Assignment 1

Kshitij Kavimandan, Pablo Alves, Pooja Mangal (Group 15)

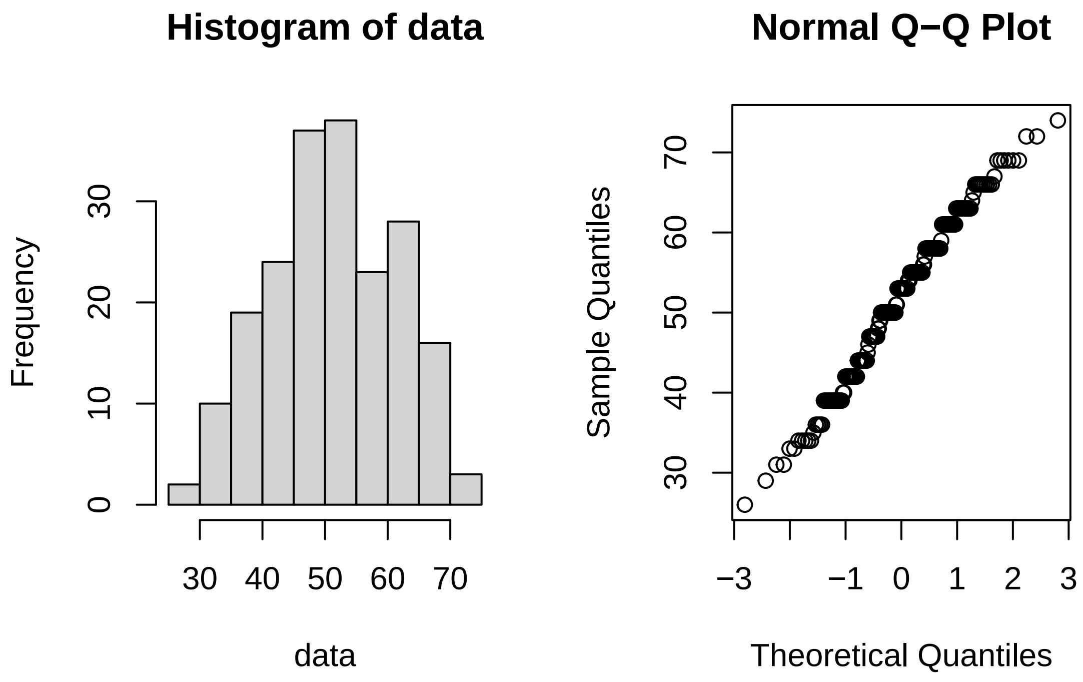
20 February 2024

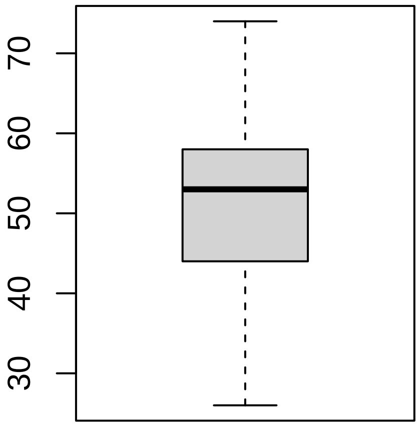
Exercise 1. Ice cream

a) Relevant plots and normality assessment:

```
# Load the dataset
ice_cream_dataset <- read.csv("Ice_cream.csv")
data = ice_cream_dataset$video
library("drcarlate")
```

```
# Do basic plots
par(mfrow=c(1,2)); hist(data); qqnorm(data); boxplot(data)
```





The histogram of the videogame scores looks normally distributed because it is roughly symetrical around 50, with values getting smaller towards its extremes. Despite this, the histogram bar for values between 60 and 65 is higher than what the corresponding analytical normal distribution should have.

Secondly, the Q-Q Plot also shows that our sample is densely populated in the center and its shape is very close to a line, which is also consistent with normality.

Finally, the boxplot is also consistent with normality, despite its median being not symetrical with its surrounding quantiles. As shown with the histogram, this can be explained with the additional values which are more frequently present in that particular region.

In short, our inspection of these basic plots suggest that this sample is normally distributed.

Because we are assuming by definition that our sample is normally distributed, we can apply the 68-95-99.7% rule. Because our sample has a mean of 51.85 and a standard deviation of 9.9, we can then see that the desired 97% confidence interval for the mean should lie inside the 6 sigma interval, which has a length of 59.9. With this rough estimation, our 97% confidence value for our mean is 51.85 +- 29.7, which is akin to the [22.15,81.55] interval.

To get the 97% CI, we need to evaluate the margin of error with the formula

margin = T(0.97)*(std(x)/sqrt(N))

where T is the t distribution with N-1 degrees of freedom, N our sample size (200)…

```
options(digits = 3)
n <- length(data)
margin <- qt(0.97, df=n-1)*sd(data)/sqrt(n)
min_mean = mean(data) - margin
max_mean = mean(data) + margin
length = (max_mean - min_mean)/2; length
```

```
## [1] 1.32
```

```
paste("97 CI: ", mean(data), " +-", length)
```

```
## [1] "97 CI:  51.85  +- 1.32428938516609"
```

Thus, the 97% CI for the mean is [50.52571,53.1749], with a length of 2.648579

In order to compute the bounded 97% CI for mu, we need to use the following formula:

```
options(digits = 3)
m_calculator<-function(p, data) sqrt(sd(data)/length(data)*(4*norminv(p+(1-p)/2)))
margin = m_calculator(p=0.97,data=data)
min_mean = mean(data) - margin
max_mean = mean(data) + margin
length = max_mean - min_mean; length
```

```
## [1] 1.31
```

```
paste("97 CI: ", mean(data)," +-", length)
```

```
## [1] "97 CI:  51.85  +- 1.31105538574369"
```

In order to compute the neccesary number of samples needed for a particular maximun length of the CI we can use the following formula

```
options(digits = 3)
n_calculator <- function(p,length,data) sd(data)*(2*norminv(p+(1-p)/2)/length)^2
samples_needed = n_calculator(p=0.97,length=3,data=data)
paste("For a margin of: ", 3 ," we need", samples_needed)
```

```
## [1] "For a margin of:  3  we need 20.7227503006415"
```

To determine the 97% bootstrap CI we can apply the following code

```
options(digits = 3)
p = 0.97
B = 100

Tstar = numeric(B)
for(i in 1:B){Tstar[i]=mean(sample(data,replace=TRUE))}

Tstar985 = quantile(Tstar,0.985)
Tstar15 = quantile(Tstar, 0.015)

c(2*mean(Tstar) - Tstar985,2*mean(Tstar)-Tstar15)
```

```
## 98.5%  1.5%
##  50.4  53.2
```

```
d = ((2*mean(Tstar)-Tstar15) - (2*mean(Tstar) - Tstar985))
paste("Length:",d)
```

```
## [1] "Length: 2.840575"
```

```
paste("Margin on each side:",d/2)
```

```
## [1] "Margin on each side: 1.4202875"
```

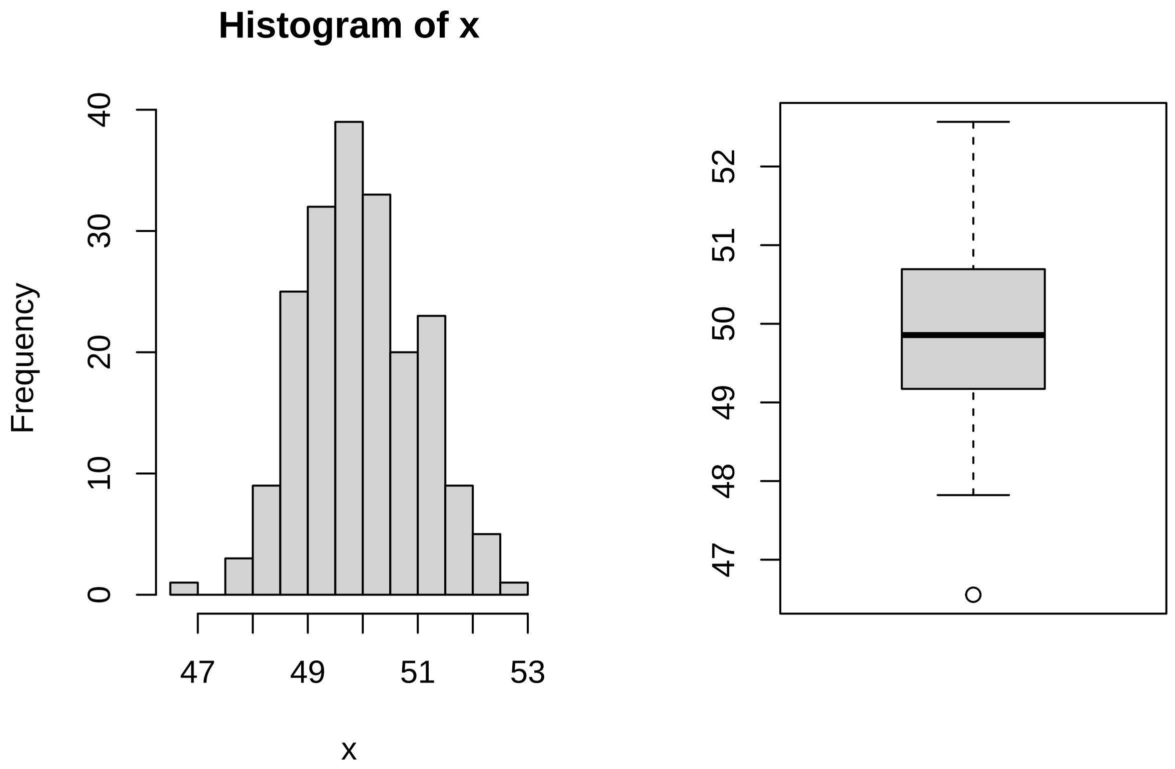
Our Bootstrap margin is bigger than our previous margins, which is consistent with the fact that the Bootstrap method does not assume normality, and thus is expected to have a bigger uncertainty on its estimation.

---

b) T-test to verify mean score:

The t-test compares the mean of the sample to the hypothesized mean and calculates a t-statistic and a p-value.

```
options(digits = 3)
mu = 50
x = rnorm(length(data), mu, 1)
par(mfrow=c(1,2))
hist(x)
boxplot(x)
```



```r
t_test_result <- t.test(x, alternative="g", mu = mu)
print(t_test_result)
```

```
##
##  One Sample t-test
##
## data:  x
## t = -0.6, df = 199, p-value = 0.7
## alternative hypothesis: true mean is greater than 50
## 95 percent confidence interval:
##  49.8  Inf
## sample estimates:
## mean of x
##        50
```

```r
t_test_result_mu_51 <- t.test(x, alternative="g", mu = 51)
t_test_result_mu_51
```

```
##
##  One Sample t-test
##
```

```
## data:  x
## t = -14, df = 199, p-value = 1
## alternative hypothesis: true mean is greater than 51
## 95 percent confidence interval:
##  49.8  Inf
## sample estimates:
## mean of x
##        50
```

- In the first test, with mu_0 = 50, the p-value is more than the significance level of 0.05 which suggests that there isn't enough evidence to reject the null hypothesis, indicating that the mean score on the video game is not significantly different from 50.
- Similarly, when testing against mu_0 = 51, the high p-value of 1 implies strong evidence in favor of the null hypothesis, meaning that the mean score is not significantly greater than 51.
- In the output, the 95% confidence interval for the first test is (49.9, infinity). This means that we are 95% confident that the true population mean lies in the above interval. The upper bound of infinity indicates that the upper limit is unbounded.
- The confidence interval for the second test is the same, as it is based on the sample mean and standard deviation. It doesn't change with the hypothesized mean mu_0.

c) Sign test and test based on ranks:

```r
options(digits = 3)
# Sign test
sign_test <- binom.test(sum(data > 50), length(data), p = 0.5, alternative = "g")
print(sign_test)
```

```
##
##  Exact binomial test
##
## data:  sum(data > 50) and length(data)
## number of successes = 108, number of trials = 200, p-value = 0.1
## alternative hypothesis: true probability of success is greater than 0.5
## 95 percent confidence interval:
##  0.479 1.000
## sample estimates:
## probability of success
##                   0.54
```

```r
# Test based on ranks (Wilcoxon signed-rank test)
wilcox_test <- wilcox.test(data, mu = 50, alternative = "g")
print(wilcox_test)
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  data
## V = 9836, p-value = 0.005
## alternative hypothesis: true location is greater than 50
```

```
# Test for fraction of scores less than 42
binom_test <- binom.test(sum(data < 42), length(data), p = 0.25, alternative = "l")
print(binom_test)
```

```
##
##  Exact binomial test
##
## data:  sum(data < 42) and length(data)
## number of successes = 31, number of trials = 200, p-value = 8e-04
## alternative hypothesis: true probability of success is less than 0.25
## 95 percent confidence interval:
##  0.000 0.203
## sample estimates:
## probability of success
##                  0.155
```

- For the median score, the sign test results in a p-value of 0.144, suggesting no significant difference between the median score and mu_0 = 50. The 95% confidence interval for the success probability of the sign test ranges from 0.479 to 1.
- Comments on comparison of the above results with the previous question (b)): *The t-test evaluates whether the mean score is significantly greater than 50, while the sign test assesses if the median score is greater than 50. Both tests yield p-values greater than 0.05, indicating that there's insufficient evidence to reject the null hypotheses. The confidence interval from the t-test provides a range of plausible values for the population mean, whereas the sign test directly assesses the proportion of observations above 50.*
- The Wilcoxon signed-rank test results in a p-value of 0.005, indicating a significant difference between the median score and mu_0 = 50. The alternative hypothesis suggests that the true location is greater than 50.
- Lastly, the binomial test to check the fraction of scores less than 42 yields a p-value of 0.0007, suggesting that the probability of success (scores less than 42) is less than 0.25.

d) Bootstrap test with test statistic T:

```
options(digits = 3)
# Bootstrap test
# Function to compute the test statistic T
compute_T <- function(x) {
  return(min(x))
}

# Bootstrap test
# Bootstrap function to calculate the test statistic T
bootstrap <- function(data) {
  min_sample <- replicate(1000, min(sample(data, 200, replace = TRUE)))
  return(min_sample)
}

# Function to perform hypothesis testing
hypothesis_test <- function(mu) {
  data_sample <- rnorm(100, mean = mu, sd = 10)  # Generate a sample from N(mu, 100)
  test_statistic <- min(data_sample)  # Compute the test statistic T

  # Bootstrap test
  bootstrap_samples <- bootstrap(data)
  p_value <- mean(bootstrap_samples <= test_statistic)

  # Return whether to reject H0 and the p-value
  return(list(reject_H0 = p_value < 0.05, p_value = p_value))
}

# Loop through mu values from 0 to 100
mu_values <- seq(0, 100, by = 1)
not_rejected_mu <- c()

for (mu in mu_values) {
  result <- hypothesis_test(mu)

  # If H0 is not rejected, print the mu value
  if (!result$reject_H0) {
    not_rejected_mu <- c(not_rejected_mu, mu)
  }
}

print(not_rejected_mu)
```

```
##  [1]  48  52  54  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71
## [20]  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
## [39]  91  92  93  94  95  96  97  98  99 100
```

The test concluded that mu values between 47 and 100 are not rejected by the null hypothesis.

```
# Apply Kolmogorov-Smirnov test
# Perform Kolmogorov-Smirnov test for different values of mu in the range [0, 100]
```

```r
mu_values_not_rejected <- c()

for (mu in seq(0, 100, by = 1)) {
  # Generate the first 100 samples from N(mu, 100)
  sample_data <- rnorm(100, mean = mu, sd = 10)

  # Compute the Kolmogorov-Smirnov statistic and p-value
  ks_test_result <- ks.test(sample_data, "pnorm", mean = mu, sd = 10)

  # Check if null hypothesis is rejected
  if (ks_test_result$p.value > 0.05) {
    mu_values_not_rejected <- c(mu_values_not_rejected, mu)
  }
}

# Display mu values for which null hypothesis is not rejected
cat("Mu values for which null hypothesis is not rejected:", mu_values_not_rejected, "\n")
```

```
## Mu values for which null hypothesis is not rejected: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

Surprisingly, the KS test resulted in every value of mu being not rejected by the null hypothesis. The KS test may not have worked as expected in this scenario due to the nature of the data or incorrect assumptions about the distribution.

e) Tests for male and female students:

```r
options(digits = 3)
data <- read.csv("Ice_cream.csv")

# Separate scores for male and female students
male_scores <- data$video[data$female == 0]
female_scores <- data$video[data$female == 1]
# Perform two-sample t-test
t_test_gender <- t.test(male_scores, female_scores, alternative = "g"); t_test_gender
```

```
##
##  Welch Two Sample t-test
##
## data:  male_scores and female_scores
## t = 2, df = 177, p-value = 0.04
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.186   Inf
## sample estimates:
## mean of x mean of y
##      53.2      50.7
```

The t-test yields a p-value of 0.04, indicating that there is a significant difference in mean scores between male and female students. The confidence interval suggests that the mean score for male students is likely higher than for female students.

```
# Perform Mann-Whitney test
mannwhitney_test <- wilcox.test(male_scores, female_scores,
alternative = "g"); mannwhitney_test
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  male_scores and female_scores
## W = 5748, p-value = 0.03
## alternative hypothesis: true location shift is greater than 0
```

The Mann-Whitney test also indicates a significant difference in the distribution of scores between male and female students, with a p-value of 0.03. This aligns with the findings of the t-test.

```
# Perform Kolmogorov-Smirnov test
ks_test_gender <- ks.test(male_scores, female_scores); ks_test_gender
```

```
##
##  Exact two-sample Kolmogorov-Smirnov test
##
## data:  male_scores and female_scores
## D = 0.2, p-value = 0.07
## alternative hypothesis: two-sided
```

The Kolmogorov-Smirnov test yields a p-value of 0.07, which is slightly higher than the significance threshold of 0.05. This suggests that there may be a difference in the distribution of scores between male and female students, but it's not as conclusive as the other tests.

f) Correlation and comparison between video game and puzzle scores:

```
options(digits = 3)
# Investigate correlation
correlation <- cor(data$video, data$puzzle); correlation
```

```
## [1] 0.465
```

This suggests that there is some degree of association between scores on the video game and scores on the puzzle, although the correlation is not extremely strong.

```
# Test if puzzle scores are higher than video game scores
wilcox_test_puzzle <- wilcox.test(data$puzzle, data$video, alternative = "g"); wilcox_test_puz
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data$puzzle and data$video
## W = 21044, p-value = 0.2
## alternative hypothesis: true location shift is greater than 0
```

Despite the moderate positive correlation between the video and puzzle scores, the statistical test does not provide strong evidence to support the hypothesis that puzzle scores are consistently higher than video game scores.

Exercise 2. Hemoglobin in trout **a)** R-code for randomization process: a) R-code for randomization process:

```
options(digits = 3)
# Load the data
hemoglobin_data <- read.table("hemoglobin.txt", header = TRUE)
# Define the number of fishes and combinations of rate and method
num_fishes <- 80
method_levels <- sample(rep(c("A","B"), 40))
rate_levels <- sample(rep(1:4, 20))

df <- data.frame(hemoglobin_data$hemoglobin,rate_levels, method_levels)
```

b) Two-way ANOVA:

```
options(digits = 3)
# Perform two-way ANOVA
rate=as.factor(hemoglobin_data$rate);
method=as.factor(hemoglobin_data$method);
# Print ANOVA table
hemoglobin_aov=lm(hemoglobin_data$hemoglobin~method*rate); anova(hemoglobin_aov)
```

```
## Analysis of Variance Table
##
## Response: hemoglobin_data$hemoglobin
##             Df Sum Sq Mean Sq F value  Pr(>F)
## method       1    2.4    2.42    1.56    0.22
## rate         3   90.6   30.19   19.47 2.4e-09 ***
## method:rate  3    4.9    1.62    1.05    0.38
## Residuals   72  111.6    1.55
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The interaction term “method:rate” has a non-significant F-value (F = 1.05, p = 0.38). This suggests that the combined effect of the method and rate on hemoglobin levels is not significantly different from what would be expected based on the individual effects of method and rate alone.

The rate has a highly significant impact on hemoglobin levels (F = 19.47, p < 0.001).

c) Influence of factors and combination yielding highest hemoglobin: options(digits = 3)

```
options(digits = 3)
# Perform two-way ANOVA
rate=as.factor(hemoglobin_data$rate);
method=as.factor(hemoglobin_data$method);
# Print ANOVA table
hemoglobin_aov=lm(hemoglobin_data$hemoglobin~method + rate); anova(hemoglobin_aov)
```

```
## Analysis of Variance Table
##
## Response: hemoglobin_data$hemoglobin
##           Df Sum Sq Mean Sq F value Pr(>F)
## method     1    2.4    2.42    1.55   0.22
## rate       3   90.6   30.19   19.43  2e-09 ***
## Residuals 75  116.5    1.55
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

So the 2nd factor (rate) has a significant effect in the additive model. This is evident from the significant F-value and very small p-value associated with the “Rate” factor.

```
# Calculate means for rate and method
mean_hemoglobin <- aggregate(hemoglobin ~ rate + method,
data = hemoglobin_data, FUN = mean)
# Find combination yielding highest hemoglobin
max_hemoglobin <- mean_hemoglobin[which.max(mean_hemoglobin$hemoglobin), ]
```

```
# Estimate mean hemoglobin value for rate 3 by using method A
mean_hemoglobin_rate3_methodA <-
mean_hemoglobin$hemoglobin[which(mean_hemoglobin$rate == 3 &
mean_hemoglobin$method == "A")]
```

```
# Estimate mean hemoglobin value for each rate
mean_hemoglobin_rate <- aggregate(hemoglobin ~ rate,
data = hemoglobin_data, FUN = mean); mean_hemoglobin_rate
```

```
##   rate hemoglobin
## 1    1       6.97
## 2    2       9.73
## 3    3       9.38
## 4    4       8.86
```

The hemoglobin rate was highest with rate 5 corresponding to the value of 2

d) One-way ANOVA:

```
options(digits = 3)
# Perform one-way ANOVA ignoring the variable method
anova_result_rate <- anova(lm(hemoglobin ~ rate, data = hemoglobin_data))
# Print ANOVA table
print(anova_result_rate)
```

```
## Analysis of Variance Table
##
## Response: hemoglobin
##           Df Sum Sq Mean Sq F value  Pr(>F)
## rate       1   27.9   27.93      12 0.00087 ***
## Residuals 78  181.6    2.33
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

The null hypothesis that hemoglobin is the same for all rates is rejected, suggesting that the rate of treatment administration has a notable impact on hemoglobin levels.

```
hemoglobin_by_rate <- tapply(hemoglobin_data$hemoglobin, hemoglobin_data$rate, mean)
hemoglobin_by_rate
```

```
##    1    2    3    4
## 6.97 9.73 9.38 8.86
```

The one-way ANOVA test provides valuable insights into the relationship between treatment rate and hemoglobin levels in this dataset. Therefore, it is deemed useful for analyzing the data and drawing meaningful conclusions.

e) Kruskal-Wallis test:

```
options(digits = 3)
# Perform Kruskal-Wallis test
attach(hemoglobin_data);
```

```
## The following objects are masked _by_ .GlobalEnv:
##
##     method, rate
```

```
kruskal_test_result <- kruskal.test(hemoglobin, rate)
# Print Kruskal-Wallis test result
print(kruskal_test_result)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  hemoglobin and rate
## Kruskal-Wallis chi-squared = 34, df = 3, p-value = 2e-07
```

The small p-value (2e-07) indicates strong evidence against the null hypothesis, suggesting that there are significant differences in hemoglobin levels among the different rates. The one-way ANOVA rejects the null hypothesis, indicating that the rate of treatment administration has a notable impact on hemoglobin levels.The Kruskal-Wallis test also rejects the null hypothesis, suggesting that there are significant differences in hemoglobin levels among the different rates of treatment administration. While both tests arrive at the same conclusion in this case, differences may arise when the assumptions of the one-way ANOVA are violated, such as in the presence of non-normal or skewed data. In such cases, the Kruskal-Wallis test, being non-parametric, might be more appropriate and may provide more robust results.

Exercise 3. Sour cream

Read the data Columns: acidity batch position starter

```
data <- read.table(file = "cream.txt", header = TRUE, sep="", dec=".")
```

a) Analyzing the data in a three-way experiment without interactions:

I decide to do 3-way additive ANOVA test, for it does not assume interaction (notice the + and not *)

```
model <- aov(acidity ~ starter + batch + position, data=data)
summary(model)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## starter      1    2.0    1.98    0.64   0.43
## batch        1    4.9    4.85    1.57   0.22
## position     1    0.2    0.23    0.07   0.79
## Residuals   21   64.7    3.08
```

Our test is intended to test the isolated effect of different factors Because for our starter factor we have a Pr(>F) of 0.431, which is big » 0.05, we cannot conclude that there is a significant difference between the effects of starter 1 and start 2 on acidity

Instead, we could do a two-sample t-test to see if the average acidity changes significantly with starter 1 or 2, which shows a p-value of 0.8482, and thus we can't reject H0 so in summary we can't say that there is a significant difference between the effects of starter 1 and starter 2 on acidity?

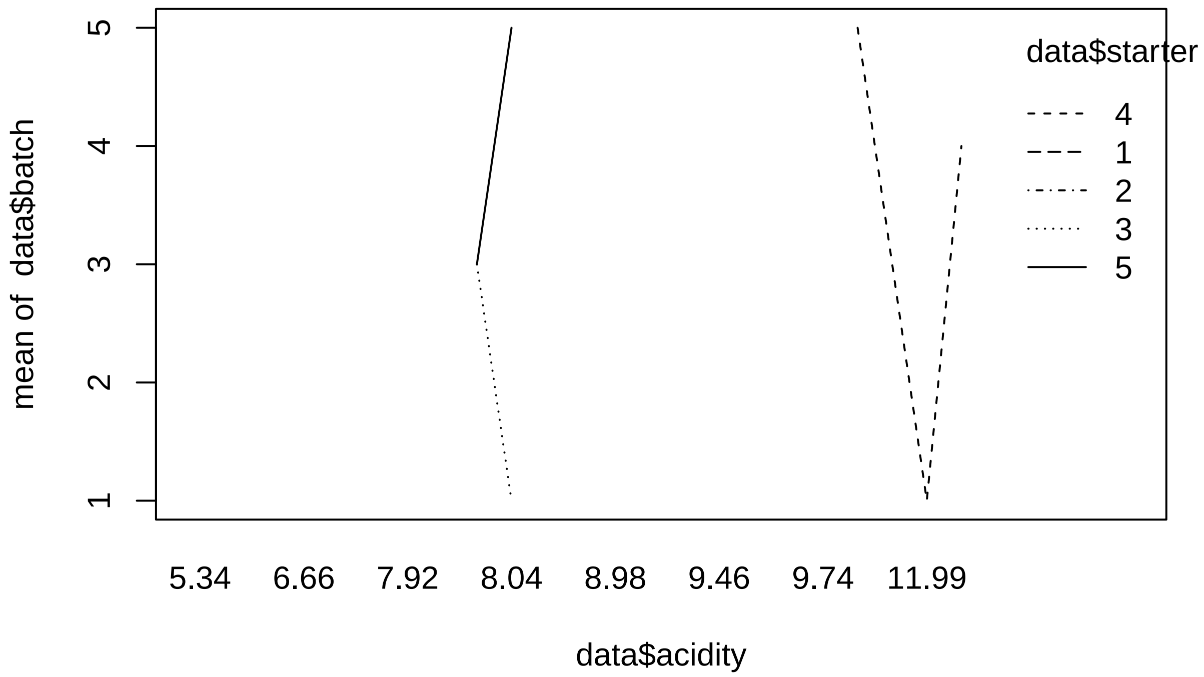
```
sample1 = data[data$starte == 1, ]
sample2 = data[data$starte == 2, ]
t.test(sample1$acidity, sample2$acidity, var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data:  sample1$acidity and sample2$acidity
## t = 0.2, df = 8, p-value = 0.8
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.6  1.9
## sample estimates:
## mean of x mean of y
##      8.42      8.27
```

EXTRA If we really wanted to go further we could also do an interaction plot, then we can see that for lines of starter 1 and 2

Like this:

```
interaction.plot(data$acidity,data$starter,data$batch)
```



b) Removing insignificant block variables and performing ANOVA:

To find insignificant block variables and thus remove them we do ANOVA again, but this time without starter

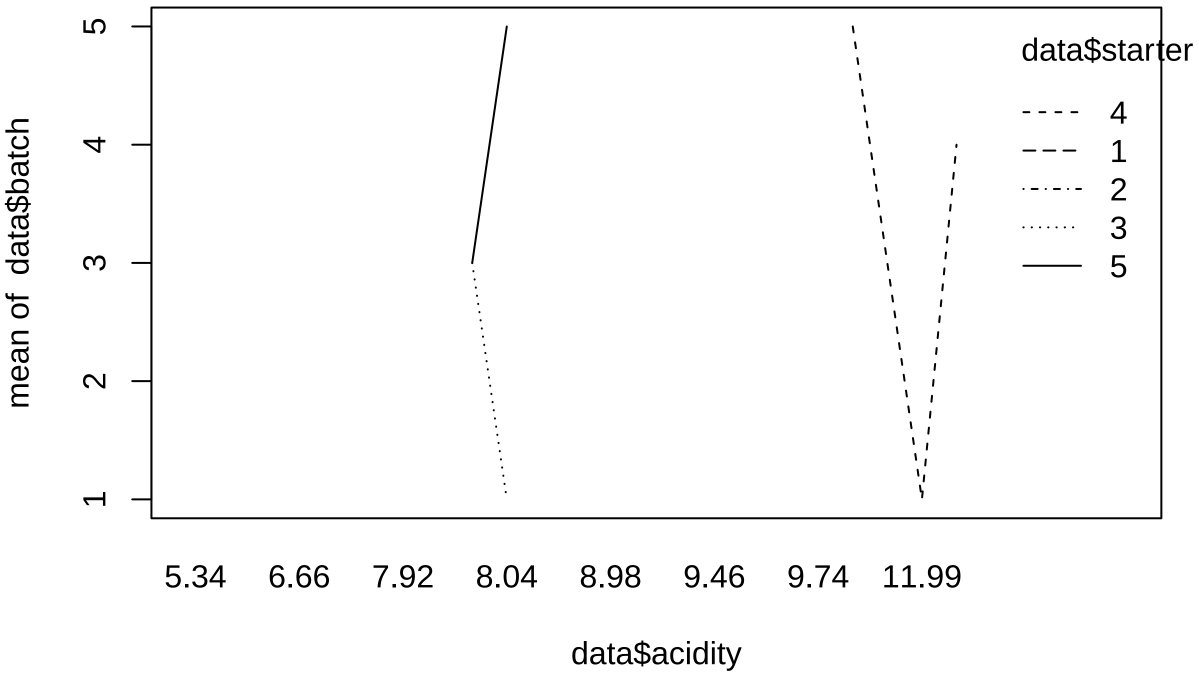
```
model2 <- aov(acidity ~ batch + position, data=data)
summary(model2)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## batch        1    4.9    4.85    1.60   0.22
## position     1    0.2    0.23    0.07   0.79
## Residuals   22   66.7    3.03
```

Here both variables have still a big p, so while technically all variables could be discarded, which would be absurd, we could just remove position, for its p-value 0.788 is greater than the 0.222 value of batch, despite both being big.

Thus, to find which starters lead to significantly different acidity we could to an interaction plot like this

```r
interaction.plot(data$acidity,data$starter,data$batch)
```



c) Applying the Friedman test:

Because we have severall batches for each starter position and not one We can't use the Friedman test

However, if we modify our data by averaging the acidity of every batch to have one element per starter sample, we could do it. Alternatively, we could also do the Friedman test five times, one for every batch

Using https://search.r-project.org/R/refmans/stats/html/friedman.test.html The syntax should be something like this (we also need to filter the extra samples)

```
# friedman_test(data, acidity ~ starter)
# friedman.test(acidity ~ starter, data = data)
```

d) Performing a mixed effects analysis:

Load library

```
#library("devtools");
#library("lme4")
```

Getting error Error: Failed to install 'lme4' from GitHub: Could not find tools necessary to compile a package

Syntax will be something like this https://www.rdocumentation.org/packages/lme4/versions/1.1-35.1/topics/lmer

```
# model <- lme(acidity ~ starter + (1|position), data = data)
```