Assignment 1

Shutong Cai, Stijn de Groot, Yichen Tong, group 20

26 February 2023

Exercise 1

a) Firstly, we import the data file into RStudio using the function `read.table()`. Next, we perform a normality test on the data using the `shapiro.test()` function. The test checks whether the data follows a normal distribution.

```
birthweight <- read.table("birthweight.txt", header = TRUE)
shapiro.test(birthweight$birthweight)
##
##  Shapiro-Wilk normality test
##
## data:  birthweight$birthweight
## W = 0.99595, p-value = 0.8995
```

After that, we construct a bounded 96% confidence interval based on the assumption that this data follows a normal distribution. The confidence interval is calculated using the sample `mean` and `std_error`, where the t-distribution has degrees of freedom `n-1`.

```
n <- length(birthweight$birthweight)
alpha <- 0.04 / 2
mean_bw <- mean(birthweight$birthweight)
std_error <- sd(birthweight$birthweight) / sqrt(n)
lower_bound <- mean_bw - qt(1-alpha, n-1) * std_error
upper_bound <- mean_bw + qt(1-alpha, n-1) * std_error
cat("The 96% CI for mean birthweight is [", round(lower_bound, 2), ", ", round(upper_bound, 2), "]\n")
## The 96% CI for mean birthweight is [ 2808.08 ,  3018.5 ]
```

We then calculate sample size needed to provide a confidence interval with a maximum length of 100. The calculation is based on the maximum allowable standard error and assumes the same level of confidence.

```
max_length <- 100
std_error_max <- max_length / (2 * qt(1-alpha, n-1))
n_min <- ceiling((2 * sd(birthweight$birthweight) / std_error_max)^2)
cat("The minimum sample size needed to achieve the maximum length of 100 for the 96% CI is", n_min, "\n
## The minimum sample size needed to achieve the maximum length of 100 for the 96% CI is 3330
```

We uses the bootstrap method to estimate a confidence interval. The bootstrap method involves resampling the dataset with replacement many times, and calculating the sample mean for each resample. The 96% confidence interval is then computed using the resulting distribution of sample means. Then we output the results of the analyses, including the 96% confidence interval for mu based on the normality assumption, the minimum sample size needed to achieve the maximum length for the confidence interval, and the 96% bootstrap confidence interval for mu. By comparing the results of the two confidence intervals, one can evaluate the suitability of the normality assumption. The sample size estimate is also useful for ensuring that the confidence interval length meets the requirement.

```
set.seed(123)
# Set random seed for reproducibility
bootstrap_mean <- replicate(10000, mean(sample(birthweight$birthweight, n, replace = TRUE)))
bootstrap_ci <- quantile(bootstrap_mean, c(alpha, 1-alpha))
cat("The 96% bootstrap CI for mean birthweight is [", round(bootstrap_ci[1], 2), ", ", round(bootstrap_c
## The 96% bootstrap CI for mean birthweight is [ 2808.4 ,  3018.71 ]
```

b) T-test: As we have proved in a) that the data *birthweight* follows the normal distribution, so the p-value from the t-test can be trusted. In this t-test, we have: H_0: $\mu \leq 2800$; H_1: $\mu > 2800$

```
t.test(birthweight,mu = 2800,alt = "g") #one sample right-sided t-test - birthwight
```

```
##
##  One Sample t-test
##
## data:  birthweight
## 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
```

Here we have the p-value = 0.014 < 0.05. So H_0 is rejected and H_1 is proved that the mean *birthweight* μ > 2800. And the confidence interval is [2829.20,$+\infty$), which is the also the acceptance interval of H_0, that is to say if μ is at this region, H_0 will be accepted. We can also say there is 95% probability that this interval contains the true value.

Sign test: Sign test can also be used to verify this claim without checking it's normally distributed or not. In this sign test, we set the test statistic as the number of samples which is bigger than 2800.

```
n = 188;
sum = sum(birthweight > 2800) #sum the number of birthweights which are bigger than 2800
binom.test(sum, n, p=0.5, alt = "g") #execute binomial test
```

```
##
##  Exact binomial test
##
## data:  sum and n
## 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
```

Then there is the p-value=0.034<0.5, thus we reject H_0 and make the same conclusion as t-test already done.

c) Here we compare the two power values of sign test and t-test by simulation, 1000times of simulation are executed and then calculate the power values with μ=2900.

```
B = 1000; n = 188; alpha = 0.05; mu = 2900; mu_0 = 2800 # set the parameters
sigma = sd(birthweight$birthweight) # calculate sample standard deviation of the birthweights
psign = numeric(B) # will contain  p-values of the sign test
pttest = numeric(B) # will contain p-values of the t-test
```

```
for(i in 1:B){
   x = rnorm(n, mean = mu, sd = sigma) # generate data under H1 with mu = 2900
   pttest[i] = t.test(x, mu = mu_0, alt = "g")[[3]] # extract p-value of the t-test
   psign[i] = binom.test(sum(x > mu_0), n, p=0.5, alt="g")[[3]] # extract p-value of the sign test
}
pwsign = sum(psign < alpha)/B # calculate the power in mu=2900 for the t-test
pwttest = sum(pttest < alpha)/B # calculate the power in mu=2900 for the sign test
```

Here are the results of the power values: pwsign=0.46; pwttest= 0.64. So with μ=2900, the power of t-test(0.64) is higher than the sign test(0.46), that is to say the t-test has better performance than the sign test in the process of testing whether the result is reliable enough to make a conclusion that H_0 is false. Because the power means the probability of rejecting H_0 when H_1 is true, the higher power value is, the more reliable the test is.

d) Let p_hat be the estimated probability that birth weight of a newborn baby is less than 2600 gram. As the sample size is big enough we can use normal approximation to calculate the confidence interval (CI) of p. The margin of error is calculated by subtracting

```
#Load in data for birth weight
data=read.table(file="birthweight.txt",header=TRUE)
birthweights = data$birthweight
#Get base statistics
mean = mean(birthweights)
sd = sd(birthweights)
n = length(birthweights)
p_hat = pnorm(2600, mean=mean, sd=sd)
#Standard error of proportion
ME = p_hat - 0.25
z = ME/sqrt((p_hat*(1-p_hat)/n))
CI = p_hat + c(-1,1)*z*sqrt(p_hat*(1-p_hat)/n)
confidence_level = pnorm(z,lower.tail=FALSE)
percentage_CI = (1- confidence_level *2)
#Manual
alpha = 0.025
z = qnorm(1-alpha/2)
manual_CI = p_hat + c(-1,1)*z*sqrt(p_hat*(1-p_hat)/n)
```

The value of p_hat in the above evaluation is 0.33 and with CI [0.25, 0.4]. P_hat represents the estimated probability that a newborn baby has a birth weight lower than 2600 grams. The confidence interval suggests that we are 0.97 confident that the true probability of a newborn baby having a birth weight less than 2600 gram lies within this range.

Exercise 2

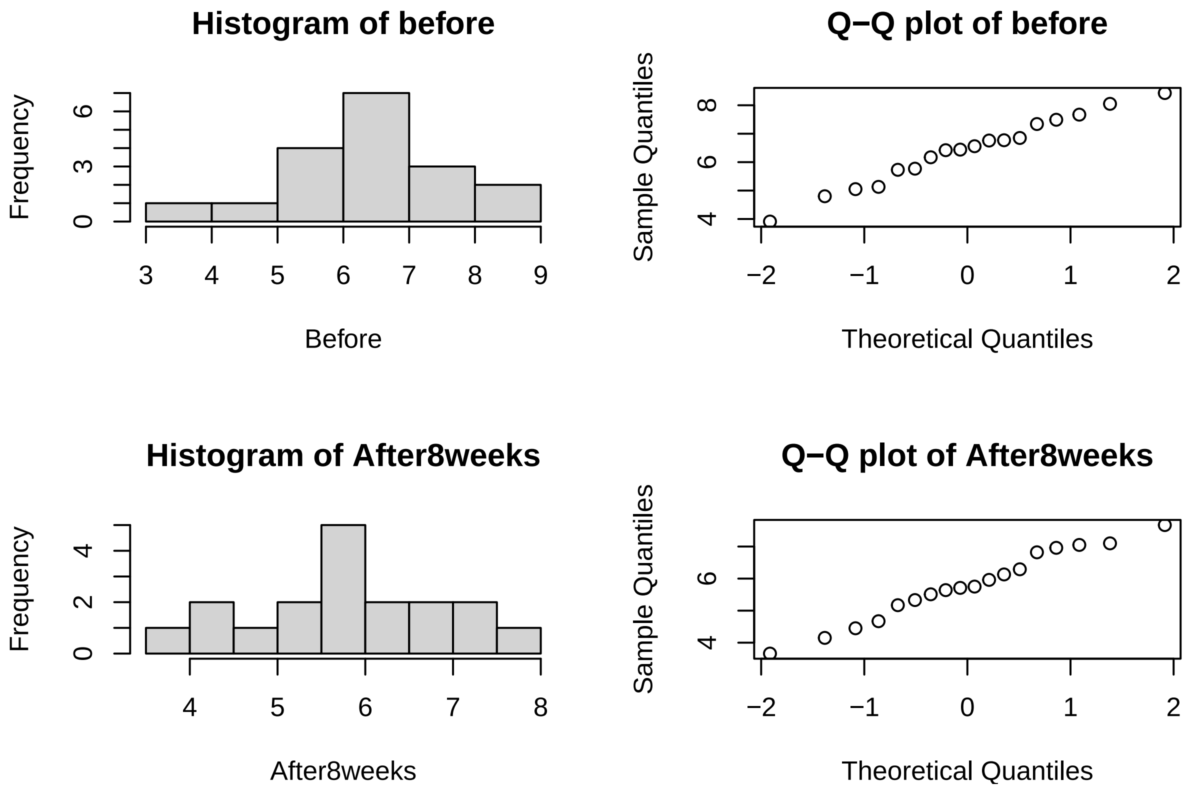
Before conducting the experiment, we shall first import the data set and attach the data to the *R* search path so that all the objects can be easily accessed by giving their names.

```
cholesterol = read.table("cholesterol.txt", header = TRUE)
attach(cholesterol) # Attach the date set to the R search path
```

a) Normality check: Two kinds relevant plots are made to check the data normality, which are the histogram and Q-Q plot.

```
par(mfrow=c(2,2)) # two plots in one row next to each other
hist(Before, main = "Histogram of before")
qqnorm(Before, main = "Q-Q plot of before")
```

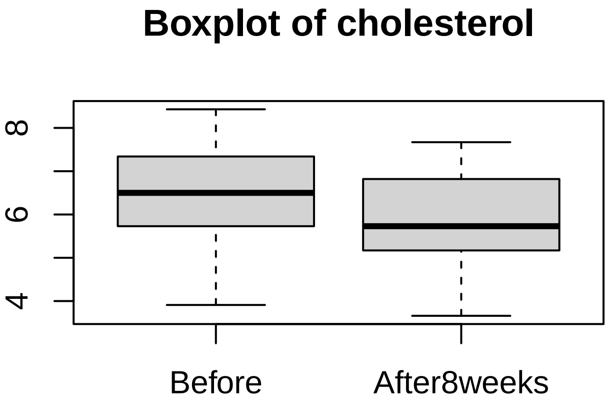
```
hist(After8weeks, main = "Histogram of After8weeks")
qqnorm(After8weeks, main = "Q-Q plot of After8weeks")
```



We can conclude that data of *Before* and *After8weeks* is normally distributed. The two histograms are bell-shaped the majority of the data is concentrated in the middle and the rest is distributed at two tails. Also, these two histograms are approximately symmetrical, although the histogram of *Before* is skewed to the right slightly. As for the Q-Q plots, the points are approximately on a straight line. Thus the data of *Before* and *After8weeks* can be assumed to follow a normal distribution.

Consistency check: Box-plots are usually used to check whether there are any inconsistencies in the data.

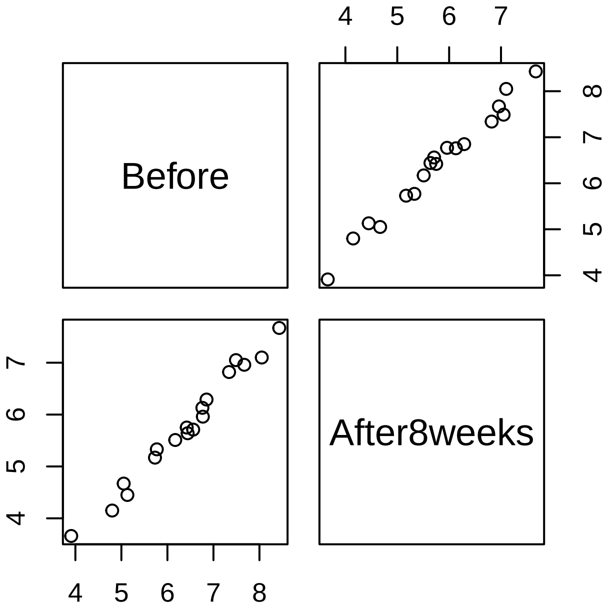
```
boxplot(Before,After8weeks,names = c("Before","After8weeks"), main = "Boxplot of cholesterol")
```



Thus we can draw the conclusion that there is no inconsistency in the data, because there is no outline shown in the box plot.

Correlation check: To check the correlation between the column *Before* and *After8weeks*, we use the scatter plot of these two variables and compute the correlation values between them.

```
pairs(cholesterol) # scatter plot of Before and After8weeks
```

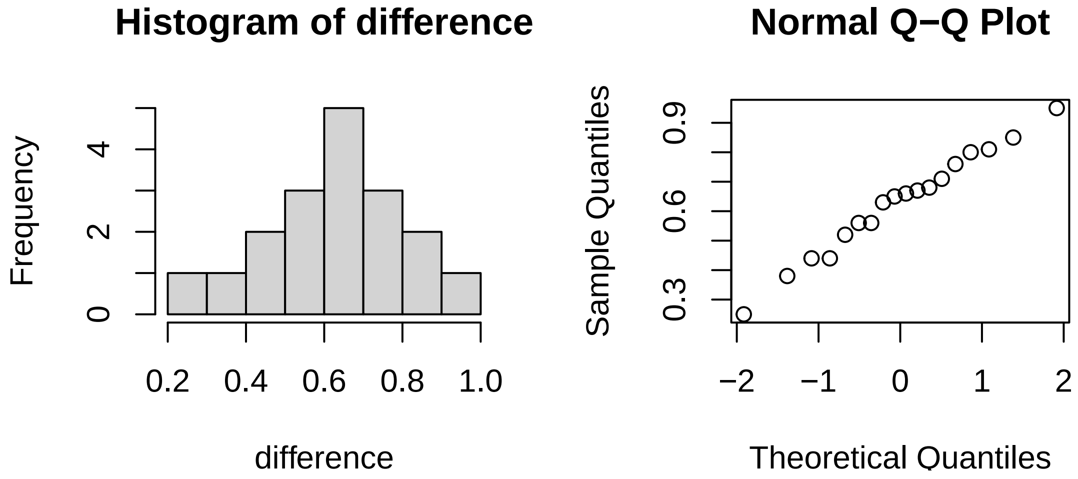


```
rho = cor(Before, After8weeks) # calculate correlation
```

From the scatter plot and the correlation value $\rho = 0.99$, we can conclude that these two columns *Before* and *After8weeks* are correlated, because the points of the scatter plot of these variables are approximately on a straight line and the correlation value ρ is very close to 1 which proves to be linear relation.

b) We use t-test and permutation test to verify whether the diet has an effect.
T-test: Before the t-test, we shall ensure that the data we test is normally distributed. Additionally, the two variables are matching data based on the cholesterol level, thus the data is paired. So we can use the paired t-test, first, check the normality of data, then execute the t-test, and finally conclude by p-value.

```
difference = Before-After8weeks
par(mfrow=c(1,2)) # two plots in one row next to each other
hist(difference)
qqnorm(difference)
```



```
shapiro.test(difference) # check the data normaliy of differences of before and after8weeks
```

```
##
```

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

```
t.test(Before - After8weeks) # execute paired t-test
```

```
##
##  One Sample t-test
##
## data:  Before - After8weeks
## t = 14.946, df = 17, p-value = 3.279e-11
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.5401131 0.7176646
## sample estimates:
## mean of x
## 0.6288889
```

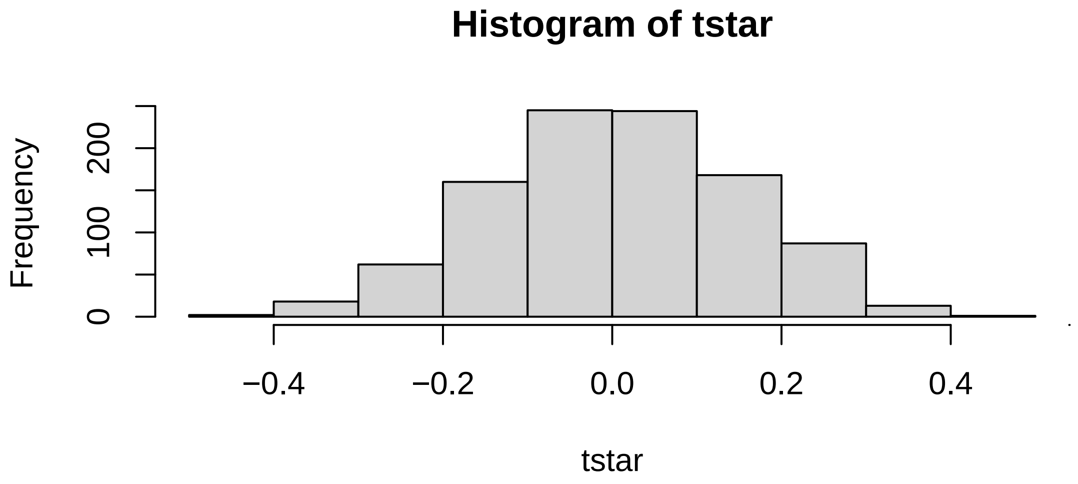
According to plots of *difference* and `p-value` of the Shapiro-Wilk test is `0.987 > 0.05`, the *difference* follows the normal distribution. So we can conclude that the diet does have an effect because the `p-value` of this paired sample test is `3.28e-11 < 0.05`, which means H_0(Difference is equal to 0) is rejected.

Permutation test: Because what we care about is whether the difference of *Before* and *After8weeks* is 0, permutation is applicable here. In this permutation test, we use the `mean(x-y)`as the test statistic.

```
mystat = function(x,y){ mean(x-y)} # set test statistic
B = 1000 # the number of T-star
tstar = numeric(B)
for(i in 1: B){
  cholesterolstar = t(apply(cbind(Before, After8weeks), 1, sample)) # generate x-star and y-star
  tstar[i] = mystat(cholesterolstar[,1], cholesterolstar[,2])} # generate T-star}
myt = mystat(Before, After8weeks) # compute original test statisc t
print(myt)
```

```
## [1] 0.6288889
```

```
# plot histogram of T-star with original test statistic t
par(mfrow = c(1,1))
hist(tstar)
lines(rep(myt,2),c(0,30), col = "red", lwd = 2)
axis(1,myt,expression(paste("t")))
```



```
pl = sum(tstar<myt)/B
pr = sum(tstar>myt)/B
p = 2*min(pl,pr) # calculate p-value
```

The p-value p =0 < 0.05, so we reject H_0, and conclude that there are differences between these two variables, that is to say, the diet has an effect on the cholesterol level.

c) Because of the central limit theorem(CLT), the mean of $\bar{X}_i$ follows the normal distribution, which means we can estimate the mean of the population $\hat{\mu}$ equals to the mean of $\bar{X}_i$, hence we can get the $\hat{\theta}$ by using the property of uniform distribution.

```
B  = 1000; a = 3; alpha = 0.05; n = 18 # set parameters
theta = max(After8weeks)
# calculate estimate theta
x_bar = numeric(B)
for(i in 1:B){
  x = runif(n, min = a, max = theta)  # generate a bunch of samples
  x_bar[i] = mean(x)
}
mu = mean(x_bar)
estitheta = 2*mu - a # use (a+b)/2 of expectaion formula in uniform distribution
# calculate 95% confidence interval
s = (estitheta - a)^2/12 # use the variance formula in uniform distribution
t_quantile = qt(1-alpha/2, df = n-1)
E = t_quantile*s/sqrt(n) # calculate margin error
cll = estitheta - E # calculate the left side CI
clr = estitheta + E # calculate the right side CI
```

Here we have $\hat{\theta}$ =7.69, 95% confidence interval(CI) of [6.78,8.61]. In order to improve the CI, here we increase the sample number to get more information from the data, thus making the CI more accurate. So we reset the sample number as50, then there is the improved CI of[7.17,8.20].

d) First we shall apply the bootstrap test to verify the H_0, then apply the Kolmogorov-Smirnov test to check whether X follows the uniform distribution.
Bootstrap test:

```
B = 1000; n =18; # set parameters
T = max(After8weeks) # calculate original test statistic T
Tstar = numeric(B)
for(i in 1:B){
  Theta = runif(1, min =3, max = 12) # generate a Theta randomly
  Xstar = runif(n, min = a, max = Theta) # generate the X-star
  Tstar[i] = max(Xstar) # calculate T-star
}
pl = sum(Tstar<T)/B
pr = sum(Tstar>T)/B
p = 2*min(pl,pr) # calculate p value
```

Here we get the p-value p =0.91> 0.05, so we accept H_0.
Kolmogorov-Smirnov test: In this case, two independent samples X_i is the generated data under H_0, and Y_i is the data follows the uniform distribution.

```
# generate data set under H0
Theta = runif(1, min =3, max = 12)
Sampleunderh0 = runif(n, min = a, max = Theta)
# ks.test
ks.test(Sampleunderh0,"punif", min = a, max = Theta)
```

```
##
##  Exact one-sample Kolmogorov-Smirnov test
##
## data:  Sampleunderh0
## D = 0.12294, p-value = 0.9182
## alternative hypothesis: two-sided
```

We get p-value $=$ `0.43` $>$ `0.05`, so we accept H_0. One thing needs to mention here is that the p-value is not fixed because the θ changes randomly, but all of them is bigger than `0.05`. Both the bootstrap test and the Kolmogorov-Smirnov test can be applied in this situation of checking whether X_i follows the uniform distribution with specif interval.

e) We first use the sign test to verify whether the median cholesterol level after 8 weeks of low fat diet is less than `6`.

```r
sumoflessthan6 = sum(After8weeks < 6)
binom.test(sumoflessthan6, 18, p=0.5, alt = "l")
```

```
##
##  Exact binomial test
##
## data:  sumoflessthan6 and 18
## 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
```

We get p-value $=$ `0.88 > 0.05`, so we accept the H_0: the median value of the data *After8weeks* is more than 6. Then to check whether the fraction of cholesterol levels after 8 weeks of low fat less than `4.5` is at most 25`%`. First we shall set H_0: the fraction `p > 25%`, then we use sign test to verify the 25% quantile.

```r
sumoflessthan4_5 = sum(After8weeks < 4.5)
binom.test(sumoflessthan4_5, n, p = 0.25, alt = "l")
```

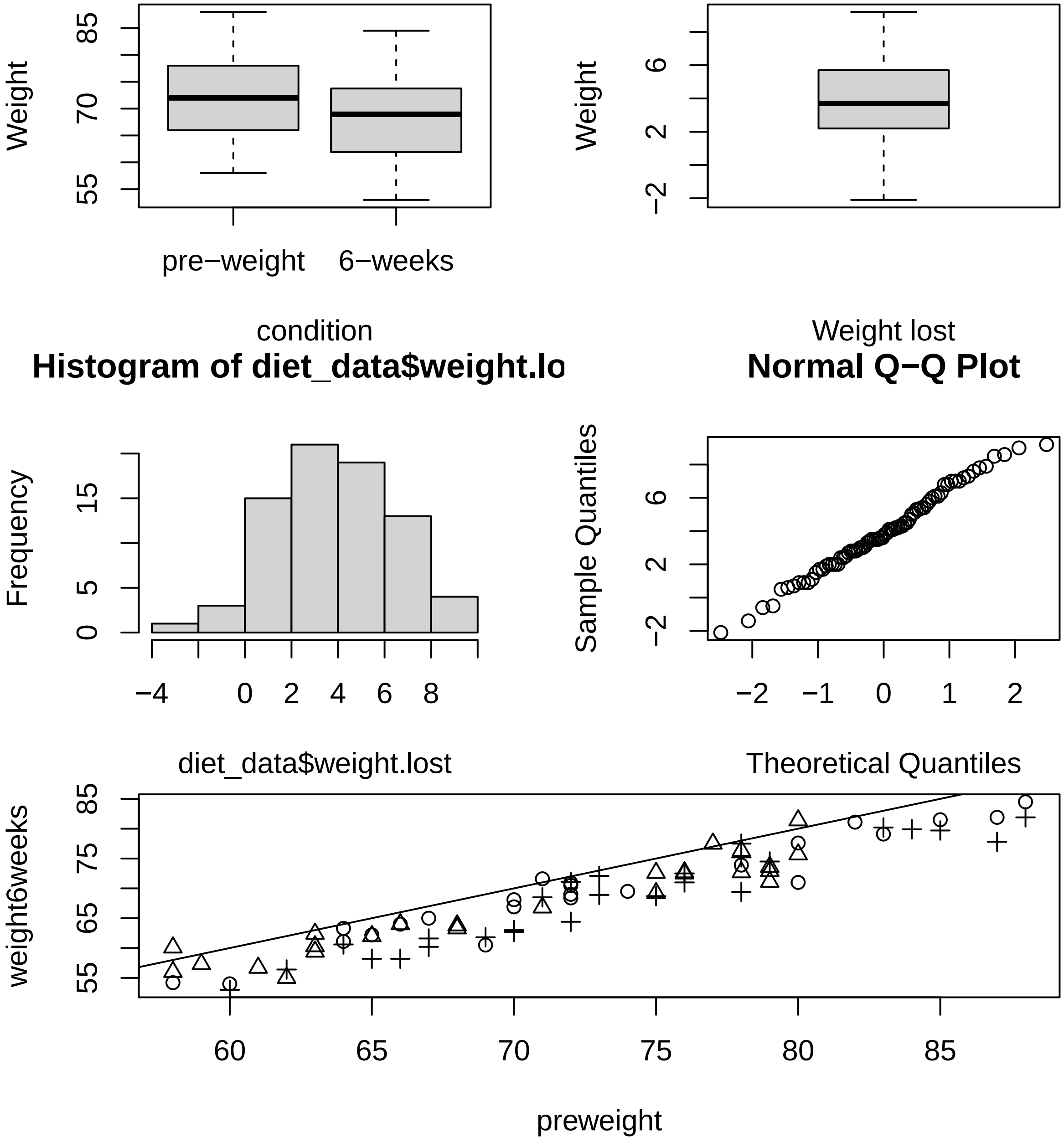
```
##
##  Exact binomial test
##
## data:  sumoflessthan4_5 and n
## number of successes = 3, number of trials = 18, p-value = 0.3057
## alternative hypothesis: true probability of success is less than 0.25
## 95 percent confidence interval:
##  0.0000000 0.3766792
## sample estimates:
## probability of success
##              0.1666667
```

We get `p-value = 0.31 > 0.05`, so we accept H_0, that is to say the fraction pf cholesterol level after 8 weeks of lo far diet less than `4.5` is more than 25%.

Exercise 3

a) Graphical summary & test preweight~weight6weeks

```r
# read in data from the file
diet_data <- read.table("diet.txt", header = TRUE); diet_data <- na.omit(diet_data)
#Create new column 'weight.lost by subtracting: preweight - weight6weeks
diet_data$weight.lost = diet_data$preweight - diet_data$weight6weeks
```



```r
#Last normality assumption - shapiro test
shap_result = shapiro.test(diet_data$weight.lost)
```

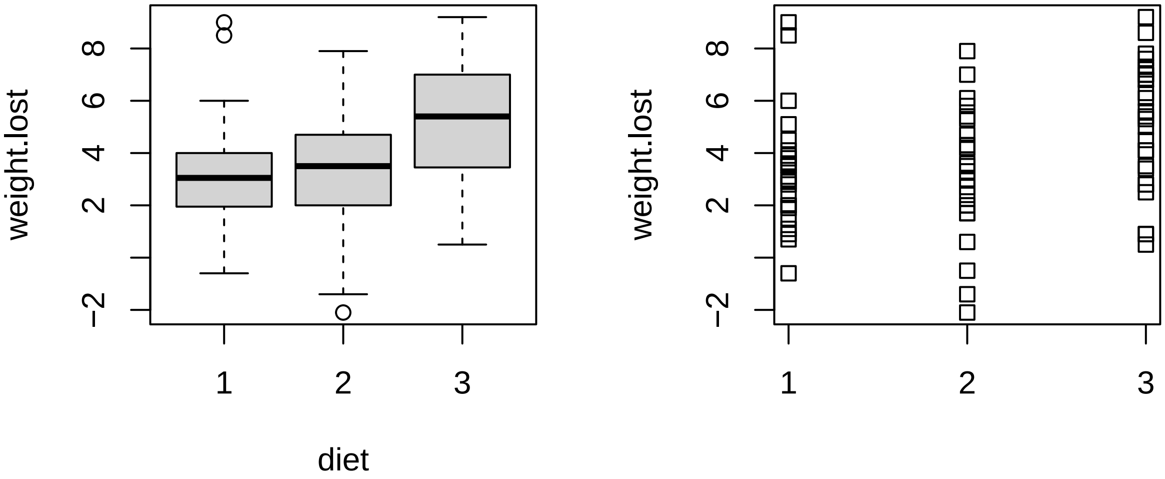
Visual inspection of the box plots suggest that the median of the pre-weight condition is higher than the condition of the participants after 6 weeks of diet. The scatter plot shows the relationship between pre-weight and after6-weeks. Visual inspection conveys that participants who were heavier in the pre-weight condition are also generally heavier in the after6-weeks condition. However, participants do seem to have a lower weight after the 6 weeks of diet compared to their pre-weight. Regarding assumptions, the bar plot suggests the data is normally distributed as it approaches a normally distributed bell curve shape. The Q-Q plot, shows a fairly straight line. This indicates that that the observed data are close to the theoretical quantiles of the normal distribution, and that the data does follow a normal distribution. Aditionally, the Shapiro-Wilk test shows a p-value of 0.79. As p is above 0.05 combined with the visual inspection, we conclude that the data is normally distributed.

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

A paired t-test shows a significant difference between the pre-weight condition and after-6-weeks condition, t(75) = 13.7284867, p < .01, suggesting that weight within participants before diet was significantly lower compared to before a diet. This confirms the experts claim.

b) One way anova: type of diet & weight loss

```
#Graphical inspection
par(mfrow=c(1,2));boxplot(weight.lost~diet,data=diet_data);stripchart(weight.lost~diet,data=diet_data,v
```

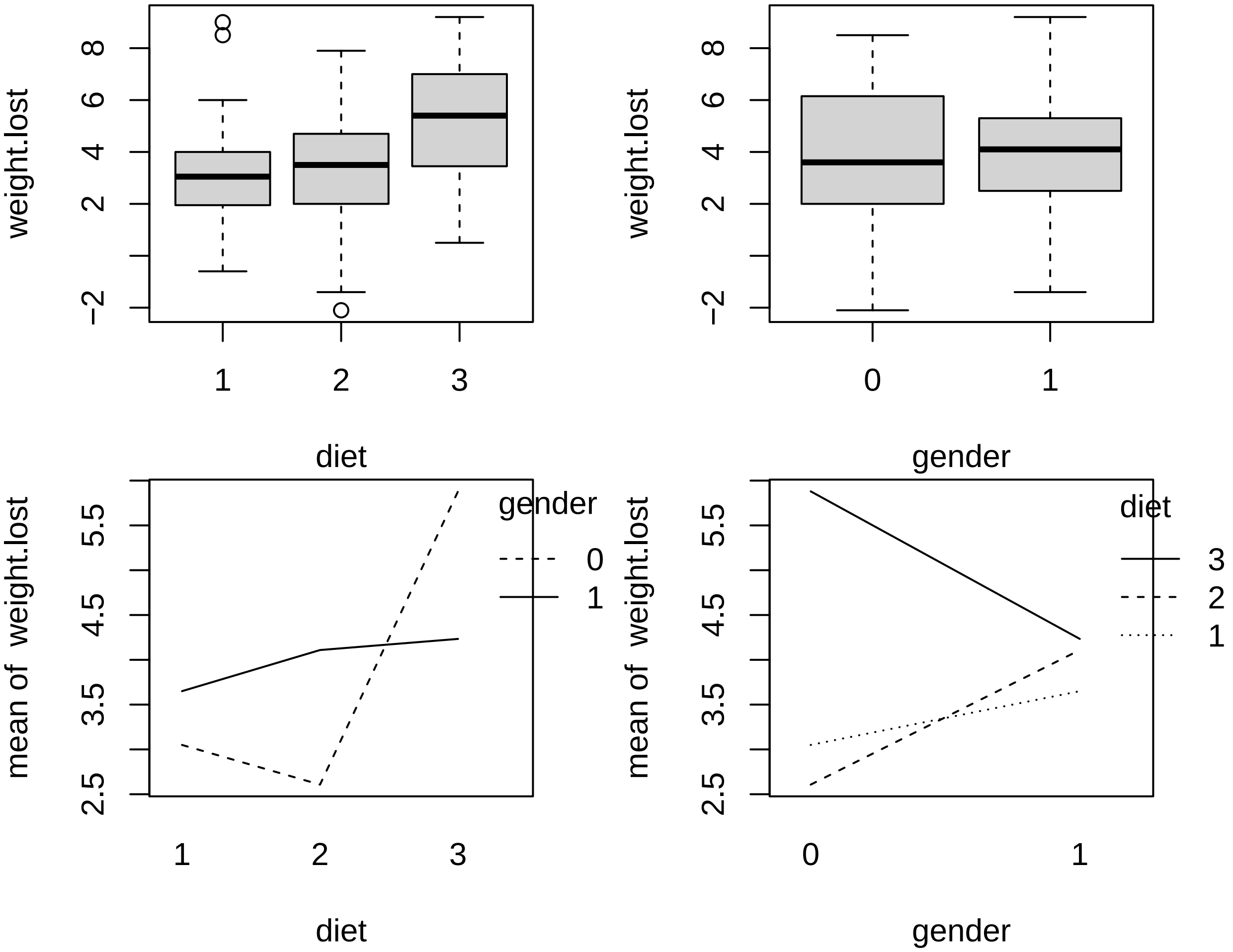


Above box plots suggest a slightly higher median for diet 2 compared to diet 1 and a more substantial difference in median, in favor of diet 3 compared to diet 1 and 2. The dot plot shows that the variance across the three groups is roughly equal as seen by the spread of the data-points. Additionally, it appears that in each group the data is normally distributed if one imagines a density curve across the groups (diets).

```
#One-way anova
# Diet column coercion to factor type
diet_data$diet = factor(diet_data$diet)
#Anova 1-way
diet_lm = lm(weight.lost~diet,data=diet_data)
diet_aov = aov(diet_lm)
anova_table = anova(diet_aov)
## Coefficients:
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.3000 0.4840 6.818 2.26e-09 ***
## diet2 -0.0320 0.6776 -0.047 0.96246
## diet3 1.8481 0.6652 2.778 0.00694 **
## Residual standard error: 2.371 on 73 degrees of freedom
## Multiple R-squared: 0.1285, Adjusted R-squared: 0.1047
## F-statistic: 5.383 on 2 and 73 DF, p-value: 0.006596
tukey.test = TukeyHSD(diet_aov)
## diff lwr upr p adj
## 2-1 -0.032000 -1.6530850 1.589085 0.99877114
## 3-1 1.848148 0.2567422 3.439554 0.01880469
## 3-2 1.880148 0.3056826 3.454614 0.01520200
```

An analysis of variance (ANOVA) was conducted to test for differences in mean weight loss among three different diet groups (diet 1, diet 2, and diet 3). The ANOVA model was significant, F(2, 73) = 5.383, p = 0.02, indicating that there was a statistically significant difference in weight loss among the diet groups. Post-hoc tests revealed that participants in diet group 3 lost significantly more weight than those in diet group 1 p = 0.019, and diet group 2 p = 0.015. And that there was not a significant difference between diet group 1 and 2, p = 0.998. These results suggest that diet 3 may be more effective for weight loss than diets 1 and 2. The Kruskal-Wallis test cannot be applied in this situation. Many of the conditions are met, however this test relies on the assumption that the samples of each group should be independent of each other. In current situation we use a within subject design, that is, multiple samples are taken from the same subject under different conditions. This violates the assumption of independence.

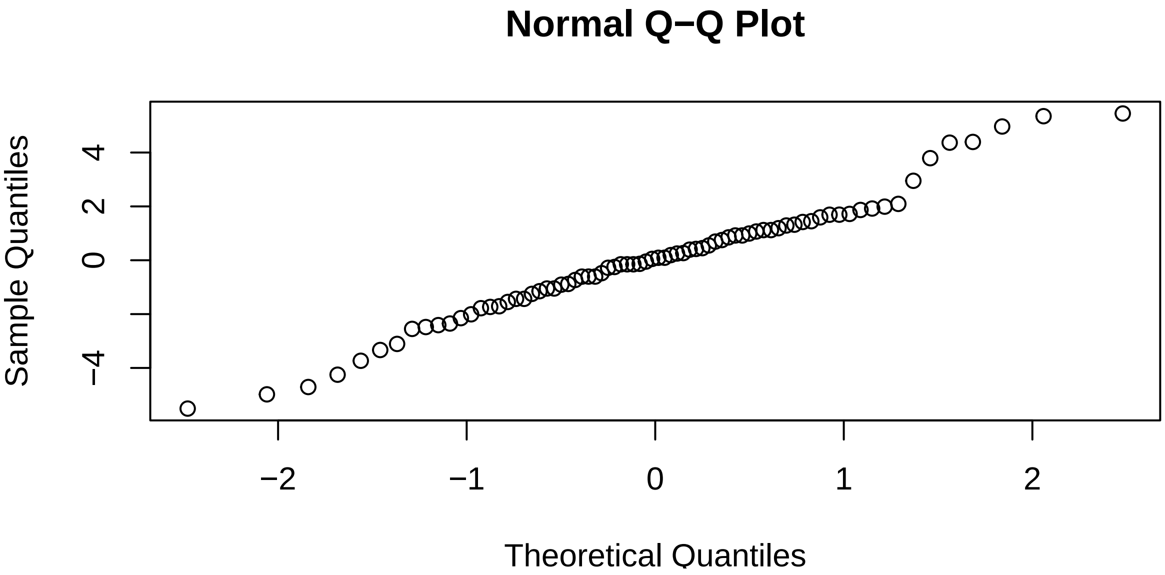
c) Two-way Anova: effect diet and gender on weight lost



Inspection of the box plot for gender suggest that the means for females and males do not differ by a big amount in lost weight. The box plot interpretation for diet is mentioned earlier. When looking at the interaction graphs, the lines are not parallel to one another. This suggests an interaction effect.

```
diet_data <- read.table("diet.txt", header = TRUE)
diet_data <- na.omit(diet_data)
diet_data$weight.lost = diet_data$preweight - diet_data$weight6weeks
diet_data$diet=as.factor(diet_data$diet); diet_data$gender=as.factor(diet_data$gender)
#Coerce predictors as factors
diet_data$diet=as.factor(diet_data$diet); diet_data$gender=as.factor(diet_data$gender)
#Create model
diet_aov=lm(weight.lost~diet*gender, data = diet_data)
```

```
#Check assumptions
par(mfrow = c(1, 1))
qqnorm(diet_aov$residuals)
```



```
#Run anova with interaction term
anova(diet_aov)
```

```
## Analysis of Variance Table
##
## Response: weight.lost
##             Df  Sum Sq Mean Sq F value   Pr(>F)
## diet         2  60.53 30.2635  5.6292 0.005408 **
## gender       1   0.17  0.1687  0.0314 0.859910
## diet:gender  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
```

When looking at the normal the Q-Q, normality seems doubtful. The results of a two-way ANOVA analysis examining the effects of gender and diet on weight lost revealed a significant interaction effect between diet and gender, F(2,70) = 3.15, p = 0.048. Therefore the main effects for factors gender and diet will not be interpreted. The significant interaction effect between diet and gender indicates that the effect of diet on weight lost is dependent on gender. Therefore, the findings suggest that the effect of a particular diet on weight loss may vary depending on whether the individual is male or female. For males diet 3 has the biggest impact on weight loss, next diet 2 and diet 1 seems to perform worst out the the three diets. However, for females diet 3 seems to work best, next diet 1 and diet 2 seems to work worst.

e) Preferred model and prediction

The preferred model is the one from C. That is, a two-way Anova with gender and diet as predictors and lost weight as dependent variable. This model is preferred because hidden effects of gender are revealed, changing the conclusion and providing additional, valuable information. Results from one-way Anova suggest that diet 3 is the most effective and diet 1 and 2 do not differ significantly. This conclusion changed and is more nuanced as described above.

```
library(emmeans)
# Obtain the estimated marginal means (EMMs) for lost.weight at each level of diet
# holding gender constant
```

```
my_emms <- as.data.frame(emmeans(diet_aov, ~ diet | gender, type = "response"))
```

Gender	diet	Predicted lost weight
0	1	3.05
0	2	2.61
0	3	5.88
1	1	3.65
1	2	4.11
1	3	4.23

The table shows the predicted weight loss for females (gender 0) and males (gender 1). The interpretation remains the same as described in section 3.c.

Exercise 4

a) We can solve this question in this way. To randomize the distribution of soil additives over plots in such a way that each soil additive is received exactly by two plots within each block, the following steps in R can be used:

```
library(MASS)
set.seed(123)  # Set seed for reproducibility

# Create a matrix to store the randomized distribution of soil additives
dist_mat <- matrix(0, nrow = 6, ncol = 4)

# Randomly assign soil additives to each block
for (i in 1:6) {
  dist_mat[i, ] <- sample(c(1,1,0,0), 4)
}

# Verify that each soil additive is received by exactly two plots within each block
colSums(dist_mat)  # The sum of each column should be 2
## [1] 2 4 2 4
rowSums(dist_mat)  # The sum of each row should be 2
## [1] 2 2 2 2 2 2

# Combine the block information and the soil additive distribution
npk_dist <- data.frame(npk, dist_mat)
```

Within this code above, firstly we set the seed for the reproducibility after load the MASS package. Then in order to store the distribution of soil additives which is randomized, a matrix is supposed to be created. After that, we need to make verification to study if every soil additive can be received by the two plots. In the end, the data of the additive distribution information is integrated and combined.

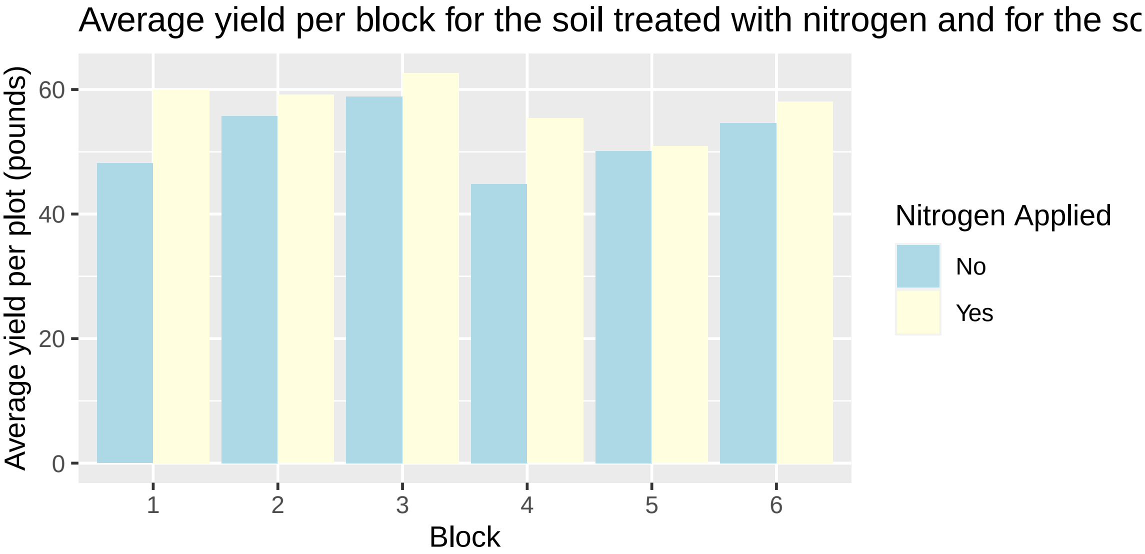
To be more specific, we use the sample function to randomly assign soil additives to each block. We specify the probability of each soil additive with 1 = applied, 0 = not applied, using a vector of length 4 with two 1's and two 0's, which we then randomly sample from using the sample function. We repeat this process for each block, and store the resulting distribution in a matrix called `dist_mat`. We then verify that each soil additive is received by exactly two plots within each block by checking the column and row sums of `dist_mat`. Finally, we combine the block information and the soil additive distribution into a data frame called `npk_dist`.

b) The method of solving this question is to make a plot showing the average yield per block for the soil treated with nitrogen and for the soil that did not receive nitrogen, we can perform the following steps to realize:

```
library(ggplot2)
rm(mean)
npk_dist_agg <- aggregate(yield ~ block + N, data = npk_dist, mean)
```

To begin with, we can calculate the average yield per block for the soil treated with nitrogen and for the soil that did not receive nitrogen by using `npk_dist_agg` with `aggregate()`.

```
ggplot(npk_dist_agg, aes(x = block, y = yield, fill = factor(N))) +
  geom_col(position = "dodge") +
  geom_bar(stat = "identity", position = "dodge") +
  scale_fill_discrete(name = "N") +
  labs(x = "Block", y = "Average yield per plot (pounds)", fill = "Nitrogen Applied") +
  scale_fill_manual(values = c("lightblue", "lightyellow"), labels = c("No", "Yes")) +
  ggtitle("Average yield per block for the soil treated with nitrogen and for the soil that did not rece
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```



After calculating the average data, we start to make a plot to realize the same target. This plot is to show the average yield per block for the soil treated with nitrogen and for the soil that did not receive nitrogen.

Basically, we use the aggregate function to calculate the average yield per block for the soil treated with nitrogen which is `N = 1` and for the soil that did not receive nitrogen which is `N = 0`. We then use the `ggplot2` package to make a grouped bar chart showing the average yield per block for the two groups, with the `x-axis` representing the block and the `y-axis` representing the average yield per plot in pounds. The fill argument is used to group the bars by whether or not nitrogen was applied, and the position argument is used to place the side of the bars by side. We also add appropriate axis labels, a title of "Average yield per block for the soil treated with nitrogen and for the soil that did not receive nitrogen", and a legend. What's more, we define adequate colours for the plot.

The purpose of taking the factor block into account is to control for any variability in yield that may be due to the location of the plots within each block. By comparing the average yield per block between the two groups `N = 1` and `N = 0`, we can see whether there is a difference in yield that is due to the application of nitrogen or to other factors, such as differences in soil conditions or weather.

c) For this question, we conduct a full two-way ANOVA model with the response variable yield and the two factors block and N as follows. To realize this, we use the `aov` function in R. To begin with, we load the required `npk` dataset from the MASS package and then make the structure of the data.

```r
library(MASS)
data(npk)
str(npk)
## 'data.frame':    24 obs. of  5 variables:
##  $ block: Factor w/ 6 levels "1","2","3","4",..: 1 1 1 1 2 2 2 2 3 3 ...
##  $ N    : Factor w/ 2 levels "0","1": 1 2 1 2 2 2 1 1 1 2 ...
##  $ P    : Factor w/ 2 levels "0","1": 2 2 1 1 1 2 1 2 2 2 ...
##  $ K    : Factor w/ 2 levels "0","1": 2 1 1 2 1 2 2 1 1 2 ...
##  $ yield: num  49.5 62.8 46.8 57 59.8 58.5 55.5 56 62.8 55.8 ...
```

From the output, we know that the dataset has 24 observations on 4 variables: `block`, `N`, `P`, and `K`. The response variable is yield, which is measured in pounds per plot. The block variable is a factor variable with 6 levels, indicating the block in which the plot was located. The `N`, `P`, and `K` variables are binary variables indicating whether nitrogen, phosphate, or potassium was applied to the plot.

```r
npk_anova_model <- aov(yield ~ block + N, data = npk)
summary(npk_anova_model)
##             Df Sum Sq Mean Sq F value Pr(>F)
## block        5  343.3   68.66   3.395 0.0262 *
## N            1  189.3  189.28   9.360 0.0071 **
## Residuals   17  343.8   20.22
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The output shows the results of the ANOVA. We can see that both the`block` and `N` variables have significant effects on yield $p < 0.05$. This indicates that both the location of the plot and the application of nitrogen have an impact on the yield of peas. We use the `aov` function to fit a full two-way ANOVA model with the response variable yield and the two factors `block` and `N`. We then use the summary function to view the results of the ANOVA, including the `F-statistic`, `p-value`, and degree of freedom for each factor and their interaction. The Friedman test is not suitable for this situation because the Friedman test is a non-parametric test that is used to compare three or more related groups, and assumes that the data are measured on an ordinal scale. In this case, we have two factors `block` and `N` that are not related, and the response variable `yield` is measured on a continuous scale. Therefore, the assumptions of the Friedman test are not met, and it is not appropriate to use this test for this analysis.

It is sensible and adequate to include the factor block in this model because the block factor represents a grouping variable that is not of primary interest, but that may have an effect on the response variable. By including the block factor in the model, we can control for any variability in yield that is due to the location of the plots within each block, and thus increase the power of the analysis to detect the effect of the `N` factor on yield.

d) This question requires me to give my favorite model and it depends on my choice. To investigate other possible models with all the factors combined, we can use the function to fit a linear regression model. Here are two possible models with one pairwise interaction term:`lm`.

The Model 1 includes the interaction between N and block and the Model 2 includes the interaction between P and block.

```r
npk_model1 <- lm(yield ~ N*block + P + K, data = npk_dist)
summary(npk_model1)
##
## Call:
## lm(formula = yield ~ N * block + P + K, data = npk_dist)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.267 -1.608  0.000  1.608  4.267
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   50.733      2.875  17.648 7.26e-09 ***
## N1            11.750      3.764   3.122   0.0108 *
## block2         7.600      3.764   2.019   0.0711 .
## block3        10.750      3.764   2.856   0.0171 *
## block4        -3.300      3.764  -0.877   0.4012
## block5         2.000      3.764   0.531   0.6068
## block6         6.450      3.764   1.714   0.1174
## P1            -1.183      1.537  -0.770   0.4590
## K1            -3.983      1.537  -2.592   0.0268 *
## N1:block2     -8.350      5.323  -1.569   0.1478
## N1:block3     -8.000      5.323  -1.503   0.1638
## N1:block4     -1.200      5.323  -0.225   0.8262
## N1:block5    -11.000      5.323  -2.067   0.0657 .
## N1:block6     -8.250      5.323  -1.550   0.1522
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.764 on 10 degrees of freedom
## Multiple R-squared:  0.8383, Adjusted R-squared:  0.6282
## F-statistic: 3.989 on 13 and 10 DF,  p-value: 0.01731
npk_model2 <- lm(yield ~ P*block + N + K, data = npk_dist)
summary(npk_model2)
##
## Call:
## lm(formula = yield ~ P * block + N + K, data = npk_dist)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.317 -2.038  0.000  2.038  4.317
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   51.083      3.138  16.280 1.59e-08 ***
## P1             4.250      4.108   1.034  0.32528
## block2         5.750      4.108   1.400  0.19189
## block3        10.350      4.108   2.519  0.03043 *
## block4         1.850      4.108   0.450  0.66210
## block5        -1.250      4.108  -0.304  0.76717
## block6         4.700      4.108   1.144  0.27927
## N1             5.617      1.677   3.349  0.00738 **
## K1            -3.983      1.677  -2.375  0.03895 *
## P1:block2     -4.650      5.810  -0.800  0.44212
## P1:block3     -7.200      5.810  -1.239  0.24356
## P1:block4    -11.500      5.810  -1.979  0.07596 .
## P1:block5     -4.500      5.810  -0.775  0.45655
## P1:block6     -4.750      5.810  -0.818  0.43267
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 4.108 on 10 degrees of freedom
## Multiple R-squared:  0.8074, Adjusted R-squared:  0.557
## F-statistic: 3.225 on 13 and 10 DF,  p-value: 0.03535
```

We include the interaction between N and block in the first model, while in the second model, we include the interaction between P and block. We also include the main effects of all three soil additives N, P, K and the block factor. To test for the presence of main effects of N, P, and K, we can use an ANOVA table to compare the two models with and without the corresponding main effect.

```
npk_model_noN <- lm(yield ~ block + P + K, data = npk_dist)
anova(npk_model_noN, npk_model1)
## Analysis of Variance Table
##
## Model 1: yield ~ block + P + K
## Model 2: yield ~ N * block + P + K
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     16 429.47
## 2     10 141.67  6     287.8 3.3859 0.0433 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

To begin with, we make the model without the main effect of N and after that we compare models with and without the main effect of N. Similarly, we can test for the main effects of P and K by comparing the models with and without the corresponding main effect.

Model 1 would be my favorite choice for this given situation, which includes the interaction between N and block. This model assumes that the effect of N on yield may depend on the block where the plot is located. The main effect of N is also included, allowing us to test whether N has a significant effect on yield, regardless of the block. The main effects of P and K are also included, controlling for any potential effects of these soil additives on yield. I prefer this model because it allows for more flexibility in modeling the relationship between the response variable and the soil additive N, which is the factor of primary interest.

e) This question mains to perform a mixed effects analysis, we will use the lmer function in the R package lme4. This function allows us to model the block variable as a random effect. Firstly we load the lme4 package. and get a mixed affects model with block as a random affect.

```
library(lme4)
## Loading required package: Matrix
npk_mixed <- lmer(yield ~ N + P + K + (1|block), data = npk_dist)
summary(npk_mixed)
## Linear mixed model fit by REML ['lmerMod']
## Formula: yield ~ N + P + K + (1 | block)
##    Data: npk_dist
##
## REML criterion at convergence: 128.1
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.79887 -0.62358  0.05054  0.65032  1.34337
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  block    (Intercept) 13.16    3.628
##  Residual             16.01    4.002
## Number of obs: 24, groups:  block, 6
##
```

```
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   54.650      2.205  24.784
## N1             5.617      1.634   3.438
## P1            -1.183      1.634  -0.724
## K1            -3.983      1.634  -2.438
##
## Correlation of Fixed Effects:
##    (Intr) N1     P1
## N1 -0.370
## P1 -0.370  0.000
## K1 -0.370  0.000  0.000
```

It is obvious the we treat block as a random effect in this code above, allowing for differences in the intercept among the blocks. We include the main effects of all three soil additives: `N`, `P`, and `K` as fixed effects.

Comparing the results from the mixed effects model to the results from the full two-way ANOVA model in the third question, we can see that the estimates of the fixed effects including `N`, `P`, and `K` are similar in both models, but the standard errors and `p-values` are slightly different. The mixed effects model assumes that the variance of the yield response may differ among the blocks, which leads to a larger estimate of the residual variance and smaller p-values for the fixed effects.

Overall, both models suggest that the soil additive has a significant effect on yield. However, the mixed effects model provides a more appropriate way to model the data, by taking into account the variation in yield among the blocks.