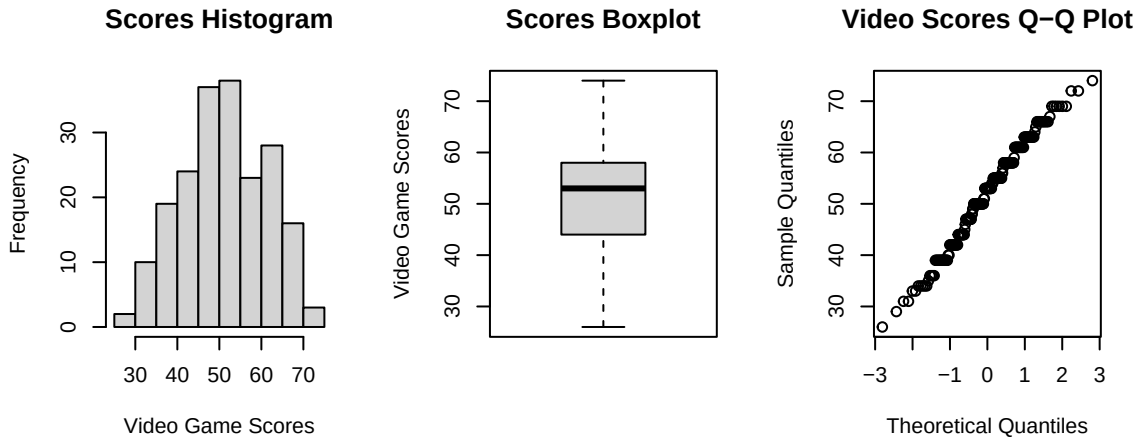
Group 3 - Assignment 1

Exercise 1. Ice Cream

a)

```
#Read in the data and extract the relevant column
ice_cream=read.csv("data/Ice_cream.csv",header=TRUE)
video=ice_cream$video

#Create relevant figures to assess distribution normality
par(mfrow=c(1,3))
hist(video, xlab="Video Game Scores", main="Scores Histogram"); boxplot(video, ylab="Video Game Scores"
```



The data for the video games scores appears to be close to normally distributed. Both the histogram and boxplot indicate that the scores are unimodal and moderately evenly distributed about the mean. The qq-plot confirms this as the data points are nearly linear. Hence, any testing or methods used with a normality assumption are justified.

```
#Constructing a bounded 97% confidence interval
video_mean = mean(video)
video_se = sd(video)/sqrt(length(video))
alpha = 0.03
degrees_freedom = length(video) - 1
t_score = qt(p=alpha/2, df=degrees_freedom, lower.tail=F)
margin_error = video_se * t_score
lower_bound = video_mean - margin_error
upper_bound = video_mean + margin_error
```

The bounds for the 97% confidence interval are [50.3197, 53.3803].

```
#Calculating the number of samples needed for a confidence interval of at most length 3
desired_margin_error = 1.5
z_score = qnorm(0.985)
```

```r
num_samples = ((z_score*sd(video))/desired_margin_error)^2
```

The number of samples needed for a confidence interval of at most length 3 is at least 206.

```r
#Creating a bootstrapped 97% confidence interval
B = 1000
Tstar = numeric(B)
for (i in 1:B) {
  Xstar = sample(video, replace = TRUE)
  Tstar[i] = mean(Xstar)
}
Tstar015 = quantile(Tstar, 0.015)  #Lower bound
Tstar985 = quantile(Tstar, 0.985)  #Upper bound

confidence_interval = c(2*mean(video) - Tstar985, 2*mean(video) - Tstar015)
```

Our bootstrapped confidence interval is (50.3349, 53.2408). We notice that this confidence interval is minutely smaller than the bounded confidence interval above. However, they are extremely similar. Since the data is already relatively normally distributed, the original confidence interval likely provides a good enough estimate and bootstrapping is unnecessary in this case.

b)

```r
#First t-test when mu_0 = 50
t.test(video, mu=50, alt="g")
```

```
##
##  One Sample t-test
##
## data:  video
## t = 2.6425, df = 199, p-value = 0.004442
## alternative hypothesis: true mean is greater than 50
## 95 percent confidence interval:
##  50.69305      Inf
## sample estimates:
## mean of x
##     51.85
```

```r
#Second t-test when mu_0 = 51
t.test(video, mu=51, alt="g")
```

```
##
##  One Sample t-test
##
## data:  video
## t = 1.2141, df = 199, p-value = 0.1131
## alternative hypothesis: true mean is greater than 51
## 95 percent confidence interval:
##  50.69305      Inf
## sample estimates:
## mean of x
##     51.85
```

Since the p-value for the first one-sided t-test when $\mu_0 = 50$ is less than 0.05, we would reject the null hypothesis that the mean video game score for the sample is equal to 50 in favor of the alternative hypothesis that it is greater than 50. Since we are doing a right-sided t-test, we have a confidence interval that extends

to the left of the mean by the standard error amount, but to the right to infinity, since we do not actually specify or care what the upper bound is for the mean, as long as it is greater than 50. In the second case, we have the same case with regard to the confidence interval. However, our t-statistic and p-value have changed. In the calculation of the t-value, we subtract μ from our calculated mean. When t differs greatly from 0, our p-value will be larger. In this case, our p-value is greater than 0.05, so we do not reject the null hypothesis that the mean is equal to 51. We also notice that our actual mean is closer to 51 than it is to 50, so intuitively we would expect that we're more likely to accept the alternative hypothesis that the mean is larger than 50 than the alternative that the mean is larger than 51.

c)

```
binom.test(sum(video>50), length(video), p=0.5, alt="g")
```

```
##
##  Exact binomial test
##
## data:  sum(video > 50) and length(video)
## number of successes = 108, number of trials = 200, p-value = 0.1444
## alternative hypothesis: true probability of success is greater than 0.5
## 95 percent confidence interval:
##  0.4793777 1.0000000
## sample estimates:
## probability of success
##                   0.54
```

```
wilcox.test(video, mu=50, alt="g")
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  video
## V = 9835.5, p-value = 0.005107
## alternative hypothesis: true location is greater than 50
```

```
binom.test(sum(video<42), length(video), p=0.25, alt="l")
```

```
##
##  Exact binomial test
##
## data:  sum(video < 42) and length(video)
## number of successes = 31, number of trials = 200, p-value = 0.0007891
## alternative hypothesis: true probability of success is less than 0.25
## 95 percent confidence interval:
##  0.0000000 0.2034054
## sample estimates:
## probability of success
##                  0.155
```

First, we conduct a sign test to determine whether or not the median is greater than 50. Our null hypothesis is that the median is less than or equal to fifty, while our alternative hypothesis is that it is greater than 50. Since our resulting p-value is greater than 0.05, we do not reject the null hypothesis that the median is less than or equal to 50. This test does not align with out result from part b); however, in this case we are comparing medians and not means. In particular, with the sign test, we are not comparing the actual data values. Therefore, t-test is more sensitive to smaller deviations from the mean and any slight skewness.

Secondly, we conduct a Wilcoxon signed rank test. Since our data is close to normal, we fulfill the requirement

of a symmetric distribution. We follow the same hypotheses as the sign test above. However, since our p-value is much less than 0.05, we reject the null hypothesis in favor of the alternative.

To perform a test to check whether the fraction of the scores less than 42 is at most 25%, we can do a modified version of the sign test for medians, but instead compare the lower quartiles. In this case, we have a null hypothesis that the lower quartile is greater than or equal to 42 and an alternative hypothesis that it is less than or equal to 42. Since our p-value is much lower than 0.05, we reject the null hypothesis in favor of the alternative that in fact the fraction of scores less than 42 is at most 25%.

d)

```
n=length(video)
t=min(video)
mus = c()
B=1000; tstar=numeric(B)
for (j in 0:100){
  for (i in 1:B){
  xstar=rnorm(200, mean=j, sd=10)
  tstar[i]=min(xstar)
  }
  pl=sum(tstar<t)/B; pr=sum(tstar>t)/B
  p=2*min(pl,pr)
  if (p >= 0.05){
    mus = c(mus, j)
  }
}
```

The values of μ for which the null hypothesis is not rejected are 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62.

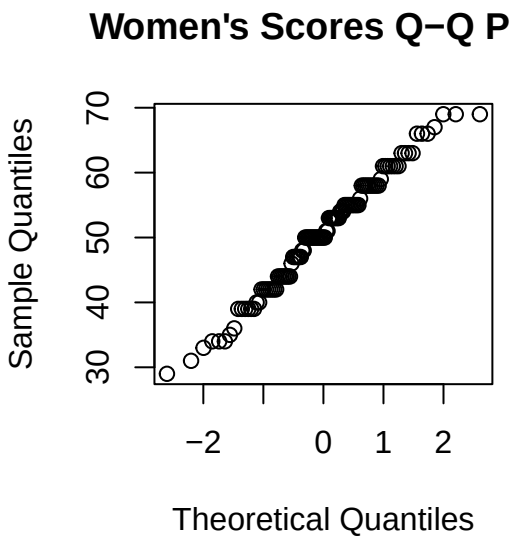
```
ks_mus = c()
suppressWarnings(for (j in 0:100){
  xstar=rnorm(200, mean=j, sd=10)
  p = ks.test(video, xstar)[[2]]
  if (p >= 0.05){
    ks_mus = c(ks_mus, j)
  }
})
```

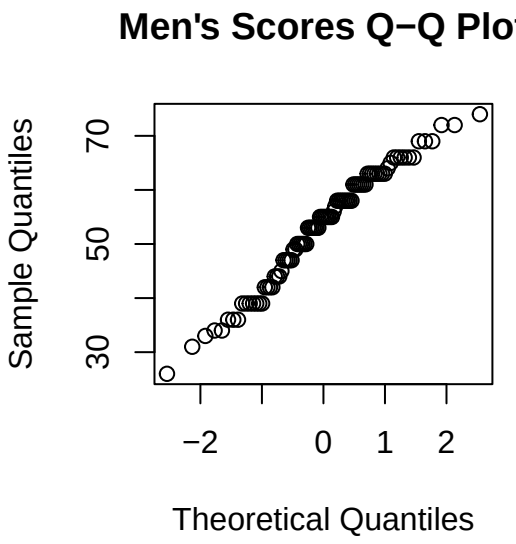
The Kolmogorov-Smirnov test is applicable in this case since we have two independent samples: one from our dataset and one from our simulated normal distribution. In this case, we can conduct an experiment similar to that above. By using a Kolmogorov-Smirnov test, the values of μ for which the null hypothesis is not rejected are 50, 51, 53.

e)

```
#Extracting the boolean data for scores from women and men
women_scores = ice_cream$video[ice_cream$female == 1]
men_scores = ice_cream$video[ice_cream$female == 0]

par(mfrow=c(1,2))
qqnorm(women_scores, main="Women's Scores Q-Q Plot"); qqnorm(men_scores, main="Men's Scores Q-Q Plot")
```






```
#Two-sample t-test
t.test(women_scores, men_scores)
```

```
##
##  Welch Two Sample t-test
##
## data:  women_scores and men_scores
## t = -1.7847, df = 176.56, p-value = 0.07603
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -5.3350452  0.2680021
## sample estimates:
## mean of x mean of y
##  50.69725  53.23077
```

In order to conduct a two-sample t-test, we assume independence in our samples. In this case that is applicable, since we assume each player plays the game without confounding factors. We also assume normality in the data. According to our qq-plots, we see that the data points are almost linear, so we can carefully make this assumption. The results of the t-test indicate a p-value that is greater than 0.05, so we do not reject the null hypothesis that the mean values of the video game scores for men and women are different.

```
#Mann-Whitney test
wilcox.test(women_scores, men_scores)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  women_scores and men_scores
## W = 4171, p-value = 0.05265
## alternative hypothesis: true location shift is not equal to 0
```

```
#Kolmogorov-Smirnov test
ks.test(women_scores, men_scores)
```

```
##
##  Exact two-sample Kolmogorov-Smirnov test
```

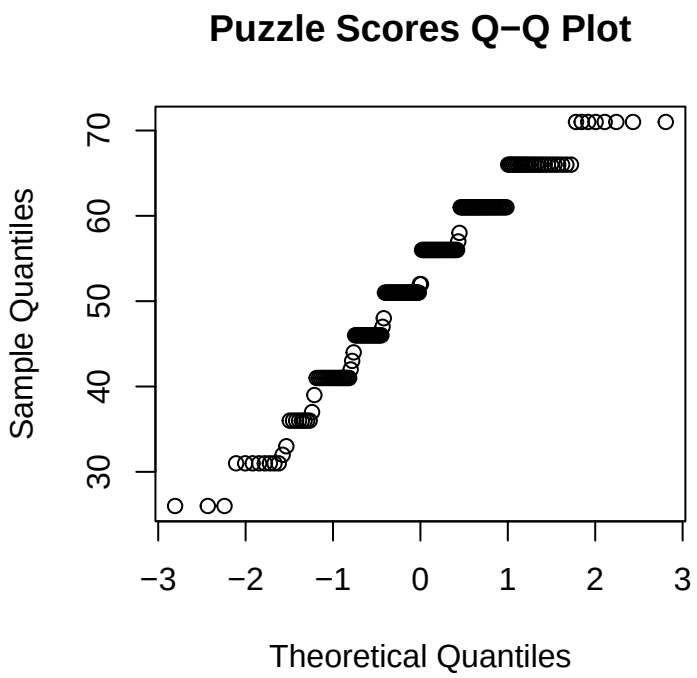
```
##
## data:  women_scores and men_scores
## D = 0.16433, p-value = 0.07148
## alternative hypothesis: two-sided
```

Similarly, the Mann-Whitney and Kolmogorov Smirnov tests assume independence of samples. In this case, both of these are applicable. In both cases, we find p-values that are greater than 0.05. For these tests, we do not reject the null hypothesis that the underlying distributions of these samples are different. In this case, the permutation test is not applicable because that requires paired samples, which these are not.

f)

```
puzzle=ice_cream$puzzle

#Check normality for the puzzle scores
qqnorm(puzzle, main="Puzzle Scores Q-Q Plot")
```




```
cor.test(video, puzzle, method="spearman")
```

```
## Warning in cor.test.default(video, puzzle, method = "spearman"): Cannot compute
## exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  video and puzzle
## S = 692269, p-value = 5.784e-13
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.4807849
```

To use Pearson's correlation test, we need to first check the normality assumption for the puzzle score data, since we have already checked the assumption for the video game score data. The data looks more like a step function than a linear function, so the normality assumption is doubtful. It is then safer to use Spearman's correlation test, which does not assume normality. In this case, the null hypothesis is that the correlation coefficient is equal to 0, i.e. there is no correlation. Since our p-value for the Spearman test is much lower than 0.05, we reject the null hypothesis in favor of the alternative, that there is a significant correlation between the two samples.

Exercise 2. Hemoglobin in trout

In this exercise, the influence of varying amounts of the antibacterial sulfamerazine on the hemoglobin levels in the blood of brown trout is explored. The statistical analysis is performed on the data set `hemoglobin.txt`, which is examined below.

```
hg_data = read.table("data/hemoglobin.txt", header=TRUE)
summary(hg_data)
```

```
##    hemoglobin          rate         method
##  Min.   : 5.500   Min.   :1.00   Length:80
##  1st Qu.: 7.350   1st Qu.:1.75   Class :character
##  Median : 8.700   Median :2.50   Mode  :character
##  Mean   : 8.736   Mean   :2.50
##  3rd Qu.:10.200   3rd Qu.:3.25
##  Max.   :11.900   Max.   :4.00
```

```
mu <- mean(hg_data$hemoglobin)
```

The set consists of 80 entries. As can be read from the data, the administering of the treatment is conducted using two different methods, A and B, and in four types of doses, labeled 1, 2, 3 and 4 and corresponding to 0, 5, 10 and 15 g of the drug. The rate (dose) and method are the factors which will be analysed, and should therefore be transformed into factor datatypes.

```
hg_data$rate = as.factor(hg_data$rate); hg_data$method = as.factor(hg_data$method)
```

a)

Upon first inspection of the data, it could be identified that the sample in the data set exhibits *balanced design*, i.e. with an equal amount of observations per factor level combination. This can be verified by creating a contingency table of the counts at each factor level combination:

```
table(hg_data$rate, hg_data$method)
```

```
##
##      A  B
##   1 10 10
##   2 10 10
##   3 10 10
##   4 10 10
```

It is evident that each factor combination contains $N = 10$ experimental units. This makes the data suitable for a two-way ANOVA test on the factors `rate` and `method`. The code below simulates how such a balanced-design sample can be acquired from a large fish population. For this purpose, a population of $M = 1000$ fish is generated, each assigned a random `rate` and `method` factor, and a hemoglobin level sampled from a normal distribution with a mean and a standard deviation equivalent to those of the original data set. The resulting data set is then sampled to obtain a balanced design of 80 fish with $N = 10$ units per combination, and the contingency table is calculated to verify the balance of the sample.

```r
set.seed(123)
M=1000; I=4; J=2

# Generate population
rates <- sample(1:I, M, replace=TRUE); methods <- sample(c('A', 'B'), M, replace=TRUE)
hemoglobin <- rnorm(M, mean(hg_data$hemoglobin), sd(hg_data$hemoglobin))
dummy_pop <- data.frame(rate=rates, method=methods, hemoglobin=hemoglobin)

# Create factor combinations
N=10
dummy_pop$combos <- interaction(dummy_pop$rate, dummy_pop$method)
combo_groups <- split(dummy_pop, dummy_pop$combos)

# Sample balanced design
balanced_groups <- lapply(combo_groups,
                          function(group) group[sample(nrow(group),min(N, nrow(group))), ])
balanced_data <- do.call(rbind, balanced_groups)
balanced_data <- subset(balanced_data, select=-combos)

# Verify balance
table(balanced_data$rate, balanced_data$method)
```

```
##
##      A  B
##   1 10 10
##   2 10 10
##   3 10 10
##   4 10 10
```

b)

Using the provided data set, a two-way ANOVA test is performed to investigate the influence of the factors `rate` and `method` on the hemoglobin levels in the trout. The test is conducted by constructing a linear model from the two factors and using it as input for the `anova` function. By default, a treatment parameterization is used.

```r
model <- lm(hemoglobin~rate*method, data=hg_data); anova(model)
```

```
## Analysis of Variance Table
##
## Response: hemoglobin
##             Df  Sum Sq Mean Sq F value    Pr(>F)
## rate         3  90.560 30.1868 19.4689 2.404e-09 ***
## method       1   2.415  2.4151  1.5576    0.2161
## rate:method  3   4.872  1.6241  1.0475    0.3769
## Residuals   72 111.637  1.5505
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

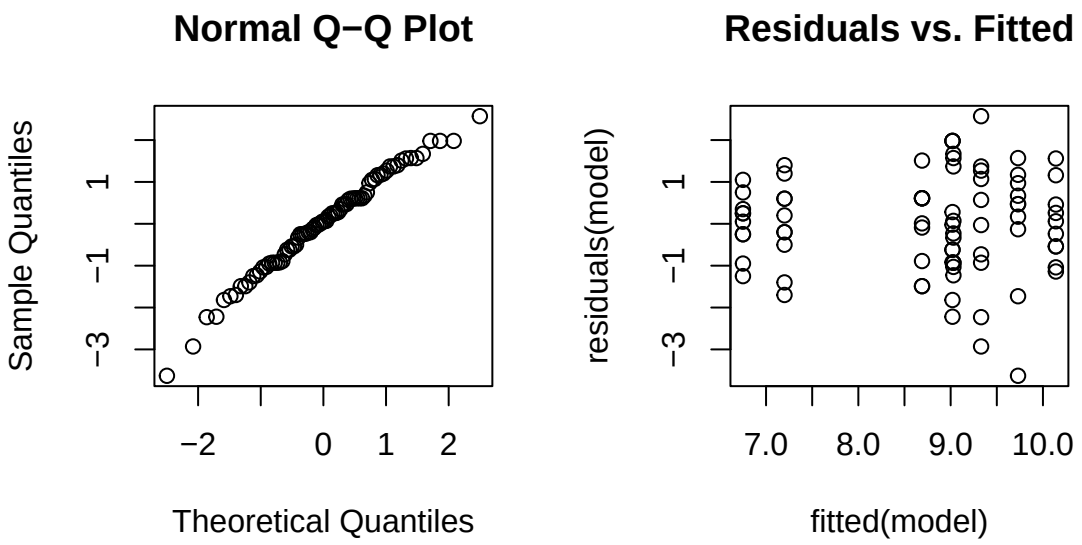
The first observation to be made is whether the H_{AB} hypothesis holds, i.e. whether an interaction between the two factors is absent. This is done by examining the p-value of the interaction term in the ANOVA table. In this case, $p = 0.3769$, which is not significant at the 5% level, and thus the null hypothesis is not rejected. This suggests that there is no evidence for interaction between the two factors. Considering the separate effect of the factors, the p-value for the `rate` factor is below the significance level, indicating a rejection of the null-hypothesis, while the p-value of the `method` factor is above the significance level, indicating a failure

to reject the null-hypothesis.

c)

To validate the test, the assumptions of the ANOVA test should be checked. The residuals of the model are examined for normality using a Q-Q-plot and are plotted against the fitted model values to ensure there is no pattern in the residuals.

```
par(mfrow=c(1,2))
qqnorm(residuals(model)); plot(fitted(model), residuals(model)); title("Residuals vs. Fitted")
```



A qualitative inspection of the plots confirms that these assumptions are met to a satisfactory degree. The validation of the model and the lack of significant interaction between the factors confirm that the indicated contribution of the `rate` factor is indeed representative for its greater effect on the response value `hemoglobin`.

Furthermore, the insignificance of the interaction allows the use of an additive model, which allows more specific comparisons between the factor levels. The `summary` function is used to extract the coefficients of the model.

```
model2 <- lm(hemoglobin~rate+method, data=hg_data); invisible(anova(model2))
summary(model2)
```

```
##
## Call:
## lm(formula = hemoglobin ~ rate + method, data = hg_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4538 -0.8881  0.0050  0.8406  2.3388
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.8012     0.3116  21.827  < 2e-16 ***
## rate2         2.7600     0.3941   7.003 9.18e-10 ***
## rate3         2.4050     0.3941   6.102 4.24e-08 ***
## rate4         1.8800     0.3941   4.770 8.86e-06 ***
```

```
## methodB       0.3475     0.2787   1.247    0.216
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.246 on 75 degrees of freedom
## Multiple R-squared:  0.4438, Adjusted R-squared:  0.4142
## F-statistic: 14.96 on 4 and 75 DF,  p-value: 4.919e-09
```

Due to the treatment parameterization, the first level of both factors is used as a reference level, so the contributions α_1 and β_1 contributions are equal to zero. As the hemoglobin levels can be seen as the sum $Y_{i,j} = \mu + \alpha_i + \beta_j + e_{j,k}$, the highest yield can be calculated by adding the highest α_i and β_j contributions to the mean μ. From the summary it is evident that these are rate 2 (5g dose) and method B. On the other hand, when extracting the `rate` and `method` factors for the highest hemoglobin measurement in the data set, the other method is present.: 11.9, 2, 1 However, this does not show contradiction, as the `method` factor is not significant in this case. Rate 2, on the other hand, is confirmed as the highest yielding rate, resulting in a mean hemoglobin level of 11.84375.

Another sample measurement is the mean hemoglobin yield under a combination of rate 3 and method A. This can be calculated by adding the contributions α_3 and β_1 (which is zero due to treatment parameterization) to the mean μ and obtaining 11.14125.

d)

Ignoring the factor `method`, the effect of the different `rate` levels can be examined in more detail using a one-way ANOVA to test the null-hypothesis that hemoglobin levels are the same for all levels.

```
model3 <- lm(hemoglobin~rate, data=hg_data); anova(model3)
```

```
## Analysis of Variance Table
##
## Response: hemoglobin
##           Df Sum Sq Mean Sq F value    Pr(>F)
## rate       3  90.56 30.1868  19.291 2.129e-09 ***
## Residuals 76 118.92  1.5648
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(model3)
```

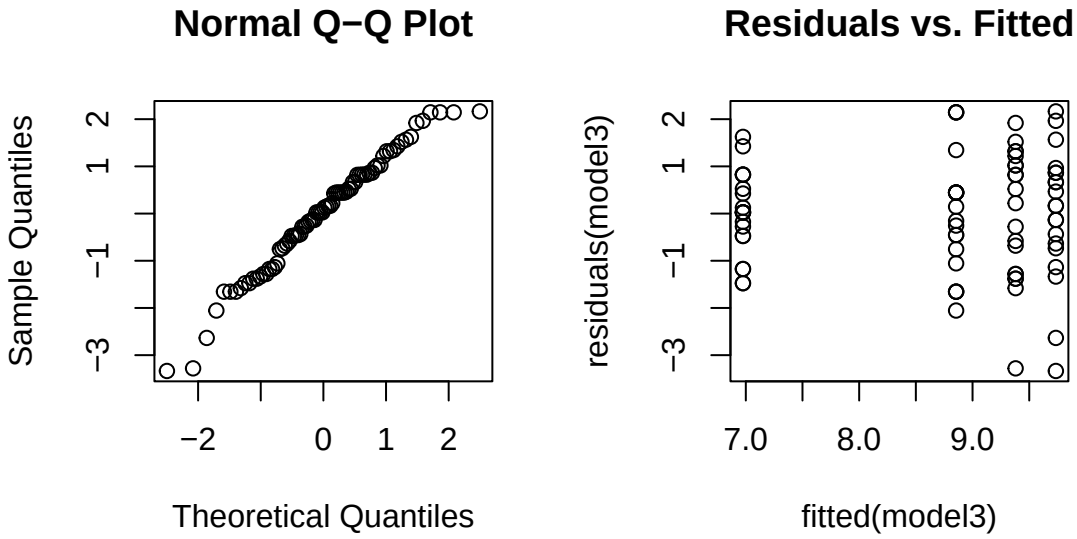
```
##
## Call:
## lm(formula = hemoglobin ~ rate, data = hg_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.335 -0.740  0.075  0.825  2.165
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.9750     0.2797  24.936  < 2e-16 ***
## rate2         2.7600     0.3956   6.977 9.69e-10 ***
## rate3         2.4050     0.3956   6.080 4.48e-08 ***
## rate4         1.8800     0.3956   4.753 9.31e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.251 on 76 degrees of freedom
```

```
## Multiple R-squared:  0.4323, Adjusted R-squared:  0.4099
## F-statistic: 19.29 on 3 and 76 DF,  p-value: 2.129e-09
```

Once again, the p-value of the `rate` factor is below the significance level, indicating a rejection of the null-hypothesis. The estimated hemoglobin values for each rate are obtained from the summary: for rate 1: 6.975, for rate 2: 11.49625, for rate 3: 11.14125 and for rate 4: 10.61625. Again, rate 2 is identified as the highest yielding one.

Whether this test is valid, remains to be seen by checking the assumptions of the ANOVA test. The residuals of the model are examined for normality using a Q-Q-plot and are plotted against the fitted model values.

```
par(mfrow=c(1,2))
qqnorm(residuals(model3)); plot(fitted(model3), residuals(model3)); title("Residuals vs. Fitted")
```



Even though a linear relationship between the theoretical and sample quantiles is visible, the tails show a more extreme deviation the QQ-plot of the two-way ANOVA, casting doubts on the normality assumption. This could be problematic for the correct interpretation of the test output.

e)

As an alternative, a Kruskal-Wallis test is performed to test the same null-hypothesis, as it operates with ranks and does not assume normality.

```
kruskal.test(hg_data$hemoglobin, hg_data$rate)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  hg_data$hemoglobin and hg_data$rate
## Kruskal-Wallis chi-squared = 34.224, df = 3, p-value = 1.777e-07
```

The resulting p-value is far below the significance level, indicating a rejection of the null-hypothesis and confirming the indication from the one-way ANOVA test that the hemoglobin levels differ across the rate levels. Given the questionable normality of the residuals, the Kruskal-Wallis test is a more robust alternative. However, it provides less detailed insight into the differences between the rate levels, as it uses a more generalised test statistic about the group medians and does not involve the more sophisticated model fitting of ANOVA.

Exercise 3. Sour Cream

Interesting facts not relevant to subquestions directly:

- as we have only one sample per combination of factors, we should not test for interactions. However, we can still test for main effects.

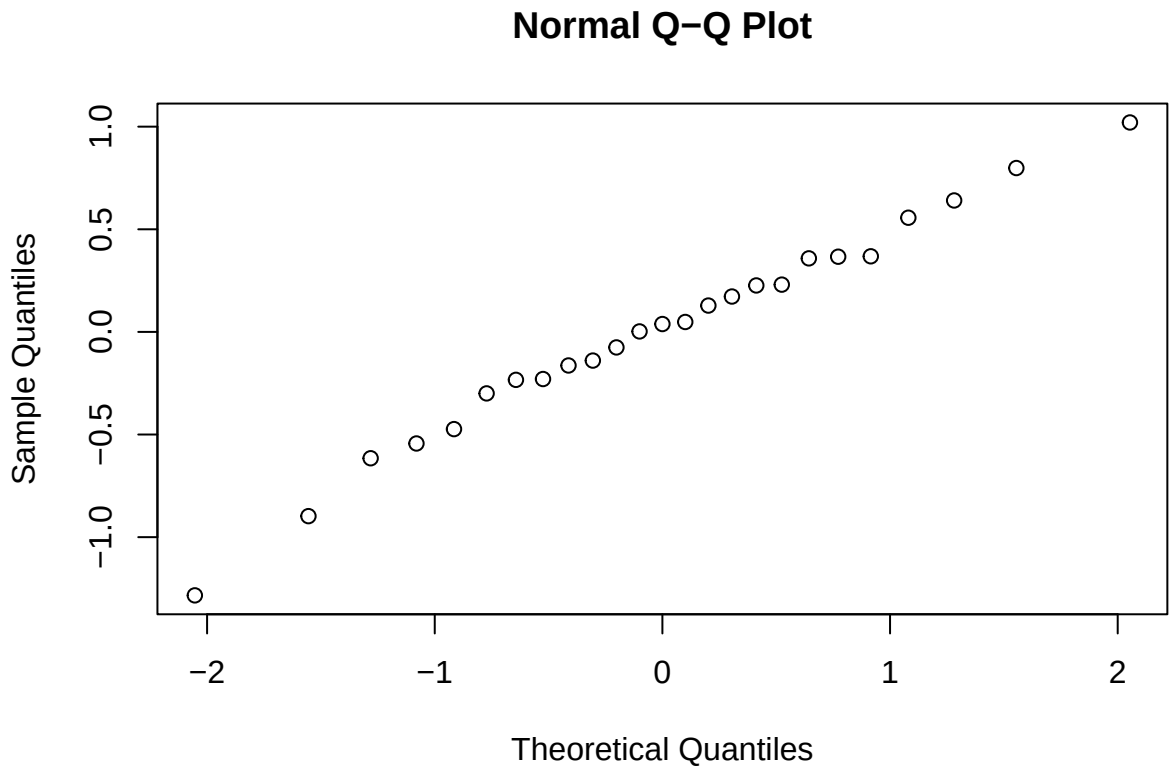
a)

Analyze the data in a three-way experiment without interactions with acidity as response and starter, batch and position as factors. By using summary command, can you tell whether there is a significant difference between the effects of starter 1 and starter 2 on acidity? Motivate your answer.

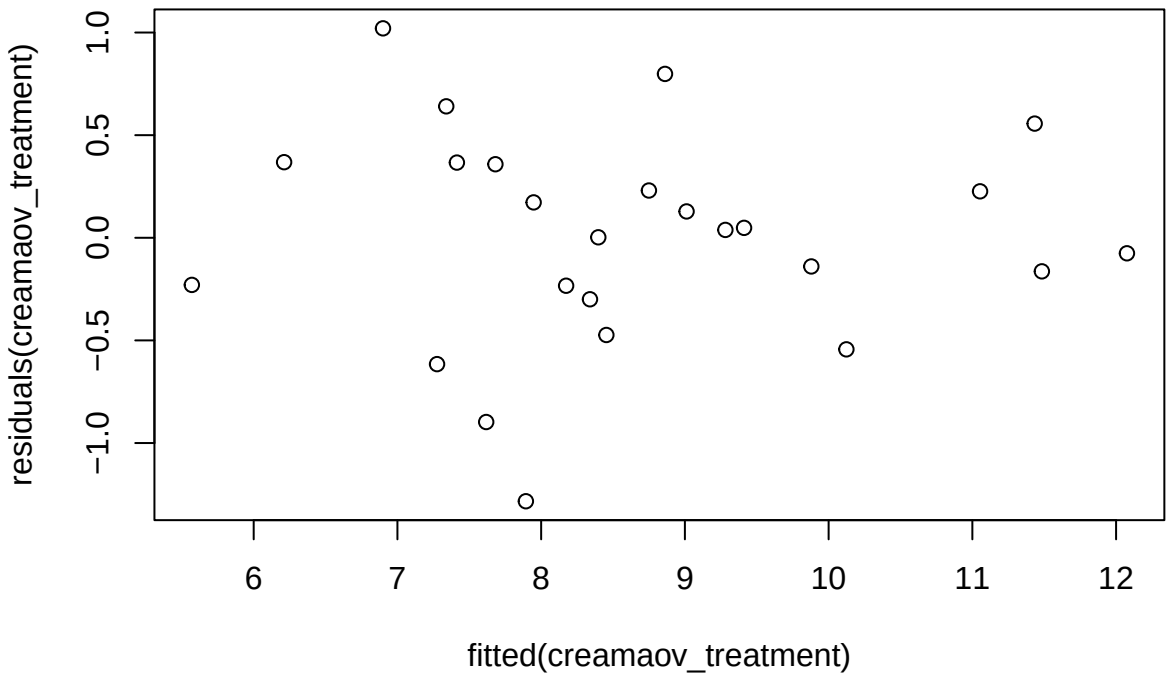
```
data <- read.csv("data/cream.txt", sep="", header=TRUE)
cream_treatment <- data.frame(acidity=data[,1], batch=factor(data[,2]), position=factor(data[,3]), star

creamaov_treatment <- lm(acidity ~ batch + position + starter, data=cream_treatment)

# Visual normality check
qqnorm(residuals(creamaov_treatment))
```




```
plot(fitted(creamaov_treatment), residuals(creamaov_treatment))
```




```
anova(creamaov_treatment)
```

```
## Analysis of Variance Table
##
## Response: acidity
##           Df Sum Sq Mean Sq F value    Pr(>F)
## batch      4 18.778  4.6944  8.5975  0.001632 **
## position   4  2.348  0.5870  1.0750  0.411191
## starter    4 44.136 11.0340 20.2080 2.904e-05 ***
## Residuals 12  6.552  0.5460
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(creamaov_treatment)
```

```
##
## Call:
## lm(formula = acidity ~ batch + position + starter, data = cream_treatment)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2836 -0.2336  0.0384  0.3584  1.0204
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.6616     0.5329  16.255 1.55e-09 ***
## batch2       -1.3480     0.4673  -2.884   0.0137 *
```

```
## batch3        0.2760     0.4673   0.591   0.5658
## batch4        1.3680     0.4673   2.927   0.0127 *
## batch5        0.2000     0.4673   0.428   0.6763
## position2    -0.6180     0.4673  -1.322   0.2107
## position3    -0.0380     0.4673  -0.081   0.9365
## position4    -0.7640     0.4673  -1.635   0.1280
## position5    -0.2640     0.4673  -0.565   0.5825
## starter2     -0.1500     0.4673  -0.321   0.7538
## starter3     -0.9800     0.4673  -2.097   0.0579 .
## starter4      2.8100     0.4673   6.013 6.10e-05 ***
## starter5     -0.4840     0.4673  -1.036   0.3208
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7389 on 12 degrees of freedom
## Multiple R-squared:  0.9088, Adjusted R-squared:  0.8175
## F-statistic:  9.96 on 12 and 12 DF,  p-value: 0.0001777
```

```
starter_difference_p_value <- summary(creamaov_treatment)$coefficients[10,4]
```

We perform a three-way ANOVA using treatment parameterization. To ensure our ANOVA results are valid, we inspect the QQ plot as well as fitted vs. residual plot to ensure normality. We pass this test based on the distribution visible there. We then look at the ANOVA coefficients to determine the significance of the means of our baseline scenario representing **starter1** and the level **starter2** sharing the same mean. As the p-value is 0.7537581, we presume that there is no significant difference between the effect of the two starters on acidity and assume they stem from the same distribution.

$$\alpha_1 = 0 \land \alpha_2 = 0 \implies \mu_{starter=1} = \mu_{starter=2} \text{ (for treatment parameterization)}$$

b)

Recall that the main interest is in the effect of starter on acidity; factors position and batch represent the block variables. Remove insignificant block variable(s) if there are such, and perform an ANOVA for the resulting "fixed effects" model. Which starter(s) lead to significantly different acidity? Motivate your answer.

```
cream_sum <- data.frame(acidity=data[,1], batch=factor(data[,2]), position=factor(data[,3]), starter=fa
contrasts(cream_sum$batch) <- contr.sum
contrasts(cream_sum$position) <- contr.sum
contrasts(cream_sum$starter) <- contr.sum

creamaov_sum <- lm(acidity ~ batch + position + starter, data=cream_sum)
anova(creamaov_sum)
```

```
## Analysis of Variance Table
##
## Response: acidity
##           Df Sum Sq Mean Sq F value    Pr(>F)
## batch      4 18.778  4.6944  8.5975  0.001632 **
## position   4  2.348  0.5870  1.0750  0.411191
## starter    4 44.136 11.0340 20.2080 2.904e-05 ***
## Residuals 12  6.552  0.5460
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

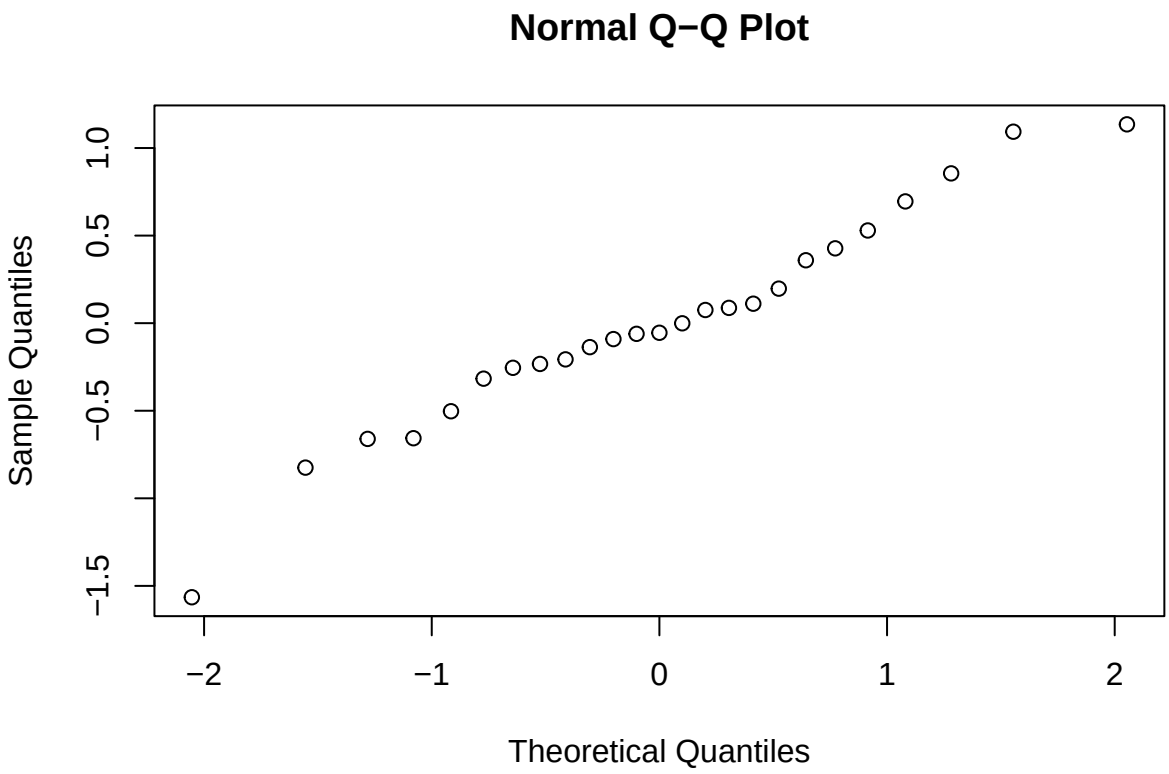
```
summary(creamaov_sum)
```

```
##
## Call:
## lm(formula = acidity ~ batch + position + starter, data = cream_sum)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2836 -0.2336  0.0384  0.3584  1.0204
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.6632     0.1478  58.620 4.01e-16 ***
## batch1       -0.0992     0.2956  -0.336 0.742952
## batch2       -1.4472     0.2956  -4.896 0.000368 ***
## batch3        0.1768     0.2956   0.598 0.560853
## batch4        1.2688     0.2956   4.293 0.001045 **
## position1     0.3368     0.2956   1.139 0.276737
## position2    -0.2812     0.2956  -0.951 0.360184
## position3     0.2988     0.2956   1.011 0.332005
## position4    -0.4272     0.2956  -1.445 0.173970
## starter1     -0.2392     0.2956  -0.809 0.434109
## starter2     -0.3892     0.2956  -1.317 0.212510
## starter3     -1.2192     0.2956  -4.125 0.001408 **
## starter4      2.5708     0.2956   8.698 1.58e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7389 on 12 degrees of freedom
## Multiple R-squared:  0.9088, Adjusted R-squared:  0.8175
## F-statistic:  9.96 on 12 and 12 DF,  p-value: 0.0001777
```

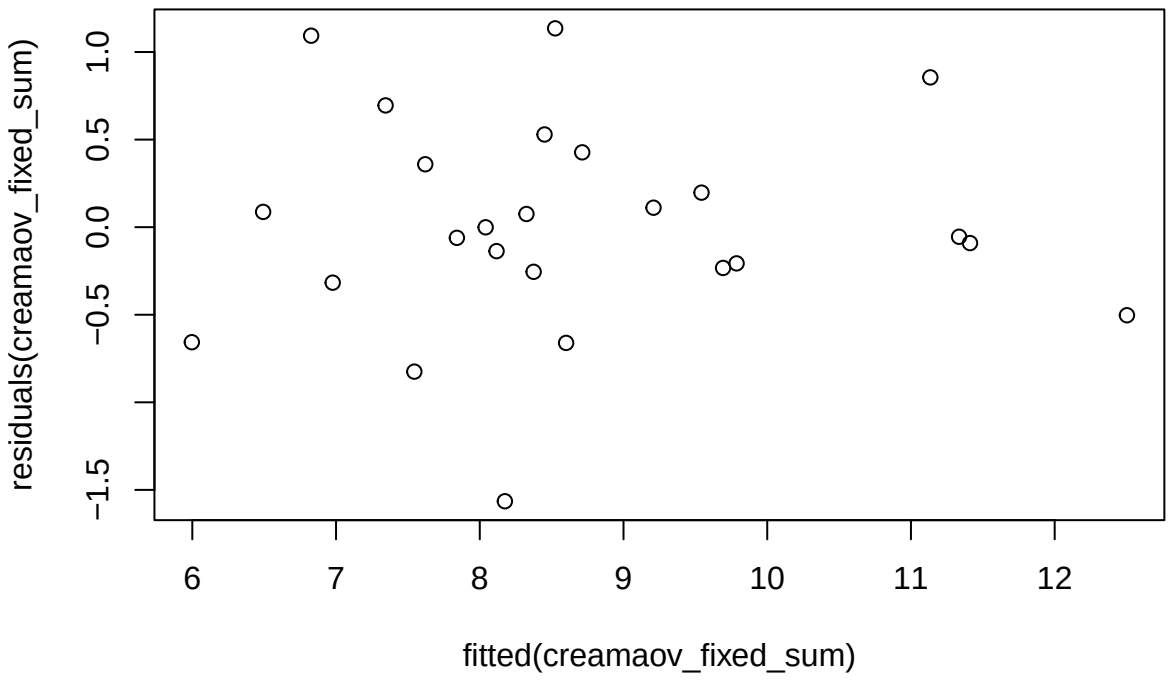
```r
p_value_batch <- anova(creamaov_sum)$'Pr(>F)'[1]
p_value_position <- anova(creamaov_sum)$'Pr(>F)'[2]
p_value_starter <- anova(creamaov_sum)$'Pr(>F)'[3]
p_value_batch3 <- summary(creamaov_sum)$coefficients[3,4]
```

```r
creamaov_fixed_sum <- lm(acidity ~ batch + starter, data=cream_sum)
# TODO: check if we are setting the right things to be factors. Block variables shouldn't be set to be
# TODO: supposedly everything is balanced here???
# we place factor of interest starter last because treatment factors needs to come last. That's because
# TODO: we could use drop1 to re-run all combinations so that the p-values will all be right

# Visual normality check
qqnorm(residuals(creamaov_fixed_sum))
```




```
plot(fitted(creamaov_fixed_sum), residuals(creamaov_fixed_sum))
```




```
anova(creamaov_fixed_sum)
```

```
## Analysis of Variance Table
##
## Response: acidity
##           Df Sum Sq Mean Sq F value    Pr(>F)
## batch      4 18.778  4.6944  8.4392 0.0007348 ***
## starter    4 44.136 11.0340 19.8360 4.816e-06 ***
## Residuals 16  8.900  0.5563
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(creamaov_fixed_sum)
```

```
##
## Call:
## lm(formula = acidity ~ batch + starter, data = cream_sum)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5648 -0.2548 -0.0548  0.3592  1.1352
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.6632     0.1492  58.078  < 2e-16 ***
## batch1       -0.0992     0.2983  -0.333 0.743816
## batch2       -1.4472     0.2983  -4.851 0.000177 ***
```

```
## batch3        0.1768     0.2983   0.593 0.561712
## batch4        1.2688     0.2983   4.253 0.000607 ***
## starter1     -0.2392     0.2983  -0.802 0.434420
## starter2     -0.3892     0.2983  -1.305 0.210487
## starter3     -1.2192     0.2983  -4.087 0.000860 ***
## starter4      2.5708     0.2983   8.617 2.09e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7458 on 16 degrees of freedom
## Multiple R-squared:  0.8761, Adjusted R-squared:  0.8141
## F-statistic: 14.14 on 8 and 16 DF,  p-value: 6.474e-06
```

Looking at the first sum parameterization ANOVA table using the full model, we find that the p-value for the factor *position* is 0.411191 and all its individual levels have coefficients corresponding to a p-value larger than `0.05` stating the lack of significance as main effect for the model. Due to the small size of the dataset, we are unable to thoroughly test for interactions, thus we will assume them to be negligible. In this case, we can remove the block variable *position* from the model to simplify.

As the *batch* variable has a factor p-value of 0.0016322 we can conclude that it is significant for the model and should be kept in our fixed effects model, further substantiated by a particularly deviating instance of **batch2** with p-value 3.6826159×10^{-4}.

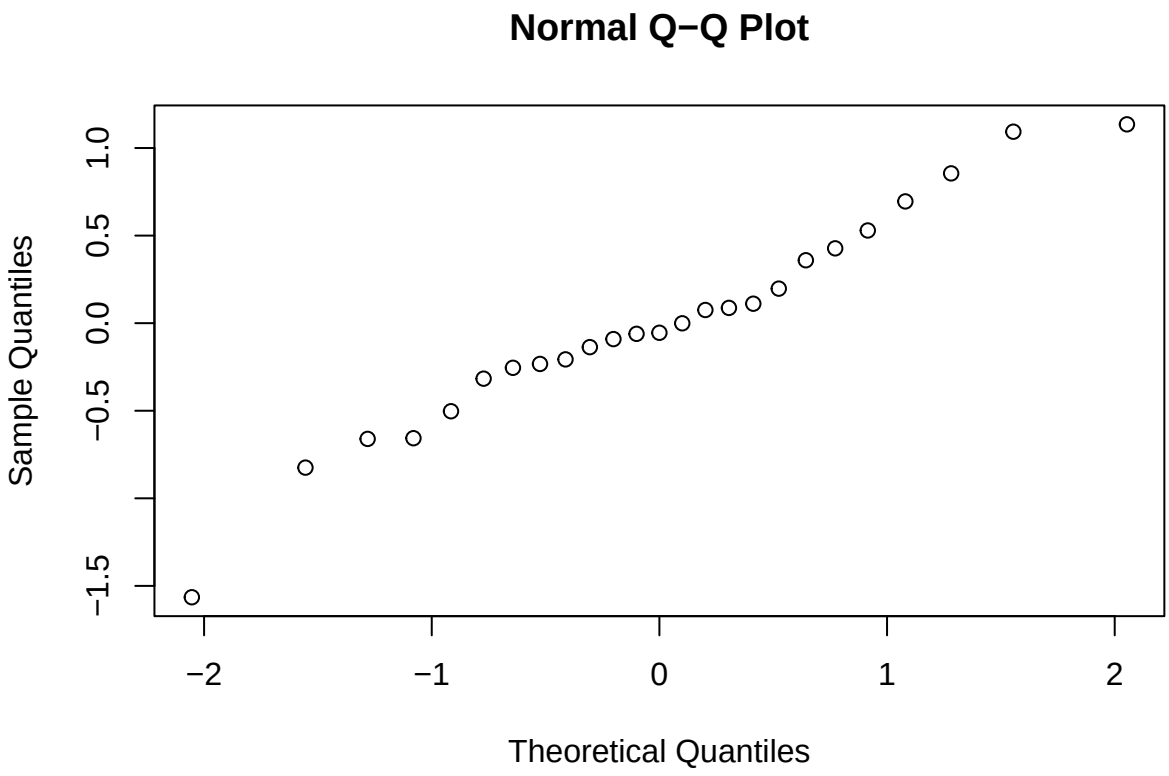
The resulting fixed effects model formula therefore becomes `acidity ~ starter + batch` ignoring interactions. In this model, we find that **starter3** and **starter4** have a very significant effect on acidity with p-values both below `0.001`.

TODO: What about starter5 in the sum paramterization model? What are we comparing here again? Double-check this makes sense.

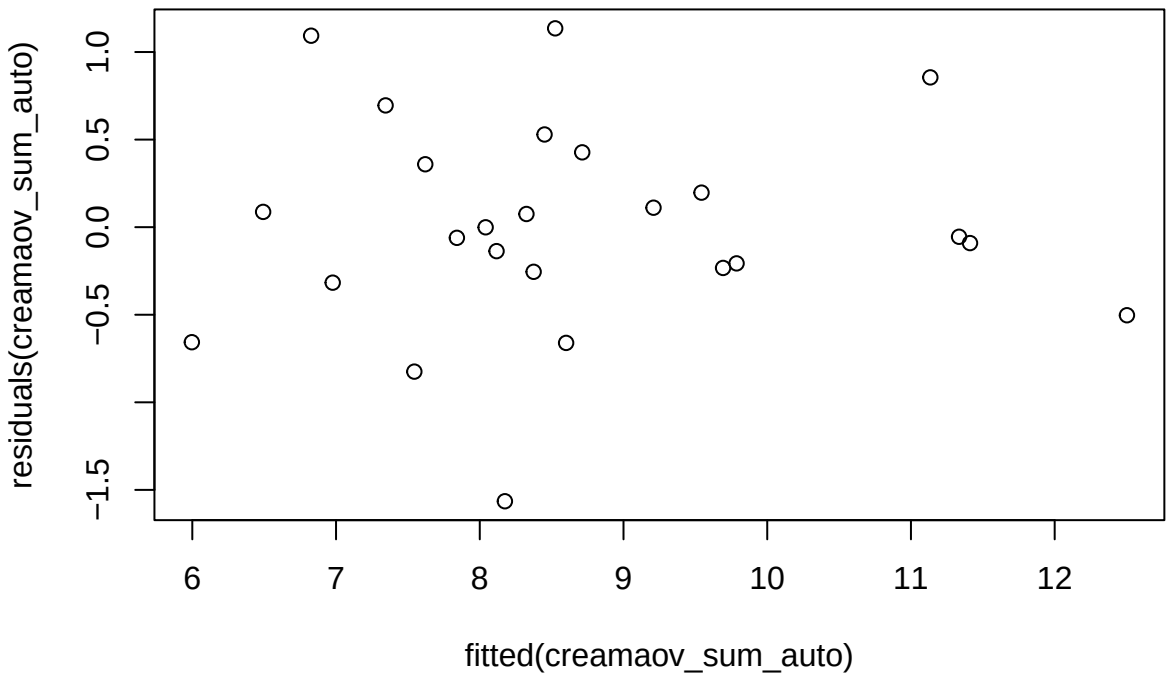
```
creamaov_sum_auto <- step(creamaov_sum)
```

```
## Start:  AIC=-7.48
## acidity ~ batch + position + starter
##
##            Df Sum of Sq    RSS    AIC
## - position  4     2.348  8.900 -7.820
## <none>                   6.552 -7.477
## - batch     4    18.778 25.330 18.328
## - starter   4    44.136 50.688 35.670
##
## Step:  AIC=-7.82
## acidity ~ batch + starter
##
##           Df Sum of Sq    RSS     AIC
## <none>                  8.900 -7.8201
## - batch    4    18.778 27.678 12.5438
## - starter  4    44.136 53.036 28.8024
```

```
# Visual normality check
qqnorm(residuals(creamaov_sum_auto))
```




```
plot(fitted(creamaov_sum_auto), residuals(creamaov_sum_auto))
```




```
anova(creamaov_sum_auto)
```

```
## Analysis of Variance Table
##
## Response: acidity
##           Df Sum Sq Mean Sq F value    Pr(>F)
## batch      4 18.778  4.6944  8.4392 0.0007348 ***
## starter    4 44.136 11.0340 19.8360 4.816e-06 ***
## Residuals 16  8.900  0.5563
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(creamaov_sum_auto)
```

```
##
## Call:
## lm(formula = acidity ~ batch + starter, data = cream_sum)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5648 -0.2548 -0.0548  0.3592  1.1352
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.6632     0.1492  58.078  < 2e-16 ***
## batch1       -0.0992     0.2983  -0.333 0.743816
## batch2       -1.4472     0.2983  -4.851 0.000177 ***
```

```
## batch3        0.1768     0.2983   0.593 0.561712
## batch4        1.2688     0.2983   4.253 0.000607 ***
## starter1     -0.2392     0.2983  -0.802 0.434420
## starter2     -0.3892     0.2983  -1.305 0.210487
## starter3     -1.2192     0.2983  -4.087 0.000860 ***
## starter4      2.5708     0.2983   8.617 2.09e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7458 on 16 degrees of freedom
## Multiple R-squared:  0.8761, Adjusted R-squared:  0.8141
## F-statistic: 14.14 on 8 and 16 DF,  p-value: 6.474e-06
```

Utilizing the step function utility in its default configuration further substantiates our previous results as the algorithm also removed the insignificant block variable *position* from the model. The resulting model of the algorithm is the same as the one we have established manually above.

c)

For the resulting model from c), can we also apply the Friedman test to test whether there is an effect of starter on acidity?

```
friedman.test(acidity ~ starter | position, data=cream_sum) # alternative syntax
```

```
##
##  Friedman rank sum test
##
## data:  acidity and starter and position
## Friedman chi-squared = 8.48, df = 4, p-value = 0.0755
```

```
# friedman.test(cream_sum$acidity, cream_sum$starter, cream_sum$position)
```

Since each combination of batch and starter exists in the dataset and there are more than two levels of both factors, we can apply the Friedman test to test whether there is an effect of starter on acidity. The Friedman test does not assume normality either so it should be more robust, albeit less sensitive.

The p-value of the Friedman test is `0.0755` which is slightly lower than the significance threshold of `0.05` and therefore we accept the null hypothesis that all means stem from the same distribution. So according to the Friedman test, we would conclude that there is no significant effect of starter on acidity. This result may be due to the small sample size and the test's lack of sensitivity. The ANOVA test should provide us with more relevant insight, but this is a result worthy to be further investigated to check our methods.

TODO: What is the difference between the Friedman test and the ANOVA test? Why do we get different results? Are we ignoring batch? Are we maybe not allowed to use it for some reason?