Group 2

Silver Lee-A-Fong Jakub Lewkowicz Domantas Pukelevičius

2023-02-20

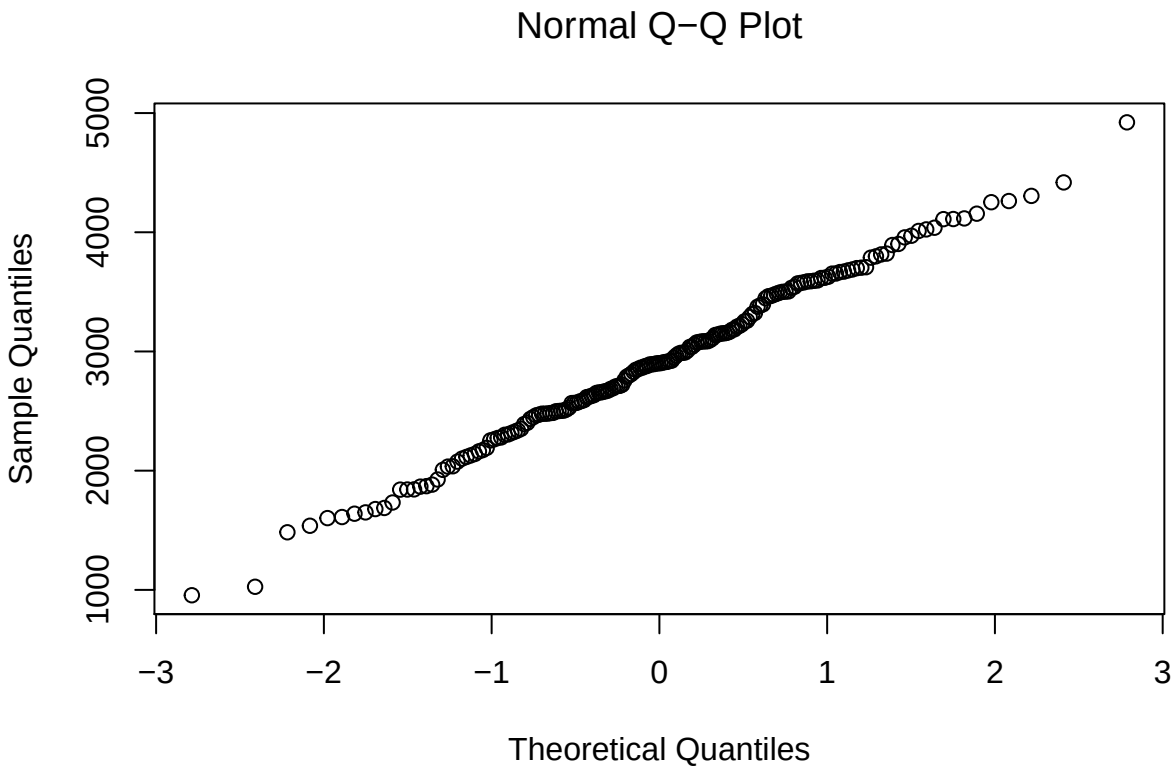
Exercise 1. Birthweights

a) QQ plot shows data distribution against the expected normal distribution. We can observe that data is normal, because observations lie approximately on a straight line. Shapiro test confirms normality of the data observed on the QQ plot (null hypothesis for Shapiro test is that data is normal, due to calculated p-value we do not reject H0).

```
df <- read.csv('Data/birthweight.txt')
shapiro.test(df$birthweight)[2]
```

```
## $p.value
## [1] 0.8995
```

```
qqnorm(df$birthweight, font.main=10)
```



We construct bounded 96%-CI by computing mean and margin of error based on standard deviation, Z score for 98th percentile and number of samples. To evaluate how many samples are needed to provide that the length of 96%-CI is at most 100, we run a loop that increments samples size and computes margin of error accordingly to a given sample size. Loop is terminated if CI length is smaller or equal to 100.

```
  n = nrow(df)
  mu = mean(df$birthweight)
  s = sd(df$birthweight)
  z_98p = 2.05
  m = z_98p*s/sqrt(n) # m = 1.96s/sqrt(n)
  bounded_CI = c(mu - m, mu + m); bounded_CI
```

```
## [1] 2809 3018
```

```
get_m = function(n) {
  s = sd(df$birthweight)
  z_98p = 2.05 # value from z score table for 98th percentile
  m = z_98p*s/sqrt(n)

  return(m)
}

for (sample_size in 1:1000) {
  lower_bound = mu - get_m(sample_size)
  upper_bound = mu + get_m(sample_size)
  CI_length = upper_bound - lower_bound
  if (CI_length <= 100) {
    break
  }
}
sample_size
```

```
## [1] 818
```

```
# bootstrap 96%-CI:
B = 1000
Tstar = 1:B
for (i in 1:B){
  Xstar = sample(df$birthweight, replace=TRUE)
  Tstar[i] = mean(Xstar)
}
Tstar20 = quantile(Tstar, 0.020)
Tstar980 = quantile(Tstar, 0.980)
sum(Tstar<Tstar20)
```

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

```
##       98%        2%
## 2810.7101 3015.1193
```

b) CI of 95% tells us that true average weight of a newborn baby in 95% situations is bigger than 2892.2 grams.

```
t.test(df$birthweight, mu=2800, alt="g")
```

```
##
##  One Sample t-test
##
## data:  df$birthweight
## t = 2.22708, df = 187, p-value = 0.013567
## alternative hypothesis: true mean is greater than 2800
## 95 percent confidence interval:
##  2829.2015       Inf
## sample estimates:
## mean of x
## 2913.2926
```

P value of 0.01357 means that H0 has to be rejected in favor of H1, which means that true mean is greater than 2800.

```
# sign test
p_value = binom.test(sum(df$birthweight > 2800), length(df$birthweight), alt='g')[3]
```

```
## [1] "0.0340"
```

c) We can compute powers of both tests by sampling from weights distribution, computing both t-tests and sign tetss for samples, accumulating results and computing final probabilities of rejecting null hypothesis. We can observe that power of t-test is bigger, due to the fact that t-tests work better for normally distributed data.

```
B = 1000
psign = numeric(B)
pttest = numeric(B)
n = 50
for(i in 1:B) {
  x = sample(df$birthweight, n)
  psign[i] = binom.test(sum(x>2800), n, alt='g')[[3]]
  pttest[i] = t.test(x, mu=2800, alt='g')[[3]]
}
power_sign = sum(psign<0.05)/B
power_ttest = sum(pttest<0.05)/B
```

```
c(power_sign, power_ttest)
```

```
## [1] 0.162 0.278
```

d) We calculated ˆ{p} (estimated mean of probability of getting weight under 2600) by taking 100 samples from initial birthweights sample and saving the proportions of weights above 2600. Then, we computed margin of error and upper bound of CI. With margin of error and standard deviation of ˆ{p}, we obtained that expert used 98% confidence level.

```r
n = 100
p_lower = 0.25
sample_probabilities = numeric(n)
for(i in 1:n){
  x = sample(df$birthweight, n)
  sample_probabilities[i] = sum(x < 2600)/n
}
s = sd(sample_probabilities)
p_estimate = mean(sample_probabilities)
m = p_estimate - p_lower;paste("margin of error:", round(m,3))
```

```
## [1] "margin of error: 0.081"
```

```r
p_upper = p_estimate + m
paste("estimated probability:", round(p_estimate,3))
```

```
## [1] "estimated probability: 0.331"
```

```r
paste("CI:", round(c(p_lower, p_upper),3))
```

```
## [1] "CI: 0.25"  "CI: 0.413"
```

```r
z_alpha = m/s;paste("estimated Z-score:",round(z_alpha,3))
```

```
## [1] "estimated Z-score: 2.512"
```

e) We decided to divide data into two subsets: weights under 2600 grams and weights above 2600 grams. We are sampling from both subsets with probabilities according to information about gender distribution. Next, we decided to perform a two-sampled t-test. Returned p-value does not indicate that expert's hypothesis is true.

```r
p_val = numeric(100)
for(i in 1:100) {
  males_u2600 = 34
  females_u2600 = 28
  males_a2600 = 61
  females_a2600 = 65
  under_2600 = df$birthweight[df$birthweight < 2600]
  above_2600 = df$birthweight[df$birthweight > 2600]
  under_2600
  samples_males_u2600_i = sample(1:length(under_2600), males_u2600)
  samples_males_u2600 = under_2600[samples_males_u2600_i]
  samples_females_u2600 = under_2600[-samples_males_u2600_i]
  samples_males_a2600_i = sample(1:length(above_2600), males_a2600)
  samples_males_a2600 = above_2600[samples_males_a2600_i]
  samples_females_a2600 = above_2600[-samples_males_a2600_i]

  samples_males = c(samples_males_a2600, samples_males_u2600)
  samples_females = c(samples_females_a2600, samples_females_u2600)
  p_val[i] = t.test(samples_males, samples_females)[[3]]
}
```

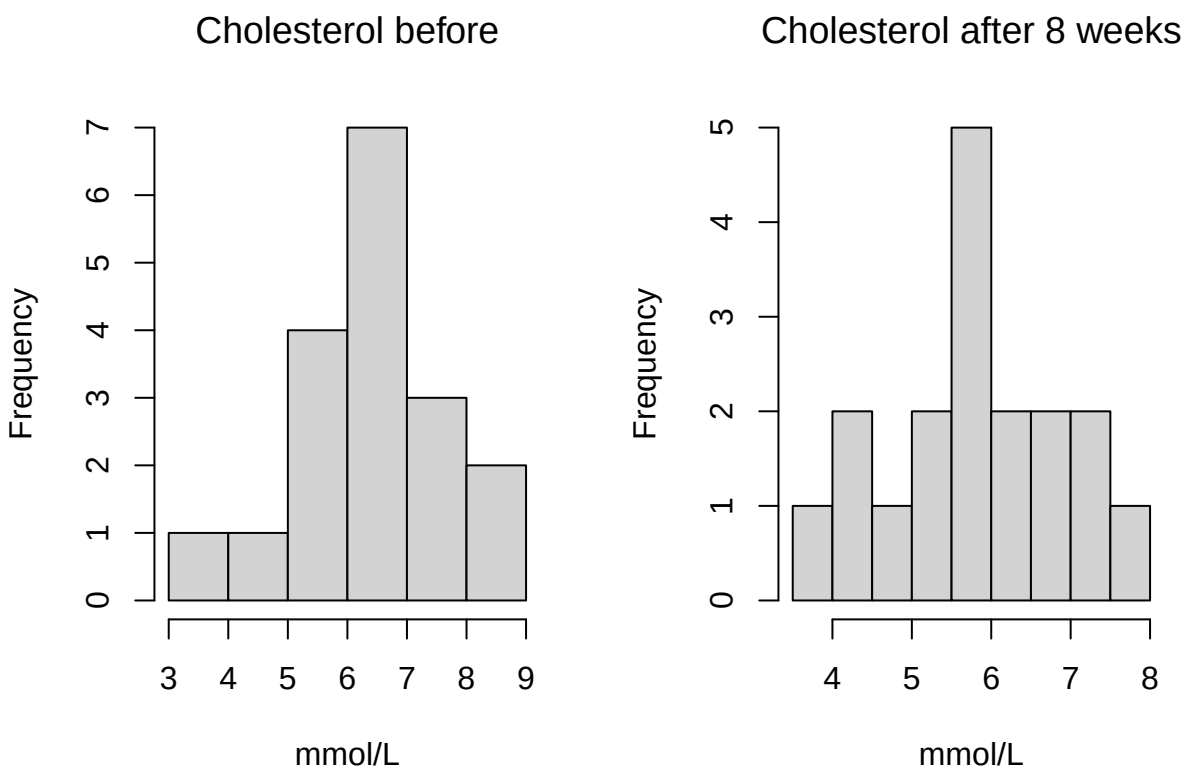
```r
mean(p_val)
```

```
## [1] 0.5539
```

Exercise 2: Cholersterol

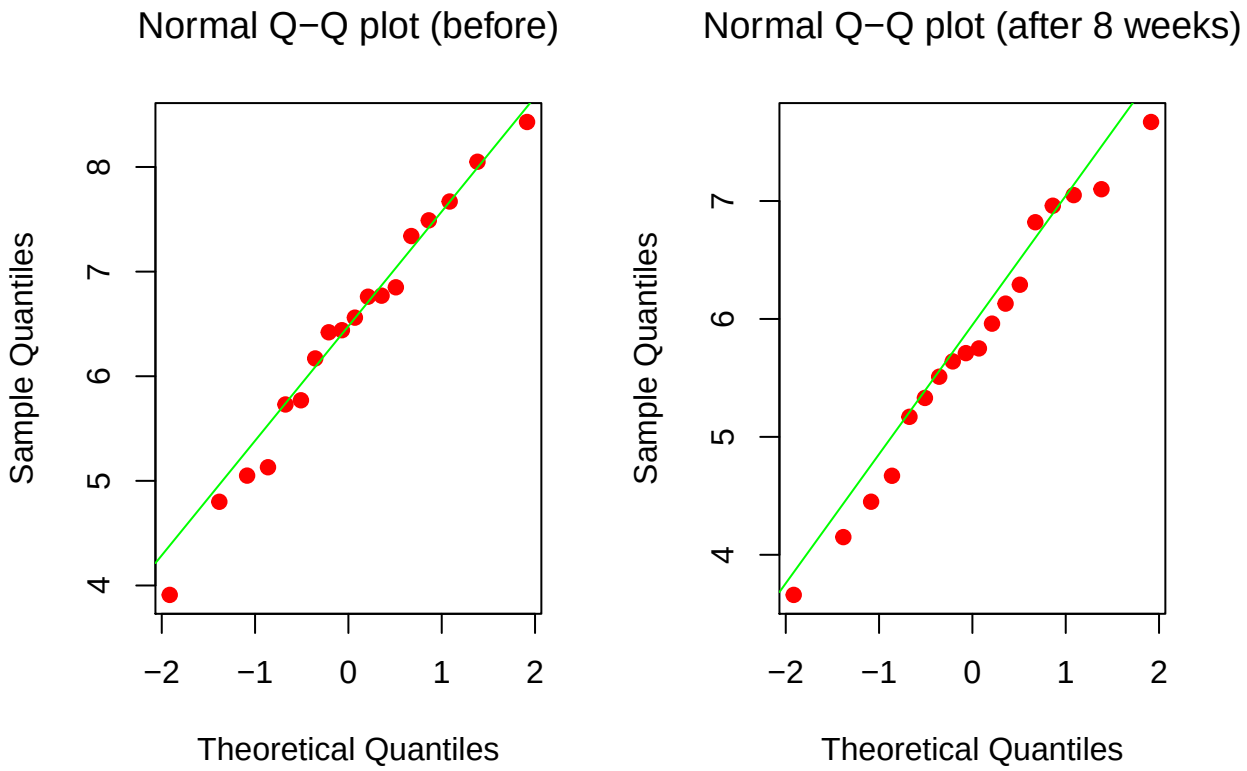
a) Histogram plots for cholesterol samples before low fat diet and after 8 weeks of low fat diet imply that data is normally distributed, since both histograms show symmetrical "bell" shape distribution. Normality is also implied by QQ plots in which a straight diagonal line show that theoretical quantiles of normal distribution match with sample quantiles (true for both samples). However, it should be taken in to the account, when assuming normality, that in both samples we only have 18 observations. Correlation between samples of cholesterol levels before, and after 8 week diet was computed to be 0.991. Since cholesterol levels were measured for the same sample of people before and after low fat diet, it can be expected that data will be highly correlated.

```r
df = read.csv('Data/cholesterol.txt', header = TRUE, sep = "")
par(mfrow=c(1,2))
hist(df$Before, main = 'Cholesterol before', xlab = 'mmol/L', font.main=10)
hist(df$After8weeks, main = 'Cholesterol after 8 weeks', xlab = 'mmol/L', font.main=10)
```

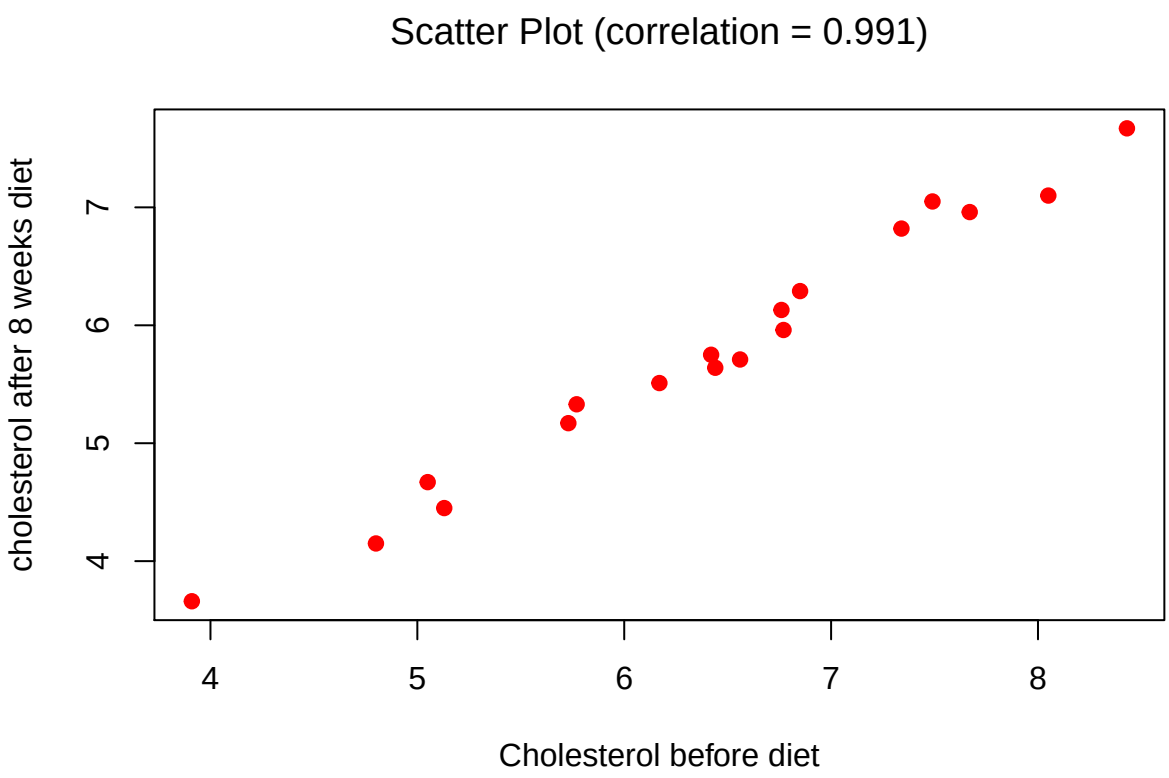



```r
par(mfrow=c(1,2))
qqnorm(df$Before, col='red', pch=19, main="Normal Q-Q plot (before)", font.main=10)
qqline(df$Before, col='green')
qqnorm(df$After8weeks, col='red', pch=19, main="Normal Q-Q plot (after 8 weeks)",
```

```
       font.main=10)
qqline(df$After8weeks, col='green')
```




```
plot(df$Before, df$After8weeks, xlab = "Cholesterol before diet",
     ylab = "cholesterol after 8 weeks diet",
     main = paste("Scatter Plot (correlation = ", round(cor(df$Before,df$After8weeks), 3),
                  collapes=")", sep=""), col='red',pch = 19, font.main=10)
```



b) To verify that low fat diet is effective in lowering cholesterol levels, paired t-test and paired Wilcoxon signed rank test were constructed, where $H_0 : \mu_{before} \leq \mu_{after8weeks}$ and $H_1 : \mu_{before} > \mu_{after8weeks}$. T-test provided us with p-value equal to 0.000 which allowed us to reject H_0, therefore we can conclude that low fat diet is indeed effective in lowering cholesterol levels. The Wilcoxon signed rank test having the same hypothesis resulted in p-value also equal to 0.000, which also allows us to confirm alternative hypothesis that $\mu_{before} > \mu_{after8weeks}$ is true. Our motives for choosing t-test and Wilcoxon signed rank test come from our data properties. The data set cholesterol features two-paired samples, in which experimental units (18 people) have two numerical outcomes (cholesterol levels (mmol/L)) - before treatment (diet) and after it. Also, it must be mentioned that both samples imply to be normally distributed (see Q-Q plots above). Therefore, two-paired nature of data and normality allows us to conduct paired t-test, and symmetry of data allows us computing Wilcoxon signed rank test.

Permutation test can also be applied in this case since we have a setting of two normally distributed paired samples.

```
ttest = t.test(df$Before, df$After8weeks, alt='g', paired=TRUE)
print(paste("p-value of two-paired t-test: ",round(ttest$p.value,3)))
```

```
## [1] "p-value of two-paired t-test:  0"
```

```
wilcox_test = wilcox.test(df$Before, df$After8weeks, alt='g',paired = TRUE)
print(paste("p-value of two-paired Wilcoxon signed rank test: ",
            round(wilcox_test$p.value,3)))
```

```
## [1] "p-value of two-paired Wilcoxon signed rank test:  0"
```

c) Assuming that $X_1, ..., X_{18} \sim Unif[3, \theta]$, where $X_1, ..., X_{18}$ is random variable from column *after8weeks*, we applied central limit theorem by drawing 18 samples with replacement from column *after8weeks* and calculating max cholesterol level in drawn sample, this step is repeated 1000 times to collect a set of maximum values. By computing mean for aforementioned maximum values set we estimate that $\hat{\theta} = 7.43$. Our computed 95% confidence interval - $[6.96, 7.67]$.

```
sample_maxs <- c()
n = 1000
for (i in 1:n){
  sample_maxs[i] = max(sample(df$After8weeks, nrow(df), replace=TRUE))
}
estimated_upper_limit = mean(sample_maxs)
cat("Estimated Theta:", estimated_upper_limit, "\n")
```

```
## Estimated Theta: 7.466
```

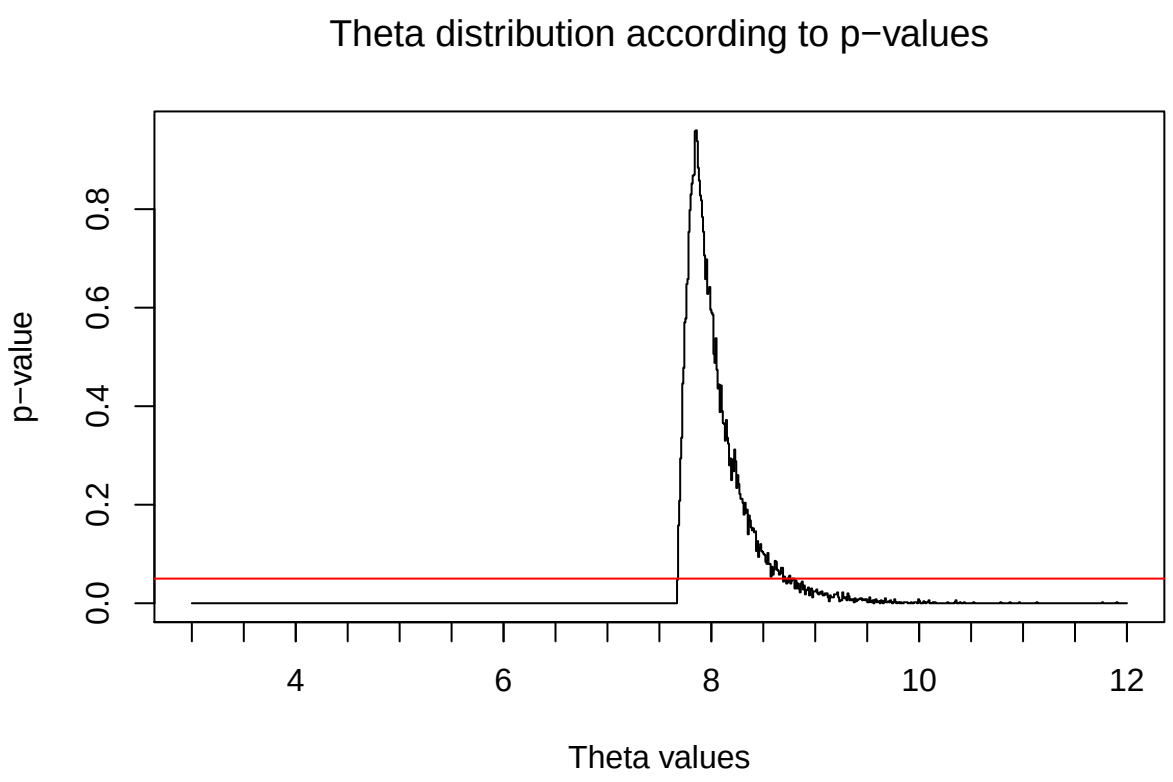
```
cat("Confidence interval:", quantile(sample_maxs, probs=c(0.025,0.975 )))
```

```
## Confidence interval: 6.96 7.67
```

d) Here, we construct bootstrap test with test statistic $T = max(X_1, ..., X_{18})$ and null hypothesis $H_0 : X_1, ..., X_{18} \sim Unif[3, \theta]$. in each bootstrap test iteration we take 18 samples from $\sim unif[3, \theta]$ distribution and we compute test statistic. This step is repeated $B = 1000$ times to obtain sample T^*, from which we then estimate p-value. At first we set $\theta = 3$ and we increment it by 0.01 after each bootstrap test, this is done until condition $\theta \leq 12$ is being satisfied. At the end we are left with a two sets of 901 θ and p-value values. Below, we plot theta and corresponding p-values, the red horizontal line marks 0.05 p-value, therefore, we can see that H_0 is not rejected when $\theta \in [7.68, 8.78]$.

Kolmogorov-Smirnov test can also be applied in this case. Let F_x denote *after8weeks* sample distribution and F_{X^*} denote distribution of $X^* \sim unif[3, \theta]$. Then with Kolmogorov-Smirnov test we would test whether we can reject $H_0 : F_x = F_{X^*}$. We would have to repeat this test for every θ in interval [3, 12] to find its values that fail to reject H_0.

```
theta = 3.00; t=max(df$After8weeks); counter = 1;B=1000;
tstar=numeric(B); p_values = c(); thetas = c();
while (theta <= 12) {
  for (i in 1:B){
    xstar = runif(n=nrow(df), min=3, max=theta)
    tstar[i]=max(xstar)
    }
  p_left=sum(tstar<t)/B; p_right=sum(tstar>t)/B;
  p_values[counter]= 2*min(p_left,p_right)
  thetas[counter] = theta
  counter = counter + 1
  theta = theta + 0.01 #increment theta by  0.01
}
plot(x=thetas, y = p_values, type = "S", xlab = "Theta values", ylab="p-value",
     main = "Theta distribution according to p-values", font.main=10)
axis(1,at=seq(0,12,0.5),labels=NA)
abline(a=0.05,b=0, col='red')
```




```
df_theta = data.frame(thetas,p_values)
df_theta = dplyr::filter(df_theta, p_values > 0.05)
interval <- c(min(df_theta$theta), max(df_theta$theta))

print(interval)
```

```
## [1] 7.69 8.76
```

e) To test whether median cholesterol level after 8 weeks of low fat diet is less than 6, we chose to conduct sign test, with $H_0 : m_x \leq 6$ and $H_1 : m_x > 6$. The obtained p-value of 0.9 fails to reject null hypothesis, therefore, we confirm that the median cholesterol levels after 8 weeks of low fat diet is less than 6.

To check whether the fraction of the cholesterol levels after 8 weeks of low fat diet less than 4.5 is at most 25%, we also chose sign test, with $H_0 : m_x \leq 4.5$ and $H_1 : m_x > 4.5$. Since sign test for median in R is done with binom.test(), we can specify the hypothesized probability of success of 0.25. The p-value of this sign test was compute to be equal to 0, therefore, we can reject null hypothesis. Thus, we confirm that the alternative hypothesis is true that the fraction of the cholesterol levels after 8 weeks of low fat diet less than 4.5 is at most 25%.

```
binom.test(sum(df$After8weeks > 6), nrow(df), alt='g')
```

```
##
##  Exact binomial test
##
```

```
## data:  sum(df$After8weeks > 6) and nrow(df)
## number of successes = 7, number of trials = 18, p-value = 0.9
## alternative hypothesis: true probability of success is greater than 0.5
## 95 percent confidence interval:
##  0.199 1.000
## sample estimates:
## probability of success
##                 0.3889
```

```
binom.test(sum(df$After8weeks > 4.5), nrow(df), alt='g', p=0.25)
```

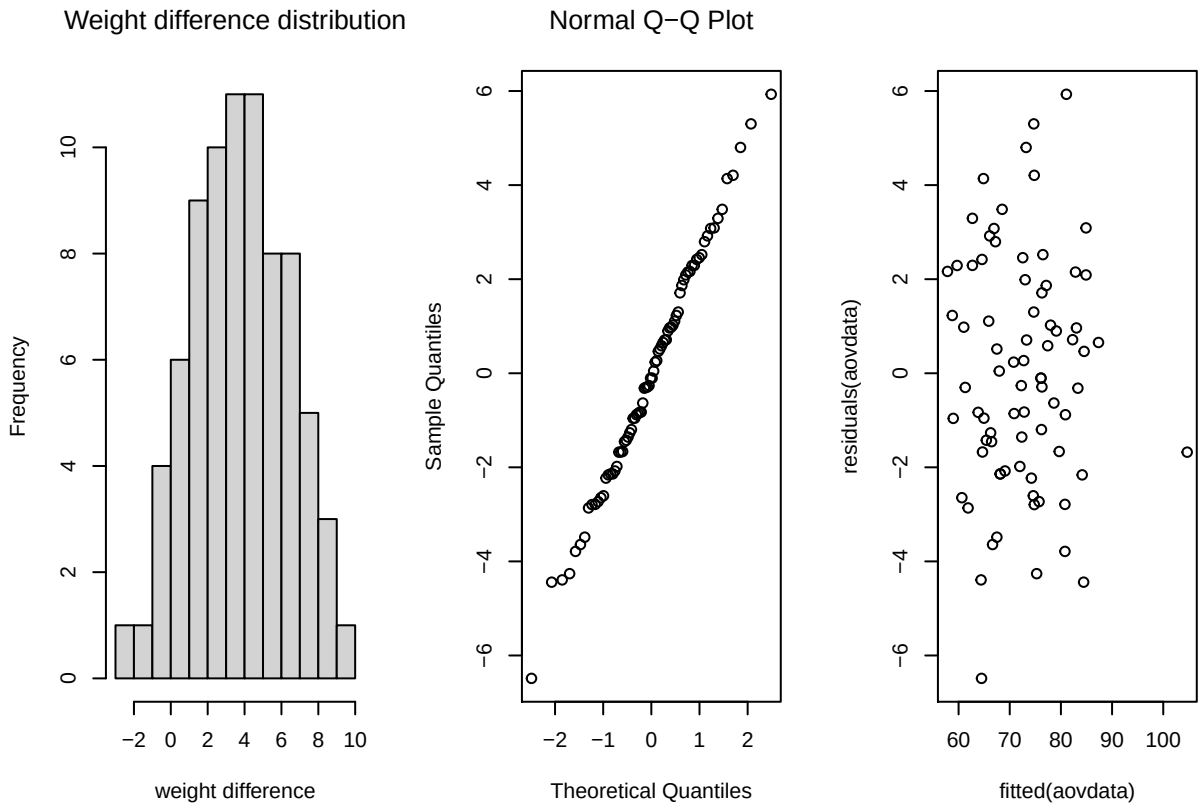
```
##
##  Exact binomial test
##
## data:  sum(df$After8weeks > 4.5) and nrow(df)
## number of successes = 15, number of trials = 18, p-value = 3e-07
## alternative hypothesis: true probability of success is greater than 0.25
## 95 percent confidence interval:
##  0.6233 1.0000
## sample estimates:
## probability of success
##                 0.8333
```

Exercise 3: Diet

```
data <- read.csv("Data/diet.txt",header = TRUE, sep = "")
```

a) To test for normality, we first create a histogram to check the distribution of the weight loss for all data. Using qq-plot to plot the residuals of the linear model of preweight data and weight6weeks data, we can observe that the data has been distributed normally. A plot of fitted data against residuals of the data shows that there is no pattern in the data, this is another confirmation for normality. The t-test results in a $p-value < 0.05$, showing that we can reject the null hypothesis saying that the mean difference equals zero.

```
aovdata = lm(preweight~weight6weeks, data=data)
par(mfrow=c(1, 3))
hist(data$preweight - data$weight6weeks, breaks=10, font.main=10,
     main="Weight difference distribution", xlab="weight difference")
qqnorm(residuals(aovdata), font.main=10)
plot(fitted(aovdata),residuals(aovdata), font.main=10)
```




```
t.test(data$preweight, data$weight6weeks, paired=TRUE)
```

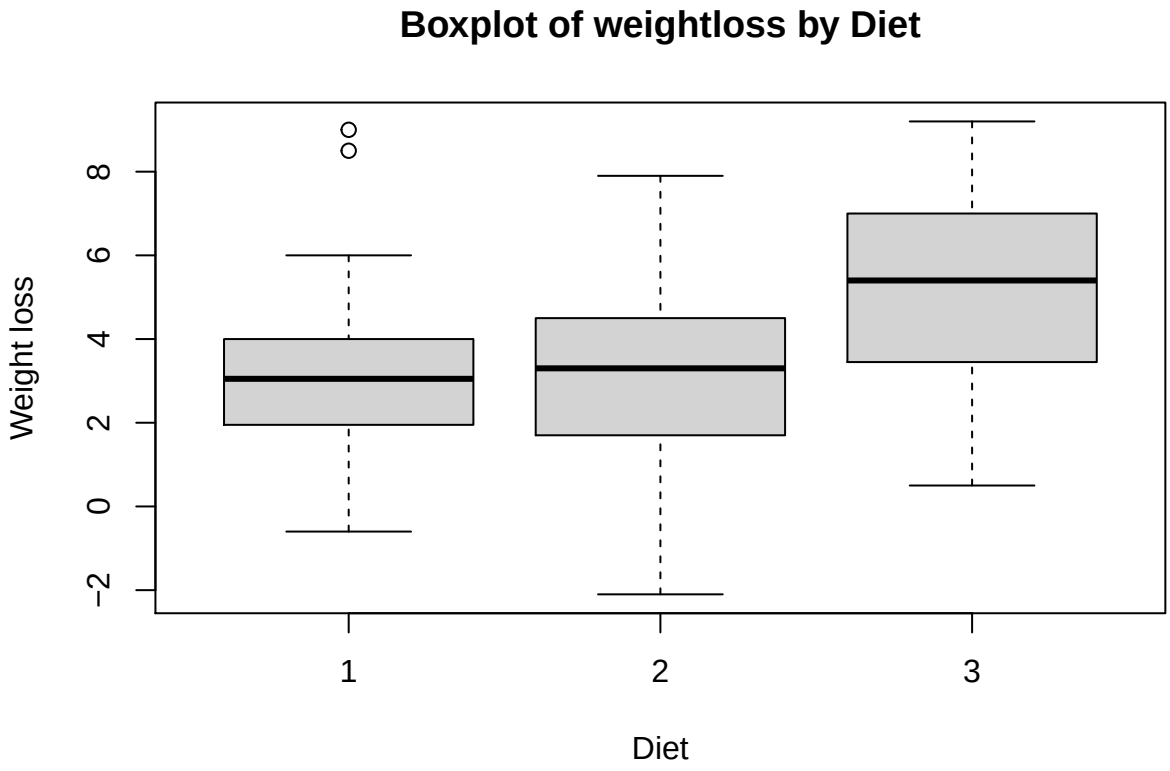
```
##
##  Paired t-test
##
## data:  data$preweight and data$weight6weeks
## t = 13, df = 77, p-value <2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  3.27 4.42
## sample estimates:
## mean of the differences
##                   3.845
```

b) We use one-way ANOVA to test whether the type of diet has an effect on the lost weight. The result of one-way ANOVA will help us to find out whether all three types of diets lead to weight loss. The result of one-way ANOVA gives us a p-value that is < 0.05, meaning that we can reject the null hypothesis stating that the mean difference is zero. From this it can be said that the mean weight loss is affected by the three types of diets. Using a box plot we visualize that diet 3 has the largest amount of weight loss, indicating that this diet is losing the most weight on average. For this situation Kruskal-Wallis can be used to test whether the medians are different for multiple groups. The outcome would describe whether the effects of the three diets are the same or not on the weight lost. Kruskal-Wallis can be applied but won't provide significant data to tell which diet was best for losing weight.

```r
model_b <- aov(preweight - weight6weeks ~ diet, data=data)
summary(model_b)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## diet         1     46    45.8    7.64 0.0072 **
## Residuals   76    455     6.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
boxplot((preweight-weight6weeks) ~ diet, data = data, xlab = "Diet",
        ylab = "Weight loss", main = "Boxplot of weightloss by Diet")
```



c) With the use of two-way ANOVA, we investigate the effect of the diet and gender (and possible interaction) on the lost weight. The results show that there is interaction between the lost weight and the gender. It is observed that the means of observations grouped by the factors are the not the same since the p-value of diet and gender is < 0.05.

```r
data$gender = as.factor(data$gender)
data$diet =  as.factor(data$diet)

model_c <- lm(preweight - weight6weeks ~ diet * gender, data = data)
anova(model_c)
```

```
## Analysis of Variance Table
##
## Response: preweight - weight6weeks
##             Df Sum Sq Mean Sq F value Pr(>F)
## diet         2     61   30.26    5.63 0.0054 **
## gender       1      0    0.17    0.03 0.8599
## diet:gender  2     34   16.95    3.15 0.0488 *
## Residuals   70    376    5.38
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

e) The two-way ANOVA model is preferred because it gives in depth information about the data that is provided. The interaction between diet and gender can be analyzed and which effect this has on the weight loss. The one-way ANOVA model does not give this much information about the dataset because we can only have one independent variable in the test. The predicted weight for diet 1 is 69.58, for diet 2 is 68.09 and for diet 3 68.48.

```
diet_1 = filter(data, diet==1)
diet_2 = filter(data, diet==2)
diet_3 = filter(data, diet==3)

avg_loss_diet_1 = mean(diet_1$preweight) - mean(diet_1$weight6weeks)
avg_loss_diet_2 = mean(diet_2$preweight) - mean(diet_2$weight6weeks)
avg_loss_diet_3 = mean(diet_3$preweight) - mean(diet_3$weight6weeks)

diet_1_predicted = mean(diet_1$preweight) - avg_loss_diet_1; diet_1_predicted
```

```
## [1] 69.58
```

```
diet_2_predicted = mean(diet_2$preweight) - avg_loss_diet_2; diet_2_predicted
```

```
## [1] 68.09
```

```
diet_3_predicted = mean(diet_3$preweight) - avg_loss_diet_3; diet_3_predicted
```

```
## [1] 68.48
```

Exercise 4: Yield of peas

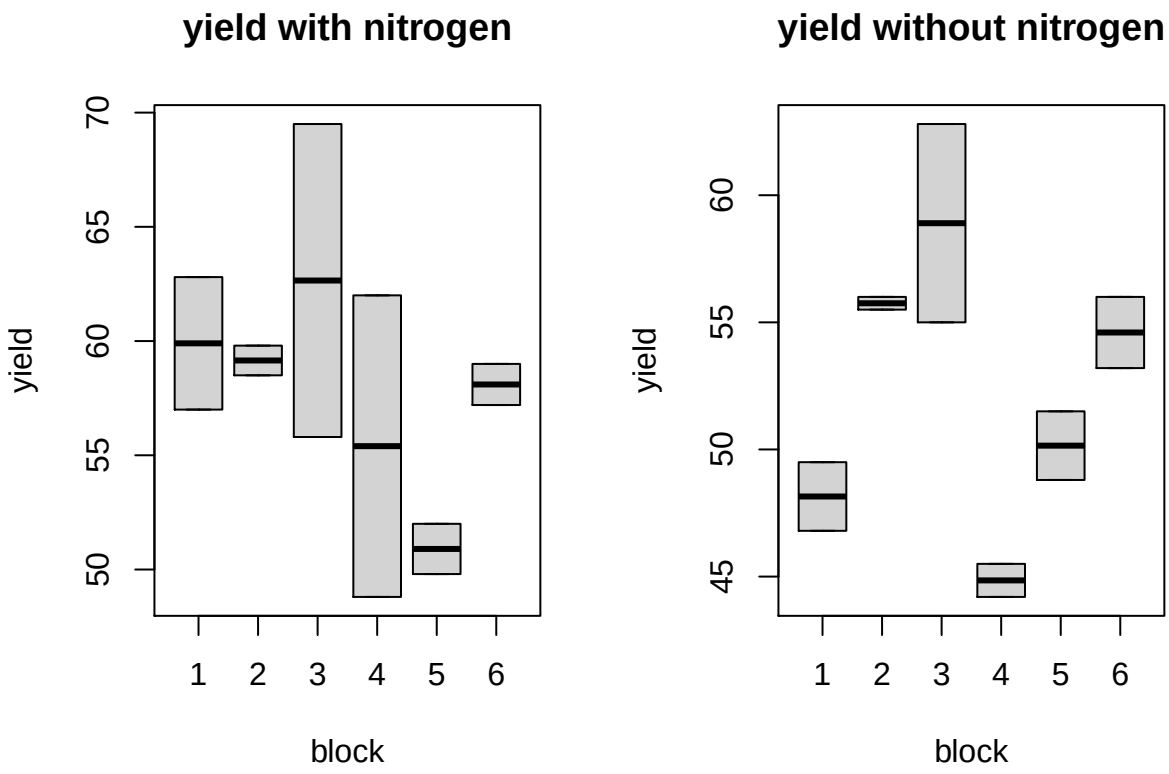
a) I - treatment levels (additives), B - blocks, N - repetitions

```
I=3; B=6; N=2
for (i in 1:B) print(sample(1:(N*I)))
```

```
## [1] 2 5 4 1 3 6
## [1] 3 1 2 6 5 4
## [1] 2 6 3 1 4 5
## [1] 4 2 5 1 6 3
## [1] 6 1 3 5 2 4
## [1] 2 4 3 6 1 5
```

b) two boxplots, one for nitrogen containing yields, and another for yields without nitrogen, show us that the average yield of plots with nitrogen within a each block is larger than average yield of plots without nitrogen within each block. The purpose to take block factor into account is to isolate any exogenous effects that may be present in certain blocks, while being absent in others, for example: one block might already contain high levels of nitrogen in the soil (naturaly), while the other block has low levels of nitrogen (also naturaly).

```
df = MASS::npk
df_nitrogen = dplyr::filter(df, N == 1)
df_no_nitrogen =  dplyr::filter(df, N == 0)
par(mfrow=c(1,2))
boxplot(yield ~ block, data=df_nitrogen, main="yield with nitrogen")
boxplot(yield ~ block, data=df_no_nitrogen, main='yield without nitrogen')
```



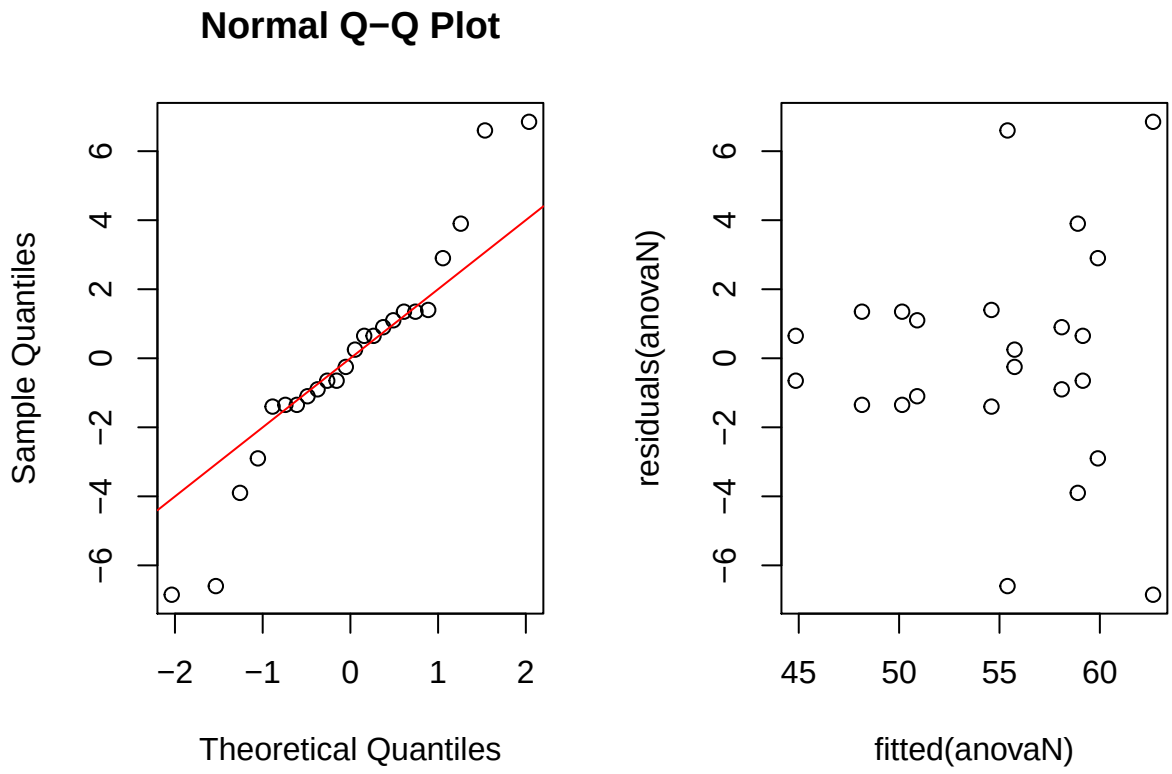
c) Based on both plots normality of residuals is doubtful. On first plot due to not forming a straight line, on second one due to the pattern than can be observed (ideally we do not want to observe any pattern). We can observe based on p-value that Nitrogen has significant effect on yield. It was sensible to include factor block in this model to only focus on a factor of our interest (Nitrogen). Friedman test cannot be applied, because a condition of having one observation per cell is not met.

```
df$block = as.factor(df$block)
df$N = as.factor(df$N)
anovaN = lm(yield ~ block * N,data=df);anova(anovaN)
```

```
## Analysis of Variance Table
```

```
##
## Response: yield
##           Df Sum Sq Mean Sq F value Pr(>F)
## block      5    343    68.7    3.36   0.04 *
## N          1    189   189.3    9.26   0.01 *
## block:N    5     99    19.7    0.96   0.48
## Residuals 12    245    20.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

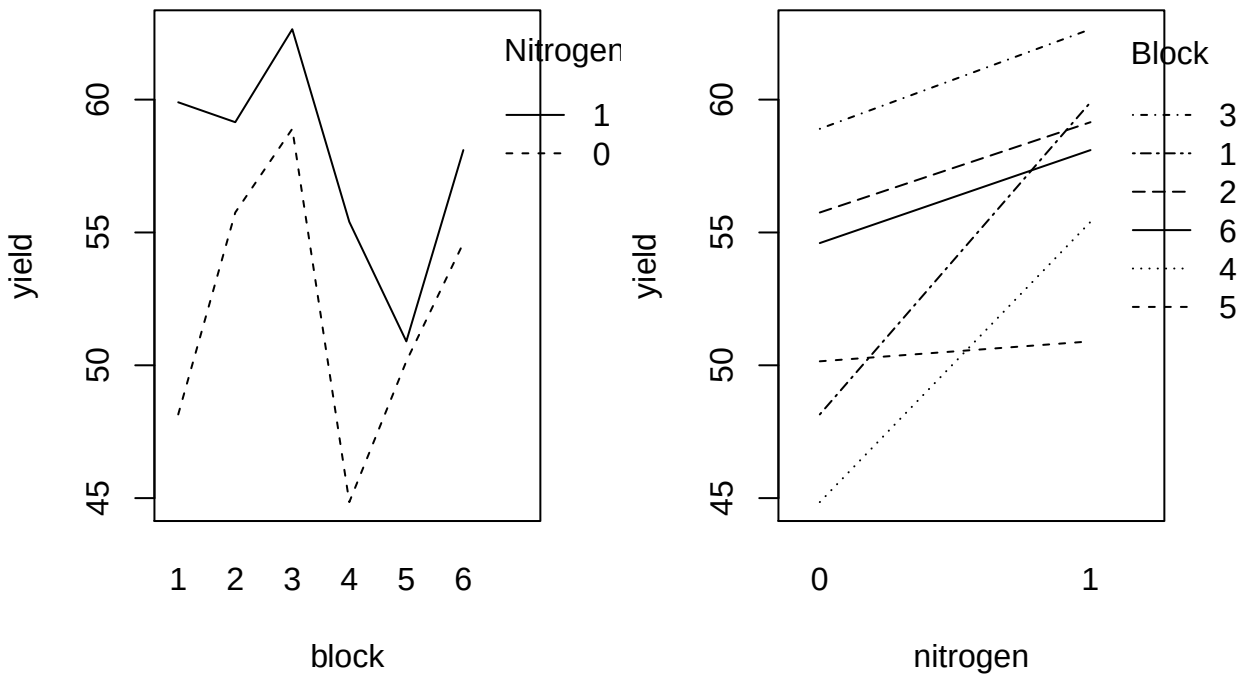
```
par(mfrow=c(1,2))
qqnorm(residuals(anovaN));qqline(residuals(anovaN),col='red')
plot(fitted(anovaN),residuals(anovaN))
```




```
additive_model = lm(yield ~ block + N,data=df);anova(additive_model)
```

```
## Analysis of Variance Table
##
## Response: yield
##           Df Sum Sq Mean Sq F value Pr(>F)
## block      5    343    68.7    3.40 0.0262 *
## N          1    189   189.3    9.36 0.0071 **
## Residuals 17    344    20.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
par(mfrow=c(1,2))
interaction.plot(df$block,df$N,df$yield, trace.label='Nitrogen',xlab='block',ylab='yield')
interaction.plot(df$N, df$block, df$yield, trace.label='Block', xlab='nitrogen', ylab='yield')
```



d) First, we constructed three ANOVA models (model_a, model_b, model_c), where we included all additives and in each model we checked the interaction between additive and block. Since we found that there is no interaction between block and the additives, we computed ANOVA with no interaction (additive model). In additive model (model_d) we see that additives N and K, and also block had p-value<0.05, which allow us to state that N, K and block do have an effect on yield. Also, we have noticed that potassium have no significant effect on yield, since its p-value in all four models was larger than 0.05.

```
df$K = as.factor(df$K)
df$P = as.factor(df$P)
model_a = lm(yield ~ N + P + block*K, data=df); anova(model_a)
```

```
## Analysis of Variance Table
##
## Response: yield
##           Df Sum Sq Mean Sq F value Pr(>F)
## N          1    189   189.3   11.14 0.0075 **
## P          1      8     8.4    0.49 0.4980
## block      5    343    68.7    4.04 0.0288 *
## K          1     95    95.2    5.60 0.0395 *
## block:K    5     70    14.1    0.83 0.5583
```

```
## Residuals 10    170    17.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_b = lm(yield ~ N + K + block*P, data=df); anova(model_b)
```

```
## Analysis of Variance Table
##
## Response: yield
##           Df Sum Sq Mean Sq F value Pr(>F)
## N          1    189   189.3   11.21 0.0074 **
## K          1     95    95.2    5.64 0.0389 *
## block      5    343    68.7    4.07 0.0282 *
## P          1      8     8.4    0.50 0.4966
## block:P    5     71    14.3    0.85 0.5473
## Residuals 10    169    16.9
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_c = lm(yield ~ K + P + block*N, data=df); anova(model_c)
```

```
## Analysis of Variance Table
##
## Response: yield
##           Df Sum Sq Mean Sq F value Pr(>F)
## K          1     95    95.2    6.72 0.0268 *
## P          1      8     8.4    0.59 0.4590
## block      5    343    68.7    4.85 0.0164 *
## N          1    189   189.3   13.36 0.0044 **
## block:N    5     99    19.7    1.39 0.3066
## Residuals 10    142    14.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_d = lm(yield ~ N + K + P + block, data=df); anova(model_d)
```

```
## Analysis of Variance Table
##
## Response: yield
##           Df Sum Sq Mean Sq F value Pr(>F)
## N          1    189   189.3   11.82 0.0037 **
## K          1     95    95.2    5.95 0.0277 *
## P          1      8     8.4    0.52 0.4800
## block      5    343    68.7    4.29 0.0127 *
## Residuals 15    240    16.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

e) we estimated that the variance of the normal population of the block effect is 54.65, which was also found to be significant. This indicates that including block as interaction variable in part C, allow us to isolate this deviation.

```
yieldlmer=lmer(yield~N+P+K+(1|block),REML=FALSE,data=df);yieldlmer
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: yield ~ N + P + K + (1 | block)
##    Data: df
##      AIC      BIC   logLik deviance df.resid
##   151.03   158.10   -69.51   139.03       18
## Random effects:
##  Groups   Name        Std.Dev.
##  block    (Intercept) 3.31
##  Residual             3.65
## Number of obs: 24, groups:  block, 6
## Fixed Effects:
## (Intercept)           N1           P1           K1
##       54.65         5.62        -1.18        -3.98
```

```
yieldlmer1=lmer(yield~(1|block),REML=FALSE,data=df)
anova(yieldlmer1,yieldlmer)
```

```
## Data: df
## Models:
## yieldlmer1: yield ~ (1 | block)
## yieldlmer: yield ~ N + P + K + (1 | block)
##            npar AIC BIC logLik deviance Chisq Df Pr(>Chisq)
## yieldlmer1    3 159 163  -76.7      153
## yieldlmer     6 151 158  -69.5      139  14.3  3     0.0025 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```