Business Statistics End of Term Assessment IB94X0 2021-2022 #1

2135320

Contents

```r
#Loading the required packages
library(tidyverse)
library(dplyr)
library(Rmisc)
library(Hmisc)
library(emmeans)
library(grid)
library(gridExtra)
library(knitr)
library(car)
library(tinytex)
options(width=100)
```

Question 1

Tutoring Data Dictionary

Variable	Description
student_ID	The unique IDs for each student at school
tutoring	Shows which student received tutoring (TRUE = Tutored, FALSE = Not tutored)
absences	The proportions(%) of classes missed by each student
score.t1	The test score at the beginning of the academic year for each student (before implementing buddying scheme)
score.t2	The test score at the end of the academic year for each student (after implementing buddying scheme)

Part 1: Analysis

Section 1: Data Preperation

```r
#Reading the file into R
tutoring_data <- read_csv("tutoring_test_data.txt")
```

```r
#Checking the structures of data to check for necessary amending
str(tutoring_data)
```

```
## spec_tbl_df [202 x 5] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ student_ID: num [1:202] 56 40 63 108 72 10 15 42 152 104 ...
##  $ tutoring  : logi [1:202] TRUE TRUE TRUE FALSE TRUE TRUE ...
##  $ absences  : num [1:202] 3.6 2.4 3.6 4.8 4.8 ...
##  $ score.t1  : num [1:202] 70.4 67.8 40.2 75.3 31 ...
##  $ score.t2  : num [1:202] 75.2 75.9 45.5 79.7 31.7 ...
```

```
##  - attr(*, "spec")=
##   .. cols(
##   ..   student_ID = col_double(),
##   ..   tutoring = col_logical(),
##   ..   absences = col_double(),
##   ..   score.t1 = col_double(),
##   ..   score.t2 = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

The datatype for tutoring variable needs to be changed to factor

```
#Checking for outliers and NA values
summary(tutoring_data)
```

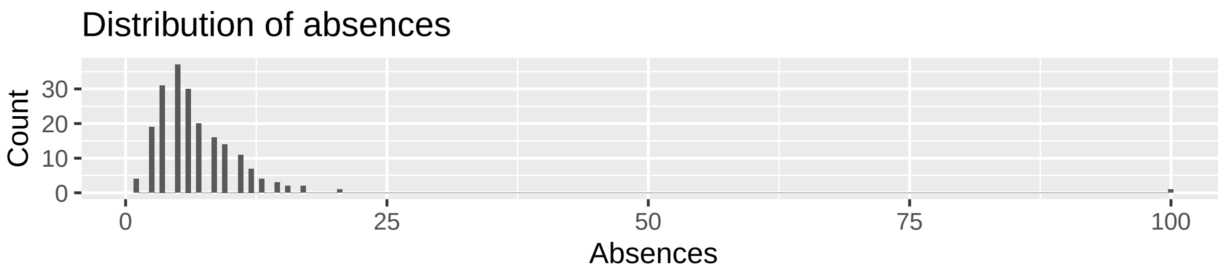
```
##    student_ID      tutoring          absences          score.t1        score.t2
##  Min.   :  1.00   Mode :logical   Min.   :  1.200   Min.   :17.31   Min.   : 11.92
##  1st Qu.: 51.25   FALSE:101       1st Qu.:  3.600   1st Qu.:46.56   1st Qu.: 46.47
##  Median :101.50   TRUE :101       Median :  6.000   Median :53.96   Median : 55.26
##  Mean   :101.50                   Mean   :  7.012   Mean   :53.89   Mean   : 56.23
##  3rd Qu.:151.75                   3rd Qu.:  8.400   3rd Qu.:62.83   3rd Qu.: 65.01
##  Max.   :202.00                   Max.   :100.000   Max.   :88.46   Max.   :200.00
##                                                                     NA's   :1
```

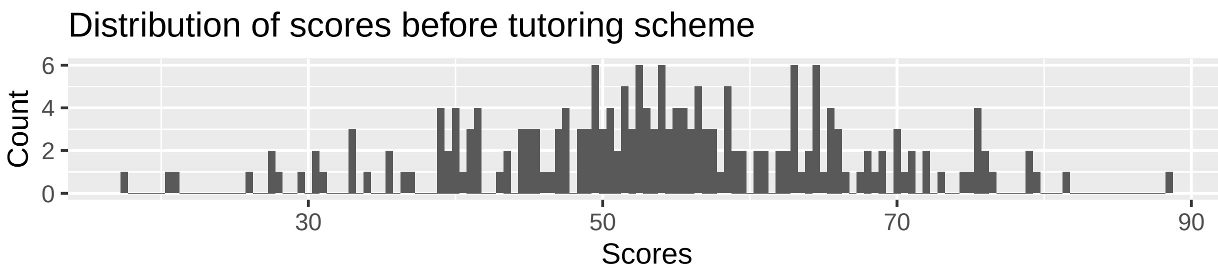
There seems to be some outliers and NA values present

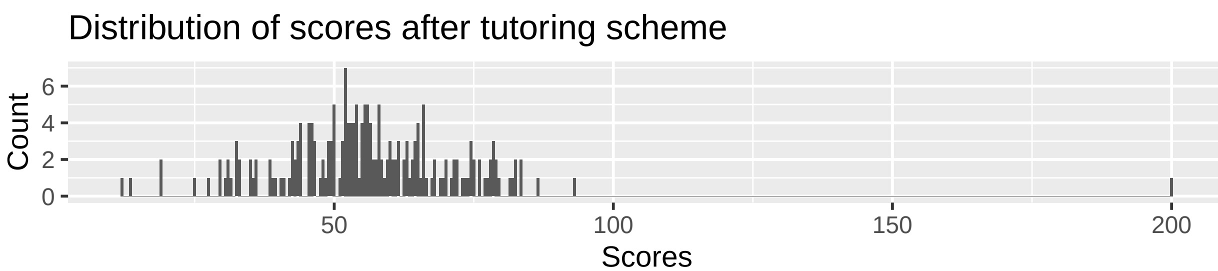
```
#Checking for duplicate entries of student ID and filtering for unique IDs
tutoring_data_new <- tutoring_data %>%
                     group_by(student_ID) %>%
                     filter(!duplicated(student_ID)) %>%
                     ungroup(student_ID)

#Checking for number of rows removed
nrow(tutoring_data) - nrow(tutoring_data_new) #No duplicates were found for this data
```

```
## [1] 0
```

```
#Checking the distribution of each variable
grid.arrange(ggplot(tutoring_data_new, aes(absences)) +
             geom_histogram(binwidth = 0.5) +
             labs(x = "Absences", y = "Count", title = "Distribution of absences"),

             ggplot(tutoring_data_new, aes(score.t1)) +
             geom_histogram(binwidth = 0.5) +
             labs(x = "Scores", y = "Count", title = "Distribution of scores before tutoring scheme"),

             ggplot(tutoring_data_new, aes(score.t2)) +
             geom_histogram(binwidth = 0.5) +
             labs(x = "Scores", y = "Count", title = "Distribution of scores after tutoring scheme")
)
```







The distributions confirm the presence of some outliers in score.t2 and absences

```
#Preparing the data based on observation of structures and visualizations
tutoring_data_new$tutoring <- factor(tutoring_data_new$tutoring,
                                     levels = c("FALSE", "TRUE"),
                                     labels = c("Non-Tutored", "Tutored"))  #Converting into categorica

levels(tutoring_data_new$tutoring)  #Checking the levels of the factors
```

```
## [1] "Non-Tutored" "Tutored"
```

```
tutoring_data_new <- tutoring_data_new %>%
                     filter(score.t2 <= 100 & absences < 25 & !is.na(score.t2)) %>% #Removing NA values
                     arrange(student_ID)  #Arranging data by student ID

str(tutoring_data_new) #Checking structure again to see the corrections
```

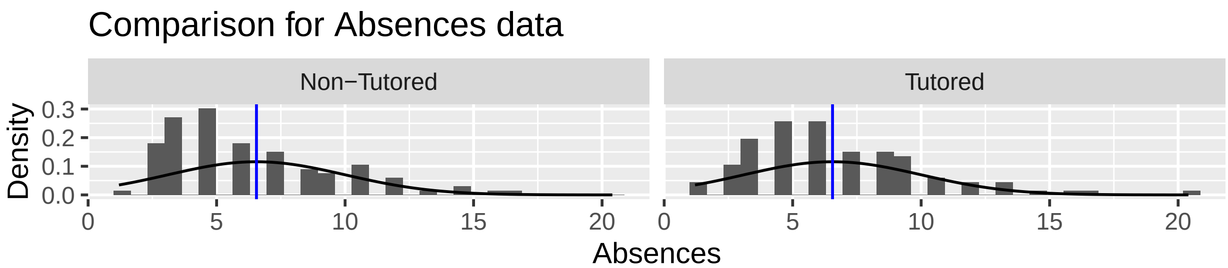
```
## tibble [200 x 5] (S3: tbl_df/tbl/data.frame)
##  $ student_ID: num [1:200] 1 2 3 4 5 6 7 8 9 10 ...
##  $ tutoring  : Factor w/ 2 levels "Non-Tutored",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ absences  : num [1:200] 4.8 6 20.4 7.2 7.2 ...
##  $ score.t1  : num [1:200] 39.4 51.8 44.6 56.7 43.4 ...
##  $ score.t2  : num [1:200] 30.8 52.1 53.1 55.3 55 ...
```

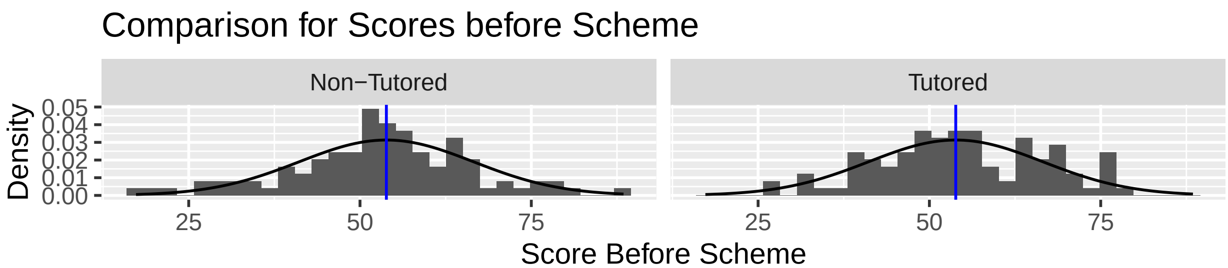
```
summary(tutoring_data_new)
```

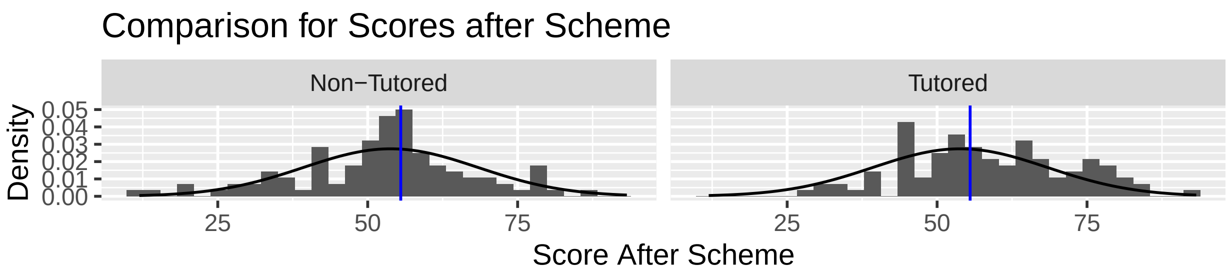
```
##    student_ID          tutoring      absences         score.t1        score.t2
##  Min.   :  1.00   Non-Tutored:100   Min.   : 1.200   Min.   :17.31   Min.   :11.92
##  1st Qu.: 50.75   Tutored    :100   1st Qu.: 3.600   1st Qu.:46.30   1st Qu.:46.45
```

```
##  Median :100.50                          Median : 6.000   Median :53.96   Median :55.20
##  Mean   :100.50                          Mean   : 6.552   Mean   :53.85   Mean   :55.51
##  3rd Qu.:150.25                          3rd Qu.: 8.400   3rd Qu.:62.87   3rd Qu.:64.88
##  Max.   :200.00                          Max.   :20.400   Max.   :88.46   Max.   :93.21
```

```
#Producing summary statistics for each item
tutoring_summary <- tutoring_data_new %>%
                    summarise(mean_absence = mean(absences), sd_absence = sd(absences),
                              mean_score.t1 = mean(score.t1), sd_score.t1 = sd(score.t1),
                              mean_score.t2 = mean(score.t2), sd_score.t2 = sd(score.t2))
  #Assigning the values to individual variables
  mean_absence <- tutoring_summary$mean_absence
  mean_score.t1 <- tutoring_summary$mean_score.t1
  mean_score.t2 <- tutoring_summary$mean_score.t2

  sd_absence <- tutoring_summary$sd_absence
  sd_score.t1 <- tutoring_summary$sd_score.t1
  sd_score.t2 <- tutoring_summary$sd_score.t2

#Comparing each variable to a normal distribution for each category
grid.arrange((ggplot(tutoring_data_new, aes(x=absences)) +
              geom_histogram(aes(y=..density..)) +
              stat_function(fun=function(x) {dnorm(x, mean=mean_absence, sd=sd_absence)}) +
              geom_vline(xintercept = mean_absence, color = "Blue") +
              facet_wrap(~tutoring) +
              labs(x="Absences", y="Density", title = "Comparison for Absences data")),

              (ggplot(tutoring_data_new, aes(x=score.t1)) +
              geom_histogram(aes(y=..density..)) +
              stat_function(fun=function(x) {dnorm(x, mean=mean_score.t1, sd=sd_score.t1)}) +
              geom_vline(xintercept = mean_score.t1, color = "Blue") +
              facet_wrap(~tutoring) +
              labs(x="Score Before Scheme", y="Density", title = "Comparison for Scores before Scheme")

              (ggplot(tutoring_data_new, aes(x=score.t2)) +
              geom_histogram(aes(y=..density..)) +
              stat_function(fun=function(x) {dnorm(x, mean=mean_score.t1, sd=sd_score.t2)}) +
              geom_vline(xintercept = mean_score.t2, color = "Blue") +
              facet_wrap(~tutoring) +
              labs(x="Score After Scheme", y="Density", title = "Comparison for Scores after Scheme")),

              nrow = 3)
```







The distributions seem fairly normal except the distribution for absence data seems slightly positively skewed. We proceed with our analysis

Section 2: Checking whether the students allocated to the tutored and non-tutored groups had similar or different average test scores before the tutoring scheme began.

NHST Approach

```
#Performing t-test
( scores.before.t.test <- t.test(score.t1 ~ tutoring, tutoring_data_new) )
```

```
##
##  Welch Two Sample t-test
##
## data:  score.t1 by tutoring
## t = -1.0467, df = 196.54, p-value = 0.2965
## alternative hypothesis: true difference in means between group Non-Tutored and group Tutored is not e
## 95 percent confidence interval:
##  -5.433861  1.665720
## sample estimates:
## mean in group Non-Tutored     mean in group Tutored
##                  52.90345                  54.78753
```

Estimation Approach

```
#Creating linear model
scores.before.lm <- lm(score.t1 ~ tutoring, tutoring_data_new)

#Extracting means and 95% confidence intervals
```

```r
scores.before.emm <- emmeans(scores.before.lm, ~tutoring)
kable(scores.before.emm, caption = "Mean scores and 95% CIs ")
```

Table 2: Mean scores and 95% CIs

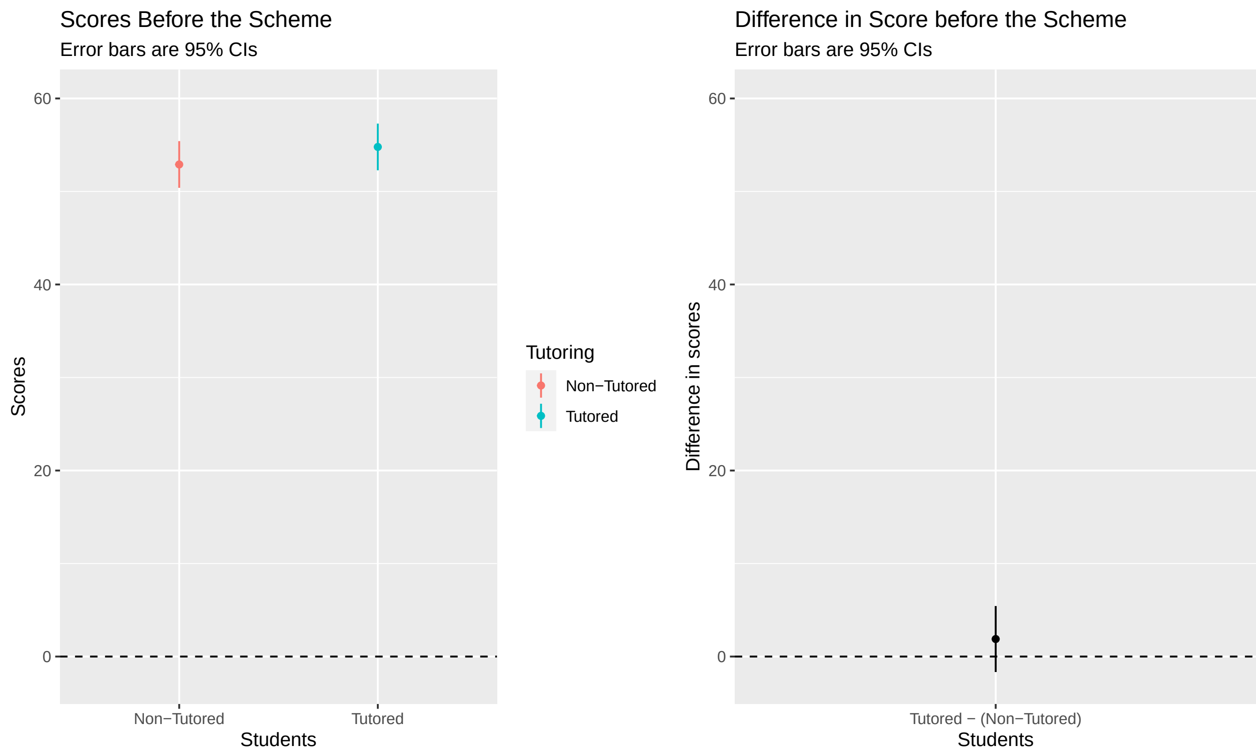
tutoring	emmean	SE	df	lower.CL	upper.CL
Non-Tutored	52.90345	1.272791	198	50.39349	55.41342
Tutored	54.78753	1.272791	198	52.27756	57.29749

```r
#Estimating the differences between means
scores.before.contrast <- confint(pairs(scores.before.emm, reverse = TRUE))
kable(scores.before.contrast, caption = "Differences between the mean scores before the scheme")
```

Table 3: Differences between the mean scores before the scheme

contrast	estimate	SE	df	lower.CL	upper.CL
Tutored - (Non-Tutored)	1.884071	1.799998	198	-1.665557	5.433699

```r
#Visualizing the estimations
avg.scores.before <- grid.arrange(
                 ggplot(summary(scores.before.emm), aes(y=emmean, x=tutoring, ymin=lower.CL, ymax=upper
                   geom_point() +
                   geom_linerange() +
                   geom_hline(yintercept=0, lty=2) +
                   labs(x="Students", y="Scores", color = "Tutoring",
                        subtitle="Error bars are 95% CIs", title="Scores Before the Scheme") +
                   ylim(-2, 60),

                 ggplot(scores.before.contrast, aes(y=estimate, x=contrast, ymin=lower.CL, ymax=upper.C
                   geom_point() +
                   geom_linerange() +
                   labs(x="Students", y="Difference in scores",
                        subtitle="Error bars are 95% CIs", title="Difference in Score before the Scheme
                   geom_hline(yintercept=0, lty=2) +
                   ylim(-2, 60),

                   ncol=2, widths = c(2,1.75))
```



Section 3: Checking if the tutored and non-tutored students had similar or different rates of absences on average

NHST Approach

```
#Performing t.test
( absences.t.test <- t.test(absences ~ tutoring, tutoring_data_new) )
```

```
##
##  Welch Two Sample t-test
##
## data:  absences by tutoring
## t = -0.98528, df = 197.6, p-value = 0.3257
## alternative hypothesis: true difference in means between group Non-Tutored and group Tutored is not
## 95 percent confidence interval:
##  -1.440721  0.480721
## sample estimates:
## mean in group Non-Tutored     mean in group Tutored
##                     6.312                     6.792
```

Estimation Approach

```
#Creating the linear model
absences.lm <- lm(absences ~ tutoring, tutoring_data_new)
summary(absences.lm)
```

```
##
## Call:
## lm(formula = absences ~ tutoring, data = tutoring_data_new)
##
## Residuals:
##    Min     1Q Median     3Q    Max
```

```
## -5.592 -2.712 -0.792  1.728 13.608
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        6.3120     0.3445  18.323   <2e-16 ***
## tutoringTutored    0.4800     0.4872   0.985    0.326
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.445 on 198 degrees of freedom
## Multiple R-squared:  0.004879,   Adjusted R-squared:  -0.0001469
## F-statistic: 0.9708 on 1 and 198 DF,  p-value: 0.3257
```

```r
#Checking for interactivity
absences.lm.scores <- lm(absences ~ tutoring + score.t1 + score.t2, tutoring_data_new)
vif(absences.lm.scores)
```

```
## tutoring score.t1 score.t2
## 1.133473 6.007094 6.248181
```

The vif scores for both score.t1 and score.t2 seem similar but are over 5. Which means the additional complexity of model is warranted to be investigated

```r
#Creating multiple regression model including score
absences.lm.score.t1 <- lm(absences ~ tutoring + score.t1, tutoring_data_new)
summary(absences.lm.score.t1)
```

```
##
## Call:
## lm(formula = absences ~ tutoring + score.t1, data = tutoring_data_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.2682 -2.2693 -0.6267  1.8597 12.6464
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      11.31919    1.00901  11.218  < 2e-16 ***
## tutoringTutored   0.65832    0.45883   1.435    0.153
## score.t1         -0.09465    0.01807  -5.239 4.13e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.235 on 197 degrees of freedom
## Multiple R-squared:  0.1266, Adjusted R-squared:  0.1177
## F-statistic: 14.27 on 2 and 197 DF,  p-value: 1.624e-06
```

```r
#Performing anova test to check whether the additional complexity improves the model
anova(absences.lm.scores, absences.lm.score.t1) #Inclusion of both score.t1 and score.t2 doesn't improv
```

```
## Analysis of Variance Table
##
## Model 1: absences ~ tutoring + score.t1 + score.t2
## Model 2: absences ~ tutoring + score.t1
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    196 2056.0
## 2    197 2062.3 -1   -6.2715 0.5979 0.4403
```

```r
anova(absences.lm, absences.lm.score.t1) #Inclusion of score.t1 does improve the model
```

```
## Analysis of Variance Table
##
## Model 1: absences ~ tutoring
## Model 2: absences ~ tutoring + score.t1
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1    198 2349.6
## 2    197 2062.3  1    287.34 27.448 4.13e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#Extracting means and 95% confidence intervals
absences.emm <- emmeans(absences.lm, ~tutoring)
kable(absences.emm, caption = "Mean absence rates and their 95% CIs by tutoring")
```

Table 4: Mean absence rates and their 95% CIs by tutoring

tutoring	emmean	SE	df	lower.CL	upper.CL
Non-Tutored	6.312	0.3444817	198	5.632676	6.991324
Tutored	6.792	0.3444817	198	6.112676	7.471324

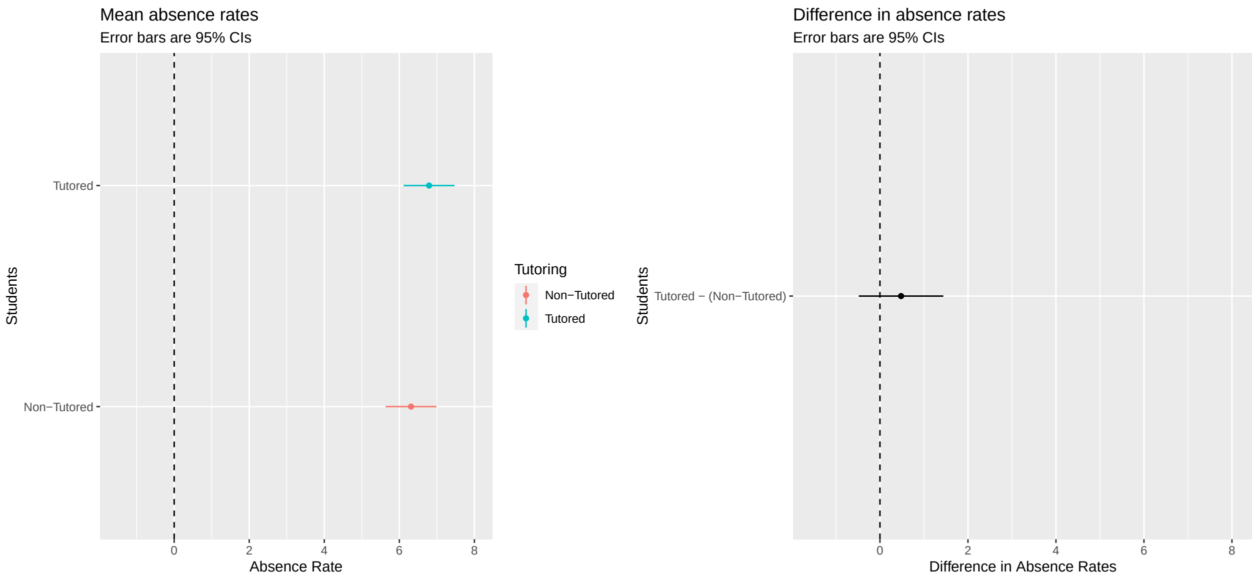
```r
#Estimating the difference in means and 95% confidence interval
absences.contrast <- confint(pairs(absences.emm, reverse = TRUE))
kable(absences.contrast, caption = "Differences in the absence rates by tutoring")
```

Table 5: Differences in the absence rates by tutoring

contrast	estimate	SE	df	lower.CL	upper.CL
Tutored - (Non-Tutored)	0.48	0.4871707	198	-0.4807091	1.440709

```r
#Visualizing the estimations
grid2.1 <- grid.arrange(
             ggplot(summary(absences.emm), aes(x = tutoring, y = emmean, ymin = lower.CL, ymax = upper.(
             geom_point() +
             geom_linerange() +
             labs(y = "Absence Rate", x = "Students", color = "Tutoring",
                  subtitle = "Error bars are 95% CIs", title = "Mean absence rates") +
             geom_hline(yintercept=0, lty=2) +
             ylim(-1.5, 8) +
             coord_flip(),

             ggplot(absences.contrast, aes(y=estimate, x=contrast, ymin=lower.CL, ymax=upper.CL)) +
             geom_point() +
             geom_linerange() +
             labs(y="Difference in Absence Rates", x="Students",
                  subtitle="Error bars are 95% CIs", title="Difference in absence rates") +
             geom_hline(yintercept=0, lty=2) +
             ylim(-1.5, 8) +
             coord_flip(),
```

```
             ncol=2)
```




```
#Extracting means and 95% confidence intervals from the model including main effect of score.t1
absences.emm.scores <- emmeans(absences.lm.score.t1, ~tutoring)
kable(absences.emm.scores, caption = "Mean absence rates and their 95% CIs by tutoring and score main e
```

Table 6: Mean absence rates and their 95% CIs by tutoring and score main effects

tutoring	emmean	SE	df	lower.CL	upper.CL
Non-Tutored	6.222839	0.3239965	197	5.583892	6.861785
Tutored	6.881161	0.3239965	197	6.242215	7.520108

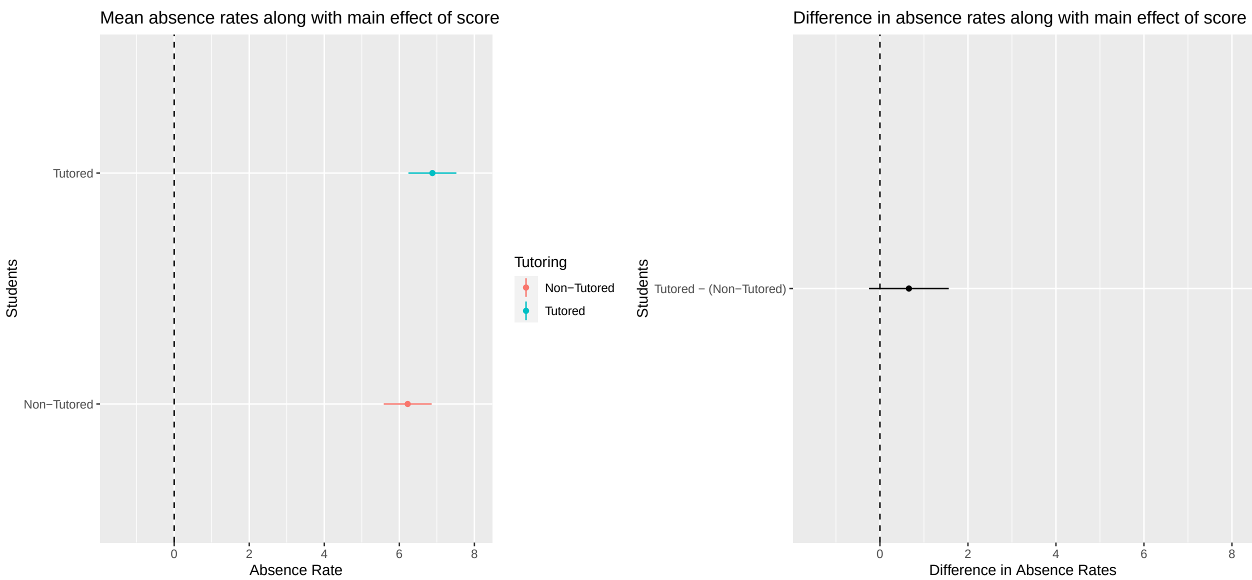
```
#Estimating the difference in means and 95% confidence interval from the model including main effect of
absences.contrast.scores <- confint(pairs(absences.emm.scores, reverse = TRUE))
kable(absences.contrast.scores, caption = "Differences in the absence rates by tutoring and score main e
```

Table 7: Differences in the absence rates by tutoring and score main effects

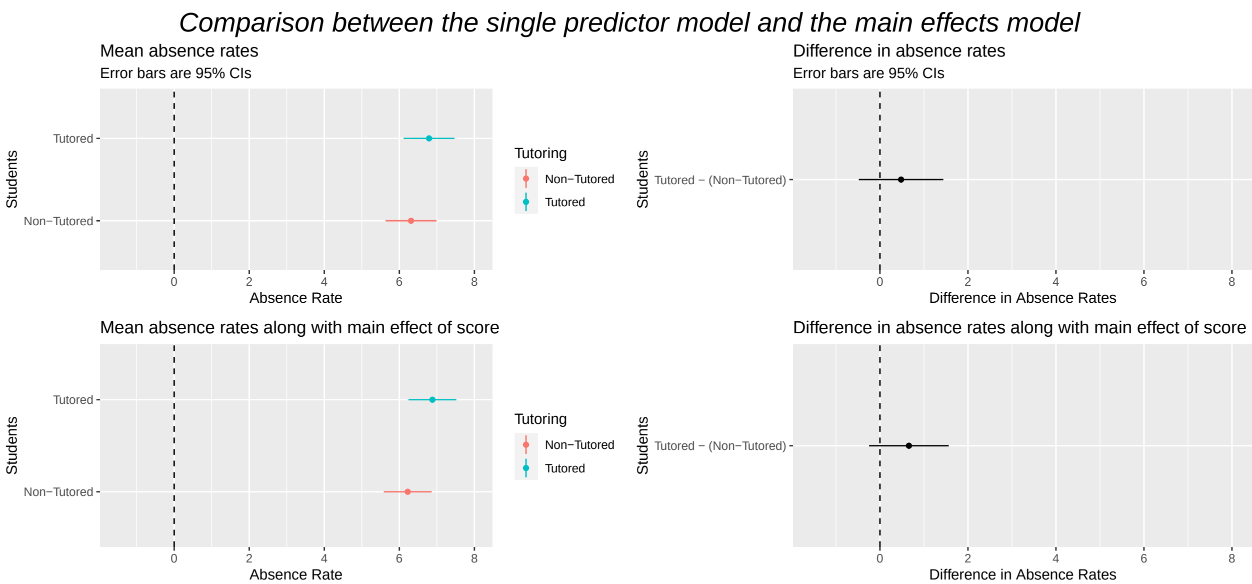
contrast	estimate	SE	df	lower.CL	upper.CL
Tutored - (Non-Tutored)	0.6583228	0.458832	197	-0.2465301	1.563176

```
grid2.2 <- grid.arrange(
             ggplot(summary(absences.emm.scores), aes(x = tutoring, y = emmean, ymin = lower.CL, ymax =
             geom_point() +
             geom_linerange() +
             labs(y = "Absence Rate", x = "Students", color = "Tutoring",
                  title = "Mean absence rates along with main effect of score") +
             geom_hline(yintercept=0, lty=2) +
             ylim(-1.5, 8) +
             coord_flip(),
```

```
ggplot(absences.contrast.scores, aes(y=estimate, x=contrast, ymin=lower.CL, ymax=upper.CL)
geom_point() +
geom_linerange() +
labs(y="Difference in Absence Rates", x="Students",
     title="Difference in absence rates along with main effect of score") +
geom_hline(yintercept=0, lty=2) +
ylim(-1.5, 8) +
coord_flip(),

ncol=2)
```




```
grid.arrange(grid2.1, grid2.2, nrow = 2, top = textGrob("Comparison between the single predictor model a
```



Section 4: Checking if the tutored students show an increase in their scores compared to the students who did not receive tutoring

Data Preparation

```
#Creating a new column to find score differences
tutoring_data_new <- tutoring_data_new %>%
                     mutate(score.diff = (score.t2 - score.t1))

#Finding the summary statistics
summary_new <- tutoring_data_new %>% group_by(tutoring) %>% dplyr::summarise(mean_diff = mean(score.dif
```

NHST Approach

```
#Performing t-test
(scores.diff.t.test <- t.test(score.diff ~ tutoring, tutoring_data_new) )
```

```
##
##  Welch Two Sample t-test
##
## data:  score.diff by tutoring
## t = -5.0811, df = 194.29, p-value = 8.78e-07
## alternative hypothesis: true difference in means between group Non-Tutored and group Tutored is not e
## 95 percent confidence interval:
##  -5.837998 -2.573164
## sample estimates:
## mean in group Non-Tutored     mean in group Tutored
##                -0.4399544                 3.7656265
```

Estimation Approach

```
#Creating the linear model
scores_diff_lm <- lm(score.diff ~ tutoring, tutoring_data_new)

#Extracting the means and 95% CIs
scores_diff_emm <- emmeans(scores_diff_lm, ~ tutoring)
kable(scores_diff_emm, caption = "Score difference means and 95% CIs by tutoring")
```

Table 8: Score difference means and 95% CIs by tutoring

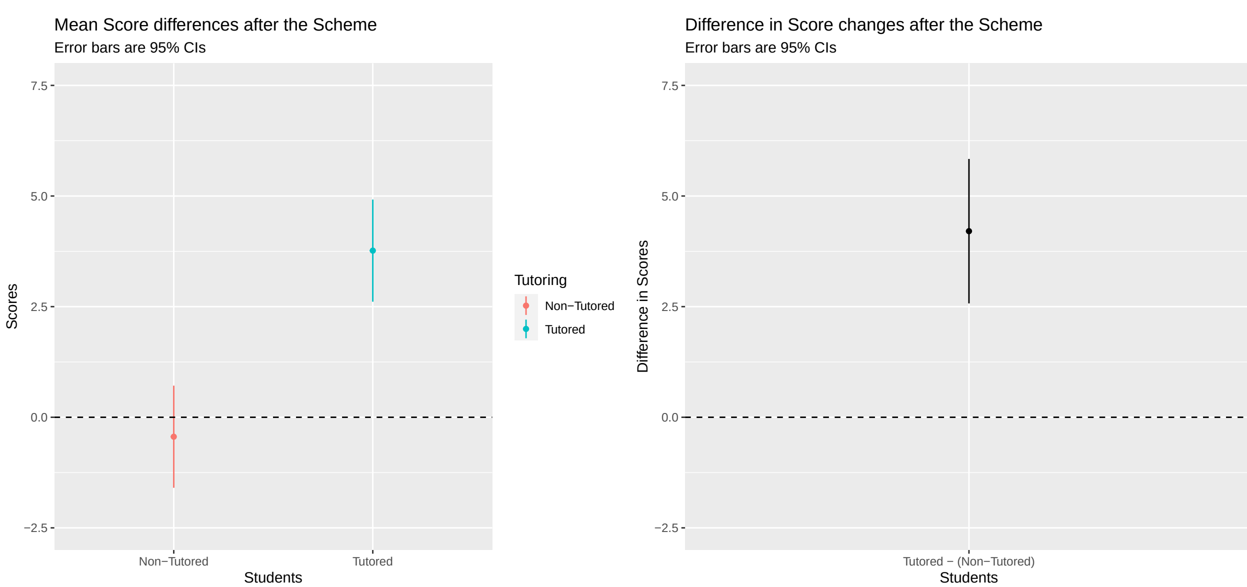
tutoring	emmean	SE	df	lower.CL	upper.CL
Non-Tutored	-0.4399544	0.5852675	198	-1.594112	0.7142034
Tutored	3.7656265	0.5852675	198	2.611469	4.9197843

```
#Estimating the difference in means and 95% CI
scores_diff_contrast <- confint(pairs(scores_diff_emm, reverse = TRUE))
kable(scores_diff_contrast, caption = "Difference in the means of score changes by tutoring")
```

Table 9: Difference in the means of score changes by tutoring

contrast	estimate	SE	df	lower.CL	upper.CL
Tutored - (Non-Tutored)	4.205581	0.8276933	198	2.573355	5.837807

```r
#Visualizing the estimations
grid2 <- grid.arrange(
          ggplot(summary(scores_diff_emm), aes(x=tutoring, y=emmean, ymin=lower.CL, ymax=upper.CL, colo
            geom_point() +
            geom_linerange() +
            labs(y="Scores", x="Students", color = "Tutoring",
                 subtitle="Error bars are 95% CIs", title="Mean Score differences after the Scheme") +
            geom_hline(yintercept=0, lty=2) +
            ylim(-2.5, 7.5),

          ggplot(scores_diff_contrast, aes(y=estimate, x=contrast, ymin=lower.CL, ymax=upper.CL)) +
            geom_point() +
            geom_linerange() +
            labs(y="Difference in Scores", x="Students",
                 subtitle="Error bars are 95% CIs", title="Difference in Score changes after the Schem
            geom_hline(yintercept=0, lty=2) +
            ylim(-2.5, 7.5),

          ncol=2)
```



Section 5: Checking for any effect of absences on the change in scores, and if this had any interaction with the effect of tutoring

```r
#Creating linear model with tutoring and absences main effects
scores.diff.absences.lm <- lm(score.diff ~ tutoring + absences, tutoring_data_new)
summary(scores.diff.absences.lm)
```

```
##
## Call:
## lm(formula = score.diff ~ tutoring + absences, data = tutoring_data_new)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.5498  -3.5452  -0.2211   3.4694  15.7400
```

```
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.3101     0.9610   0.323    0.747
## tutoringTutored   4.2626     0.8298   5.137 6.69e-07 ***
## absences         -0.1188     0.1208  -0.984    0.326
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.853 on 197 degrees of freedom
## Multiple R-squared:  0.1197, Adjusted R-squared:  0.1107
## F-statistic: 13.39 on 2 and 197 DF,  p-value: 3.525e-06
```

```
#Using anova to check if including absences improves the model
anova(scores_diff_lm, scores.diff.absences.lm)
```

```
## Analysis of Variance Table
##
## Model 1: score.diff ~ tutoring
## Model 2: score.diff ~ tutoring + absences
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    198 6782.3
## 2    197 6749.1  1    33.177 0.9684 0.3263
```

The inclusion of main effects of tutoring and absences do not improve the model according to the anova test

```
#Creating linear model with tutoring and absences having interaction
scores.diff.absences.lm.inter <- lm(score.diff ~ tutoring * absences, tutoring_data_new)
summary(scores.diff.absences.lm.inter)
```

```
##
## Call:
## lm(formula = score.diff ~ tutoring * absences, data = tutoring_data_new)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.6208  -3.5477  -0.2268   3.5930  15.7730
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               0.42453    1.25171   0.339   0.7349
## tutoringTutored           4.03560    1.79026   2.254   0.0253 *
## absences                 -0.13696    0.17517  -0.782   0.4352
## tutoringTutored:absences  0.03471    0.24235   0.143   0.8863
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.868 on 196 degrees of freedom
## Multiple R-squared:  0.1198, Adjusted R-squared:  0.1063
## F-statistic:  8.89 on 3 and 196 DF,  p-value: 1.495e-05
```

```
#Using anova to check if interactivity improves the model with no interaction
anova(scores_diff_lm, scores.diff.absences.lm.inter)
```

```
## Analysis of Variance Table
##
## Model 1: score.diff ~ tutoring
```

```
## Model 2: score.diff ~ tutoring * absences
##   Res.Df    RSS Df Sum of Sq     F Pr(>F)
## 1    198 6782.3
## 2    196 6748.4  2    33.883 0.492 0.6121
```

The inclusion of interactive effect of tutoring and absences also do not improve the model according to the anova test

Part 2: Report

Finding 1: Students allocated to the tutored and non-tutored groups didn't conclusively have similar or different average test scores before the tutoring scheme.

In order to check whether the students allocated to the tutored and non-tutored groups had similar or different average test scores before the tutoring scheme began, we performed a two-sample t-test to predict their scores(score.t1) by tutoring status:

```
##
##  Welch Two Sample t-test
##
## data:  score.t1 by tutoring
## t = -1.0467, df = 196.54, p-value = 0.2965
## alternative hypothesis: true difference in means between group Non-Tutored and group Tutored is not e
## 95 percent confidence interval:
##  -5.433861  1.665720
## sample estimates:
## mean in group Non-Tutored     mean in group Tutored
##                  52.90345                  54.78753
```

We conclude from our sample of 200 students that the mean score for tutored student group is **not significantly** greater than that of non-tutored student group, **Welch t(197) = 1.05, p = 0.2965.**

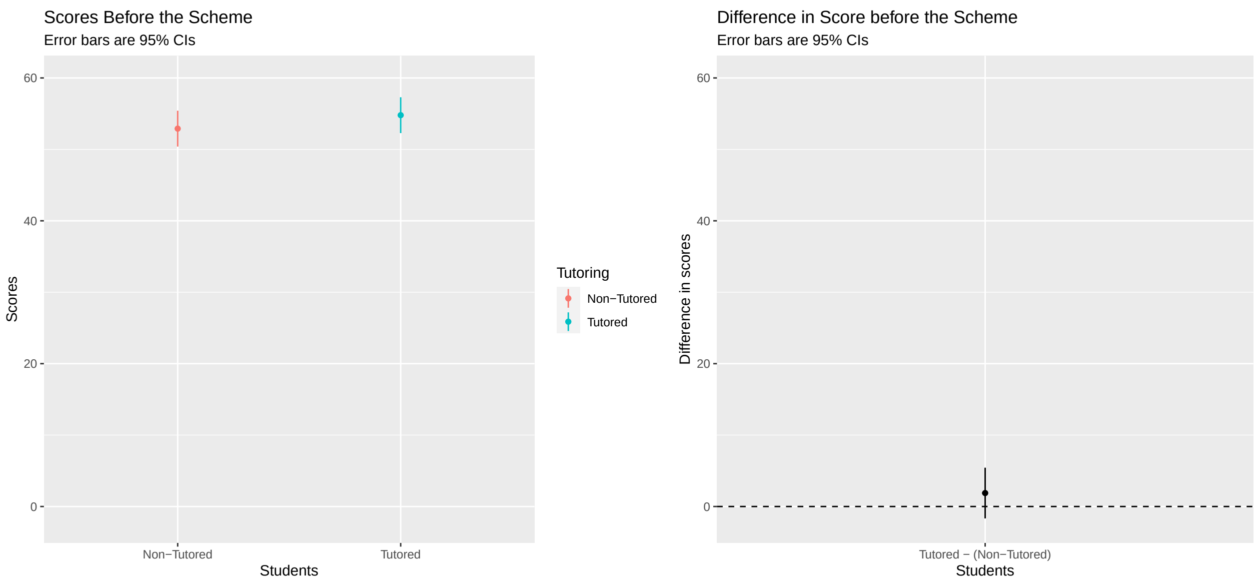
For getting a better estimation of average test scores before the tutoring scheme, we look at the following means and confidence intervals:

Table 10: The mean test scores and their 95% CIs before the tutoring scheme

tutoring	emmean	SE	df	lower.CL	upper.CL
Non-Tutored	52.90345	1.272791	198	50.39349	55.41342
Tutored	54.78753	1.272791	198	52.27756	57.29749

Table 11: Difference in the test scores before the tutoring scheme

contrast	estimate	SE	df	lower.CL	upper.CL
Tutored - (Non-Tutored)	1.884071	1.799998	198	-1.665557	5.433699



The mean score for non-tutored students is **52.9 95% CI [50.4 - 55.4]**. The mean score for tutored students is **54.8 95% CI [52.3 - 57.3]**. The difference is 1.88 **95% CI [-1.7 - 5.4]** greater for the tutored student group.

Thus, both the p-value and the CI of difference of the scores of the two groups indicate that **we cannot say conclusively whether students allocated to the tutored and non-tutored groups had similar or different scores before the tutoring scheme began.**

Finding 2: Cannot conclusively say if the tutored and non-tutored students had similar or different rates of absences on average

In order to check whether the tutored and non-tutored groups had similar or different rates of absences on average, we performed a two-sample t-test to predict their absences by tutoring status:

```
##
##  Welch Two Sample t-test
##
## data:  absences by tutoring
## t = -0.98528, df = 197.6, p-value = 0.3257
## alternative hypothesis: true difference in means between group Non-Tutored and group Tutored is not
## 95 percent confidence interval:
##  -1.440721  0.480721
## sample estimates:
## mean in group Non-Tutored     mean in group Tutored
##                     6.312                     6.792
```

We conclude from our sample of 200 students that the mean absence rate for tutored student group is **not significantly** greater than that of non-tutored student group, **Welch t(197) = 0.98, p = 0.3257.**

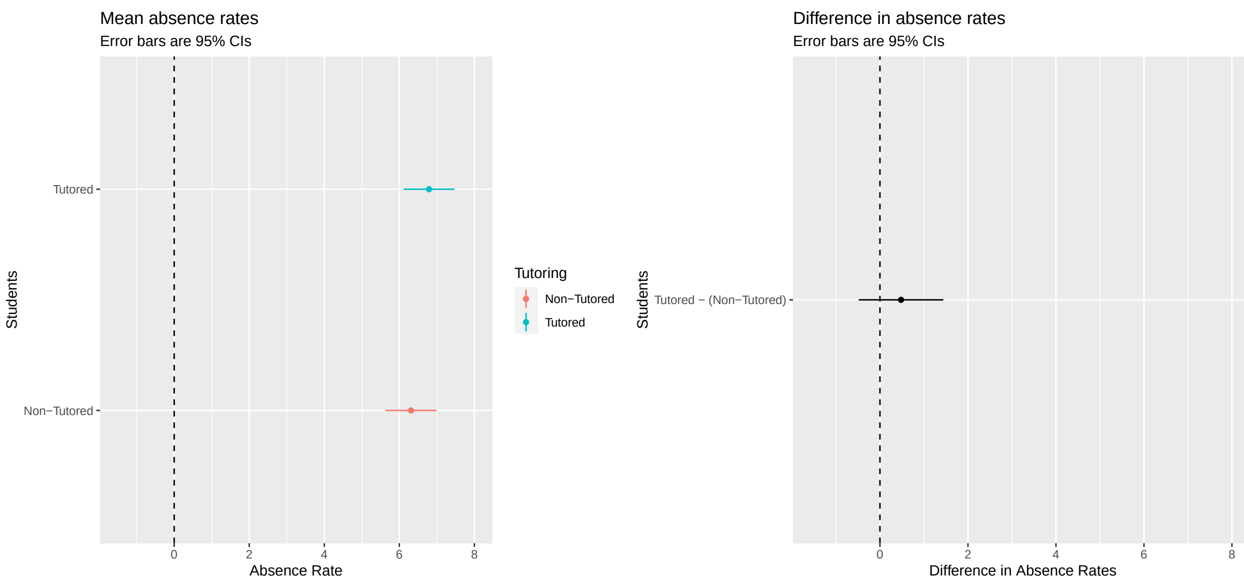
For getting a better estimation of average absence rates, we look at the following means and confidence intervals:

Table 12: Average absences and 95% CI based on tutoring categories

tutoring	emmean	SE	df	lower.CL	upper.CL
Non-Tutored	6.312	0.3444817	198	5.632676	6.991324
Tutored	6.792	0.3444817	198	6.112676	7.471324

Table 13: Differences in means and 95% CI

contrast	estimate	SE	df	lower.CL	upper.CL
Tutored - (Non-Tutored)	0.48	0.4871707	198	-0.4807091	1.440709



The mean absence rate for non-tutored students is **6.3 95% CI [5.6 - 6.9]**. The mean absence rate for tutored students is **6.8 95% CI [6.1 - 7.5]**. The difference is **0.48 95% CI [-0.5 - 1.4]** greater for the tutored student group.

Thus, both the p-value and the CI of difference of the scores of the two groups indicate that **we cannot say conclusively whether tutored and non-tutored groups had similar or different rates of absences on average.**

Further analysis showed that there is merit to include additional complexity in our model by using the main effect of either of the student scores along with the tutoring status while predicting the absence rates. However, the better model, though it changes the means and the 95% CIs, cannot also draw any different conclusion.

Finding 3: Tutored students do show an increase in their scores compared to the students who did not receive tutoring

In order to check whether tutored students show an increase in their scores compared to the students who did not receive tutoring, we found the difference between their scores at the beginning and the end of the academic year and performed two-sample t-test and estimation process on the data:

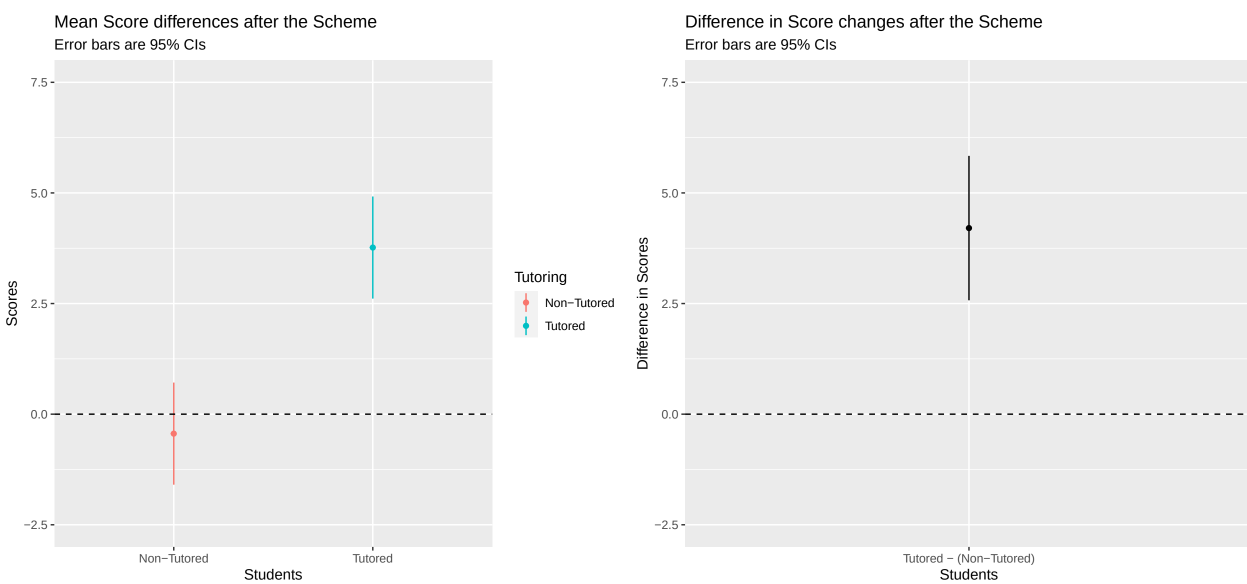
```
##
##  Welch Two Sample t-test
##
## data:  score.diff by tutoring
## t = -5.0811, df = 194.29, p-value = 8.78e-07
## alternative hypothesis: true difference in means between group Non-Tutored and group Tutored is not
## 95 percent confidence interval:
##  -5.837998 -2.573164
## sample estimates:
## mean in group Non-Tutored     mean in group Tutored
##                -0.4399544                 3.7656265
```

We conclude from the t-test on our sample of 200 students that the mean score difference for tutored student group is **significantly greater** than that of non-tutored student group, **Welch** $\mathbf{t(194) = 5.08, p < 0.05.}$

For getting a better estimation of average absence rates, we look at the following means and confidence intervals:

```
##  tutoring    emmean    SE  df lower.CL upper.CL
##  Non-Tutored  -0.44 0.585 198    -1.59    0.714
##  Tutored       3.77 0.585 198     2.61    4.920
##
## Confidence level used: 0.95
```

```
##  contrast                estimate    SE  df lower.CL upper.CL
##  Tutored - (Non-Tutored)     4.21 0.828 198     2.57     5.84
##
## Confidence level used: 0.95
```



The mean score decrease for non-tutored students is **0.44 95% CI** [**-1.59 - 0.71**]. The mean score increase for tutored students is **3.77 95% CI** [**2.61 - 4.92**]. The difference is a score of **4.21 95% CI** [**2.57 - 5.84**] greater for the tutored student group.

Thus, both the p-value and the CI of difference of the scores of the two groups indicate that **we can say conclusively that tutored students show an increase in their scores compared to the non-tutored students after the tutoring scheme was implemented.**

Finding 4: No effect of absences on the change in scores, and no interaction with the effect of tutoring

In order to check for the effect of absences on the change in scores, we first check how the scores behave when either tutoring status or absence rate is held constant (i.e. not changing), then we observe for for any interaction between tutoring status and absence rate:

```
##
## Call:
## lm(formula = score.diff ~ tutoring + absences, data = tutoring_data_new)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -14.5498  -3.5452  -0.2211   3.4694  15.7400
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.3101     0.9610   0.323    0.747
## tutoringTutored    4.2626     0.8298   5.137 6.69e-07 ***
## absences          -0.1188     0.1208  -0.984    0.326
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.853 on 197 degrees of freedom
## Multiple R-squared:  0.1197, Adjusted R-squared:  0.1107
## F-statistic: 13.39 on 2 and 197 DF,  p-value: 3.525e-06
```

Thus, the results of the regression show that there is **no significant main effect** of absence rate on scores **(tutoring = -0.11, t(197) = 0.98, p = 0.326)** but there was a **significant main effect** of tutoring on scores **(absences = 4.26, t(197) = 5.14, p < 0.0001)**

```
## Analysis of Variance Table
##
## Model 1: score.diff ~ tutoring
## Model 2: score.diff ~ tutoring + absences
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    198 6782.3
## 2    197 6749.1  1    33.177 0.9684 0.3263
```

This is supported by the our anova test, which indicates the model is not significantly improved by the additional complexity of including the effect of absence.

Now, we check for any interactivity between tutoring status and absence rate:

```
##
## Call:
## lm(formula = score.diff ~ tutoring * absences, data = tutoring_data_new)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.6208  -3.5477  -0.2268   3.5930  15.7730
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                0.42453    1.25171   0.339   0.7349
## tutoringTutored            4.03560    1.79026   2.254   0.0253 *
## absences                  -0.13696    0.17517  -0.782   0.4352
## tutoringTutored:absences  0.03471    0.24235   0.143   0.8863
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.868 on 196 degrees of freedom
## Multiple R-squared:  0.1198, Adjusted R-squared:  0.1063
## F-statistic:  8.89 on 3 and 196 DF,  p-value: 1.495e-05
```

Thus, the results of the regression show that there is **no significant main effect** of absence rate on scores **(tutoring = -0.14, t(196) = 0.78, p = 0.4352)** but there was a **significant main effect** of tutoring on scores **(absences = 4.04, t(196) = 2.25, p = 0.0253).** There was also **no significant interaction** between tutoring status and absence, with the positive effect of absence being larger when tutoring status was 'Tutored' **(absences = 0.03, t(196) = 0.14, p = 0.8863)**

```
## Analysis of Variance Table
##
## Model 1: score.diff ~ tutoring
## Model 2: score.diff ~ tutoring * absences
##   Res.Df    RSS Df Sum of Sq     F Pr(>F)
## 1    198 6782.3
## 2    196 6748.4  2    33.883 0.492 0.6121
```

This conclusion is also supported by the anova test, which indicates the model is not significantly improved by the additional complexity of including the interactivity effect of absence.

Thus, we can conclude by saying that there was no significant effect of absences on change in scores, neither did it have any interaction with the effect of tutoring.

Question 2a

Beer Data Dictionary

Variable	Description
Name	The name of the beer
Style	The style of the beer
Brewery	The name of the manufacturer of the beer
ABV	Abbreviation for 'Alcohol by volume' - indicates how much of the total volume of liquid in a beer is made up of alcohol
rating	The rating of the beer in a scale of 1-5
minIBU	Abbreviation for 'International Bitterness Units' - indicates the minimum level of a beer's bitterness
maxIBU	Abbreviation for 'International Bitterness Units' - indicates the maximum level of a beer's bitterness
Astringency	Beer astringency is an off flavor and is perceived as a dry grainy, mouth-puckering, tannic sensation.
Body	Describes how heavy or light the beer is
Alcohol	The alcohol content in the beer
Bitter	The bitterness of a beer
Sweet	The sweetness of a beer
Sour	The sourness of a beer
Salty	The saltiness of a beer
Fruits	The fruitiness of a beer
Hoppy	The Hops content of a beer
Spices	The Spice content of a beer
Malty	The Malt content of a beer

Part 1: Analysis

Section 1: Data Preparation

```
#Reading the file into R
beer_data <- read_csv("Craft-Beer_data_set.txt")
```

```r
#Checking the structure of the dataset
str(beer_data)
```

```
## spec_tbl_df [5,558 x 18] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Name       : chr [1:5558] "Amber" "Double Bag" "Long Trail Ale" "Doppelsticke" ...
##  $ Style      : chr [1:5558] "Altbier" "Altbier" "Altbier" "Altbier" ...
##  $ Brewery    : chr [1:5558] "Alaskan Brewing Co." "Long Trail Brewing Co." "Long Trail Brewing Co."
##  $ ABV        : num [1:5558] 5.3 7.2 5 8.5 5.3 7.2 6 5.3 5 4.8 ...
##  $ rating     : num [1:5558] 3.65 3.9 3.58 4.15 3.67 3.78 4.1 3.46 3.6 4.1 ...
##  $ minIBU     : num [1:5558] 25 25 25 25 25 25 25 25 25 25 ...
##  $ maxIBU     : num [1:5558] 50 50 50 50 50 50 50 50 50 50 ...
##  $ Astringency: num [1:5558] 13 12 14 13 21 25 22 28 18 25 ...
##  $ Body       : num [1:5558] 32 57 37 55 69 51 45 40 49 35 ...
##  $ Alcohol    : num [1:5558] 9 18 6 31 10 26 13 3 5 4 ...
##  $ Bitter     : num [1:5558] 47 33 42 47 63 44 46 40 37 38 ...
##  $ Sweet      : num [1:5558] 74 55 43 101 120 45 62 58 73 39 ...
##  $ Sour       : num [1:5558] 33 16 11 18 14 9 25 29 22 13 ...
##  $ Salty      : num [1:5558] 0 0 0 1 0 1 1 0 0 1 ...
##  $ Fruits     : num [1:5558] 33 24 10 49 19 11 34 36 21 8 ...
##  $ Hoppy      : num [1:5558] 57 35 54 40 36 51 60 54 37 60 ...
##  $ Spices     : num [1:5558] 8 12 4 16 15 20 4 8 4 16 ...
##  $ Malty      : num [1:5558] 111 84 62 119 218 95 103 97 98 97 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Name = col_character(),
##   ..   Style = col_character(),
##   ..   Brewery = col_character(),
##   ..   ABV = col_double(),
##   ..   rating = col_double(),
##   ..   minIBU = col_double(),
##   ..   maxIBU = col_double(),
##   ..   Astringency = col_double(),
##   ..   Body = col_double(),
##   ..   Alcohol = col_double(),
##   ..   Bitter = col_double(),
##   ..   Sweet = col_double(),
##   ..   Sour = col_double(),
##   ..   Salty = col_double(),
##   ..   Fruits = col_double(),
##   ..   Hoppy = col_double(),
##   ..   Spices = col_double(),
##   ..   Malty = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```r
#Checking for any NA values
summary(beer_data)
```

```
##      Name              Style              Brewery                ABV              rating
##  Length:5558        Length:5558        Length:5558        Min.   : 0.000   Min.   :1.27
##  Class :character   Class :character   Class :character   1st Qu.: 5.000   1st Qu.:3.59
##  Mode  :character   Mode  :character   Mode  :character   Median : 6.000   Median :3.82
##                                                           Mean   : 6.634   Mean   :3.76
##                                                           3rd Qu.: 7.900   3rd Qu.:4.04
##                                                           Max.   :57.500   Max.   :4.83
```

```
##      minIBU          maxIBU        Astringency        Body           Alcohol
##  Min.   : 0.00   Min.   :  0.00   Min.   : 0.00   Min.   :  0.00   Min.   :  0.00
##  1st Qu.:10.00   1st Qu.: 25.00   1st Qu.: 8.00   1st Qu.: 25.00   1st Qu.:  5.00
##  Median :20.00   Median : 35.00   Median :14.00   Median : 38.00   Median : 10.00
##  Mean   :20.72   Mean   : 38.45   Mean   :15.94   Mean   : 42.75   Mean   : 15.98
##  3rd Qu.:25.00   3rd Qu.: 45.00   3rd Qu.:22.00   3rd Qu.: 55.00   3rd Qu.: 20.00
##  Max.   :65.00   Max.   :100.00   Max.   :83.00   Max.   :197.00   Max.   :139.00
##      Bitter           Sweet             Sour            Salty            Fruits
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   Min.   : 0.000   Min.   :  0.00
##  1st Qu.: 13.00   1st Qu.: 27.00   1st Qu.:  9.00   1st Qu.: 0.000   1st Qu.: 10.00
##  Median : 29.00   Median : 49.50   Median : 21.00   Median : 0.000   Median : 28.00
##  Mean   : 34.32   Mean   : 53.63   Mean   : 34.61   Mean   : 1.314   Mean   : 39.38
##  3rd Qu.: 51.00   3rd Qu.: 74.00   3rd Qu.: 44.00   3rd Qu.: 1.000   3rd Qu.: 61.75
##  Max.   :150.00   Max.   :263.00   Max.   :323.00   Max.   :66.000   Max.   :222.00
##      Hoppy            Spices            Malty
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.00
##  1st Qu.: 14.00   1st Qu.:  4.00   1st Qu.: 33.00
##  Median : 30.00   Median :  9.00   Median : 65.00
##  Mean   : 38.41   Mean   : 17.58   Mean   : 68.59
##  3rd Qu.: 56.00   3rd Qu.: 22.00   3rd Qu.: 99.00
##  Max.   :193.00   Max.   :184.00   Max.   :304.00
```

```r
#Removing NA values
beer_data <- na.omit(beer_data)

#Checking for duplicate data
nrow(distinct(beer_data)) #No duplicates as the distinct entries are equal to the number of rows in the
```

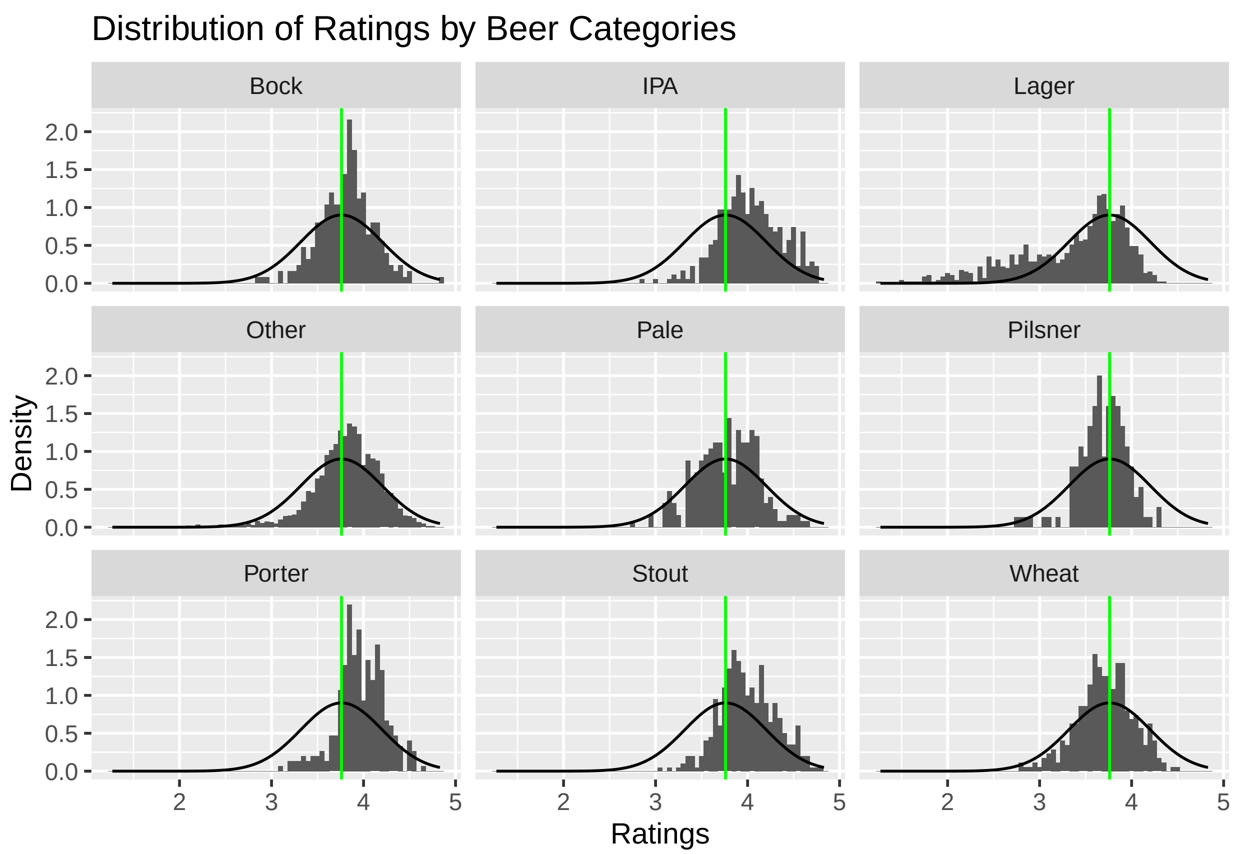
```
## [1] 5556
```

```r
#If duplicates existed, then the following code could be used to remove them:
                                    #beer_data <- beer_data[which(beer_data == distinct(beer_data)),]
```

```r
#Renaming the categories of beer according to requirement
beer_data[grepl("IPA", beer_data$Style), "Style"] <- "IPA"
beer_data[grepl("Lager", beer_data$Style), "Style"] <- "Lager"
beer_data[grepl("Porter", beer_data$Style), "Style"] <- "Porter"
beer_data[grepl("Stout", beer_data$Style), "Style"] <- "Stout"
beer_data[grepl("Wheat", beer_data$Style), "Style"] <- "Wheat"
beer_data[grepl("Pale", beer_data$Style), "Style"] <- "Pale"
beer_data[grepl("Pilsner", beer_data$Style), "Style"] <- "Pilsner"
beer_data[grepl("Bock", beer_data$Style), "Style"] <- "Bock"
beer_data[beer_data$Style!= "IPA" &
          beer_data$Style != "Lager" &
          beer_data$Style != "Porter" &
          beer_data$Style != "Stout" &
          beer_data$Style != "Wheat" &
          beer_data$Style != "Pale" &
          beer_data$Style != "Pilsner" &
          beer_data$Style != "Bock" , "Style"] <- "Other"

#Converting the beer category column into factor
beer_data$Style <- as.factor(beer_data$Style)

#Checking if the levels of factors are set properly
levels(beer_data$Style)
```

```
## [1] "Bock"     "IPA"      "Lager"    "Other"    "Pale"     "Pilsner" "Porter"   "Stout"    "Wheat"
```

```
#Generating summary statistics
beer_data_summary <- beer_data %>%
                     group_by(Style) %>%
                     summarise(mean_rating = mean(rating), std_rating = sd(rating))
  #Storing the summary values to individual variables
  rating_sd <- beer_data_summary$std_rating
  rating_mean <- beer_data_summary$mean_rating

#Visualizing the distribution of rating data by beer categories and comparing to normal distribution
ggplot(beer_data, aes(x=rating)) +
       geom_histogram(aes(y=..density..), binwidth = 0.05) +
       stat_function(fun=function(x) {dnorm(x, mean=rating_mean, sd=rating_sd)}) +
       geom_vline(xintercept = beer_data_summary$mean_rating, color = "Green") +
       facet_wrap(~Style) +
       labs(x="Ratings", y="Density",
            title = "Distribution of Ratings by Beer Categories")
```



The distributions seem fairly normal, except a few show slightly positive or negative skewness. Overall, the distrubutions seems fit to continue with our analysis

Section 2: Calculating the mean rating and 95% confidence intervals of the rating within each category using a linear model.

Estimation Approach

```r
#Creating the linear model
(beer.lm <- lm(rating ~ Style, beer_data) )
```

```
##
## Call:
## lm(formula = rating ~ Style, data = beer_data)
##
## Coefficients:
##  (Intercept)      StyleIPA    StyleLager    StyleOther     StylePale  StylePilsner   StylePorter
##     3.811520      0.217909     -0.454156     -0.004102     -0.037400     -0.121187      0.155880
##   StyleStout    StyleWheat
##     0.187530     -0.100320
```

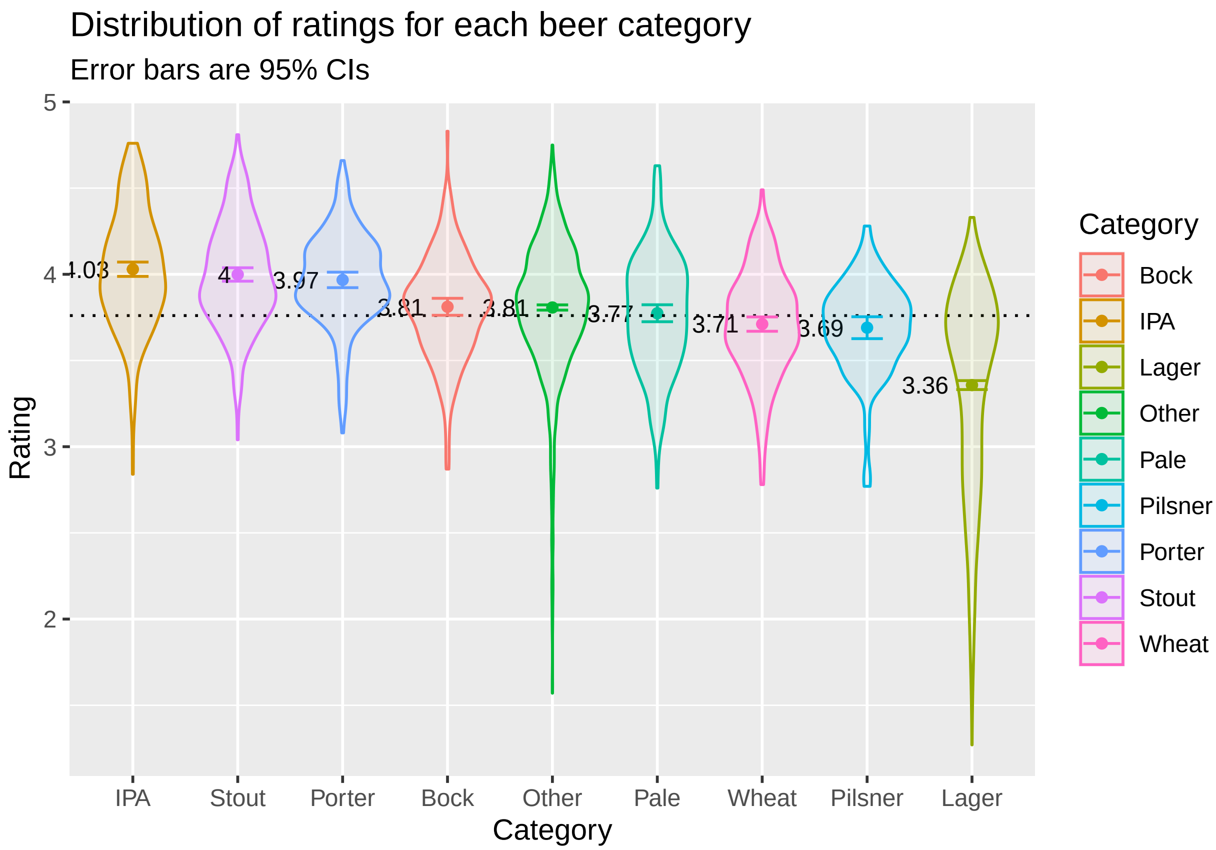
```r
#Extracting the means and 95% CIs
beer.emm <- emmeans(beer.lm, ~ Style)

#Storing the data as data frame and preparing it for reordering
beer.emm <- as.data.frame(beer.emm)
beer.emm <- beer.emm %>%
            group_by(emmean) %>%
            arrange(desc(emmean))
#Creating a table to show the estimations
(kable1 <- kable(beer.emm, caption = "Mean rating and 95% CIs within each category") )
```

Table 15: Mean rating and 95% CIs within each category

Style	emmean	SE	df	lower.CL	upper.CL
IPA	4.029429	0.0212219	5547	3.987825	4.071032
Stout	3.999050	0.0198512	5547	3.960134	4.037966
Porter	3.967400	0.0229222	5547	3.922463	4.012337
Bock	3.811520	0.0251101	5547	3.762294	3.860746
Other	3.807418	0.0077758	5547	3.792175	3.822662
Pale	3.774120	0.0251101	5547	3.724894	3.823346
Wheat	3.711200	0.0212219	5547	3.669597	3.752803
Pilsner	3.690333	0.0324169	5547	3.626783	3.753883
Lager	3.357364	0.0132415	5547	3.331405	3.383322

Section 3: Plot that displays, on a single axes, the distribution of the ratings within each category, the mean ratings and 95% confidence intervals

```r
#Visualizing the data
ggplot(beer.emm, aes(x=reorder(Style, -emmean),  #Arranging means in descending order
                     y=emmean, ymin=lower.CL, ymax=upper.CL,
                     color = Style)) +
       geom_point() +
       geom_hline(aes(yintercept=rating_mean), linetype = "dotted") +
       geom_errorbar(width = 0.3, stat = "identity") +
       labs(x="Category", y="Rating", color = "Category", fill = "Category",
            subtitle="Error bars are 95% CIs", title = "Distribution of ratings for each beer category"
       geom_text(aes(label = round(emmean, 2)), hjust = 1.5, size = 3, color = "Black") +  #Inserting l
       geom_violin(data=beer_data,
                   mapping=aes(x=Style, y=rating, ymin=NULL, ymax=NULL, color = Style, fill = Style),
                   alpha = 0.1) #Adding the distribution of the entire dataset
```



Part 2: Report

Finding 1: The mean rating and 95% confidence intervals of the rating within each category using a linear model

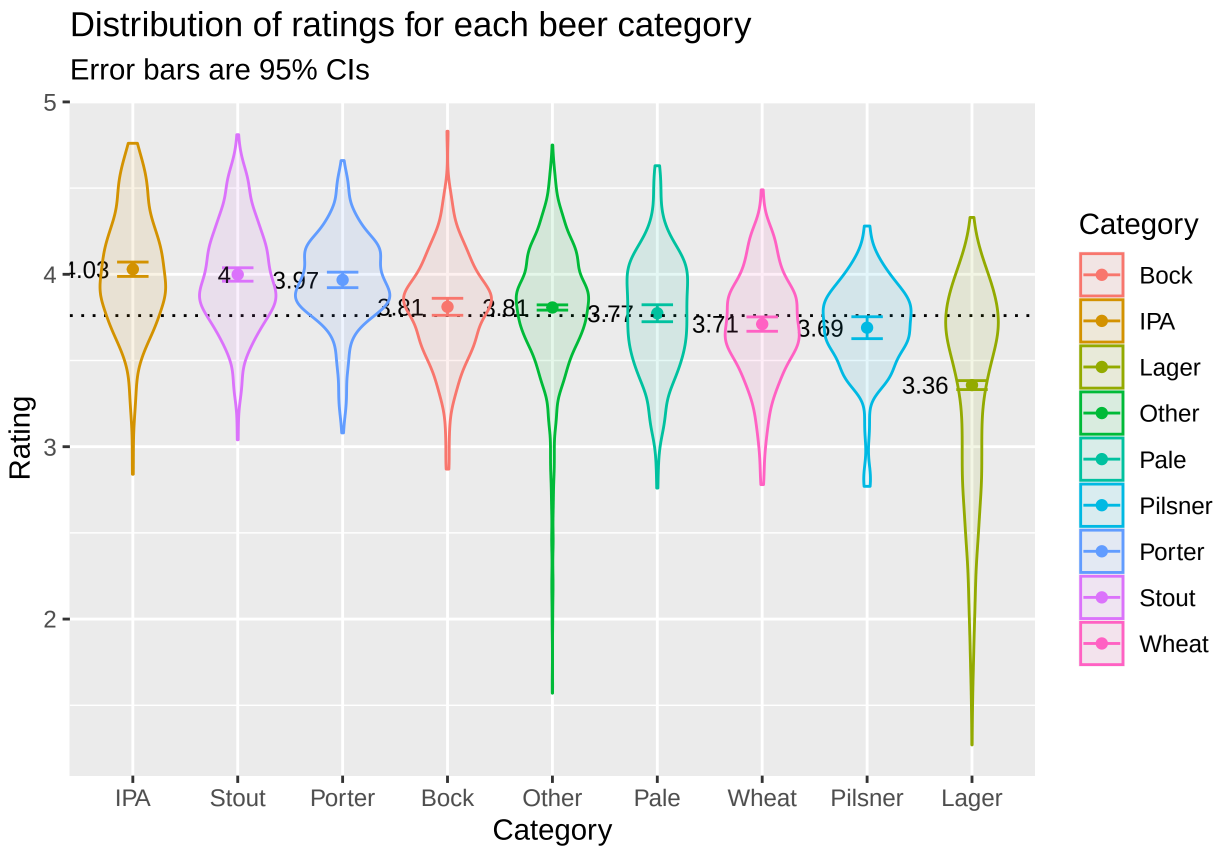
A linear model was used to predict beer rating by category. Using the model, the following mean ratings and 95% CIs were found for each category of beer:

Table 16: Mean rating and 95% CIs within each category

Style	emmean	SE	df	lower.CL	upper.CL
IPA	4.029429	0.0212219	5547	3.987825	4.071032
Stout	3.999050	0.0198512	5547	3.960134	4.037966
Porter	3.967400	0.0229222	5547	3.922463	4.012337
Bock	3.811520	0.0251101	5547	3.762294	3.860746
Other	3.807418	0.0077758	5547	3.792175	3.822662
Pale	3.774120	0.0251101	5547	3.724894	3.823346
Wheat	3.711200	0.0212219	5547	3.669597	3.752803
Pilsner	3.690333	0.0324169	5547	3.626783	3.753883
Lager	3.357364	0.0132415	5547	3.331405	3.383322

Finding 2: A plot that displays, on a single axes, the distribution of the ratings within each category, the mean ratings and 95% confidence intervals

A violin plot is used to show the distribution of the ratings within each category and the mean ratings and 95% CIs are shown using the error bars in the following figure:



Question 2b

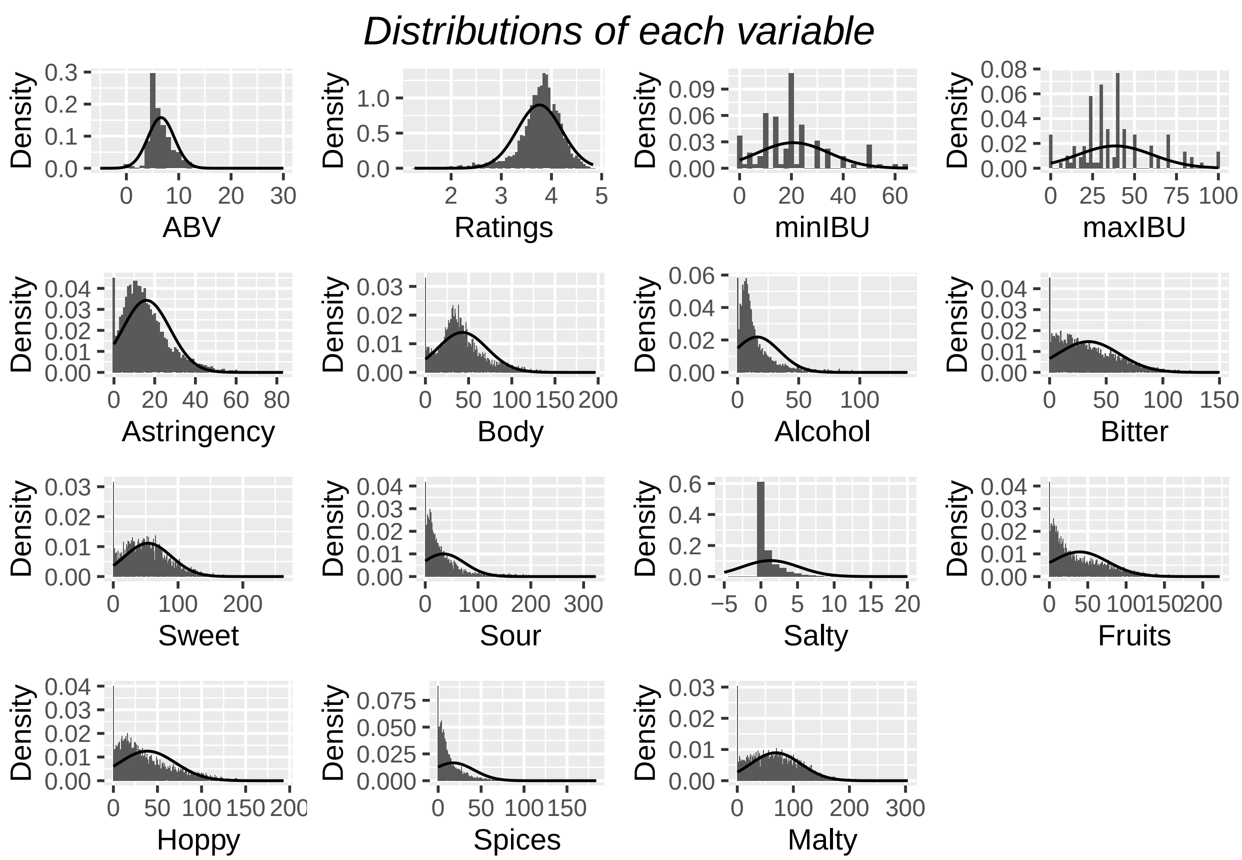
Part 1: Analysis

Section 1: Data Preparation

```
#Generating the summary statistics
beer_data_summary2 <- beer_data %>% summarise(mean_abv = mean(ABV), sd_abv = sd(ABV),
                                              mean_rating = mean(rating), sd_rating = sd(rating),
                                              mean_minIBU = mean(minIBU), sd_minIBU = sd(minIBU),
                                              mean_maxIBU = mean(maxIBU), sd_maxIBU = sd(maxIBU),
                                              mean_Astringency = mean(Astringency), sd_Astrigency = sd(
                                              mean_Body = mean(Body), sd_Body = sd(Body),
                                              mean_Alcohol = mean(Alcohol), sd_Alcohol = sd(Alcohol),
                                              mean_Bitter = mean(Bitter), sd_Bitter = sd(Bitter),
                                              mean_Sweet = mean(Sweet), sd_Sweet = sd(Sweet),
                                              mean_Sour = mean(Sour), sd_Sour = sd(Sour),
                                              mean_Salty = mean(Salty), sd_Salty = sd(Salty),
                                              mean_Fruits = mean(Fruits), sd_Fruit = sd(Fruits),
                                              mean_Hoppy = mean(Hoppy), sd_Hoppy = sd(Hoppy),
                                              mean_Spices = mean(Spices), sd_Spices = sd(Spices),
                                              mean_Malty = mean(Malty), sd_Malty = sd(Malty)
                                              )
```

```
#Checking the distribution of all the related variables and comparing to their normal distributions
grid.arrange((ggplot(beer_data, aes(x=ABV)) +
     geom_histogram(aes(y=..density..), binwidth = 1) +
     stat_function(fun=function(x) {dnorm(x, mean=beer_data_summary2$mean_abv, sd=beer_data_summary2$sd_
     labs(x="ABV", y="Density")) +
     xlim(-5,30), #Not including the outliers for visualizing

     (ggplot(beer_data, aes(x=rating)) +
     geom_histogram(aes(y=..density..), binwidth = 0.05) +
     stat_function(fun=function(x) {dnorm(x, mean=beer_data_summary2$mean_rating, sd=beer_data_summary2$
     labs(x="Ratings", y="Density")),

     (ggplot(beer_data, aes(x=minIBU)) +
     geom_histogram(aes(y=..density..), binwidth = 2) +
     stat_function(fun=function(x) {dnorm(x, mean=beer_data_summary2$mean_minIBU, sd=beer_data_summary2$
     labs(x="minIBU", y="Density")),

     (ggplot(beer_data, aes(x=maxIBU)) +
     geom_histogram(aes(y=..density..), binwidth = 2) +
     stat_function(fun=function(x) {dnorm(x, mean=beer_data_summary2$mean_maxIBU, sd=beer_data_summary2$
     labs(x="maxIBU", y="Density")),

     (ggplot(beer_data, aes(x=Astringency)) +
     geom_histogram(aes(y=..density..), binwidth = 1) +
     stat_function(fun=function(x) {dnorm(x, mean=beer_data_summary2$mean_Astringency, sd=beer_data_summ
     labs(x="Astringency", y="Density")),

     (ggplot(beer_data, aes(x=Body)) +
     geom_histogram(aes(y=..density..), binwidth = 1) +
     stat_function(fun=function(x) {dnorm(x, mean=beer_data_summary2$mean_Body, sd=beer_data_summary2$s
     labs(x="Body", y="Density")),

     (ggplot(beer_data, aes(x=Alcohol)) +
     geom_histogram(aes(y=..density..), binwidth = 1) +
     stat_function(fun=function(x) {dnorm(x, mean=beer_data_summary2$mean_Alcohol, sd=beer_data_summary
     labs(x="Alcohol", y="Density")),

     (ggplot(beer_data, aes(x=Bitter)) +
     geom_histogram(aes(y=..density..), binwidth = 1) +
     stat_function(fun=function(x) {dnorm(x, mean=beer_data_summary2$mean_Bitter, sd=beer_data_summary2$
     labs(x="Bitter", y="Density")),

     (ggplot(beer_data, aes(x=Sweet)) +
     geom_histogram(aes(y=..density..), binwidth = 1) +
     stat_function(fun=function(x) {dnorm(x, mean=beer_data_summary2$mean_Sweet, sd=beer_data_summary2$
     labs(x="Sweet", y="Density")),

     (ggplot(beer_data, aes(x=Sour)) +
     geom_histogram(aes(y=..density..), binwidth = 1) +
     stat_function(fun=function(x) {dnorm(x, mean=beer_data_summary2$mean_Sour, sd=beer_data_summary2$s
     labs(x="Sour", y="Density")),

     (ggplot(beer_data, aes(x=Salty)) +
```

```
geom_histogram(aes(y=..density..), binwidth = 1) +
stat_function(fun=function(x) {dnorm(x, mean=beer_data_summary2$mean_Salty, sd=beer_data_summary2$
labs(x="Salty", y="Density")) +
xlim(-5, 20),

(ggplot(beer_data, aes(x=Fruits)) +
geom_histogram(aes(y=..density..), binwidth = 1) +
stat_function(fun=function(x) {dnorm(x, mean=beer_data_summary2$mean_Fruits, sd=beer_data_summary2$
labs(x="Fruits", y="Density")),

(ggplot(beer_data, aes(x=Hoppy)) +
geom_histogram(aes(y=..density..), binwidth = 1) +
stat_function(fun=function(x) {dnorm(x, mean=beer_data_summary2$mean_Hoppy, sd=beer_data_summary2$
labs(x="Hoppy", y="Density")),

(ggplot(beer_data, aes(x=Spices)) +
geom_histogram(aes(y=..density..), binwidth = 1) +
stat_function(fun=function(x) {dnorm(x, mean=beer_data_summary2$mean_Spices, sd=beer_data_summary2$
labs(x="Spices", y="Density")),

(ggplot(beer_data, aes(x=Malty)) +
geom_histogram(aes(y=..density..), binwidth = 1) +
stat_function(fun=function(x) {dnorm(x, mean=beer_data_summary2$mean_Malty, sd=beer_data_summary2$
labs(x="Malty", y="Density")),

top = textGrob("Distributions of each variable",gp=gpar(fontsize=15,font=3)))
```

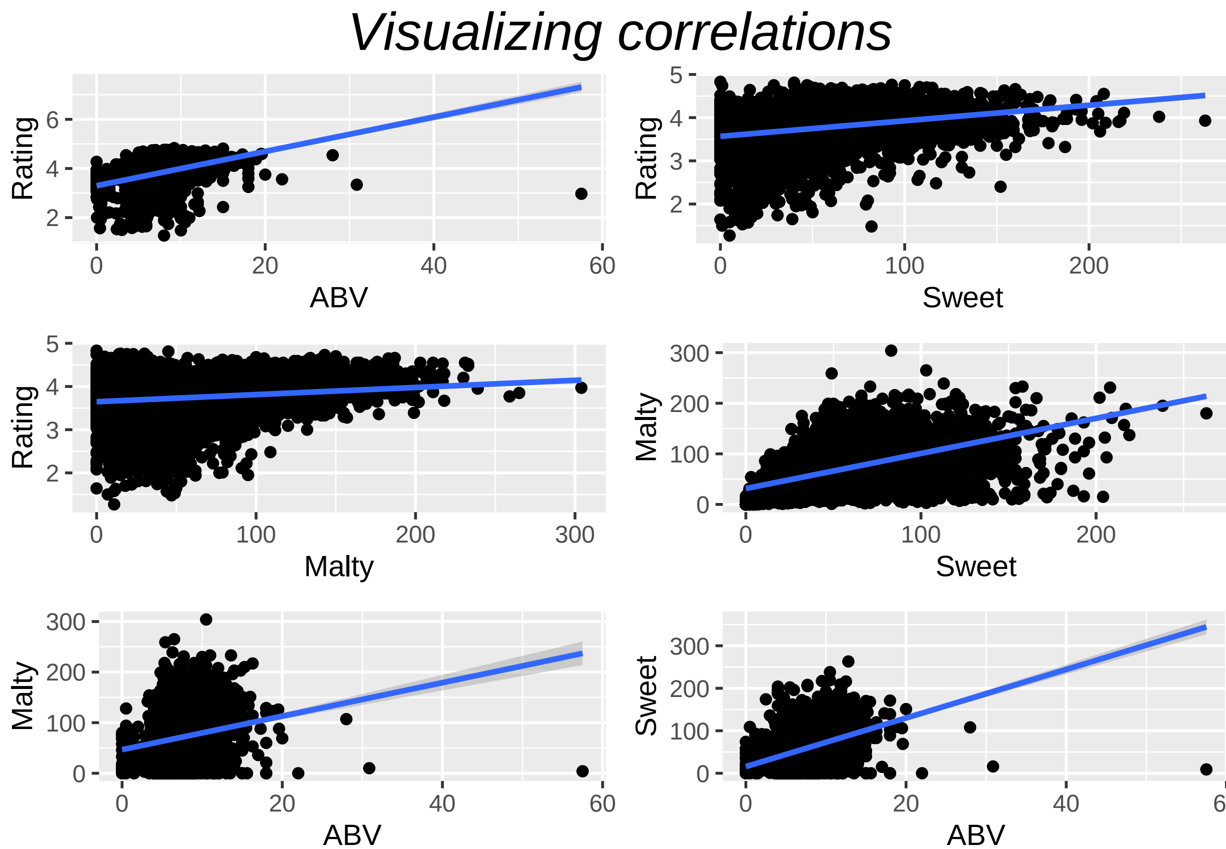


We are more interested in the distributions of ratings, ABV, Sweet and Malty. Ratings and ABV seem fairly normal, however, Sweet and Malty have a lot of positive skewness. We will keep this in mind while performing the analysis.

```
#Checking correlation for relevant variables
rcorr(as.matrix(select(beer_data, rating, ABV, Sweet, Malty), type = "pearson"))
```

```
##         rating  ABV Sweet Malty
## rating    1.00 0.40  0.29  0.17
## ABV       0.40 1.00  0.40  0.19
## Sweet     0.29 0.40  1.00  0.56
## Malty     0.17 0.19  0.56  1.00
##
## n= 5556
##
##
## P
##         rating ABV Sweet Malty
## rating          0   0     0
## ABV      0          0     0
## Sweet    0      0         0
## Malty    0      0   0
```

```
#Visualizing the correlations
grid.arrange(ggplot(beer_data, aes(x = ABV, y = rating)) +
             geom_point() +
             geom_smooth(method = lm) +
             labs(y = "Rating"),

             ggplot(beer_data, aes(x = Sweet, y = rating)) +
             geom_point() +
             geom_smooth(method = lm) +
             labs(y = "Rating"),

             ggplot(beer_data, aes(x = Malty, y = rating)) +
             geom_point() +
             geom_smooth(method = lm) +
             labs(y = "Rating"),

             ggplot(beer_data, aes(x = Sweet, y = Malty)) +
             geom_point() +
             geom_smooth(method = lm),

             ggplot(beer_data, aes(x = ABV, y = Malty)) +
             geom_point() +
             geom_smooth(method = lm),

             ggplot(beer_data, aes(x = ABV, y = Sweet)) +
             geom_point() +
             geom_smooth(method = lm),

             top = textGrob("Visualizing correlations",gp=gpar(fontsize=20,font=3))
)
```



Section 2: Checking whether, on average, a beer receives a higher rating if it has a higher or lower ABV.

NHST Approach

```
#Creating the linear model with ABV as predictor
abv_lm <- lm(rating ~ ABV, beer_data)
summary(abv_lm)
```

```
##
## Call:
## lm(formula = rating ~ ABV, data = beer_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3418 -0.1463  0.0526  0.2322  1.0088
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.297235   0.015351  214.79   <2e-16 ***
## ABV         0.069819   0.002163   32.28   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4064 on 5554 degrees of freedom
## Multiple R-squared:  0.158,  Adjusted R-squared:  0.1578
## F-statistic:  1042 on 1 and 5554 DF,  p-value: < 2.2e-16
```

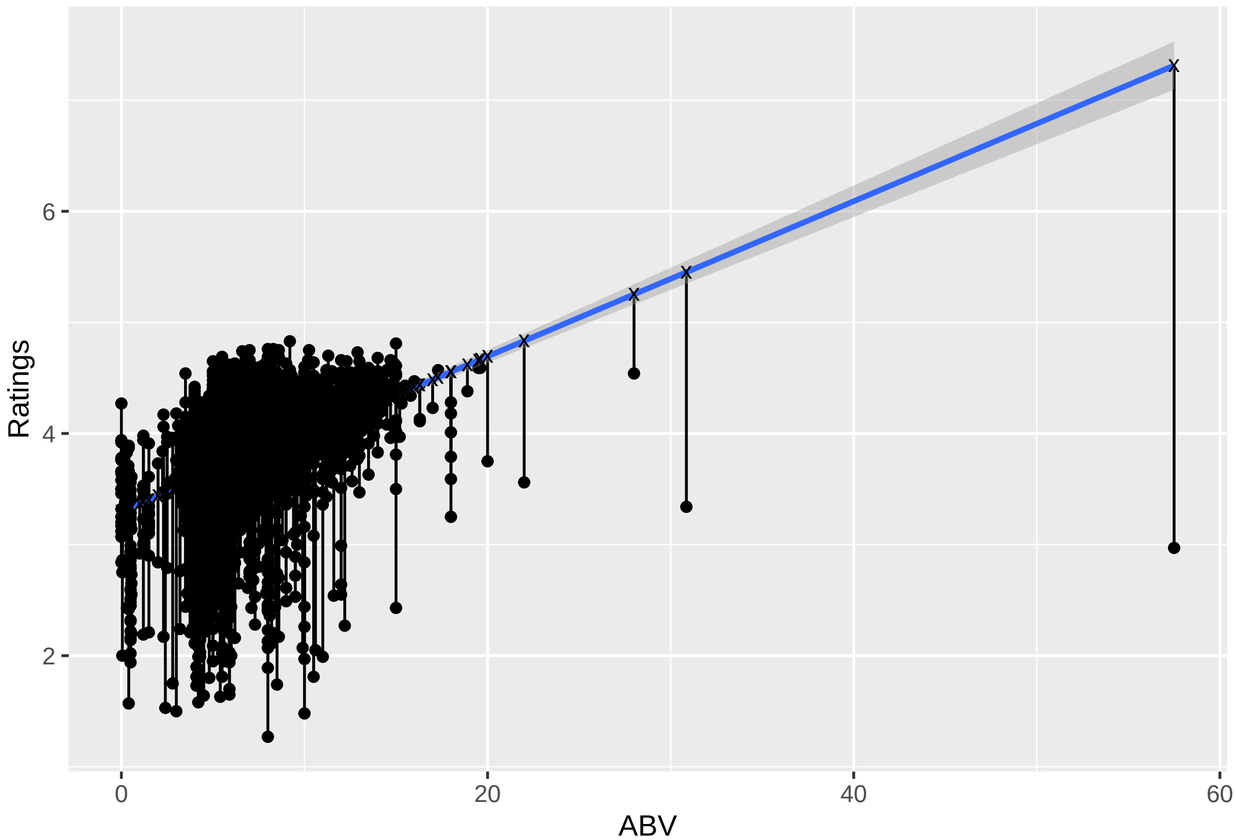
```r
#Creating another linear model with no predictor
abv_base <- lm(rating ~ 1, beer_data)

#Checking whether the model is improved by the use of ABV as predictor
anova(abv_base, abv_lm)
```

```
## Analysis of Variance Table
##
## Model 1: rating ~ 1
## Model 2: rating ~ ABV
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   5555 1089.4
## 2   5554  917.3  1    172.11 1042.1 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Comparison with the baseline linear model and the linear model with predictor in anova test shows that the additional complexity of including ABV makes our model significantly better.

Estimation Approach

```r
#Extracting the coefficient and 95% CIs from the linear model
abv_est <- cbind(coef(abv_lm), confint(abv_lm))

#Using the linear model to generate rating predictions based on existing ABV data
beer_data <- beer_data %>% mutate(rating.hat = predict(abv_lm))

#Visualizing the residuals
ggplot(beer_data, aes(x = ABV, y = rating, ymin = rating, ymax = rating.hat)) +
  geom_point() +
  geom_linerange() +
  geom_smooth(method = lm) +
  geom_point(aes(y = rating.hat), shape = "x", size = 3) +
  labs(y = "Ratings")
```



Section 3: Checking if having more or less Sweet or Malty elements in the flavour results in higher or lower ratings

Sweet Flavor

```
#Creating linear model for Sweet flavor:
flavor_lm.s <- lm(rating ~ ABV + Sweet, beer_data)
summary(flavor_lm.s)
```

```
##
## Call:
## lm(formula = rating ~ ABV + Sweet, data = beer_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6915 -0.1553  0.0460  0.2329  1.0836
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.266784   0.015380  212.41   <2e-16 ***
## ABV         0.058734   0.002333   25.18   <2e-16 ***
## Sweet       0.001939   0.000164   11.83   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4014 on 5553 degrees of freedom
## Multiple R-squared:  0.1787, Adjusted R-squared:  0.1784
## F-statistic:   604 on 2 and 5553 DF,  p-value: < 2.2e-16
```

```r
#Creating linear model for Sweet flavor with interaction:
flavor_lm_inter.s <- lm(rating ~ ABV * Sweet, beer_data)
summary(flavor_lm_inter.s)
```

```
##
## Call:
## lm(formula = rating ~ ABV * Sweet, data = beer_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1734 -0.1570  0.0453  0.2329  1.0866
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.1889925  0.0243314 131.065  < 2e-16 ***
## ABV          0.0700376  0.0035981  19.465  < 2e-16 ***
## Sweet        0.0034591  0.0004035   8.574  < 2e-16 ***
## ABV:Sweet   -0.0002007  0.0000487  -4.122 3.81e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4008 on 5552 degrees of freedom
## Multiple R-squared:  0.1812, Adjusted R-squared:  0.1807
## F-statistic: 409.5 on 3 and 5552 DF,  p-value: < 2.2e-16
```

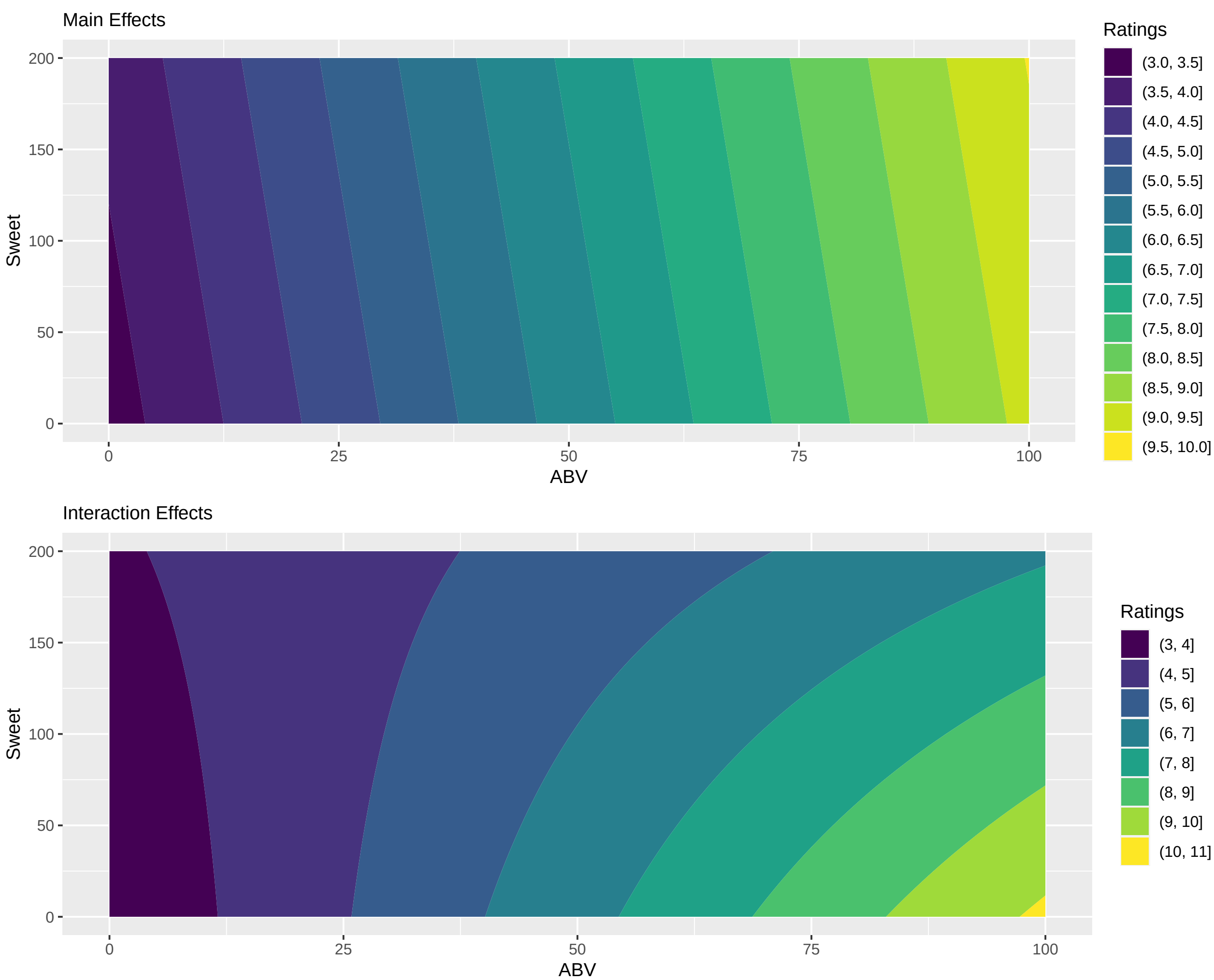
```r
#Checking to see which model is better
anova(flavor_lm.s, flavor_lm_inter.s)
```

```
## Analysis of Variance Table
##
## Model 1: rating ~ ABV + Sweet
## Model 2: rating ~ ABV * Sweet
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1   5553 894.77
## 2   5552 892.04  1      2.73 16.991 3.81e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

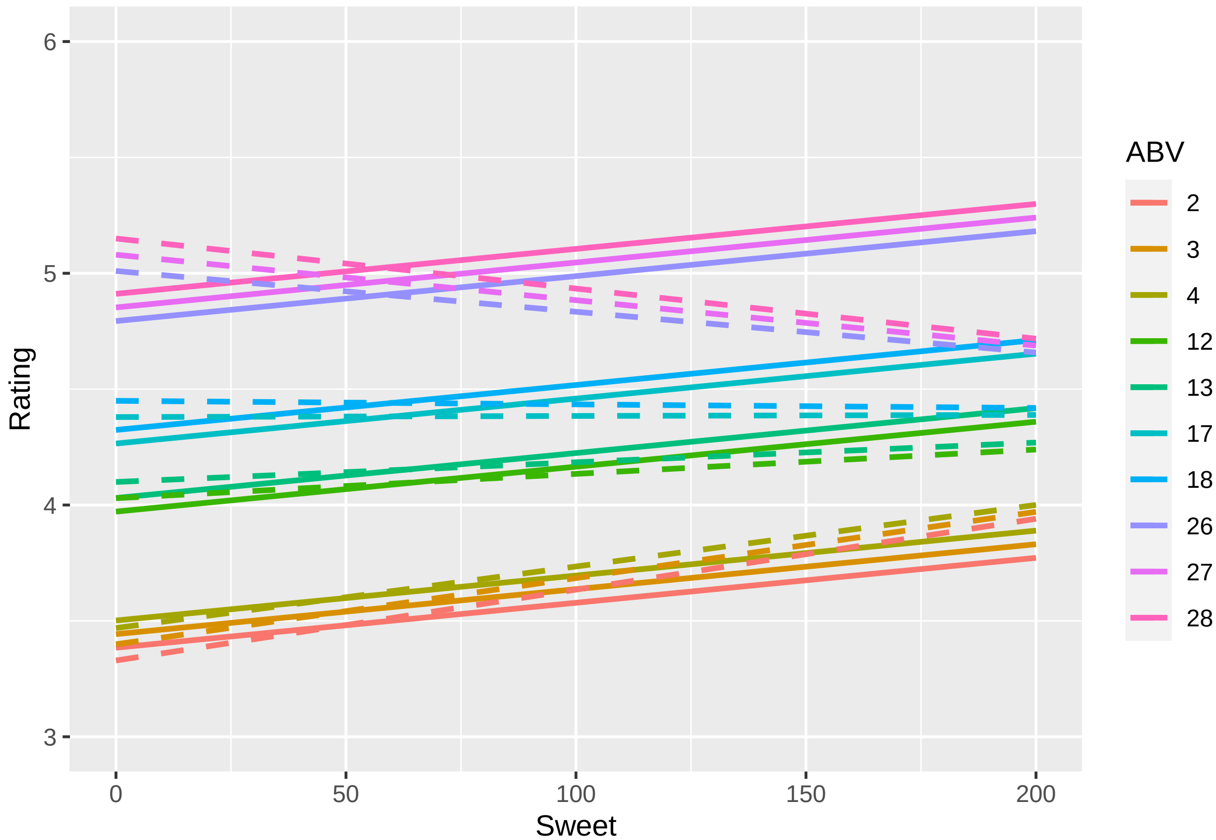
The anova test concludes that the additional complexity of including interaction effect of Sweet significantly improves the model. We will focus on this model while making interpretations.

```r
#Creating a tibble
intr.surf.data.s <- tibble(ABV = unlist(expand.grid(seq(0, 100, 1), seq(0, 200, 5))[1]),
                          Sweet = unlist(expand.grid(seq(0, 100, 1), seq(0, 200, 5))[2]))

#Adding some prediction points
intr.surf.data.s <- mutate(intr.surf.data.s,
                          main.hat = predict(flavor_lm.s, intr.surf.data.s),
                          intr.hat = predict(flavor_lm_inter.s, intr.surf.data.s))

#Visualizing the surfaces
surf.main.s <- ggplot(intr.surf.data.s, aes(ABV, Sweet)) +
               geom_contour_filled(aes(z = main.hat)) +
               labs(subtitle = "Main Effects")  +
               guides(fill=guide_legend(title="Ratings"))
```

```
surf.intr.s <- ggplot(intr.surf.data.s, aes(ABV, Sweet)) +
               geom_contour_filled(aes(z = intr.hat)) +
               labs(subtitle = "Interaction Effects") +
               guides(fill=guide_legend(title="Ratings"))

grid.arrange(surf.main.s, surf.intr.s, nrow = 2)
```




```
#Visualizing the predictions as constant ABV levels
(effect.s <- filter(intr.surf.data.s, ABV %in% c(2, 3, 4, 12, 13, 17 , 18, 26, 27, 28)) %>%
  mutate(ABV = factor(ABV)) %>%
  ggplot() +
  geom_line(aes(Sweet, main.hat, colour = ABV), size = 1) +
  geom_line(aes(Sweet, intr.hat, colour = ABV), linetype = "dashed", size = 1) +     #, show.legend = FA
  ylim(3,6) +
  ylab("Rating"))
```



Malty Flavor

```
#Creating linear model for Malty flavor:
flavor_lm.m <- lm(rating ~ ABV + Malty, beer_data)
summary(flavor_lm.m)
```

```
##
## Call:
## lm(formula = rating ~ ABV + Malty, data = beer_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1210 -0.1520  0.0417  0.2235  1.0655
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.2530643  0.0163266 199.249  < 2e-16 ***
## ABV         0.0666807  0.0021905  30.441  < 2e-16 ***
## Malty       0.0009474  0.0001238   7.653 2.31e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4043 on 5553 degrees of freedom
## Multiple R-squared:  0.1668, Adjusted R-squared:  0.1665
## F-statistic: 555.7 on 2 and 5553 DF,  p-value: < 2.2e-16
```

```
#Creating linear model for with both flavors included
flavor_lm.m.s <- lm(rating ~ ABV + Malty + Sweet, beer_data)
```

```r
vif(flavor_lm.s)
```

```
##      ABV    Sweet
## 1.192518 1.192518
```

```r
vif(flavor_lm.m)
```

```
##      ABV    Malty
## 1.036326 1.036326
```

```r
vif(flavor_lm.m.s)
```

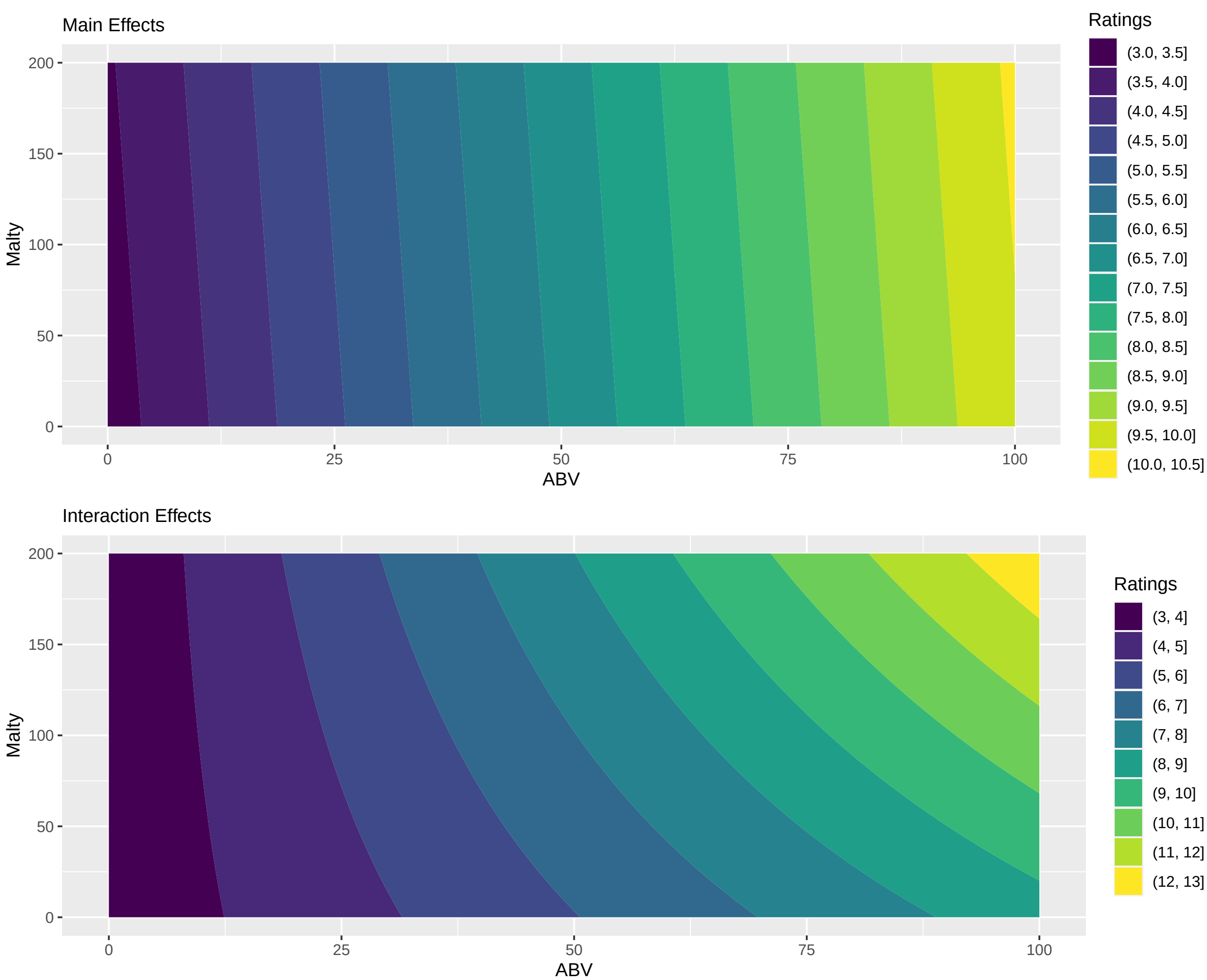
```
##      ABV    Malty    Sweet
## 1.195358 1.455948 1.675384
```

```r
##Creating linear model for Malty flavor with interaction:
flavor_lm_inter.m <- lm(rating ~ ABV * Malty, beer_data)
summary(flavor_lm_inter.m)
```

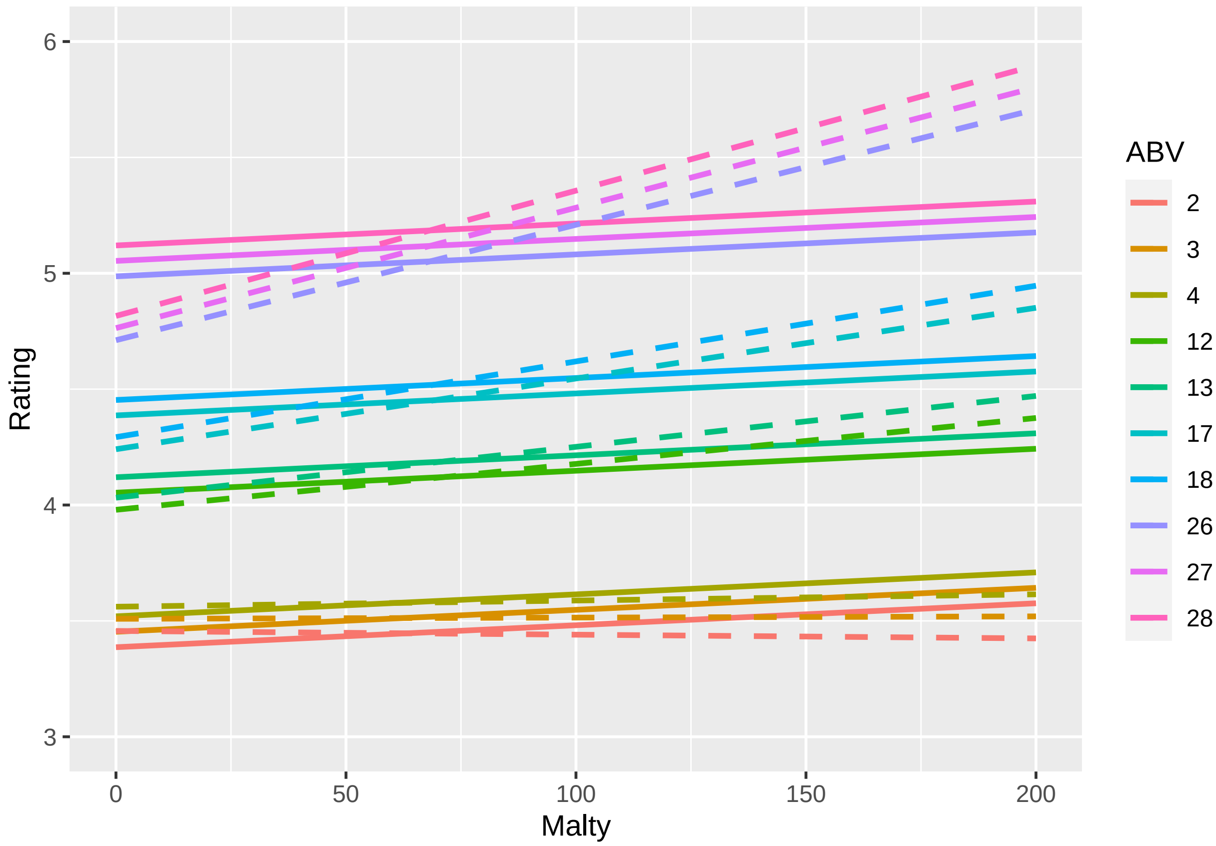
```
##
## Call:
## lm(formula = rating ~ ABV * Malty, data = beer_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4349 -0.1510  0.0430  0.2277  1.0475
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.352e+00  2.547e-02 131.606  < 2e-16 ***
## ABV          5.228e-02  3.590e-03  14.560  < 2e-16 ***
## Malty       -5.897e-04  3.281e-04  -1.797   0.0724 .
## ABV:Malty    2.142e-04  4.236e-05   5.057  4.4e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4034 on 5552 degrees of freedom
## Multiple R-squared:  0.1706, Adjusted R-squared:  0.1701
## F-statistic: 380.6 on 3 and 5552 DF,  p-value: < 2.2e-16
```

```r
#Creating a tibble
intr.surf.data.m <- tibble(ABV = unlist(expand.grid(seq(0, 100, 1), seq(0, 200, 5))[1]),
                         Malty = unlist(expand.grid(seq(0, 100, 1), seq(0, 200, 5))[2]))

intr.surf.data.m <- mutate(intr.surf.data.m,
                         main.hat = predict(flavor_lm.m, intr.surf.data.m),
                         intr.hat = predict(flavor_lm_inter.m, intr.surf.data.m))

#Visualizing the surfaces
surf.main.m <- ggplot(intr.surf.data.m, aes(ABV, Malty)) +
               geom_contour_filled(aes(z = main.hat)) +
               labs(subtitle = "Main Effects")  +
               guides(fill=guide_legend(title="Ratings"))

surf.intr.m <- ggplot(intr.surf.data.m, aes(ABV, Malty)) +
               geom_contour_filled(aes(z = intr.hat)) +
               labs(subtitle = "Interaction Effects") +
```

```
                guides(fill=guide_legend(title="Ratings"))

grid.arrange(surf.main.m, surf.intr.m, nrow = 2)
```




```
#Visualizing the predictions as constant ABV levels
(effect.m <- filter(intr.surf.data.m, ABV %in% c(2, 3, 4, 12, 13, 17 , 18, 26, 27, 28)) %>%
  mutate(ABV = factor(ABV)) %>%
  ggplot() +
  geom_line(aes(Malty, main.hat, colour = ABV), size = 1) +
  geom_line(aes(Malty, intr.hat, colour = ABV), linetype = "dashed", size = 1) +
  ylim(3,6) +
  ylab("Rating"))
```



Final Comparison

Part 2: Report

Finding 1: Beers receive a higher rating if it has a higher ABV and receive a lower rating if it has a lower ABV

In order to find whether, on average, a beer receives a higher rating if it has a higher or lower ABV, we looked into the possible correlation between Ratings and ABV.

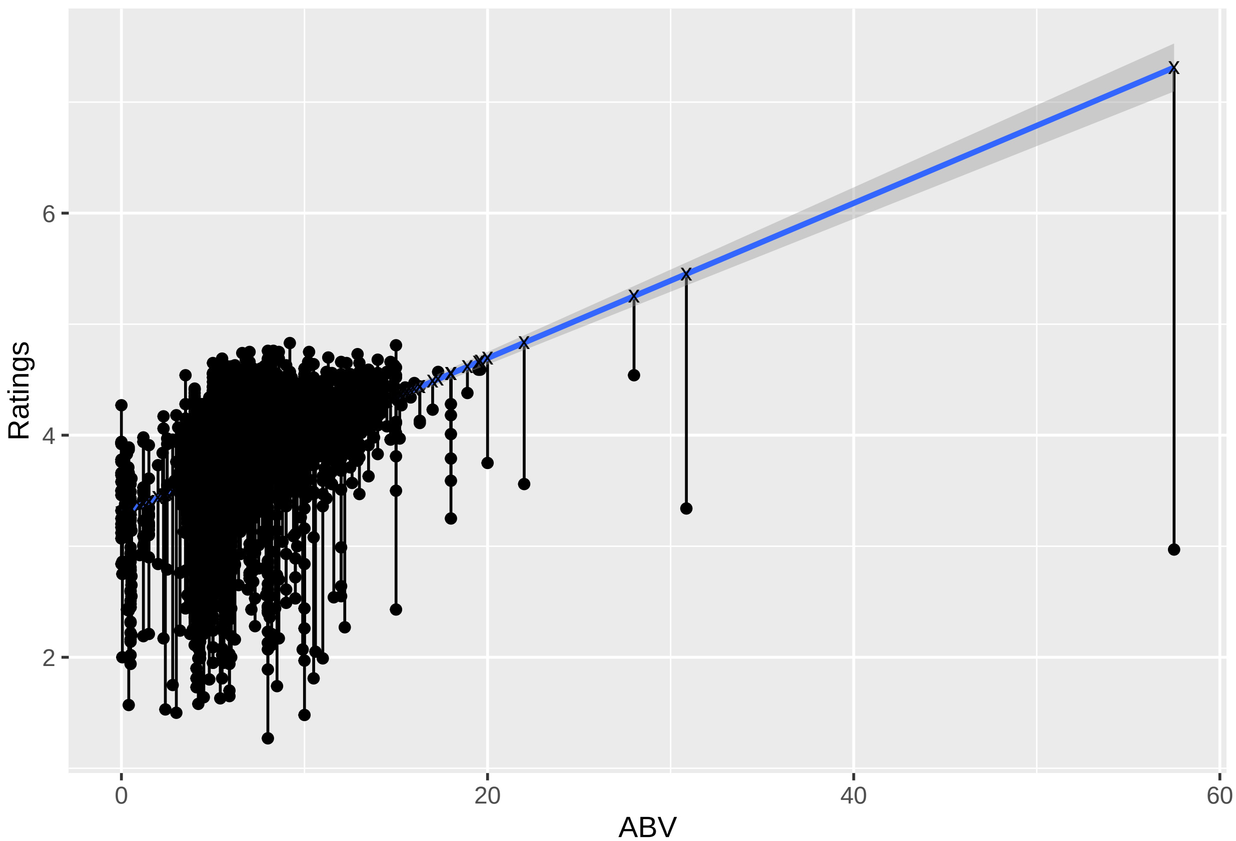
The test showed evidence of significant data-based multicollinearity between these variables. Specifically, we found a significant ($p = 0$) correlation of 0.4 between Ratings and ABV. Thus, we used a linear regression to examine whether this relationship is significant in a linear model.

```
##
## Call:
## lm(formula = rating ~ ABV, data = beer_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3418 -0.1463  0.0526  0.2322  1.0088
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.297235   0.015351  214.79   <2e-16 ***
## ABV         0.069819   0.002163   32.28   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.4064 on 5554 degrees of freedom
## Multiple R-squared:  0.158,  Adjusted R-squared:  0.1578
## F-statistic:  1042 on 1 and 5554 DF,  p-value: < 2.2e-16
```

The model shows that there are **0.07 extra rating points for every increase in ABV**. This increase is **significantly different from zero**, **t(5554)=32.28, p<.0001.**

We checked how using this linear model would be effective in predicting ratings by comparing against the observed ratings, ie, the residuals. The line that minimizes the square of the residuals has been chosen.



Hence, due to the nature of the positive correlation between rating and ABV, **we can conclude by saying that, on average, a beer with higher ABV has higher rating and beer with lower ABV has lower rating.** The figure shows the ability of the model to predict beer ratings based on ABV for newer data.

Finding 2: Interactive effect of Sweet and Malty flavours

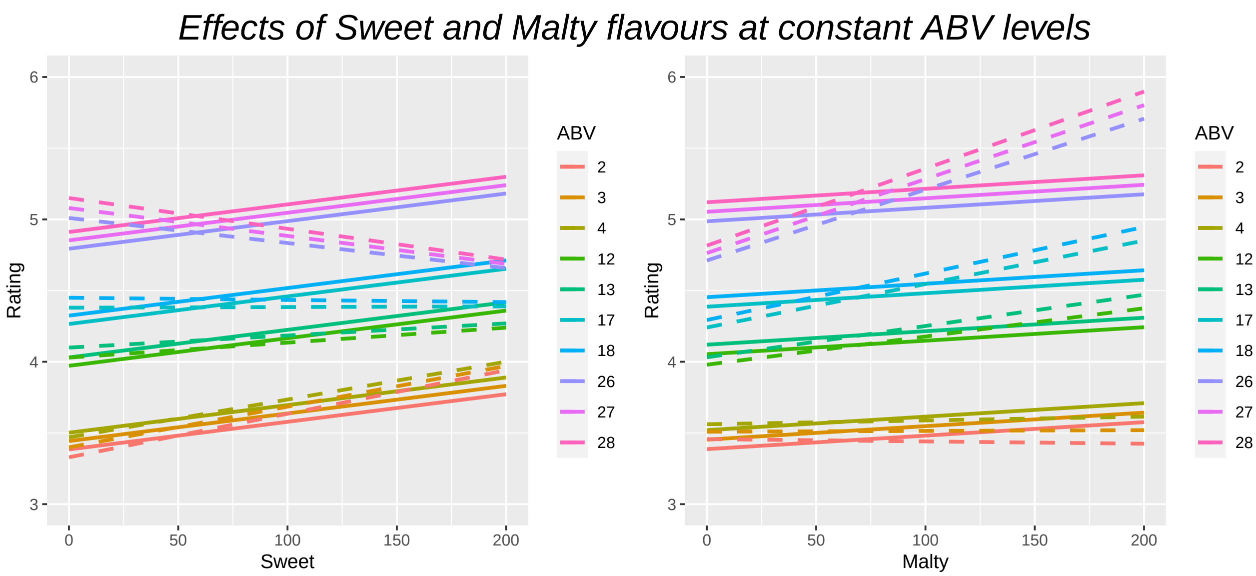
In order to investigate, we created multiple linear regression models, where we looked into:

1. Regression model with main effects of ABV and Sweet flavor.
2. Regression model with main effects of ABV and Malty flavor.
3. Regression model with interaction between ABV and Sweet flavor.
4. Regression model with interaction between ABV and Malty flavor.

From the main effects model, we have found that the **ratings increase** by a **statistically significant 0.06, t(5553)=25.18, p<.0001**, for every extra ABV value, holding the **Sweetness measure constant**. On the other hand, when **controlling for the ABV values**, the **ratings increased** by **0.002 for every extra unit of sweetness**, which is **significantly different from zero t(5553)=11.83, p<.0001**.

Again, the ratings increase by a **statistically significant 0.07, t(5553)=30.44, p<0.0001**, for every extra ABV value, holding the **Maltiness measure constant**. On the other hand, when controlling for the ABV values, the **ratings increased by 0.001 for every extra unit of maltiness**, which is **significantly different from zero t(5553)=7.65, p<.0001.**

However, our findings indicate that the **model including the interaction between the flavors and ABV are significantly better** and we will focus our interpretations on that. We used the models to enter multiple predictors to give us the best possible understanding of the effect of the two flavors (Sweet and Malty) with ABV held constant at various levels. The following figure shows us how the ratings will behave accordingly:



In the above figure, the **dashed lines show predictions based upon interaction between the interaction model**, and **the solid lines show predictions based upon the main effects models**. In the **main effects model**, the slopes of the flavors against Rating are always parallel for all different values of ABV. In the **interaction models**, the slopes of Sweet against Rating are steeper for low values of ABV and shallower for high values of ABV, while the slopes of Malty against Rating are shallower for low values of ABV and shallower for high values of ABV.

From the analysis, we can conclude that:

1. In order to maximize ratings, the **company should use more Malty flavor with beers having high ABV and use more Sweet flavor in beers with low ABV.**
2. As the interaction model including ABV and Malty shows that at each higher ABV levels there is greater positive correlation between ratings and Malty, this flavor should be used if they are **creating a high ABV beer.**
3. As the interaction model including ABV and Sweet shows that at each lower ABV levels there is greater positive correlation between ratings and Sweet, this flavor should be used if they are **creating a low ABV beer.**

This is to certify that the work I am submitting is my own. All external references and sources are clearly acknowledged and identified within the contents. I am aware of the University of Warwick regulation concerning plagiarism and collusion.

No substantial part(s) of the work submitted here has also been submitted by me in other assessments for accredited courses of study, and I acknowledge that if this has

been done an appropriate reduction in the mark I might otherwise have received will be made.