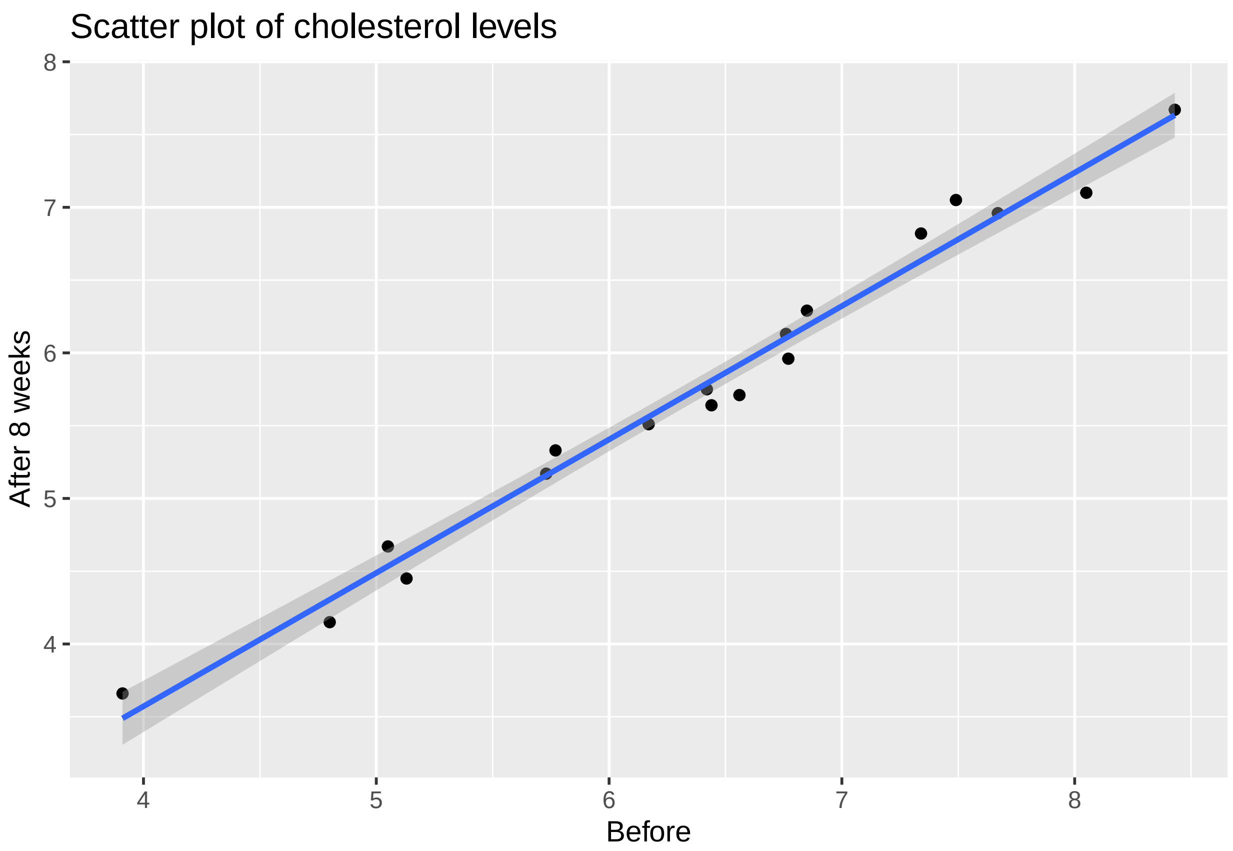
The data seems correlated and normally distributed.

```
#library('tidyverse')
cholesterol <- read.table("cholesterol.txt", header = TRUE)

library(ggplot2)

summary(cholesterol)
```

```
##      Before        After8weeks   
##  Min.   :3.910   Min.   :3.660  
##  1st Qu.:5.740   1st Qu.:5.210  
##  Median :6.500   Median :5.730  
##  Mean   :6.408   Mean   :5.779  
##  3rd Qu.:7.218   3rd Qu.:6.688  
##  Max.   :8.430   Max.   :7.670
```

```
# Regression the 2 variables are highly correlated
ggplot(cholesterol, aes(x = Before, y = After8weeks)) +
  geom_point() +
  geom_smooth(method = "lm") +
  labs(x = "Before", y = "After 8 weeks", title = "Scatter plot of cholesterol levels")
```

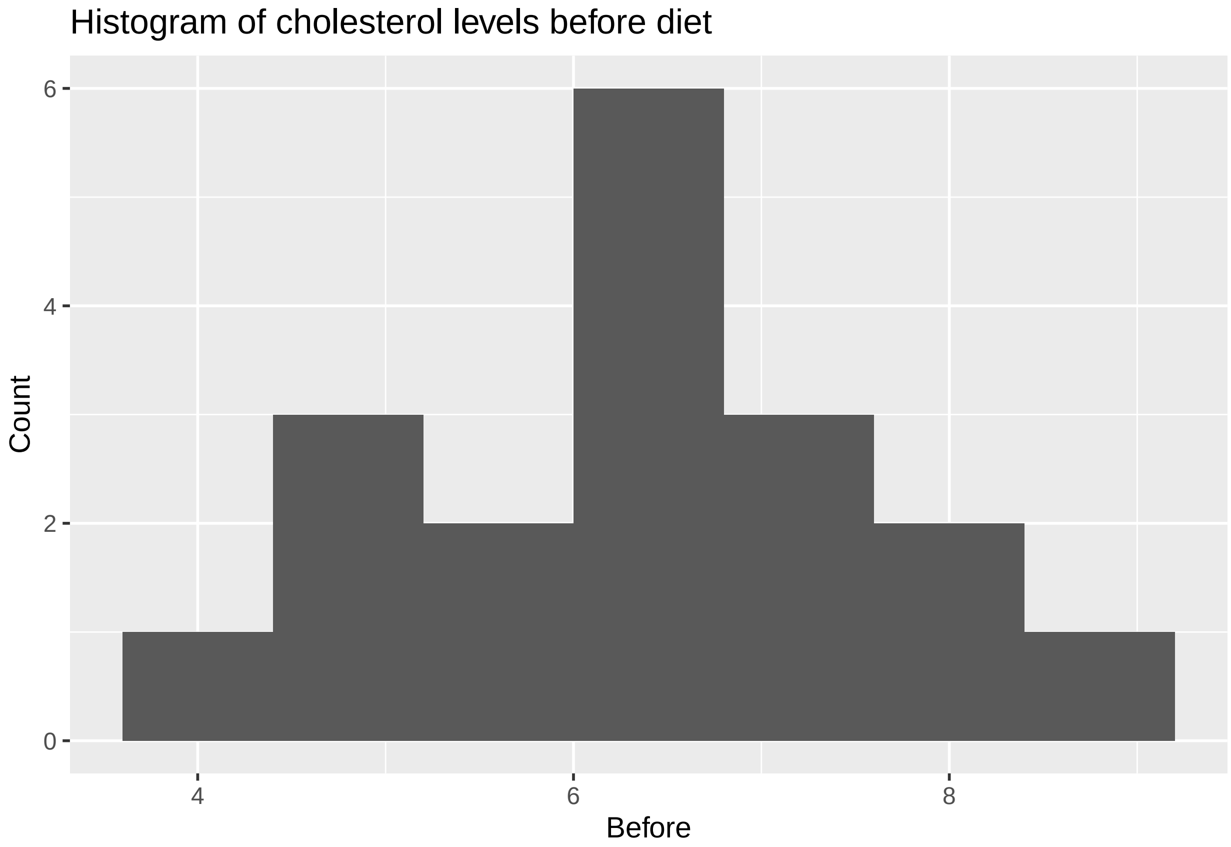
```
## `geom_smooth()` using formula = 'y ~ x'
```



```
cor(cholesterol$Before, cholesterol$After8weeks)
```

```
## [1] 0.9908885
```

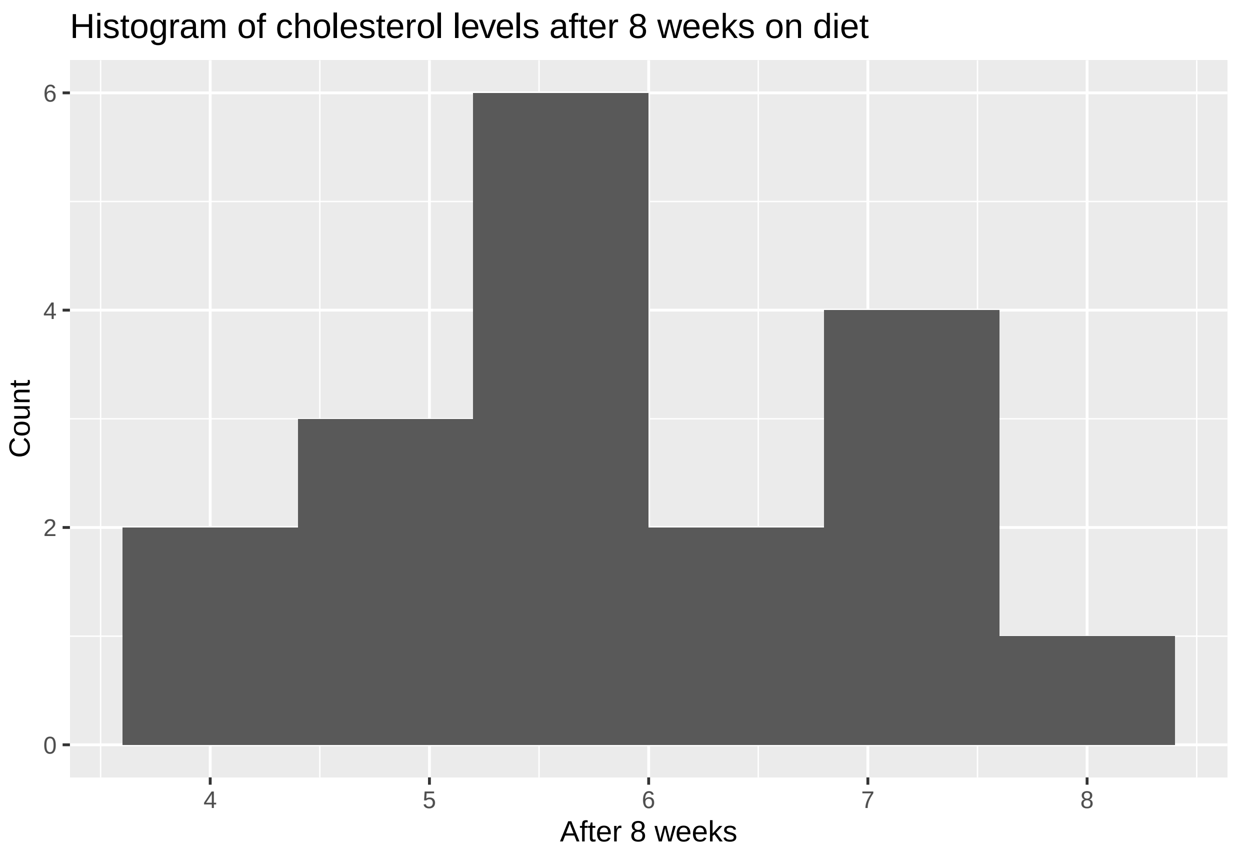
```
# creating histograms
ggplot(cholesterol, aes(x = Before)) +
  geom_histogram(binwidth = 0.8) +
  labs(x = "Before",
       y = "Count",
       title = "Histogram of cholesterol levels before diet")
```



```
shapiro.test(cholesterol$Before) # accept null, data is normal
```

```
##
##  Shapiro-Wilk normality test
##
## data:  cholesterol$Before
## W = 0.9819, p-value = 0.9675
```

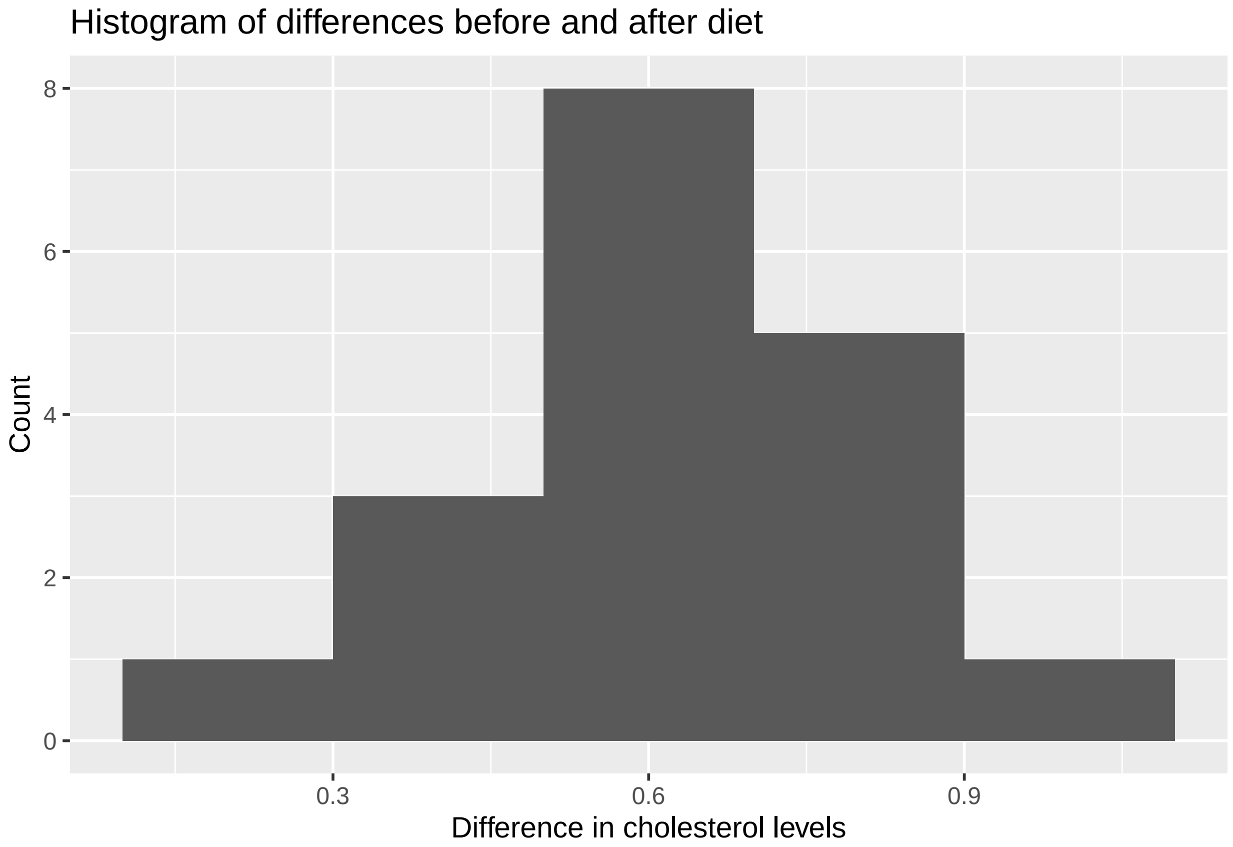
```
ggplot(cholesterol, aes(x = After8weeks)) +
  geom_histogram(binwidth = 0.8) +
  labs(x = "After 8 weeks",
       y = "Count",
       title = "Histogram of cholesterol levels after 8 weeks on diet")
```



```
shapiro.test(cholesterol$After8weeks) # accept null, data is normal
```

```
##
##  Shapiro-Wilk normality test
##
## data:  cholesterol$After8weeks
## W = 0.97733, p-value = 0.9183
```

```
# checking difference
diff <- cholesterol$Before - cholesterol$After8weeks

ggplot(cholesterol, aes(x = Before - After8weeks)) +
  geom_histogram(binwidth = 0.2) +
  labs(
    x = "Difference in cholesterol levels",
       y = "Count",
       title = "Histogram of differences before and after diet")
```



```
shapiro.test(diff) # accept null, data is normal
```

```
##
##  Shapiro-Wilk normality test
##
## data:  diff
## W = 0.98501, p-value = 0.9869
```

B) Given that the data set contains measurements of cholesterol levels before and after a low fat low cholesterol diet with the same individuals, the data are paired. Therefore, we can use paired tests to analyze the data.

```
wilcox.test(cholesterol$Before, cholesterol$After8weeks, paired = TRUE)
```

```
##
##  Wilcoxon signed rank exact test
##
## data:  cholesterol$Before and cholesterol$After8weeks
## V = 171, p-value = 7.629e-06
## alternative hypothesis: true location shift is not equal to 0
```

```
#strong evidence to suggest that the margarine diet #had an effect on cholesterol levels.
```

in this case, a permutation test is not applicable since we only have one group of paired data. Permutation tests are typically used for comparing two independent groups of data.

C)

```
# Calculate estimate for theta
theta_hat <- 2*mean(cholesterol$After8weeks) - 3
sigma <- sqrt((theta_hat-3)^2/12)

n <- length(cholesterol$After8weeks)
z <- qnorm(0.975)

lower <- theta_hat - z*(sigma/sqrt(n))
upper <- theta_hat + z*(sigma/sqrt(n))
#confidence interval
c(lower, upper)
```

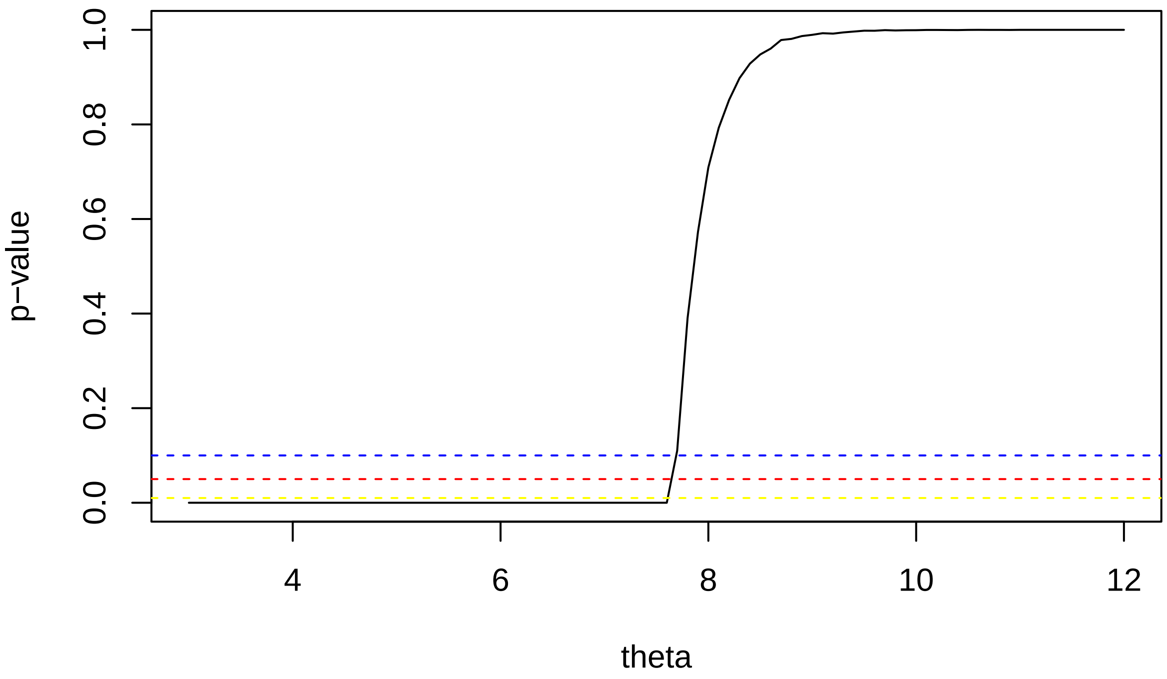
```
## [1] 7.816600 9.298956
```

D)

$H_0 : X_1, \cdots, X_{18} \sim Unif[3, 12]$

$T = \max(X_1, \cdots, X_{18})$

```
after = cholesterol$After8weeks
# Define the test statistic function
T <- function(x) max(x)

# Set up the range of theta values to test
theta_vals <- seq(3, 12, by = 0.1)

# Create a vector to store the p-values for each theta value
p_vals <- rep(NA, length(theta_vals))

# Perform the bootstrap test for each theta value
for (i in seq_along(theta_vals)) {
  # Generate bootstrap samples
  boot_samples <- replicate(10000, runif(18, min = 3, max = theta_vals[i]))
  # Calculate test statistics for bootstrap samples
  boot_t <- apply(boot_samples, 2, T)
  # Calculate p-value
  p_vals[i] <- mean(boot_t >= max(after))
}

# Plot the p-values as a function of theta
plot(theta_vals, p_vals, type = "l", xlab = "theta", ylab = "p-value")
abline(h = 0.01, col = "yellow", lty = 2)
abline(h = 0.05, col = "red", lty = 2)
abline(h = 0.1, col = "blue", lty = 2)
```



We can see that H_0 is not rejected, at $\alpha = 5\%$, for values of theta between approximately 3 and 7.5, but is rejected for larger values of theta. Therefore, we can conclude that the data is consistent with a uniform distribution on [3, 7.5].

Can the Kolmogorov-Smirnov test be also applied for this situation? No, the Kolmogorov-Smirnov test is not applicable in this case since the null hypothesis is not a continuous distribution.

E)

```
# Median test
binom.test(sum(cholesterol$After8weeks<6),length(cholesterol$After8weeks),p=0.5,alt="less")
```

```
##
##  Exact binomial test
##
## data:  sum(cholesterol$After8weeks < 6) and length(cholesterol$After8weeks)
## number of successes = 11, number of trials = 18, p-value = 0.8811
## alternative hypothesis: true probability of success is less than 0.5
## 95 percent confidence interval:
##  0.0000000 0.8010467
## sample estimates:
## probability of success
##              0.6111111
```

```
binom.test(sum(cholesterol$After8weeks < 4.5), length(cholesterol$After8weeks), p = 0.25, alternative =
```

```
##
```

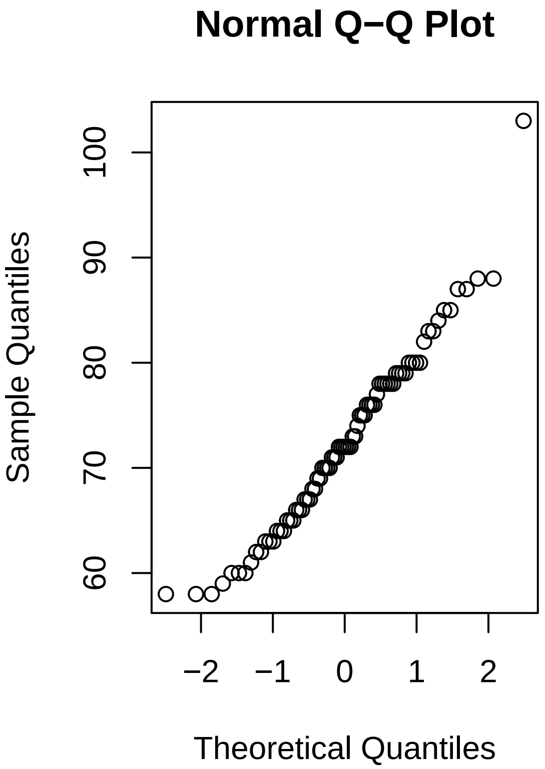
```
##  Exact binomial test
##
## data:  sum(cholesterol$After8weeks < 4.5) and length(cholesterol$After8weeks)
## number of successes = 3, number of trials = 18, p-value = 0.3057
## alternative hypothesis: true probability of success is less than 0.25
## 95 percent confidence interval:
##  0.0000000 0.3766792
## sample estimates:
## probability of success
##              0.1666667
```

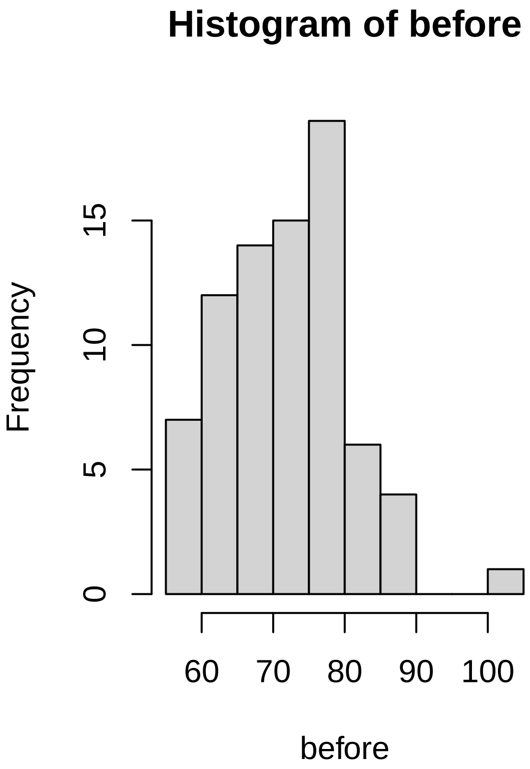
Exercise 3

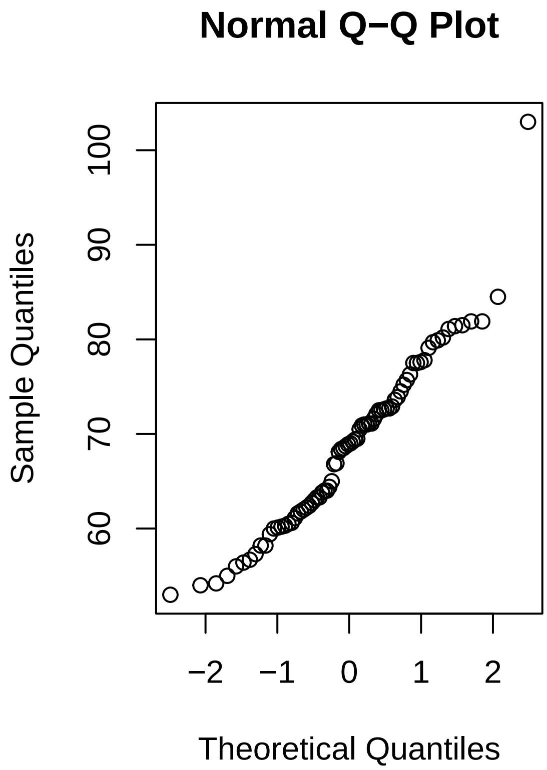
First, we add the response variable weight.lost.

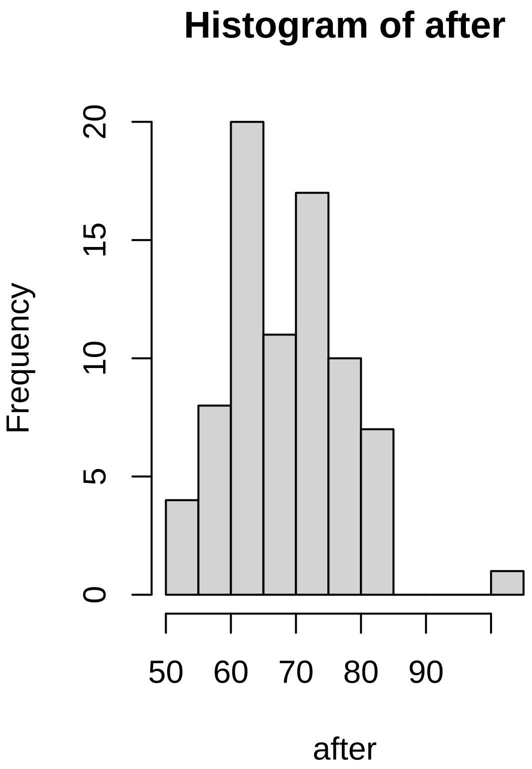
a)

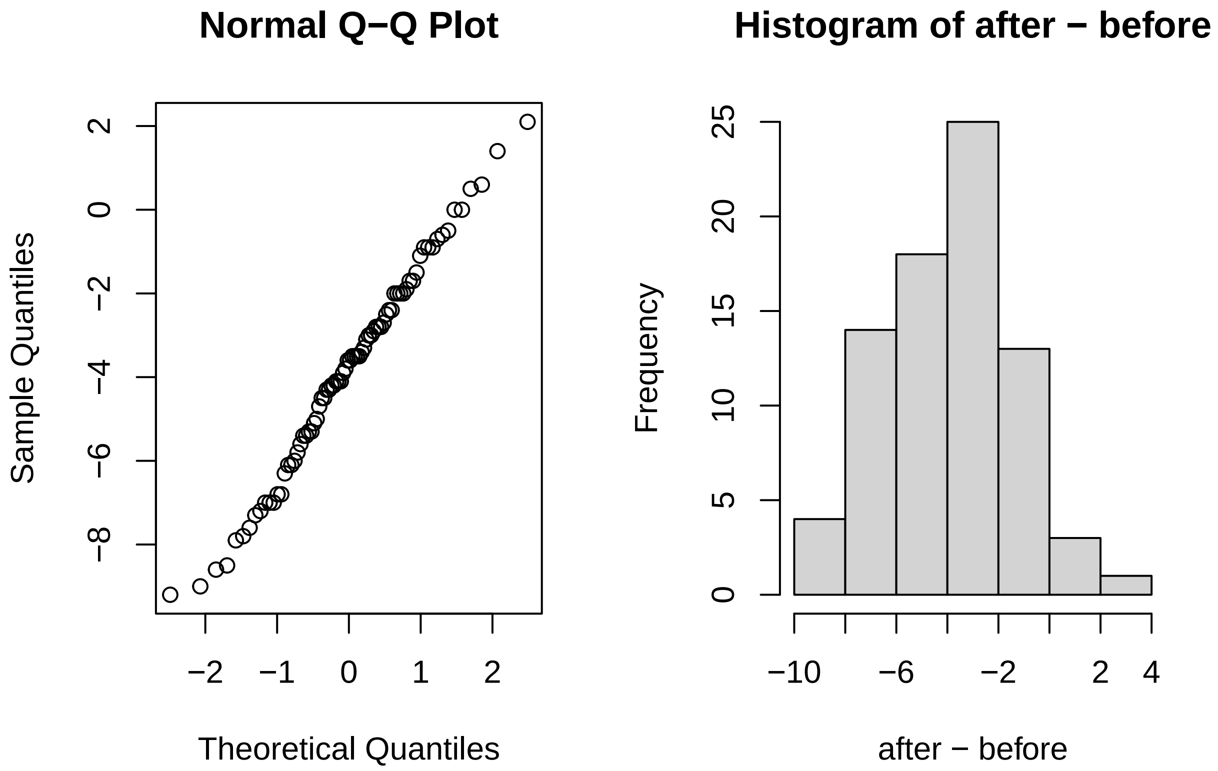
Both histograms do not look normal, but the difference of after-before from the Shapiro test does mostly look normal. So we can use the paired t-test. According to the t-test, the diet does affect the weight loss because of a p-value < 2.2e-16.






Normal Q−Q Plot
Sample Quantiles
60
70
80
90
100
−2
−1
0
1
2
Theoretical Quantiles


Histogram of after
Frequency
0
5
10
15
20
50
60
70
80
90
after



```
##
##  Paired t-test
##
## data:  before and after
## t = 13.309, df = 77, p-value < 2.2e-16
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  3.269602 4.420141
## sample estimates:
## mean difference
##        3.844872
```

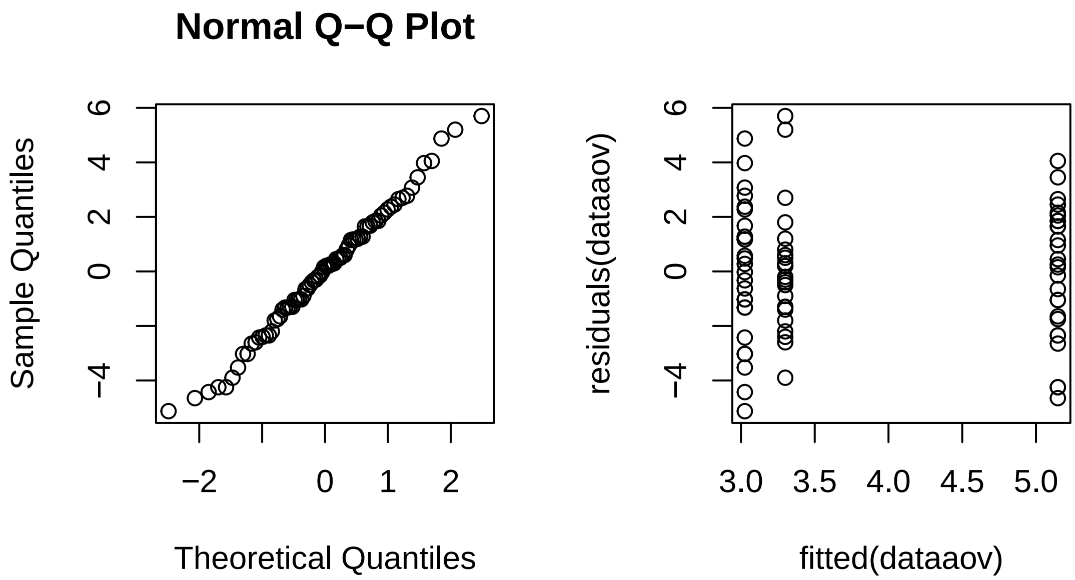
b)

We perform one-way ANOVA and it gives a p-value of 0.003. This means that diet has an effect on weight loss.

```
data$diet=factor(data$diet)
dataaov=lm(weight.lost~diet,data=data)
anova(dataaov)
## Analysis of Variance Table
##
## Response: weight.lost
##           Df Sum Sq Mean Sq F value   Pr(>F)
## diet       2  71.09  35.547  6.1974 0.003229 **
## Residuals 75 430.18   5.736
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We will check the assumptions. We perform a QQ-plot with the residuals for normality. The QQ-plot looks normal. The fitted values plot also does not really show a pattern.

```
par(mfrow=c(1,2)); qqnorm(residuals(dataaov))
plot(fitted(dataaov),residuals(dataaov))
```



We use the summary to be able to check the estimates. The group means are all >0, which means that the diets lead to weight loss and diet 3 is the best for losing weight.

```
summary(dataaov) #can we use fitted(dataaov here as well? )
##
## Call:
## lm(formula = weight.lost ~ diet, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1259 -1.3815  0.1759  1.6519  5.7000
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.3000     0.4889   6.750 2.72e-09 ***
## diet2        -0.2741     0.6719  -0.408  0.68449
## diet3         1.8481     0.6719   2.751  0.00745 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.395 on 75 degrees of freedom
## Multiple R-squared:  0.1418, Adjusted R-squared:  0.1189
## F-statistic: 6.197 on 2 and 75 DF,  p-value: 0.003229
```

c)

The two-way ANOVA test shows that diet and gender have an effect on the lost weight. The p-value of 0.049, shows that there is not a significant interaction between diet and gender.

```
genderaov=lm(weight.lost~gender*diet,data=data)
anova(genderaov)
## Analysis of Variance Table
##
## Response: weight.lost
##             Df Sum Sq Mean Sq F value   Pr(>F)
## gender       1   0.28  0.2785  0.0518 0.820623
## diet         2  60.42 30.2086  5.6190 0.005456 **
## gender:diet  2  33.90 16.9520  3.1532 0.048842 *
## Residuals   70 376.33  5.3761
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

e)

We prefer b, because we think diet and lost weight are the most relevant variables. Gender is a “fixed” variable which you cannot really change that easily.

The predicted weight loss for all diets: 3.300kg, 3.026kg and 5.148kg

```
preddata=data.frame(diet=c("1","2","3"))
predict(dataaov,preddata)
##        1        2        3
## 3.300000 3.025926 5.148148
```

Exercise 4

Load the MASS package and view the npk dataset

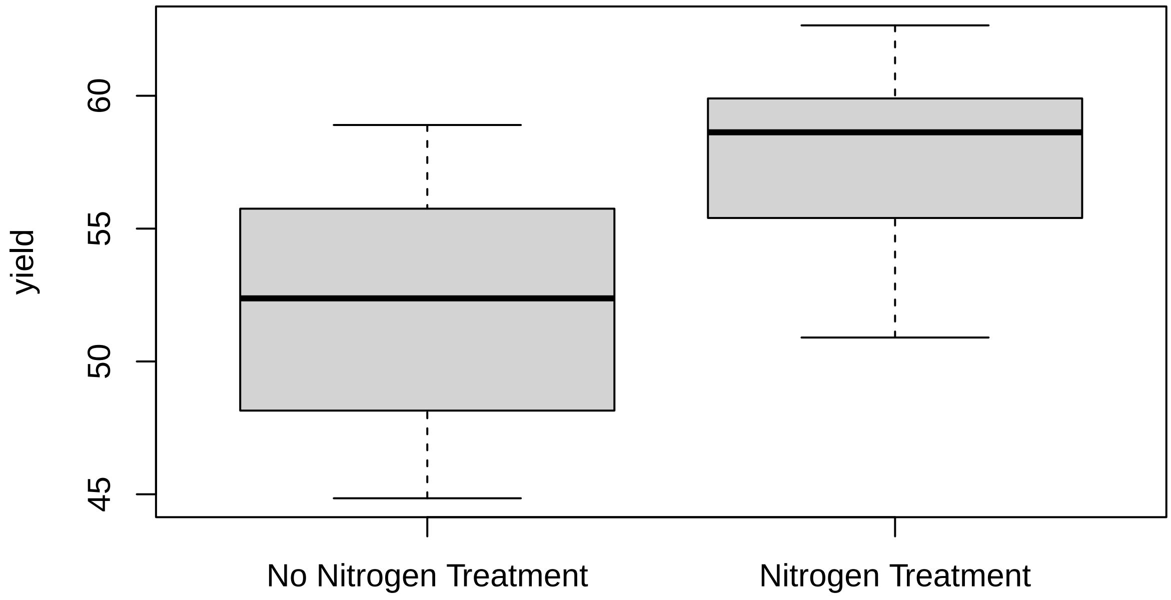
A)

```
#Create a matrix of random plots
random_plots_matrix <- cbind(rep(1:24),rep(1:6, each = 4),
                             replicate(3, c(replicate(6, sample(c(1,1,0,0))))))
#Convert the matrix to a data frame
data_frame <- data.frame(random_plots_matrix)
#Set column names
header <- c("plot", "block", "N", "P", "K")
colnames(data_frame) <- header
#View the resulting data frame
head(data_frame)
```

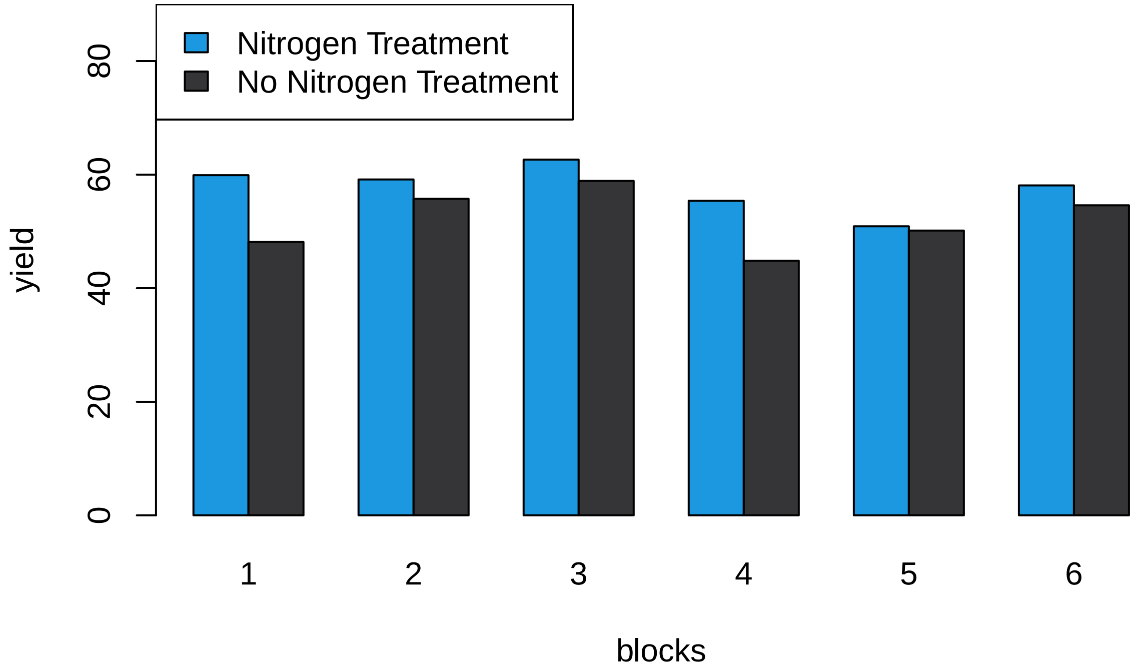
```
##   plot block N P K
## 1    1     1 0 0 0
## 2    2     1 0 0 1
## 3    3     1 1 1 1
## 4    4     1 1 1 0
## 5    5     2 1 0 0
## 6    6     2 1 1 1
```

B)

```
# Subset the data to only include nitrogen treatment
npk_nitrogen  <- npk[npk$N == 1,]
# Calculate the average yield per block for nitrogen treatment
nitrogen_avg_yield  <- aggregate(yield ~ block, data = npk_nitrogen, FUN = mean)
# Subset the data to only include plots without nitrogen treatment
npk_no_nitrogen  <- npk[npk$N == 0,]
# Calculate the average yield per block for plots without nitrogen
#treatment
no_nitrogen_avg_yield  <- aggregate(yield ~ block, data = npk_no_nitrogen, FUN = mean)
# Make a boxplot comparing the average yield per block for nitrogen and no nitrogen
#treatments
boxplot(no_nitrogen_avg_yield$yield, nitrogen_avg_yield$yield,
        names = c("No Nitrogen Treatment", "Nitrogen Treatment"), ylab = "yield")
```



```
#legend("topleft", legend = c("Nitrogen Treatment", "No Nitrogen Treatment"))


avg_yield <- apply(cbind(nitrogen_avg_yield[[2]], no_nitrogen_avg_yield[[2]]),
                   1,
                   function(x) unname(unlist(x)))
rownames(avg_yield)<- c("nitrogen_avg_yield","no_nitrogen_avg_yield")

barplot(avg_yield, col = c("#1b98e0", "#353436"),beside = TRUE,
        xlab="blocks",ylab = "yield", ylim=c(0,90),
        names.arg=c("1", "2", "3", "4", "5", "6"))

legend("topleft", legend = c("Nitrogen Treatment", "No Nitrogen Treatment"),
       fill =c("#1b98e0", "#353436"))
```



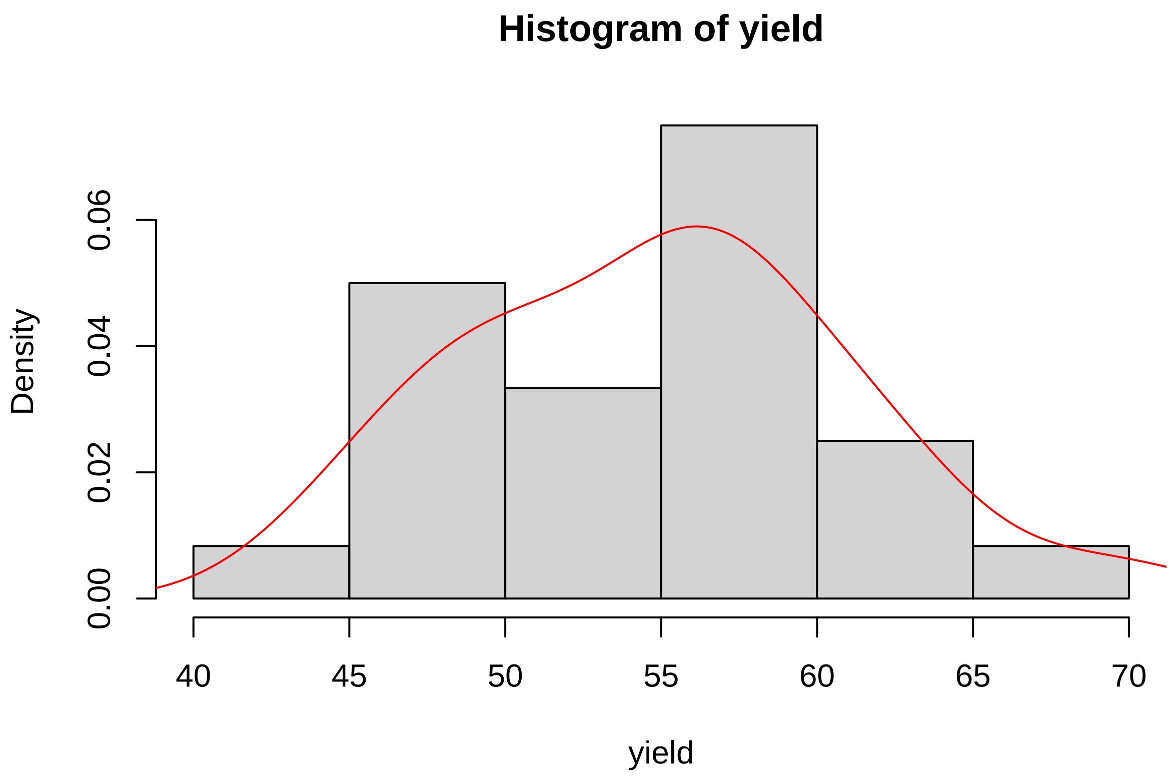
we can see that the average yield per block for the soil treated with nitrogen is higher than the average yield per block for the soil that did not receive nitrogen. This suggests that nitrogen had a positive effect on the yield of the crops.

It is important to take the factor block into account because it helps to control for any differences in yield that may be due to variation in the environment across different blocks. By including block as a factor, we can estimate the effect of nitrogen on yield while controlling for any potential confounding effects of block. This helps to increase the accuracy and reliability of the estimated effect of nitrogen on yield.

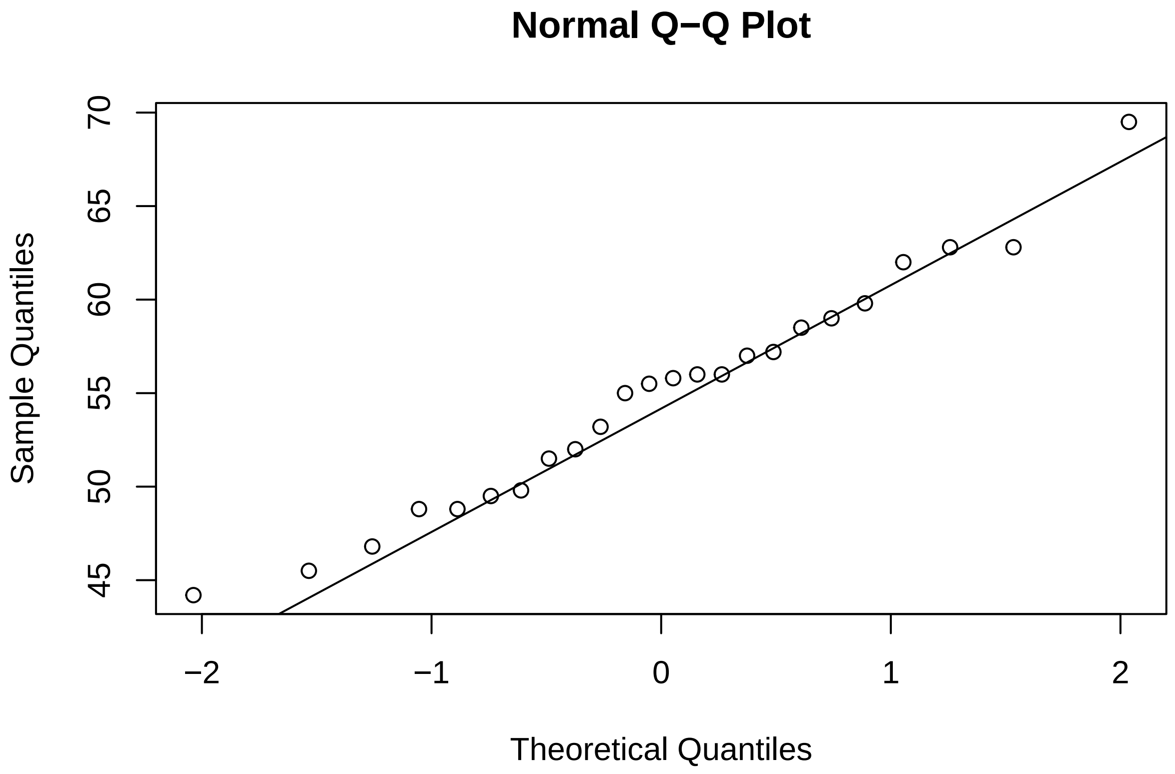
C)

Check for normality with QQ plot,histogram, and boxplot

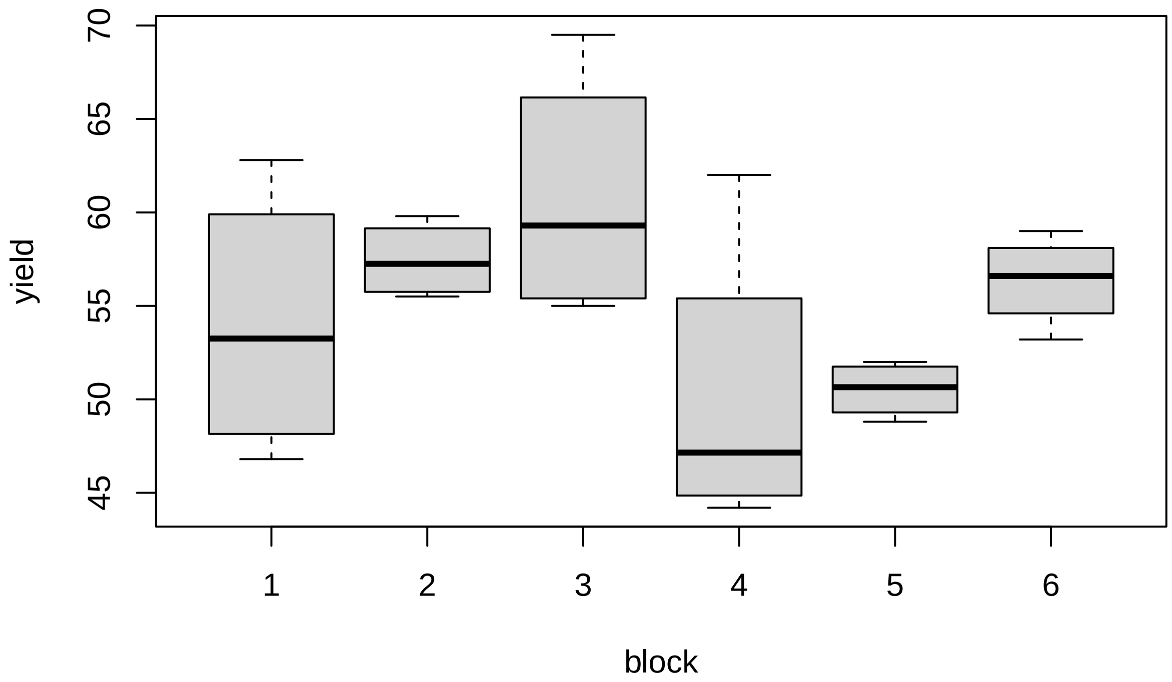
```
# Check for normality with histogram
hist(x = npk$yield , breaks=7 ,main = "Histogram of yield",
     xlab = " yield ",freq = FALSE)
lines(x= density(x= npk$yield), col=" red ")
```

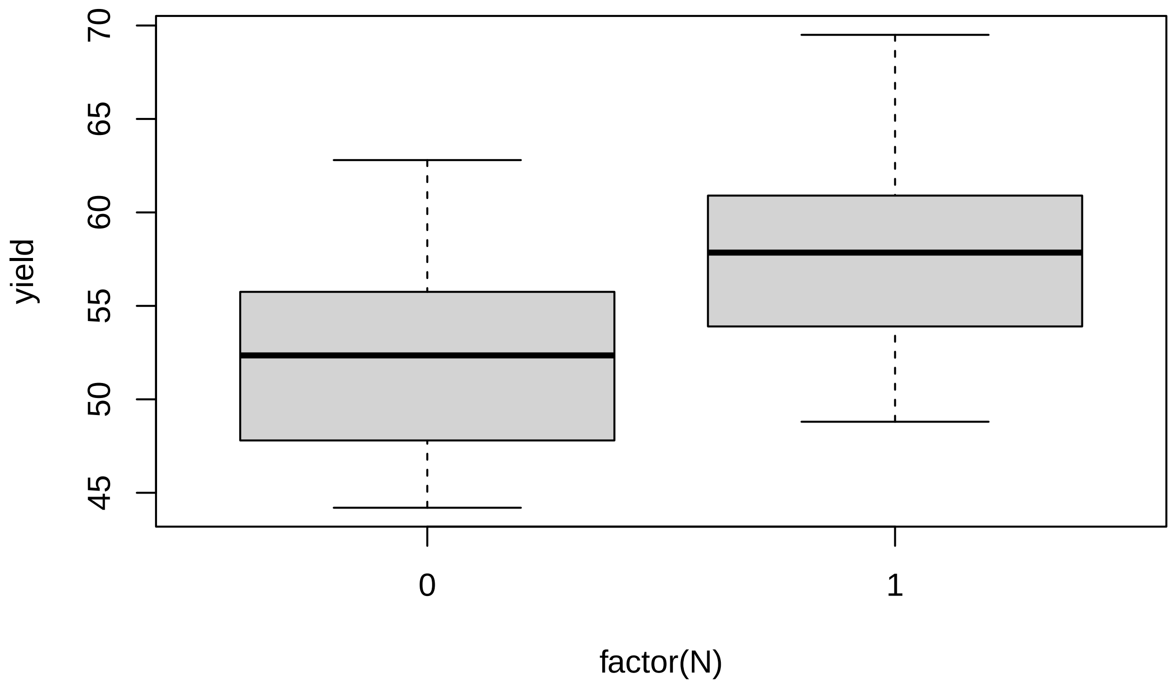


```
qqnorm(npk$yield)
qqline(npk$yield)
```



```
# Check for normality with boxplot
plot(yield ~ block + factor(N), data = npk)
```


yield
70
65
60
55
50
45
1
2
3
4
5
6
block



```
# Shapiro-Wilk test
shapiro.test(npk$yield)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  npk$yield
## W = 0.97884, p-value = 0.8735
```

the density displays a bell-shaped curve that resembles a normal distribution. The QQ-plot also appears to follow a straight line, indicating normality of the data.the Shapiro-Wilk test p-value of the test is greater than 0.05, we can assume that the data is normally distributed. In this case, the p-value is 0.87, which suggests that the data is normally distributed. The boxplot exhibits differences in the mean per block.

Given that the data appears to be normally distributed, we can proceed with conducting a two-way ANOVA with the response variable yield and the two factors block and N:

```
twoANOVA <- aov(yield ~ block * factor(N), data = npk)
summary(twoANOVA)
```

```
##                 Df Sum Sq Mean Sq F value Pr(>F)
## block            5  343.3   68.66   3.359 0.0397 *
## factor(N)        1  189.3  189.28   9.261 0.0102 *
## block:factor(N)  5   98.5   19.70   0.964 0.4769
## Residuals       12  245.3   20.44
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Based on the ANOVA table, it was sensible to include factor block into this model as it has a significant effect on yield with a p-value of 0.0397. The factor N also appears to have a significant effect on yield with a p-value of 0.0102.

The interaction term between block and N is not statistically significant, as its p-value is 0.4769. This suggests that the effect of block on yield does not depend on the level of N, and vice versa.

In conclusion, the ANOVA suggests that both block and N have significant effects on yield, and that it was sensible to include both factors in the model.

The Friedman test cannot be applied in this situation since the Friedman test requires more than one observation per combination of the factors. In this case, there is only one observation per combination of block and N, so the Friedman test cannot be used.

D)

```
model1 = lm(yield ~ N:block + P + K,data=npk)
model2 = lm(yield ~ P:block + N + K,data=npk)
model3 = lm(yield ~ K:block + P + N,data=npk)

print('model 1: ')
```

```
## [1] "model 1: "
```

```
anova(model1)
```

```
## Analysis of Variance Table
##
## Response: yield
##           Df Sum Sq Mean Sq F value  Pr(>F)
## P          1   8.40   8.402  0.5931 0.45904
## K          1  95.20  95.202  6.7201 0.02684 *
## N:block   11 631.09  57.372  4.0498 0.01785 *
## Residuals 10 141.67  14.167
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
print('model 2: ')
```

```
## [1] "model 2: "
```

```
anova(model2)
```

```
## Analysis of Variance Table
##
## Response: yield
##           Df Sum Sq Mean Sq F value   Pr(>F)
## N          1 189.28 189.282 11.2143 0.007381 **
## K          1  95.20  95.202  5.6404 0.038947 *
## P:block   11 423.10  38.463  2.2788 0.102669
## Residuals 10 168.79  16.879
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
print('model 3: ')
```

```
## [1] "model 3: "
```

```
anova(model3)
```

```
## Analysis of Variance Table
##
## Response: yield
##           Df Sum Sq Mean Sq F value   Pr(>F)
## P          1   8.40   8.402  0.4945 0.497989
## N          1 189.28 189.282 11.1397 0.007521 **
## K:block   11 508.77  46.251  2.7220 0.062935 .
## Residuals 10 169.92  16.992
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_add = lm(yield ~ N+P+K+block, data=npk)
print('model all sum: ')
```

```
## [1] "model all sum: "
```

```
anova(model_add)
```

```
## Analysis of Variance Table
##
## Response: yield
##           Df Sum Sq Mean Sq F value  Pr(>F)
## N          1 189.28 189.282 11.8210 0.00366 **
## P          1   8.40   8.402  0.5247 0.47999
## K          1  95.20  95.202  5.9455 0.02767 *
## block      5 343.29  68.659  4.2879 0.01272 *
## Residuals 15 240.19  16.012
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

model with all factors summed is prefered, given that it is the model with most significant factors.

e) According to the p-value, nitrogen has a significant effect on the yield. The fixed effects model in C shows similar results, where nitrogen also has a significant effect on the yield.

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
npklmer= lmer(yield~N+(1|block), REML=FALSE, data=npk)
npklmer2=lmer(yield~(1|block), REML=FALSE, data=npk)
npkmaov=anova(npklmer2, npklmer)
npkmaov
```

```
## Data: npk
## Models:
## npklmer2: yield ~ (1 | block)
## npklmer: yield ~ N + (1 | block)
##          npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## npklmer2    3 159.38 162.91 -76.690   153.38
## npklmer     4 153.48 158.20 -72.742   145.48 7.8953  1   0.004956 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```