Assignment 1: Group 45

Daniel Engbert, Rik Timmer, Koen van der Pool

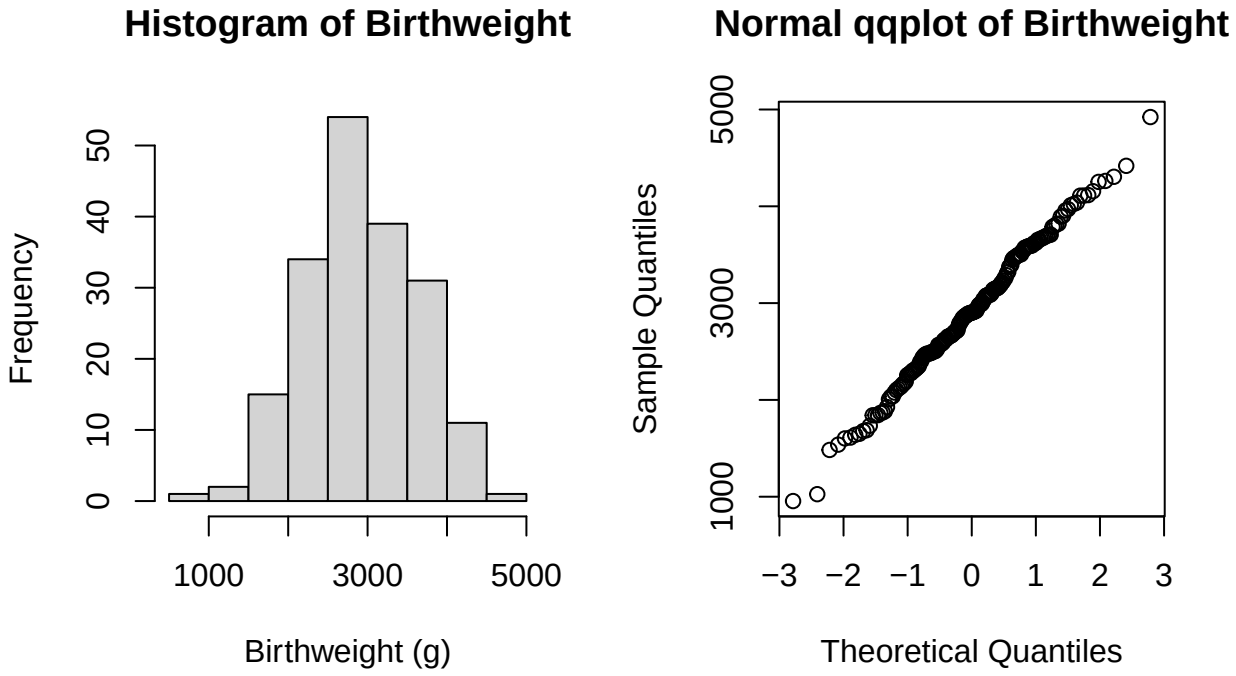
11 February 2023

Note: we made a function `checkNorm()` which prints a histogram, qqplot, and p-value from the shapiro-wilk normality test. And we made a function `printPval()` which simply prints a given p-value to 3 significant figures. We utilize both functions throughout this assignment.

Exercise 1: Birthweights

1 a)

```
data = read.table("birthweight.txt", header=T)
par(mfrow=c(1,2))
# use x as alias for the dataset
x = data$birthweight

# checking normality
pval = checkNorm(x, "Birthweight", unit="g")
```




```
## [1] "Shapiro-Wilk normality p-value for Birthweight: 0.900"
```

```
res = t.test(x, conf.level=0.96)
print(sprintf("96%% confidence interval: [%.3f, %.3f]", res$conf.int[1], res$conf.int[2]))
```

```
## [1] "96% confidence interval: [2808.084, 3018.501]"
```

```
# calculate min sample size needed
tval = qt(0.98, length(x)-1)
m = 50
n = (tval * sd(x) / m) ** 2
print(sprintf("min sample size, n = %.3f", n))
```

```
## [1] "min sample size, n = 832.382"
```

The birthweight data appears to be normal based on a normal-appearing histogram, the straight line in the qqplot, and the shapiro-wilk normality test (having $p = 0.900 > 0.05$).

$$m = \frac{t * s}{\sqrt{n}} \rightarrow n = \left(\frac{t * s}{m}\right)^2$$

The equation above, gives the sample size, n, needed for the confidence interval to have a width of 100 (meaning $m = 50$), where t is the t-score for the quantile 0.02 (such that both tails of the distribution have total area $1 - 0.96 = 0.04$). Our R code above computes n=832.382, which rounded up indicates the min sample size is $n = 833$ (for a 96% CI of length at most 100).

```
B = 20000
Tstar = numeric(B)
for (i in 1:B){
  # sample with replacement, for a new sample of same length(x)
  Xstar = sample(x, replace=TRUE)
  Tstar[i] = mean(Xstar)
}

Tstar02 = quantile(Tstar, 0.02)
Tstar04 = quantile(Tstar, 0.98)
print(sprintf("Bootstrap 96%% CI = [%.3f, %.3f]", Tstar02, Tstar04))
```

```
## [1] "Bootstrap 96% CI = [2808.662, 3017.016]"
```

The output of our bootstrap 96% CI indicates that we can say with 96% confidence that the true mean weight of newborn babies is between [2808.66, 3017.02] grams. This is approximately consistent with the CI calculated previously (as expected).

1 b)

```
res = t.test(x, mu=2800, alt="g")
sprintf("95%% confidence interval: [%.3f, %.3f]", res$conf.int[[1]], res$conf.int[[2]])
```

```
## [1] "95% confidence interval: [2829.202, Inf]"
```

```
printPval(res$p.value)
```

```
## [1] "p-value = 0.014"
```

The output of the confidence interval here denotes that we can say with 95% confidence that the true mean is in the interval [2829.202, Inf], and with a p-value of $0.014 < 0.05$, we reject the null hypothesis, as there's sufficient evidence to support the claim that the mean birthweight $\mu > 2800$ grams.

As we previously determined, the data is approximately normally distributed, and symmetric. Therefore the difference between the mean and median is minimal, and we can use a Wilcoxon signed rank test for one sample.

```
res = wilcox.test(x, mu=2800, alt="g")
printPval(res$p.value)
```

```
## [1] "p-value = 0.015"
```

The resulting p-value of $0.015 < 0.05$ also suggests there's sufficient evidence to support the claim that the mean birthweight $\mu > 2800$ grams.

1 c)

```
B = 10000; n = length(x)
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) {
  y = rnorm(n, mean=mean(x),sd=sd(x)) ## generate data under H1 with mu=0.5
  mu = 2800
  pttest[i] = t.test(y, mu=mu, alternative="greater")[[3]] ## extract p-value
  psign[i] = wilcox.test(y, mu=mu, alt="g")[[3]]
} ## extract p-value
signPower = sum(psign<0.05) / B; tPower = sum(pttest<0.05) / B

print(sprintf("wilcox signed rank test power = %.3f, t-test power = %.3f", signPower, tPower))
```

```
## [1] "wilcox signed rank test power = 0.699, t-test power = 0.716"
```

```
#par(mfrow=c(1,2)); hist(psign); hist(pttest)
```

Our simulation results suggest that for this problem, the power of the t-test is 0.716, and the power of Wilcoxon sign test is 0.699. So the t-test is slightly better fitted for use with this data (since the data is normal).

1 d)

The equation below shows how to solve for $z_{\alpha/2}$ given the low bound (0.25) of a confidence interval for $\hat{p}$. The `pnorm` function can then be used to compute α from $z_{\alpha/2}$.

$$0.25 = \hat{p} - z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \rightarrow z_{\alpha/2} = \frac{\hat{p} - 0.25}{\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}}$$

```
pHat = sum(x < 2600) / length(x) # compute pHat based on the data
n = length(x)
z = (pHat - 0.25) / sqrt((pHat * (1-pHat)) / n) # compute z_{\alpha/2}
sprintf("pHat = %.5f, z = %.5f", pHat, z)
```

```
## [1] "pHat = 0.32979, z = 2.32696"
```

```
alpha = 2 * pnorm(z, lower.tail = F) # compute alpha
ciLevel = 1 - alpha
sprintf("confidence interval level = %.3f", ciLevel)
```

```
## [1] "confidence interval level = 0.980"
```

```
ciUpper = pHat + z * sqrt((pHat * (1-pHat)) / n)
ciLower = pHat - z * sqrt((pHat * (1-pHat)) / n)
sprintf("final confidence interval = [%.3f, %.3f]", ciLower, ciUpper)
```

```
## [1] "final confidence interval = [0.250, 0.410]"
```

We compute $\hat{p}$ from the data, use it to solve for $z_{\alpha/2}$, allowing us to compute that $\alpha = 0.01$ (note z-distribution can be used here thanks to the central limit theorem because $n = 188$ is very large). So the original confidence interval was at a 98% confidence level, and in our code above we compute the full confidence interval = [0.250, 0.410]

1 e)

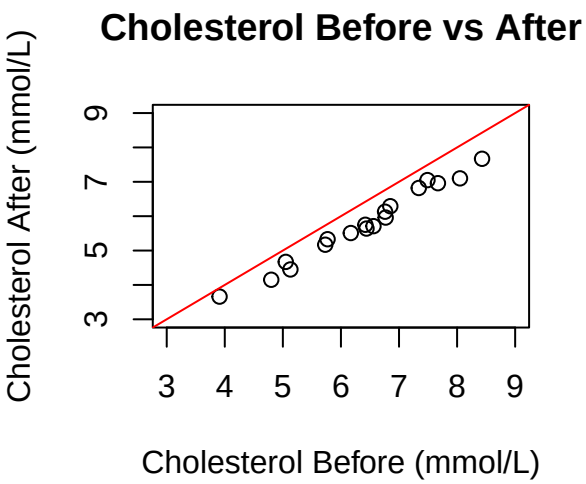
```
res = prop.test(c(34, 28), c(95, 93)); printPval(res$p.value)
```

```
## [1] "p-value = 0.501"
```

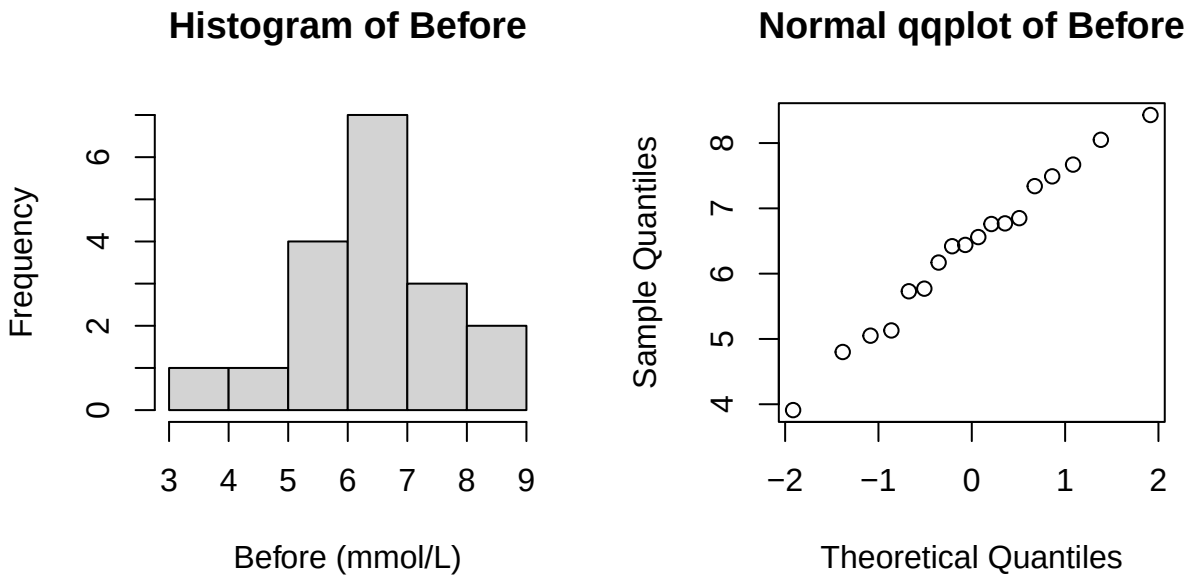
Considering male and female babies to be two separate populations, we'll call a "success" when a baby is born < 2600 grams. So we do a proportion test comparing the "successes" of each respective population (34/95 vs 28/93), and the resulting p-value $= 0.501 > 0.05$ indicates we fail to reject the H_0 that the proportion of babies born < 2600 grams differs between the male and female populations.

Exercise 2: Cholesterol

2 a)

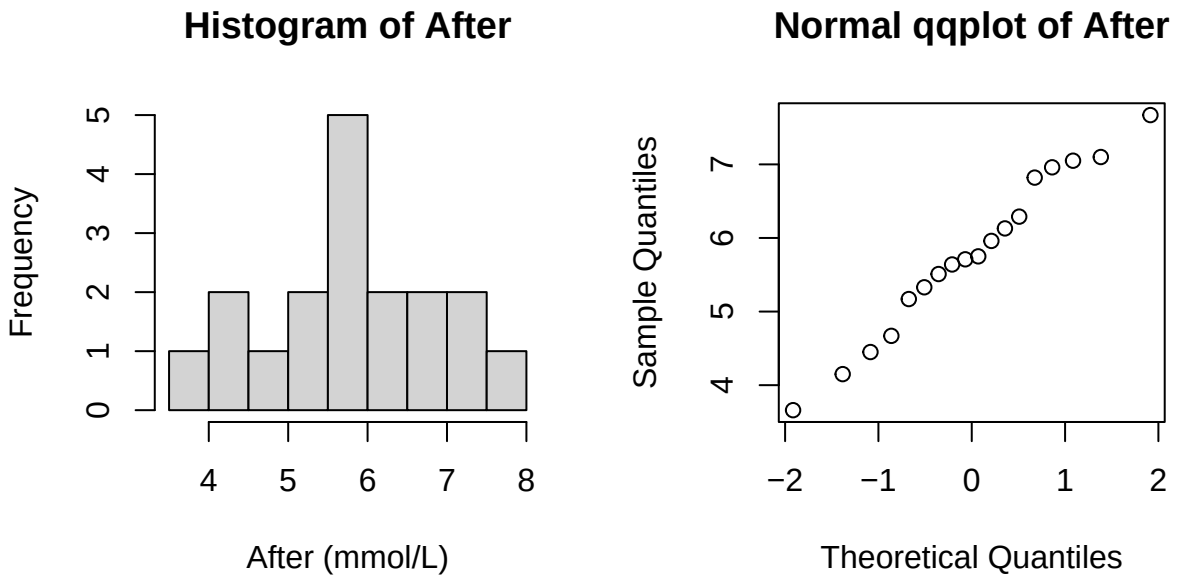


```
par(mfrow=c(1,2))
pval = checkNorm(data$Before, "Before", unit="mmol/L")
```




```
## [1] "Shapiro-Wilk normality p-value for Before: 0.967"
```

```
pval = checkNorm(data$After8weeks, "After", unit="mmol/L")
```




```
## [1] "Shapiro-Wilk normality p-value for After: 0.918"
```

Based on the histograms, qqplots and Shapiro-Wilk normality test results (both variables having $p > 0.05$), both variables (`Before`, `After8Weeks`) appear to be normally distributed.

As both variables appear to be normal, we can use the Pearson's correlation test (instead of Spearman's) to test the correlation between them.

```
res = cor.test(data$Before, data$After8weeks)
sprintf("correlation = %.3f", res[[4]])
```

```
## [1] "correlation = 0.991"
```

The Pearson's correlation test produces a correlation coefficient of 0.991, which is very close to 1.0, which suggests that indeed there is a strong positive correlation between the cholesterol levels `Before` and `After8Weeks`.

Based on the scatter plot there appears to be no inconsistencies (outliers) in the dataset.

2 b)

In this experiment, the data are paired because a given index i into the two variables, contains a value measured from the same experimental unit (person) before and after receiving a treatment (a low fat diet in this case).

To investigate whether the diet had an effect, we can use a paired t-test (due to normality), and a sign test (which makes no assumptions). The null hypothesis, H_0: is that the difference between `Before` and `After8Weeks` is 0 (there's no effect).

```
res = t.test(data$After8weeks, y=data$Before, alt="two.sided", paired=TRUE)
printPval(res$p.value)
```

```
## [1] "p-value = 0.000"
```

```
difference = data$After8weeks - data$Before
res = binom.test(sum(difference < 0), length(difference), p=0.5, alt="two.sided")
printPval(res$p.value)
```

```
## [1] "p-value = 0.000"
```

The p-values $\approx 0 < 0.05$ for both tests, so there's sufficient evidence to support the claim that the diet does have an effect on cholesterol level.

The Permutation test could also be used here because we have two paired samples, and no assumptions are needed.

2 c)

Based on the sample of cholesterol after 8 weeks, we can compute a CI for the population's average using:

$$\bar{X} \pm t_{\alpha/2} \frac{s}{\sqrt{n}}$$

And given the uniform distribution $unif(3, \theta)$ has mean $\mu = \frac{\theta+3}{2} \rightarrow \theta = 2 * \mu + 3$, so we can compute theta from our CI for $\bar{x}$.

```
summary(data$After8weeks)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.660   5.210   5.730   5.779   6.688   7.670
```

```
xbar = mean(data$After8weeks)
s = sd(data$After8weeks)
n = length(data$After8weeks)
```

```r
m = qt(0.975, df=n-1) * (s / sqrt(n))
# compute CI for \bar{x}
xBarLow = xbar - m; xBarHigh = xbar + m
sprintf("CI for xbar = [%.3f, %.3f]", xBarLow, xBarHigh)
```

```
## [1] "CI for xbar = [5.231, 6.327]"
```

```r
# use results to compute CI for \theta
thetaLow = 2 * xBarLow - 3; thetaHigh = 2 * xBarHigh - 3
sprintf("CI for theta = [%.3f, %.3f]", thetaLow, thetaHigh)
```

```
## [1] "CI for theta = [7.462, 9.654]"
```

This CI can be slightly improved, because given the max value in After8Weeks is 7.6700, we know that θ must be ≥ 7.670 in order to have generated this data.

So we can slightly adjust our CI to be:

```r
sprintf("max(data$After8weeks) = %.3f", max(data$After8weeks))
```

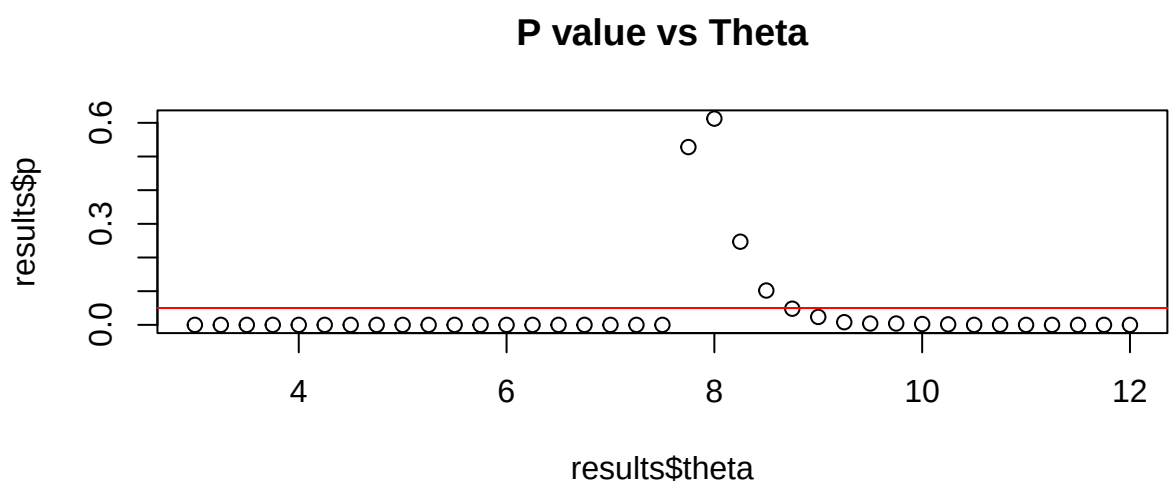
```
## [1] "max(data$After8weeks) = 7.670"
```

```r
sprintf("improved CI = [%.3f, %.3f]",  max(data$After8weeks), thetaHigh)
```

```
## [1] "improved CI = [7.670, 9.654]"
```

2 d)

```r
# returns TRUE if X ~Unif(3, theta) is plausible (meaning p > 0.05)
doBootstrap = function (x, theta, makePlot=FALSE) {
  n = length(x); t = max(x); B = 4000
  tstar = numeric(B)
  for (i in 1:B) {
    xstar = runif(n, min=3, max=theta)
    tstar[i] = max(xstar)
  }
  pl = sum(tstar<t) / B; pr = sum(tstar>t) / B
  p = 2 * min(pl, pr) # e.g. 0.038
  return(data.frame(theta=theta, p=p, valid=(p>=0.05)))
}

results = NULL
for (theta in seq(3,12,0.25)){
  cur = doBootstrap(data$After8weeks, theta)
  if (is.null(results)) {
    results = cur
  } else {
    results = rbind(results, cur)
  }
}
plot(results$theta, results$p, main="P value vs Theta")
abline(h = 0.05, col = "red") # plot y=0.05 for reference
```




```
res = dplyr::filter(results, p>0.05); res # print signficant results
```

```
##   theta      p valid
## 1  7.75 0.5280  TRUE
## 2  8.00 0.6125  TRUE
## 3  8.25 0.2470  TRUE
## 4  8.50 0.1020  TRUE
```

Based on the results for $\theta \in [7.750, 8.500]$ we fail to reject H_0. (Note that we tested θ values at intervals of every 0.25, so we can only comment on intervals that are a multiple of 0.25).

You could use the Kolmogorv-Smirnov test here as well, by generating a sample of values $(Y_1, \ldots, Y_n) \sim Unif(3, \theta)$ for a given θ, allowing you to evaluate if the original data sample could also be from the same distribution. You could also run this test many times (similar to a boostrap test), for many generated samples Y, allowing you to more robustly verify the results.

2 e)

```
res = binom.test(sum(data$After8weeks >= 6), length(data$After8weeks), p=0.5, alt="less")
printPval(res$p.value)
```

```
## [1] "p-value = 0.240"
```

The p-value from the sign-test is $0.240 > 0.05$ so we fail to reject $H_0 : m \geq 6$.

To test if the fraction of cholesterol levels After8weeks less than 4.5 is at most 25%, we propose using a modified version of the sign test which tests the 25th percentile of the data (rather than the median 50th percentile). This test is implemented as a binomial test with $p = 0.25$.

```
res = binom.test(sum(data$After8weeks < 4.5), length(data$After8weeks), p=0.25, alt="l")
printPval(res$p.value)
```

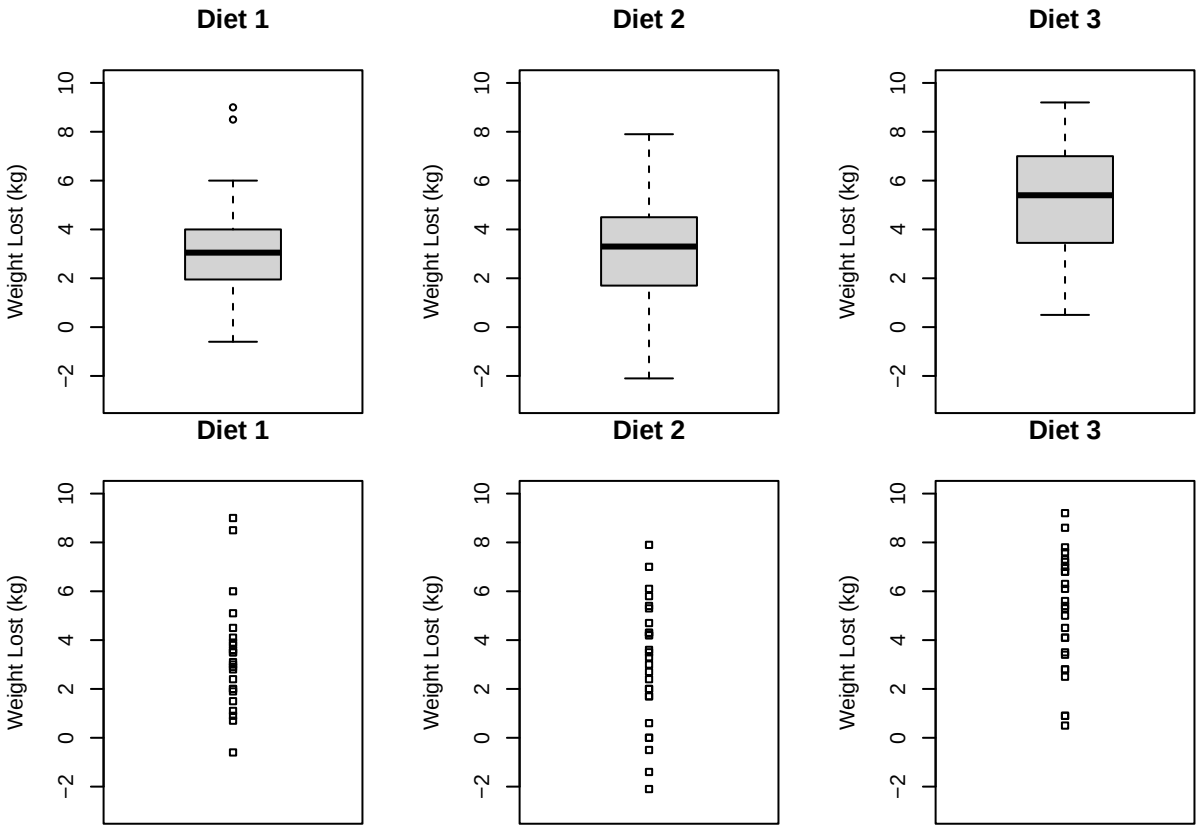
```
## [1] "p-value = 0.306"
```

The p-value from the sign-test is $0.306 > 0.05$ so there's not enough evidence to reject H_0 (that the fraction of the cholesterol levels after 8 weeks of low fat diet less than 4.5 is $\leq 25\%$).

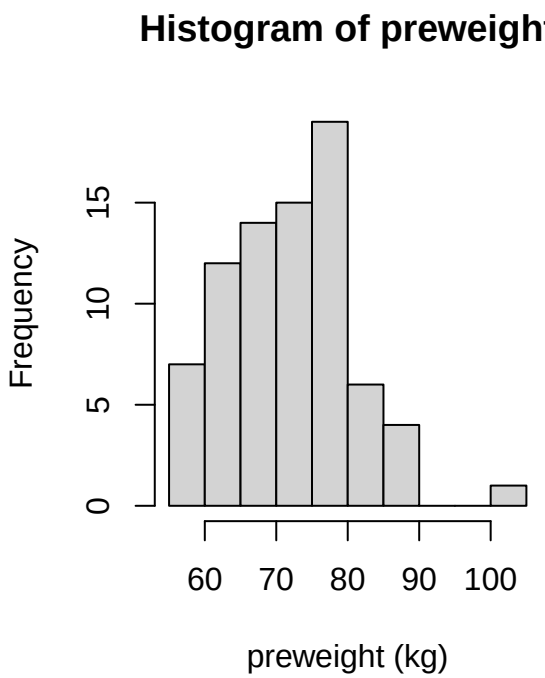
Exercise 3: Diet

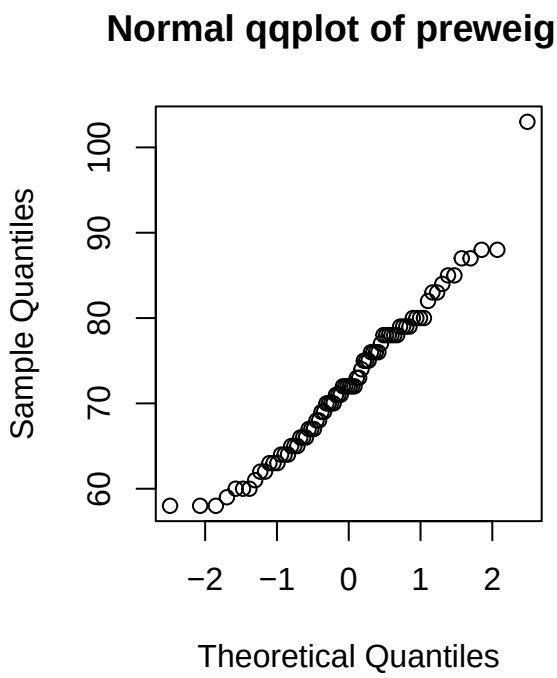
3 a)

```
data = read.table("diet.txt", header=T)
# compute weight lost
data$weightlost = data$preweight - data$weight6weeks
diet1 = dplyr::filter(data, diet == 1)
diet2 = dplyr::filter(data, diet == 2)
diet3 = dplyr::filter(data, diet == 3)
```



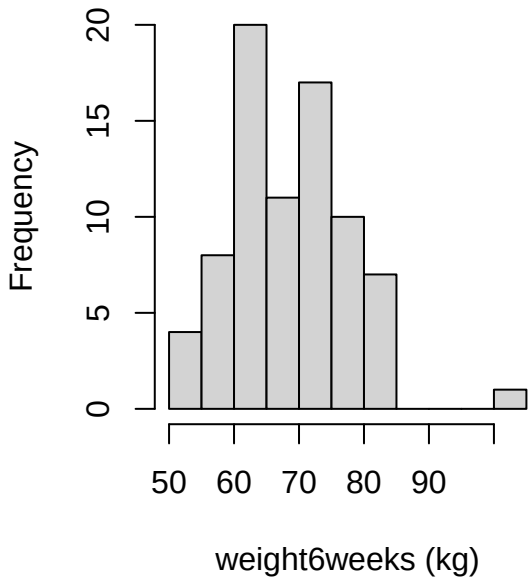

```
par(mfrow=c(1,2))
pval = checkNorm(data$preweight, "preweight", unit="kg")
```

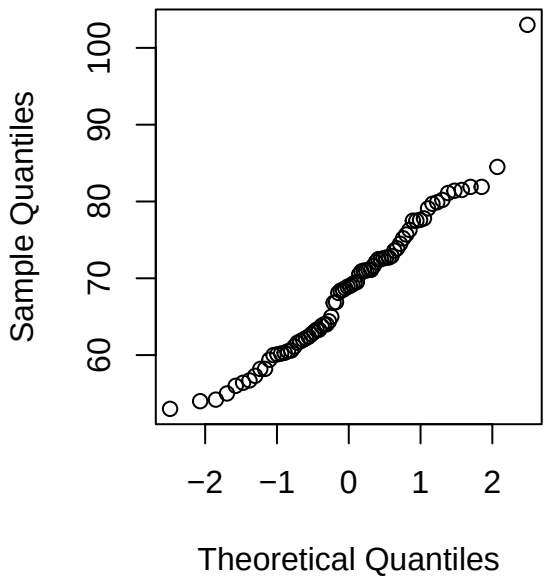





```
## [1] "Shapiro-Wilk normality p-value for preweight: 0.055"
```

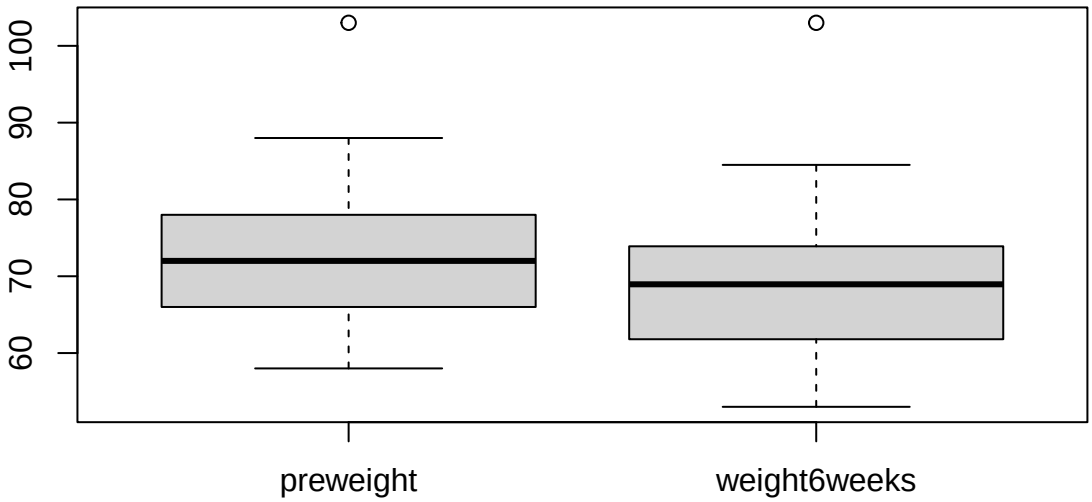
```
pval = checkNorm(data$weight6weeks, "weight6weeks", unit="kg")
```






```
## [1] "Shapiro-Wilk normality p-value for weight6weeks: 0.011"
```

```
par(mfrow=c(1,1))
boxplot(data.frame(preweight=data$preweight, weight6weeks=data$weight6weeks), names=c("preweigh
```



Based on the histograms, normal qq plots, and Shapiro-Wilk test results (with weight6weeks having $p < 0.05$), normality seems doubtful for these variables (certainly for weight6weeks at least).

Additionally, while the two variables appear symmetric based on the boxplots, the histogram for preweight casts some doubt into the symmetry of the data, so we'll use the sign test as a safe way to compare the median difference between these two variables.

(Note that we assume this question wants us to combine the sub-datasets of the 3 diets, into a larger dataset that doesn't consider the type of diet).

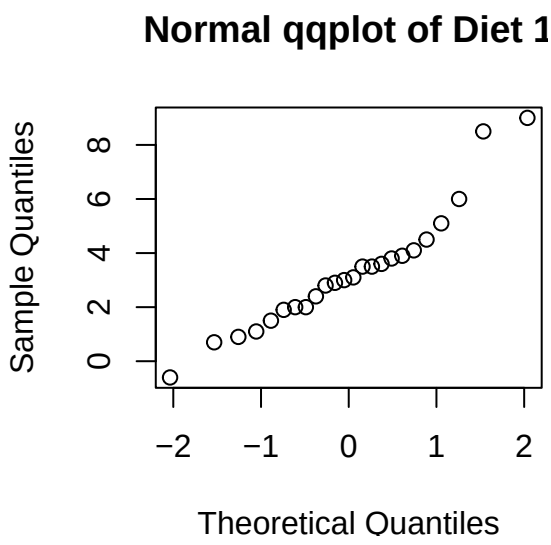
```
res = binom.test(sum(data$weightlost > 0), length(data$weightlost), p=0.5, alternative="g")
```

The results of the sign test with p=0.000 < 0.05 led us to reject the $H_0 : m \leq 0$ (that the median difference in weight before and after the diet is ≤ 0).

3 b)

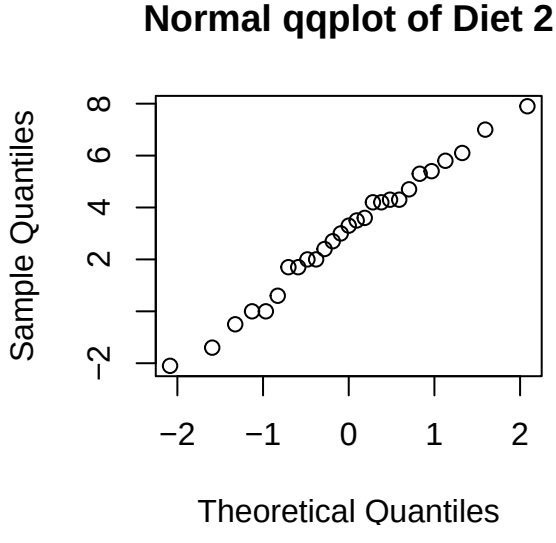
First we check for normality in the diet datasets:

```
par(mfrow = c(1, 2))
pval = checkNorm(diet1$weightlost, "Diet 1")
```



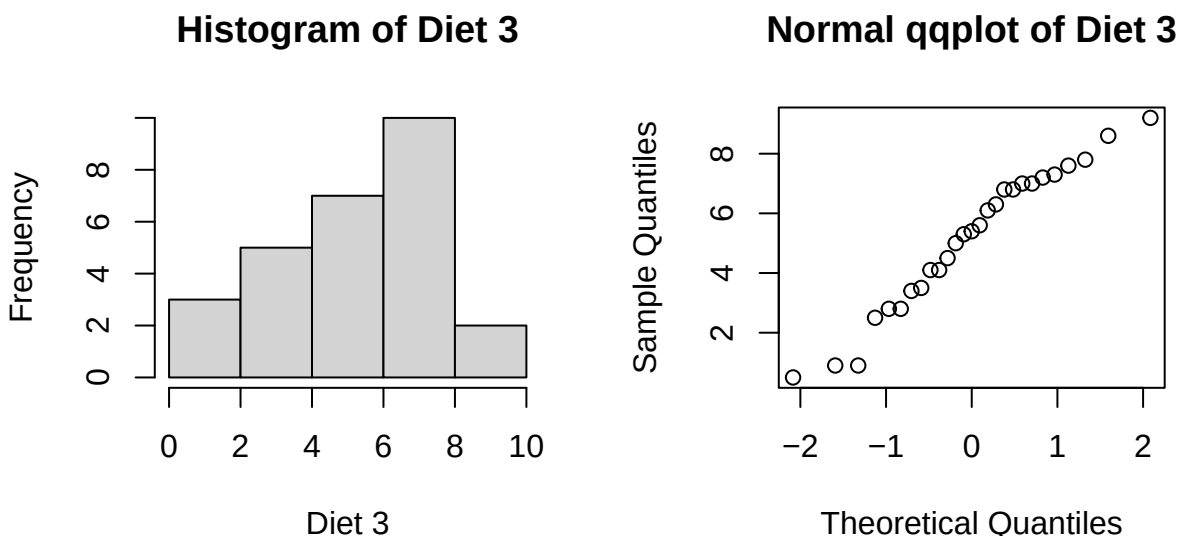

```
## [1] "Shapiro-Wilk normality p-value for Diet 1: 0.077"
```

```
pval = checkNorm(diet2$weightlost, "Diet 2")
```




```
## [1] "Shapiro-Wilk normality p-value for Diet 2: 0.961"
```

```
pval = checkNorm(diet3$weightlost, "Diet 3")
```




```
## [1] "Shapiro-Wilk normality p-value for Diet 3: 0.372"
```

Based on the boxplots and qqplots, normality appears doubtful (for at least diet 1 and 3). We will proceed with ANOVA anyways as instructed.

```
# tell R that certain columns are factors and not just numbers
data$diet = factor(data$diet)
dietaov = lm(weightlost~diet, data=data)
danova = anova(dietaov)
print("summary(dietaov):"); summary(dietaov)
```

```
## [1] "summary(dietaov):"
```

```
##
## Call:
## lm(formula = weightlost ~ 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
```

```
print("anova:"); danova
```

```
## [1] "anova:"
```

```
## Analysis of Variance Table
##
## Response: weightlost
##           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
```

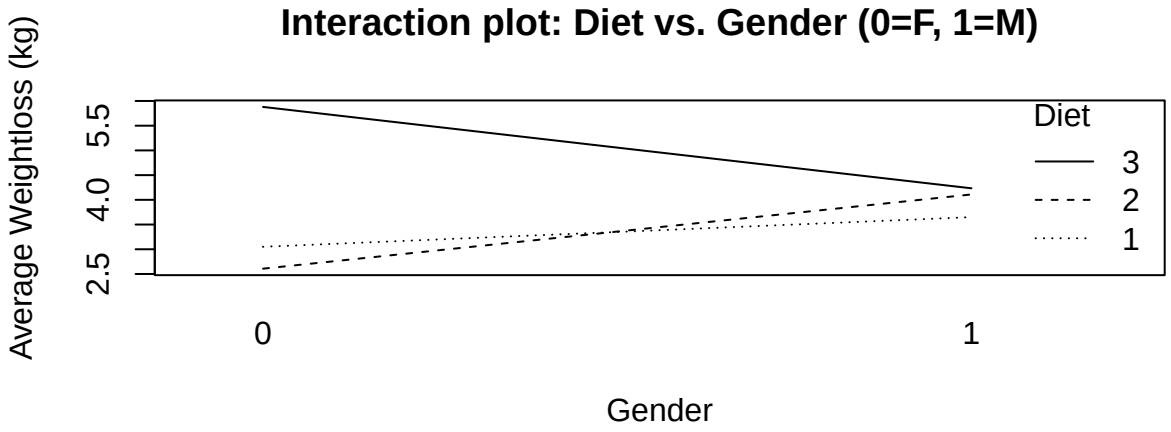
The p-value from ANOVA $0.003 < 0.05$ suggests that diet does have a significant effect on weight loss.

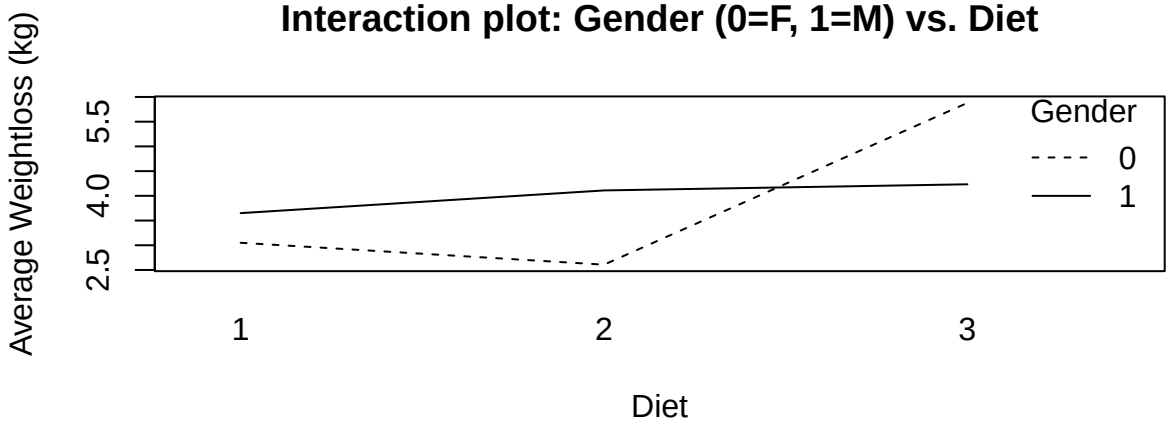
All 3 diets appear to lead to weight loss (based on their average values, which can be seen by adding the intercept (diet1's mean), to the respective α values for diet2 and diet3. Also diet3 appeared to be the best for losing weight, with the highest average weight loss of 3.3.

The Kruskal-Wallis test can be applied here because $n_i > 5$ for all diet groups, and the test doesn't rely on normality so we don't have to be concerned about normality.

3 c:

```
# diet, gender
data$gender = factor(data$gender)
par(mfrow = c(2, 1))
# remove people with no gender data
data2 = dplyr::filter(data, !is.na(gender))
interaction.plot(data2$gender, data2$diet, data2$weightlost, main ="Interaction plot: Diet vs.

interaction.plot(data2$diet, data2$gender, data2$weightlost, main ="Interaction plot: Gender (
```






```
gaov = lm(weightlost~diet*gender, data=data2)
#print("summary(gaov):"); summary(gaov)
ganova = anova(gaov)
print("anova:"); ganova
```

```
## [1] "anova:"
```

```
## Analysis of Variance Table
##
## Response: weightlost
##             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
```

The p-value for interaction, $0.049 < 0.05$, indicates we reject H_{AB} as there's sufficient evidence to suggest there's an interaction between the diet and gender variables.

Based on the ANOVA results, we fail to reject the null hypothesis H_B (that gender has no effect on

weightlost), however we reject H_A (that diet has no effect on weightlost).

In summary, there's sufficient data to suggest that diet has an effect on weightloss, but insufficient to suggest gender does, and gender and diet appear to have an interaction on weightlost.

3 e): Given the data, it seems more intuitive to us to use the approach from b). This is because the data has these three defining diets, and the one-way ANOVA model allows for a more fine-grained approach into checking factor effects. This is further confirmed by gender not being a statistically significant factor into weight loss, as per c).

```
sprintf("diet1 predicted weightlost: %.3f", dietaov$coefficients[1])
```

```
## [1] "diet1 predicted weightlost: 3.300"
```

```
sprintf("diet2 predicted weightlost: %.3f", dietaov$coefficients[1] + dietaov$coefficients[2])
```

```
## [1] "diet2 predicted weightlost: 3.026"
```

```
sprintf("diet2 predicted weightlost: %.3f", dietaov$coefficients[1] + dietaov$coefficients[3])
```

```
## [1] "diet2 predicted weightlost: 5.148"
```

The output above gives the predicted weightloss for each diet (using the model fit in part 3b). The average predicted weightloss for diets 2 and 3 are computed by adding their coefficient (in the linear model) to the intercept (the predicted diet1 weightlost).

Exercise 4: Yield of Peas

4 a)

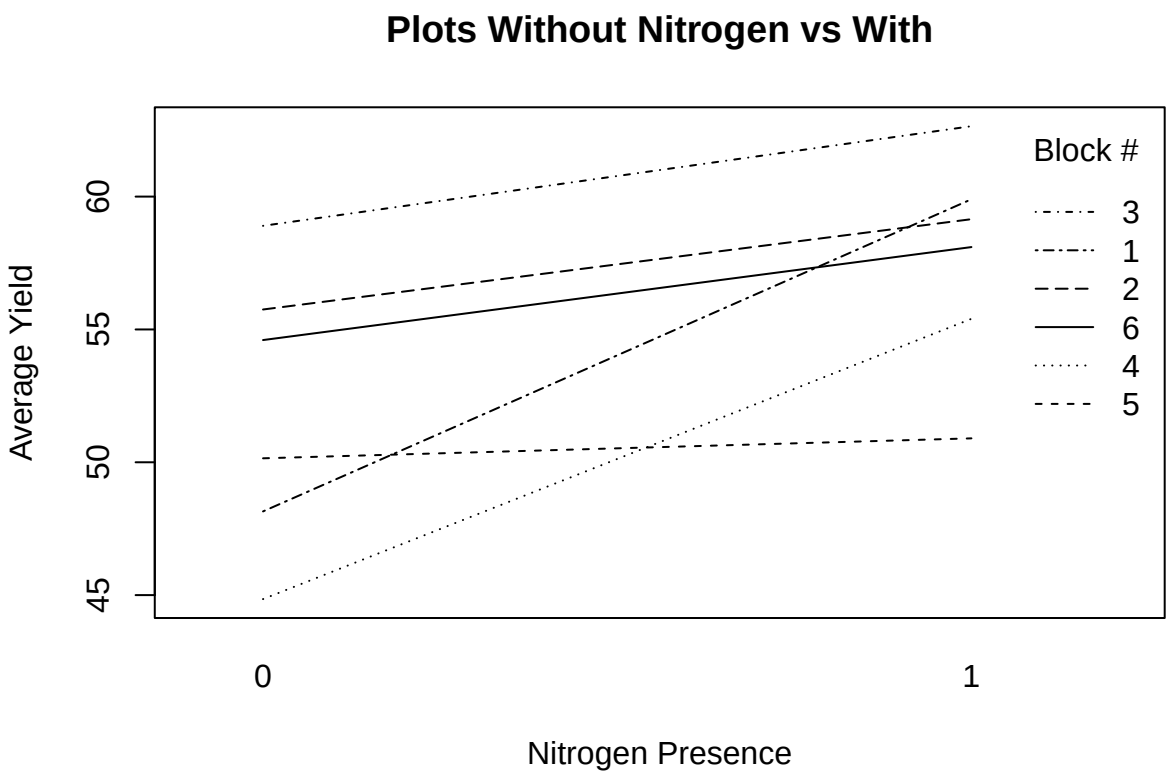
```
library(MASS)
data = npk

randomize = function(n_blocks){
  t = c(1,1,0,0)
  t_temp = c()
  for (x in 1:n_blocks){
    t_temp = c(t_temp, sample(t))
  }
  return(t_temp)
}
#Generate random columns where each block has random
#  permutations of exactly two of each soil additive
N = randomize(6); P = randomize(6); K = randomize(6)
block = c()
for (x in 1:6){
  block = c(block, rep(x,4))
}
random_df = data.frame(block = block, N = N, P = P, K = K)
# print top and bottom rows of dataframe for visualizing
head(random_df,4)
```

```
##   block N P K
## 1     1 0 0 0
## 2     1 1 1 0
## 3     1 1 0 1
## 4     1 0 1 1
```

```
tail(random_df,4)
```

```
##    block N P K
## 21     6 1 0 1
## 22     6 0 0 0
## 23     6 1 1 0
## 24     6 0 1 1
```

4 b)



4 c)

```
naov = lm(yield~block+N, data=data)
#print("summary(naov):"); summary(naov)
nanova = anova(naov)
print("nanova:"); nanova
```

```
## [1] "nanova:"
```

```
## Analysis of Variance Table
##
## Response: yield
##           Df Sum Sq Mean Sq F value   Pr(>F)
```

```
## block      5 343.29  68.659  3.3951 0.026173 *
## N          1 189.28 189.282  9.3598 0.007095 **
## Residuals 17 343.79  20.223
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As we can see from the two-way ANOVA above, the block factor has $p < 0.05$, so we reject the null hypothesis that the block factor has no effect on the yield, suggesting that it is sensible to keep it in the model.

The Friedman test can't be used here, because it's for complete block design experiments, and this dataset was the result of an incomplete block design (not all possible combinations of soil additives were applied to each farm plot).

4 d)

```
testModel = function(raov, coef="") {
  #print("raov:"); print(raov)
  #print("summary(raov):"); print(summary(raov))
  ranova = anova(raov)
  print("anova results:"); print(ranova)
}
testModel(lm(yield~block*N+P+K, data=data), coef="block:N")
```

```
## [1] "anova results:"
## Analysis of Variance Table
##
## Response: yield
##           Df Sum Sq Mean Sq F value   Pr(>F)
## block      5 343.29  68.659  4.8465 0.016439 *
## N          1 189.28 189.282 13.3611 0.004423 **
## P          1   8.40   8.402  0.5931 0.459045
## K          1  95.20  95.202  6.7201 0.026843 *
## block:N    5  98.52  19.704  1.3908 0.306583
## Residuals 10 141.67  14.167
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
testModel(lm(yield~block*P+N+K, data=data))
```

```
## [1] "anova results:"
## Analysis of Variance Table
##
## Response: yield
##           Df Sum Sq Mean Sq F value   Pr(>F)
## block      5 343.29  68.659  4.0678 0.028234 *
## P          1   8.40   8.402  0.4978 0.496588
## N          1 189.28 189.282 11.2143 0.007381 **
## K          1  95.20  95.202  5.6404 0.038947 *
## block:P    5  71.40  14.280  0.8460 0.547341
## Residuals 10 168.79  16.879
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
testModel(lm(yield~block*K+N+P, data=data))
```

```
## [1] "anova results:"
## Analysis of Variance Table
##
## Response: yield
##           Df Sum Sq Mean Sq F value   Pr(>F)
## block      5 343.29  68.659  4.0407 0.028799 *
## K          1  95.20  95.202  5.6028 0.039477 *
## N          1 189.28 189.282 11.1397 0.007521 **
## P          1   8.40   8.402  0.4945 0.497989
## block:K    5  70.27  14.054  0.8271 0.558263
## Residuals 10 169.92  16.992
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Based on the ANOVA results, there's no significant interaction between the block and any of the fertilizers (N, P, and K). This is indicated by the p-values for `block:N, block:P, block:K` in each of the respective models.

```
raov = lm(yield~block+N+P+K, data=data)
#print("summary(raov):"); print(summary(raov))
ranova = anova(raov)
print("anova results:"); print(ranova)
```

```
## [1] "anova results:"
```

```
## Analysis of Variance Table
##
## Response: yield
##           Df Sum Sq Mean Sq F value  Pr(>F)
## block      5 343.29  68.659  4.2879 0.01272 *
## 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 *
## Residuals 15 240.19  16.012
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Based on the p-values from the ANOVA output, P (phosphorous) is not a significant factor (effecting yield), while N, K, and block all appear to be significant with $p < 0.05$.

From the models we investigated, the linear combination `lm(yield~block+N+P+K, data=data)` is our favorite because it doesn't include an interaction between block and any of the other factors (which was found to be insignificant). Therefore this model has a simpler level of complexity while still allowing the individual effects of N, P, and K to be evaluated.

4 e)

```
library(lme4)
```

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

```
model1 = lmer(yield~N+(1|block), REML=FALSE,data=data)
model2 = lmer(yield~(1|block), REML=FALSE,data=data)
anova(model2, model1)
```

```
## Data: data
## Models:
## model2: yield ~ (1 | block)
## model1: yield ~ N + (1 | block)
##        npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## model2    3 159.38 162.91 -76.690   153.38
## model1    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
```

By the ANOVA comparison above, the p-value $0.005 < 0.05$, indicates there's sufficient evidence to support the claim that nitrogen (N) effects yield.

Compared to the fixed effects model in part 4c, both models agree on supporting the claim of nitrogen effecting yield.