Business Statistics

Tatiana Hernandez

Table of Contents

```
knitr::opts_chunk$set(echo = TRUE)

#install.packages("kableExtra")
#install.packages("Rmisc")
#install.packages("emmeans")
#install.packages("gridExtra")
#install.packages("Hmisc")
#install.packages("afex")
#install.packages("car")
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse
2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.4.4     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.0
## ✓ purrr     1.0.2
## ── Conflicts ──────────────────────────────────────────
tidyverse_conflicts() ──
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors

library(kableExtra)

## Warning: package 'kableExtra' was built under R version 4.3.3

##
## Attaching package: 'kableExtra'
##
## The following object is masked from 'package:dplyr':
##
##     group_rows

library(tidyr)
library(dplyr)
library(Rmisc) # for CI()
```

```
## Warning: package 'Rmisc' was built under R version 4.3.3

## Loading required package: lattice
## Loading required package: plyr
## ---------------------------------------------------------------------------
----
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first,
then dplyr:
## library(plyr); library(dplyr)
## ---------------------------------------------------------------------------
----
##
## Attaching package: 'plyr'
##
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
##
## The following object is masked from 'package:purrr':
##
##     compact
```

```
library(emmeans) # for emmeans() and pairs()
```

```
## Warning: package 'emmeans' was built under R version 4.3.3

## Welcome to emmeans.
## Caution: You lose important information if you filter this package's
results.
## See '? untidy'
```

```
library(gridExtra) # for grid.arrange()
```

```
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(Hmisc) # for correlation functions
```

```
## Warning: package 'Hmisc' was built under R version 4.3.3

##
## Attaching package: 'Hmisc'
##
## The following objects are masked from 'package:plyr':
##
##     is.discrete, summarize
```

```
##
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
##
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```r
library(ggplot2)
library(afex)
```

```
## Warning: package 'afex' was built under R version 4.3.3
```

```
## Loading required package: lme4
```

```
## Warning: package 'lme4' was built under R version 4.3.3
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Warning in check_dep_version(): ABI version mismatch:
## lme4 was built with Matrix ABI version 1
## Current Matrix ABI version is 0
## Please re-install lme4 from source or restore original 'Matrix' package
```

```
## ************
## Welcome to afex. For support visit: http://afex.singmann.science/
## - Functions for ANOVAs: aov_car(), aov_ez(), and aov_4()
## - Methods for calculating p-values with mixed(): 'S', 'KR', 'LRT', and
'PB'
## - 'afex_aov' and 'mixed' objects can be passed to emmeans() for follow-up
tests
## - Get and set global package options with: afex_options()
## - Set sum-to-zero contrasts globally: set_sum_contrasts()
## - For example analyses see: browseVignettes("afex")
## ************
##
## Attaching package: 'afex'
##
## The following object is masked from 'package:lme4':
##
##     lmer
```

```r
library(car)
```

```
## Warning: package 'car' was built under R version 4.3.3
## Loading required package: carData

## Warning: package 'carData' was built under R version 4.3.3

## 
## Attaching package: 'car'
## 
## The following object is masked from 'package:dplyr':
## 
##     recode
## 
## The following object is masked from 'package:purrr':
## 
##     some

options(width=100)
```

Question 1

Section 1

```
# Reading the database
tutoring_data <- read_csv("tutoring_test_data.csv")
```

Data Dictionary

Variable	Description
student_ID	student ID
absences	Attendance in regular classes as a proportion of class time that was missed (%) in regular classes
score.t1	Scores obtained at the beginning of the academic year (range 0-100)
score.t2	Scores obtained at the end of the academic year (range 0-100)
tutoring	Whether the student received tutoring classes (FALSE /TRUE)

Types of variables:

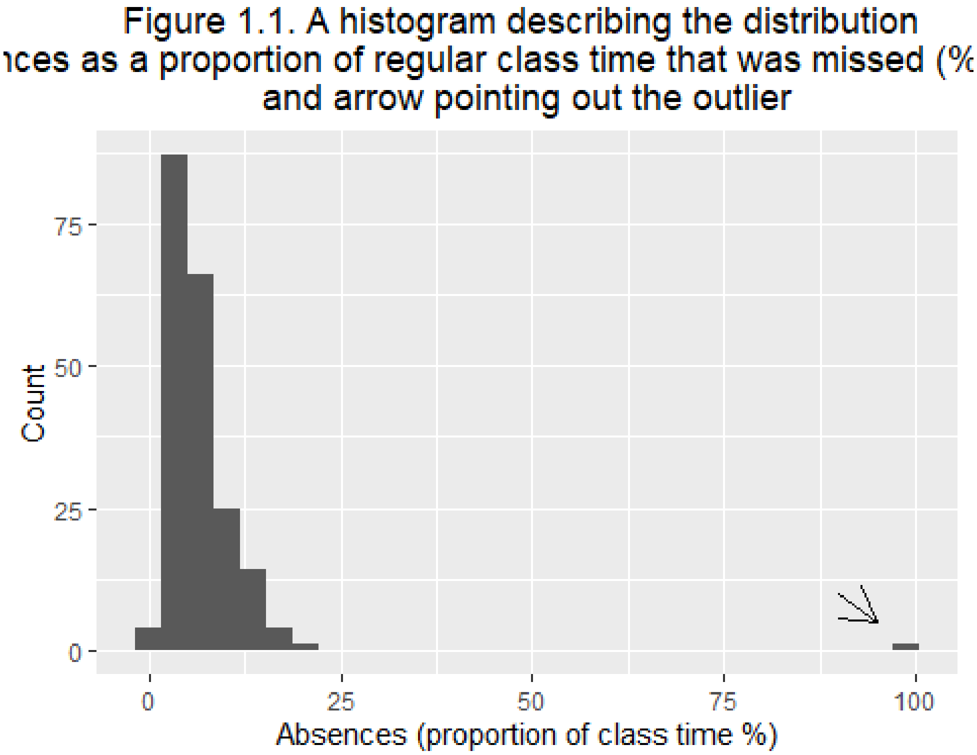
1. Typeless: student_ID

2. Numerical:absences,score.t1, and score.t2

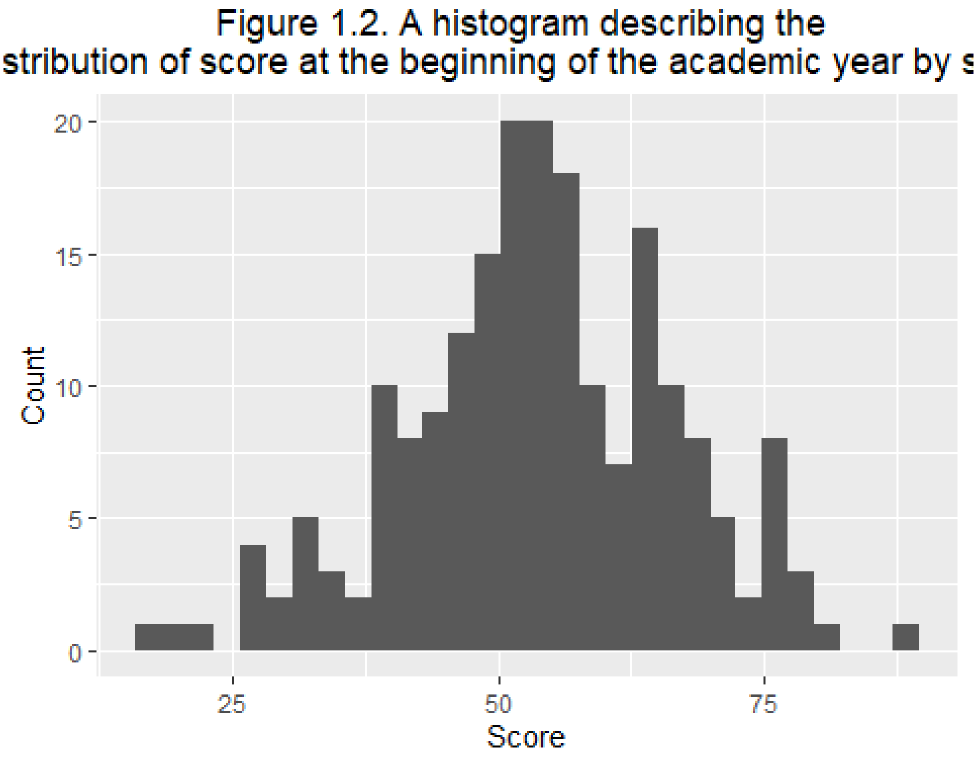
3. Categorical: tutoring.

```
#Checking our data summary

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

#Reemplazing NA value with 0 from the variable score.t2

tutoring_data$score.t2[is.na(tutoring_data$score.t2)] <- 0

#Before performing the required analysis, plot and examine our given
variables (numerical variables:absences,score.t1,score.t2)

tutoring_data %>% ggplot() + geom_histogram(aes(absences))+labs(x="Absences
(proportion of class time %)", y="Count",title="Figure 1.1. A histogram
describing the distribution
of Absences as a proportion of regular class time that was missed (%) by
student,
and arrow pointing out the outlier")+theme(plot.title = element_text(hjust =
0.5))+ geom_segment(aes(x=90, xend=95, y=10, yend=5),arrow = arrow(length=
unit(0.5,"cm")))
```

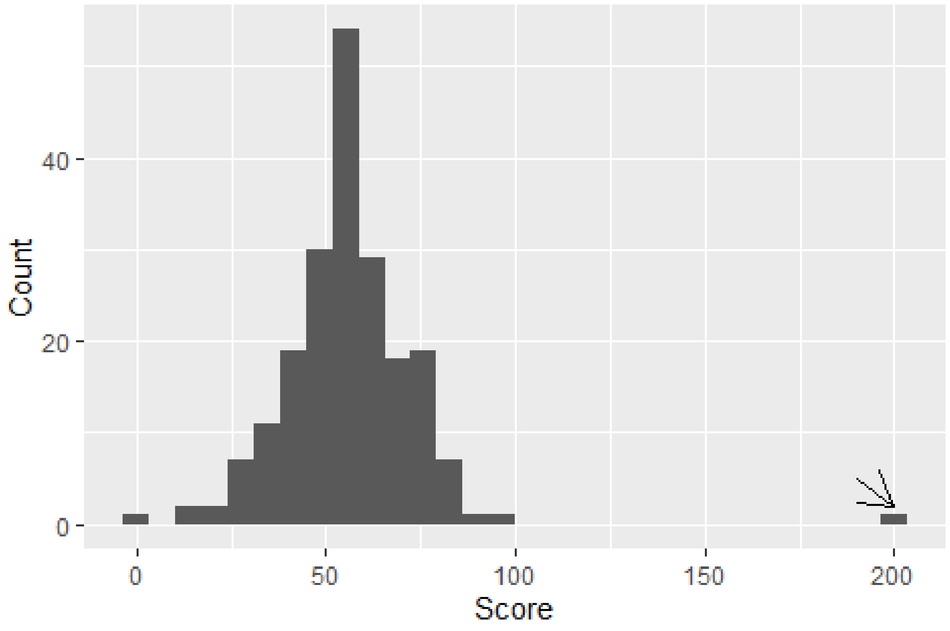


```
tutoring_data %>% ggplot() + geom_histogram(aes(score.t1))+labs(x="Score",
y="Count",title="Figure 1.2. A histogram describing the
distribution of score at the beginning of the academic year by
student")+theme(plot.title = element_text(hjust = 0.5))
```



```
tutoring_data %>% ggplot() + geom_histogram(aes(score.t2))+labs(x="Score",
y="Count",title="Figure 1.3. A histogram describing the
distribution of score at the end of the academic year by student,
and arrow pointing out the outlier")+theme(plot.title = element_text(hjust =
0.5))+ geom_segment(aes(x=190, xend=200, y=5, yend=2),arrow = arrow(length=
unit(0.5,"cm")))
```

Figure 1.3. A histogram describing the
distribution of score at the end of the academic year by stud
and arrow pointing out the outlier



```
# Checking categorical variables (tutoring) by using table() function:
# Having considered the total number of observations: 202. We have 101
students who received tutoring whereas 101 who did not, then the proportion
is 50% and 50% respectively.

table(tutoring_data$tutoring)

##
## FALSE  TRUE
##   101   101
```

Having plotted each variable we have the following:

```
#it seems that there was a student (ID 201) who was absent in regular classes
during the whole academic year (outlier), but at the same time took tutoring
classes.

tutoring_data %>% filter(absences > 90)

## # A tibble: 1 × 5
##   student_ID tutoring absences score.t1 score.t2
##        <dbl> <lgl>       <dbl>    <dbl>    <dbl>
## 1        201 TRUE          100     53.7        0

#Revising the outlier found on score.t2 (student ID number 202). Assuming a
maximum score of 100 which a student can obtain.
```

```
tutoring_data %>% filter(score.t2 >100)

## # A tibble: 1 × 5
##   student_ID tutoring absences score.t1 score.t2
##        <dbl> <lgl>       <dbl>    <dbl>    <dbl>
## 1        202 FALSE        6.00     62.2      200
```

Treating Outliers

```
#Identifying row position of the outlier (student ID 201)

which(tutoring_data$absences >90)

## [1] 12

#Removing absenteeism equal to 100 from the student number 201(row 12)

tutoring_data <- tutoring_data[-12,]

#Identifying row position of the outlier (student ID 202)

which(tutoring_data$score.t2 == 200)

## [1] 151

#Removing score at the end of the academic year equal to 200 from the student
number 202 (row 152)

tutoring_data <- tutoring_data[-151,]

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.: 50.75   FALSE:100       1st Qu.: 3.600   1st Qu.:46.30   1st
Qu.:46.45
##  Median :100.50   TRUE :100       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

#Re-coding the column "tutoring" into an integer. Tutoring TRUE = 1 and
Tutoring FALSE = 0

tutoring_data$tutoring[tutoring_data$tutoring=="FALSE"] <- "0"
```

```
tutoring_data$tutoring[tutoring_data$tutoring=="TRUE"] <- "1"

#Finally ,let's save the tutoring variable into a factor since variables for
use in lm() should either be numeric or factors. Categorical data (factor of
2 levels FALSE AND TRUE)

tutoring_data <- mutate(tutoring_data, tutoring = as.factor(tutoring))
str(pull(tutoring_data, tutoring))

##  Factor w/ 2 levels "0","1": 2 2 2 1 2 2 2 2 1 1 ...

#Removing the student ID variable since it is not relevant for an analysis

tutoring_data$student_ID <- NULL
```

The Request

1 Check whether the students allocated to the tutored and non-tutored groups had similar or different average test scores before the tutoring scheme began.

Null hypothesis significance testing (NHST)

As it can be seen from the previous histograms the score.t1 tends to be normally distributed, then let's start performing our t-test.

```
#T-test comparing test score before the tutoring scheme between tutored and
non-tutored students

test_t1_vs_tutoring <- tutoring_data %>% filter(tutoring %in% c("0", "1"))

# Running t-test

t.test(score.t1~tutoring, data=test_t1_vs_tutoring)

##
##  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 0 and group
1 is not equal to 0
## 95 percent confidence interval:
##  -5.433861  1.665720
## sample estimates:
## mean in group 0 mean in group 1
##        52.90345        54.78753
```

Creating linear regression model for Score at the beginning of the academic year and tutoring as independent variable

$$Score.t1 = \beta_{Intercept} + \beta_{tutoring} \times tutoring + \epsilon$$

```
m.scorest1.by.tutoring <- lm(score.t1~tutoring, data= tutoring_data)

#Summary of our lm model
summary(m.scorest1.by.tutoring)

##
## Call:
## lm(formula = score.t1 ~ tutoring, data = tutoring_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -35.595  -7.414  -0.194   9.067  35.554
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   52.903      1.273  41.565   <2e-16 ***
## tutoring1      1.884      1.800   1.047    0.297
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.73 on 198 degrees of freedom
## Multiple R-squared:  0.005503,   Adjusted R-squared:  0.0004801
## F-statistic: 1.096 on 1 and 198 DF,  p-value: 0.2965
```

Estimation approach

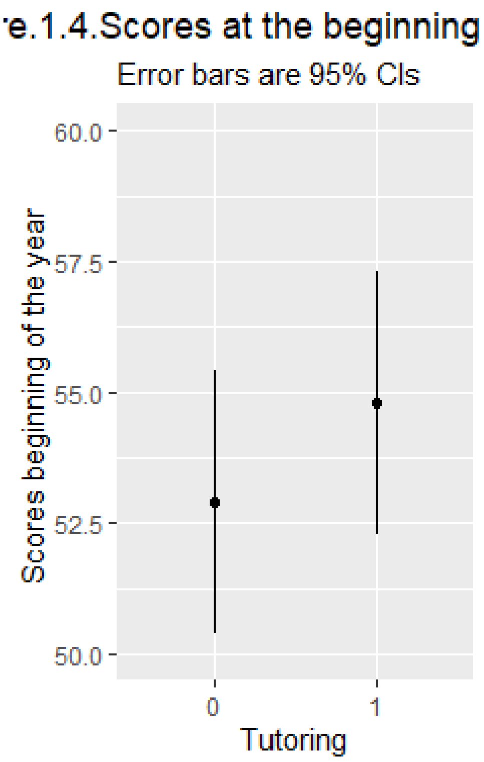
Here we make a model object `m.scorest1.by.tutoring` and then use emmeans to get confidence intervals for the scores at the beginning of the year for tutored and non-tutored pupils and `pairs()` and `confint()` to *contrast* the tutoring and non-tutoring.

```
#calculating the means and confidence intervals scores at the beginning of
the year for each tutoring category (1,0)

(  m.scorest1.by.tutoring.emm <- emmeans(m.scorest1.by.tutoring, ~tutoring)
)

##  tutoring emmean   SE  df lower.CL upper.CL
##  0          52.9 1.27 198     50.4     55.4
##  1          54.8 1.27 198     52.3     57.3
##
## Confidence level used: 0.95

#Contrast between scores at the beginning and tutoring
(  m.scorest1.by.tutoring.contrast <-
confint(pairs(m.scorest1.by.tutoring.emm))  )

##  contrast              estimate  SE  df lower.CL upper.CL
##  tutoring0 - tutoring1    -1.88 1.8 198    -5.43     1.67
```

```
##
## Confidence level used: 0.95

#Calculating the confidence interval for our tutoring coefficient in our
linear model which tries to explain the variance of score.t1 by using
tutoring as a predictor.

cbind(coefficients=coef(m.scorest1.by.tutoring),confint(m.scorest1.by.tutorin
g))

##             coefficients     2.5 %    97.5 %
## (Intercept)    52.903455 50.393488 55.413421
## tutoring1       1.884071 -1.665557  5.433699
```

Reporting 95% confidence interval difference:

```
grid.arrange(
    ggplot(summary(m.scorest1.by.tutoring.emm), aes(x=tutoring, y=emmean,
ymin=lower.CL, ymax=upper.CL)) +
        geom_point() + geom_linerange() +
        labs(y="Scores beginning of the year ", x="Tutoring", subtitle="Error
bars are 95% CIs", title="Figure.1.4.Scores at the beginning of the year") +
ylim(50,60)+theme(plot.title = element_text(hjust = 0.5)),

    ggplot(m.scorest1.by.tutoring.contrast, aes(x=contrast, y=estimate,
ymin=lower.CL, ymax=upper.CL)) +
        geom_point() + geom_linerange() +
        labs(y="Difference in scores beginning of the year ", x="Contrast",
subtitle="Error bars are 95% CIs", title="Figure.1.5.Difference in scores
             beginning of the year") + ylim(-6,2) +
        geom_hline(yintercept=0, lty=2)+theme(plot.title = element_text(hjust
= 0.5)),
    ncol=2
)
```


'e.1.4.Scores at the beginning

Error bars are 95% CIs

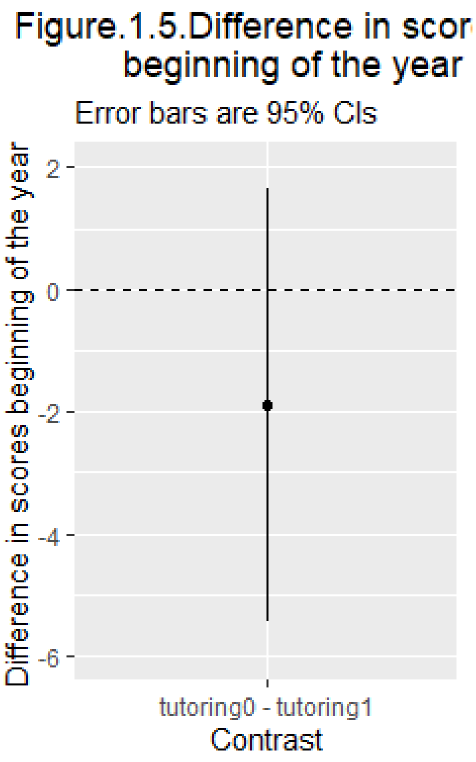


Figure.1.5.Difference in scor
beginning of the year

Error bars are 95% CIs

2 Did the tutored and non-tutored students have similar or different rates of absences on average?

Null hypothesis significance testing (NHST)

As it can be seen from the previous histograms the variable absences tends to be normally distributed, then let's start performing our t-test.

```
#T-test comparing absence proportions (regular classes) between tutored and non-tutored students

absence_vs_tutoring <- tutoring_data %>% filter(tutoring %in% c("0", "1"))


t.test(absences~tutoring, data=absence_vs_tutoring)

##
##  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 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -1.440721  0.480721
## sample estimates:
```

```
## mean in group 0 mean in group 1
##           6.312           6.792
```

Creating linear regression model for absences and tutoring as independent variable

$$absences = \beta_{Intercept} + \beta_{tutoring} \times tutoring + \epsilon$$

```
m.absences.by.tutoring <- lm(absences~tutoring, data=absence_vs_tutoring)

#Summary of our lm model

summary(m.absences.by.tutoring)

## 
## Call:
## lm(formula = absences ~ tutoring, data = absence_vs_tutoring)
## 
## 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 ***
## tutoring1     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
```

Estimation approach

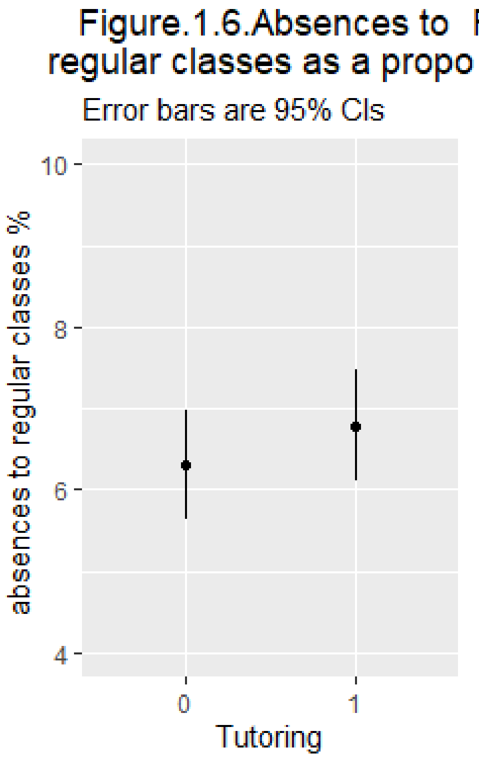
Here we make a model object `m.absences.by.tutoring` and then use emmeans to get confidence intervals for the absences for tutored and non-tutored pupils and `pairs()` and `confint()` to *contrast* the tutoring and non-tutoring.

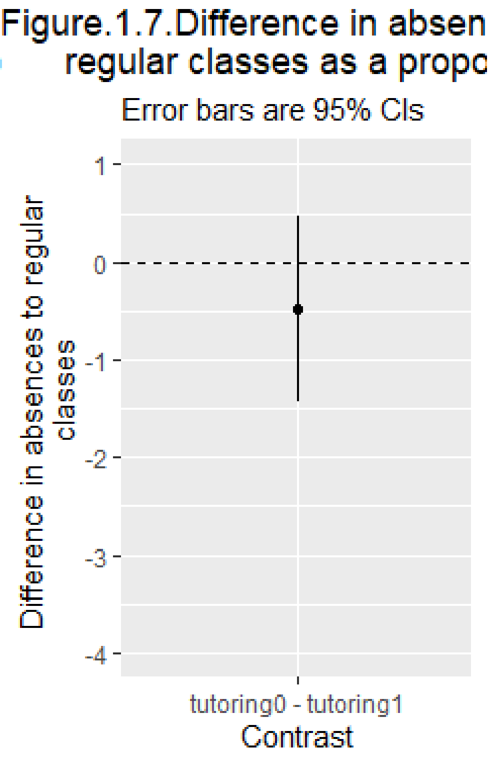
```
#calculating the means and confidence intervals for the proportion of
absences for each tutoring category (1,0)

(  m.absences.by.tutoring.emm <- emmeans(m.absences.by.tutoring, ~tutoring)
)

##  tutoring emmean    SE  df lower.CL upper.CL
##  0          6.31 0.344 198     5.63     6.99
##  1          6.79 0.344 198     6.11     7.47
## 
## Confidence level used: 0.95
```

```
#Contrast between tutoring absences 0 and 1

(  m.absences.by.tutoring.contrast <-
confint(pairs(m.absences.by.tutoring.emm))  )

##  contrast              estimate    SE  df lower.CL upper.CL
##  tutoring0 - tutoring1    -0.48 0.487 198    -1.44    0.481
##
## Confidence level used: 0.95

#Calculating the confidence interval for our tutoring coefficient tutoring in
our linear model which tries to explain the variance of absences by using
tutoring as a predictor.

cbind(coefficients=coef(m.absences.by.tutoring),confint(m.absences.by.tutorin
g))

##             coefficients      2.5 %   97.5 %
## (Intercept)        6.312  5.6326761 6.991324
## tutoring1          0.480 -0.4807091 1.440709
```

Reporting 95% confidence interval difference:

```
grid.arrange(
    ggplot(summary(m.absences.by.tutoring.emm), aes(x=tutoring, y=emmean,
ymin=lower.CL, ymax=upper.CL)) +
        geom_point() + geom_linerange() +
        labs(y="absences to regular classes % ", x="Tutoring",
subtitle="Error bars are 95% CIs", title="Figure.1.6.Absences to
             regular classes as a proportion") + ylim(4,10)+theme(plot.title
= element_text(hjust = 0.5)),

    ggplot(m.absences.by.tutoring.contrast, aes(x=contrast, y=estimate,
ymin=lower.CL, ymax=upper.CL)) +
        geom_point() + geom_linerange() +
        labs(y="Difference in absences to regular
            classes", x="Contrast", subtitle="Error bars are 95% CIs",
title="Figure.1.7.Difference in absences to
            regular classes as a proportion ") + ylim(-4,1) +
        geom_hline(yintercept=0, lty=2)+theme(plot.title = element_text(hjust
= 0.5)),
    ncol=2
)
```

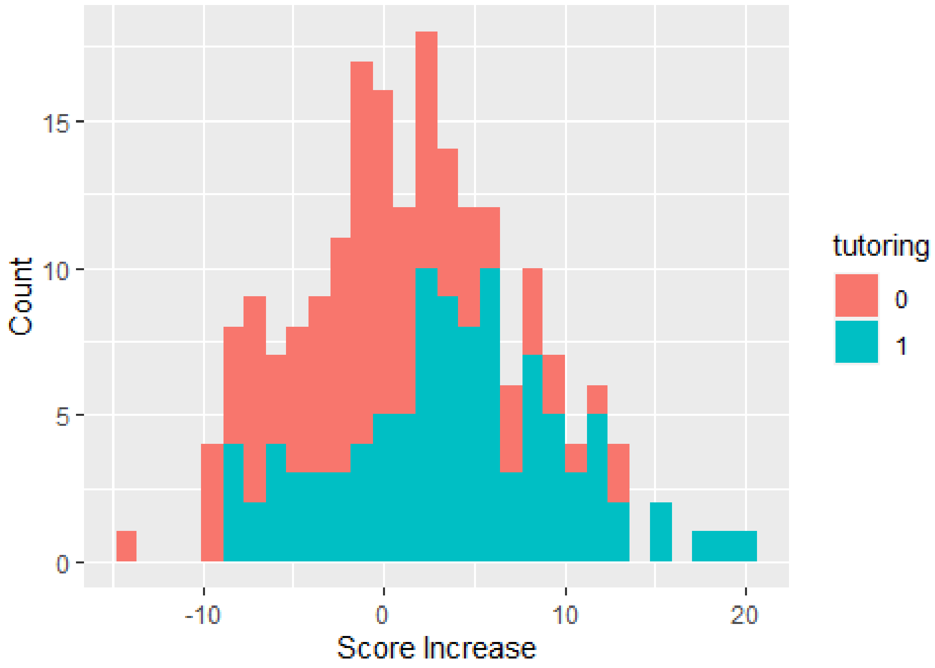




***3** Did the tutored students show an increase in their scores compared to the students who did not receive tutoring?*

```
#Calculating the difference between scores at the beginning of the academic
year and at the end of the academic year as a new variable called
Score_Increase

tutoring_data <- tutoring_data %>% mutate(Score_Increase = score.t2-score.t1)

#Plotting and checking our new calculated variable by plotting a histogram:

tutoring_data %>% ggplot() + geom_histogram(aes(x=Score_Increase,
fill=tutoring))+labs(x="Score Increase", y="Count",title="Figure 1.8. A
histogram describing the distribution
of the new variable called Score Increase by tutoring")+theme(plot.title =
element_text(hjust = 0.5))

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

Figure 1.8. A histogram describing the distribution of the new variable called Score Increase by tutoring



Null hypothesis significance testing (NHST)

As it can be seen from the previous histograms that our new variable Score_Increase tends to be normally distributed, then let's start performing our t-test.

```
#T-test comparing score at the beginning and score at the end difference
(score.t2-score.t1) between tutored and non-tutored students

scoret1t2_vs_tutoring <- tutoring_data %>% filter(tutoring %in% c("0", "1"))


t.test(Score_Increase~tutoring, data=scoret1t2_vs_tutoring)

##
##  Welch Two Sample t-test
##
## data:  Score_Increase by tutoring
## t = -5.0811, df = 194.29, p-value = 8.78e-07
## alternative hypothesis: true difference in means between group 0 and group
1 is not equal to 0
## 95 percent confidence interval:
##  -5.837998 -2.573164
## sample estimates:
## mean in group 0 mean in group 1
##      -0.4399544       3.7656265
```

Creating linear regression model for Score Increase and tutoring as independent variable

$$Score_{I}ncrease = \beta_{Intercept} + \beta_{tutoring1} \times tutoring1 + \epsilon$$

tutoring1=TRUE

```
m.difference.by.tutoring <- lm(Score_Increase~tutoring,
data=scoret1t2_vs_tutoring)

#Summary of our lm model

summary(m.difference.by.tutoring)

##
## Call:
## lm(formula = Score_Increase ~ tutoring, data = scoret1t2_vs_tutoring)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.0850  -3.7445  -0.0812   3.2597  15.9767
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.4400     0.5853  -0.752    0.453
## tutoring1     4.2056     0.8277   5.081 8.65e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.853 on 198 degrees of freedom
## Multiple R-squared:  0.1154, Adjusted R-squared:  0.1109
## F-statistic: 25.82 on 1 and 198 DF,  p-value: 8.652e-07
```

Estimation approach

Here we make a model object `m.difference.by.tutoring` and then use emmeans to get confidence intervals for the difference in scores (t1 and t2) for tutored and non-tutored pupils and `pairs()` and `confint()` to *contrast* the tutoring and non-tutoring.

```
#Calculating the means and confidence intervals Score_Increase for each
tutoring category (1,0)

(  m.difference.by.tutoring.emm <- emmeans(m.difference.by.tutoring,
~tutoring)  )

##  tutoring emmean    SE  df lower.CL upper.CL
##  0         -0.44 0.585 198    -1.59    0.714
##  1          3.77 0.585 198     2.61    4.920
##
## Confidence level used: 0.95

#Contrast between tutoring 1 and 0 Score Increase
```

```
(  m.difference.by.tutoring.contrast <-
confint(pairs(m.difference.by.tutoring.emm)))
```

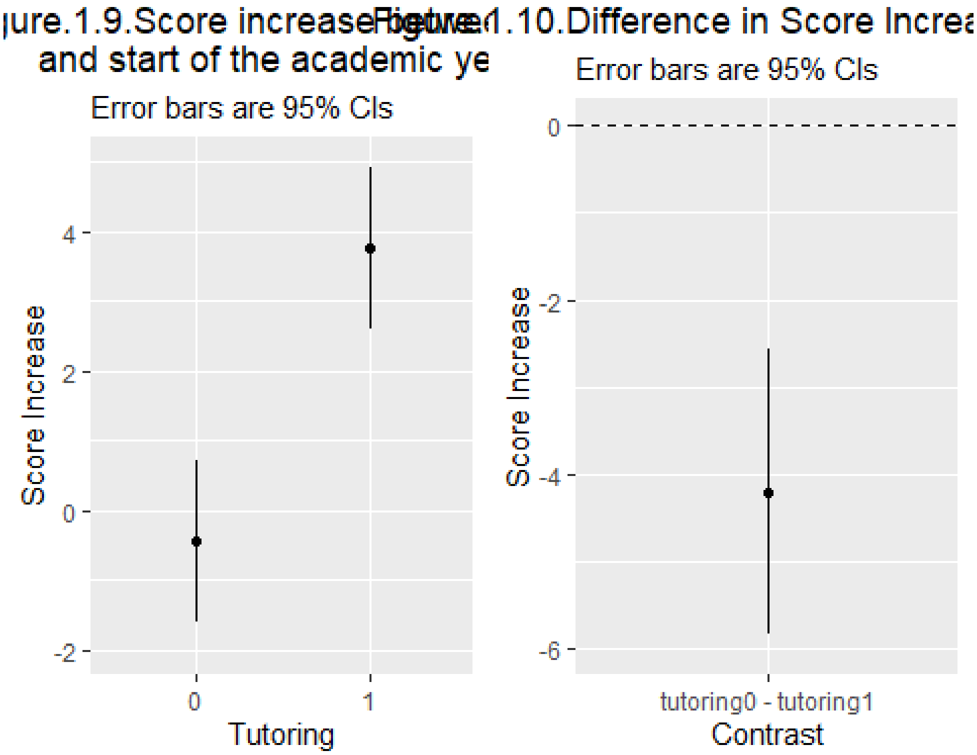
```
##  contrast              estimate    SE  df lower.CL upper.CL
##  tutoring0 - tutoring1    -4.21 0.828 198    -5.84    -2.57
##
## Confidence level used: 0.95
```

```
#Calculating the confidence interval for our coefficient for tutoring in our
linear model which tries to explain the variance of Score Increase by using
tutoring as a predictor.


cbind(coefficients=coef(m.difference.by.tutoring),confint(m.difference.by.tut
oring))
```

```
##              coefficients     2.5 %    97.5 %
## (Intercept)    -0.4399544 -1.594112 0.7142034
## tutoring1       4.2055809  2.573355 5.8378065
```

Reporting 95% confidence interval difference:

```
grid.arrange(
    ggplot(summary( m.difference.by.tutoring.emm), aes(x=tutoring, y=emmean,
ymin=lower.CL, ymax=upper.CL)) +
        geom_point() + geom_linerange() +
        labs(y="Score Increase", x="Tutoring", subtitle="Error bars are 95%
CIs", title="Figure.1.9.Score increase between end
and start of the academic year") + ylim(-2,5)+theme(plot.title =
element_text(hjust = 0.5)),

    ggplot(m.difference.by.tutoring.contrast, aes(x=contrast, y=estimate,
ymin=lower.CL, ymax=upper.CL)) +
        geom_point() + geom_linerange() +
        labs(y="Score Increase", x="Contrast", subtitle="Error bars are 95%
CIs", title="Figure.1.10.Difference in Score Increase contrast") + ylim(-6,0)
+
        geom_hline(yintercept=0, lty=2)+theme(plot.title = element_text(hjust
= 0.5)),
    ncol=2
)
```



4 was there any effect of absences on the change in scores?

we could see that our new variable Score Increase is normally distributed, then we can use the following approach and technique for calculating the correlation

NHSTing regression coefficients

Regression coefficients

The regression coefficient tells us about the best-fitting straight line through the data. The slope tells us about the effect a unit change in our independent variable (on the x axis which is absences) has on our dependent variable (on the y axis which is the Score Increase)

Creating a linear regression model for Score Increase and absences as independent variable

`Score_Increase~absences` means predict the outcome variable on the left using the predictor(s) on the right

This produces a model object that we are saving here as `m.scoreincrease.by.absences`

$$Score_{I}ncrease = \beta_{Intercept} + \beta_{absences} \times absences + \epsilon$$

```
m.scoreincrease.by.absences <- lm(Score_Increase~absences,
data=tutoring_data)
summary(m.scoreincrease.by.absences)
```

```
##
## Call:
## lm(formula = Score_Increase ~ absences, data = tutoring_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.5013  -4.0944  -0.0555   3.9418  17.9472
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1575     0.9466   2.279   0.0237 *
## absences     -0.0755     0.1280  -0.590   0.5558
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.217 on 198 degrees of freedom
## Multiple R-squared:  0.001756,   Adjusted R-squared:  -0.003286
## F-statistic: 0.3482 on 1 and 198 DF,  p-value: 0.5558
```

Estimation approach to regression coefficients

For the estimation approach we are calculating our 98% confidence interval:

```
cbind(coefficients=coef(m.scoreincrease.by.absences),confint(m.scoreincrease.
by.absences))
```

```
##             coefficients      2.5 %    97.5 %
## (Intercept)   2.15751104  0.2908417 4.0241803
## absences     -0.07549985 -0.3278114 0.1768117
```

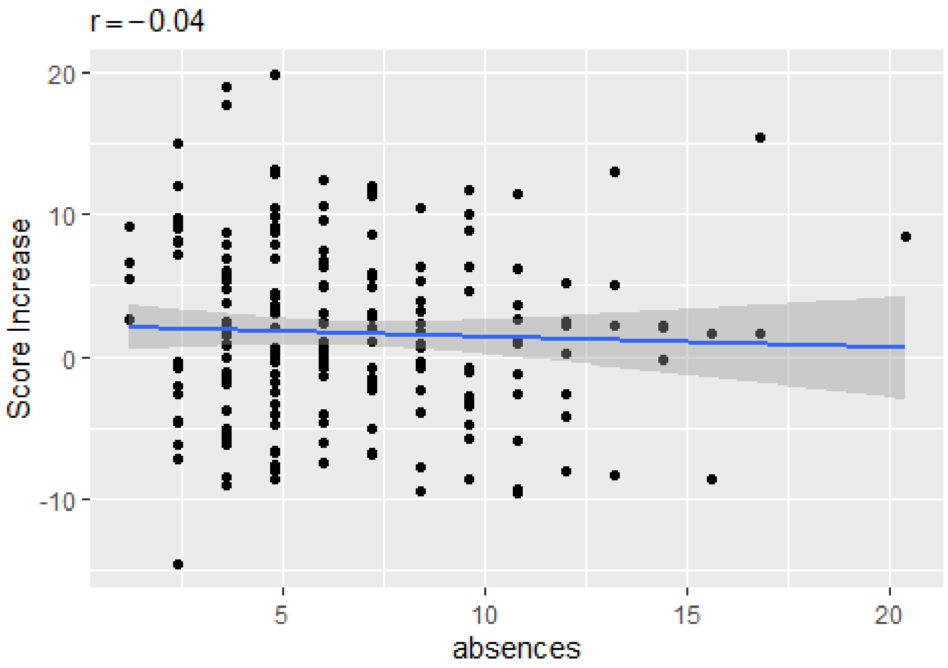
Plot showing a linear regression model between score difference and absences: correlation of -0.04 shows a very weak negative relationship between these variables (p value from the correlation matrix shows that this relationship is not significant).

In this case, the line is relatively flat, indicating no overall increase or decrease in score difference based on a change in absences. Besides that, the confidence intervals are wider compared to the size of the fluctuations in the trend line, and it includes zero. Therefore, the strongest conclusion is that there is no relationship.

```
ggplot(tutoring_data, aes(y=Score_Increase, x=absences)) + geom_point() +
labs(x="absences", y="Score Increase", title="Figure.1.11.linear regression
showing the relationship between Score Increase
and abscenses", subtitle = expression(r== -0.04)) + geom_smooth(method=lm)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

Figure.1.11.linear regression showing the relationship and abscenses



Correlation

The only two significant predictor for score difference is tutoring with 0.34 of relationship, and score at the end of the academic year as a structural correlation.

```
rcorr(as.matrix(tutoring_data))

##                tutoring absences score.t1 score.t2 Score_Increase
## tutoring           1.00     0.07     0.07     0.21           0.34
## absences           0.07     1.00    -0.34    -0.32          -0.04
## score.t1           0.07    -0.34     1.00     0.91           0.07
## score.t2           0.21    -0.32     0.91     1.00           0.49
## Score_Increase     0.34    -0.04     0.07     0.49           1.00
##
## n= 200
##
##
## P
##                tutoring absences score.t1 score.t2 Score_Increase
## tutoring                0.3257   0.2965   0.0029   0.0000
## absences       0.3257            0.0000   0.0000   0.5558
## score.t1       0.2965   0.0000            0.0000   0.2950
## score.t2       0.0029   0.0000   0.0000            0.0000
## Score_Increase 0.0000   0.5558   0.2950   0.0000
```

Comparing if our model for explaining Score increase improves by adding absences variable:

Comparing if our model for explaining score difference change improves by adding absences variable:The result is that it does not improve since absences has no significant effect on the score differences.$F(1,198) = 0.348, p > .05$

```
anova(m.scoreincrease.by.absences)
```

```
## Analysis of Variance Table
##
## Response: Score_Increase
##            Df Sum Sq Mean Sq F value Pr(>F)
## absences    1   13.5  13.459  0.3482 0.5558
## Residuals 198 7653.1  38.652
```

The estimation approach to regression coefficients

We can use `coef()` to get the coefficients and `confint()` to get confidence intervals for our coefficients

```
#Calculating the confidence interval for our absences coefficient in our
linear model which tries to explain the variance of score increase by using
absences as a predictor.

cbind(coefficient=coef(m.scoreincrease.by.absences),
confint(m.scoreincrease.by.absences))
```

```
##              coefficient      2.5 %    97.5 %
## (Intercept)  2.15751104  0.2908417 4.0241803
## absences    -0.07549985 -0.3278114 0.1768117
```

Note the difference between the NHST approach and the estimation approach

- The NHST approach tells us that the extra score difference is zero
- The estimation approach tells us that the confidence intervals are wide compared to the size of the fluctuations in the trend line, and that it includes zero.

5.did the effect of absences on the change in scores have any interaction with the effect of tutoring?

Discrete predictor: interaction effect

Sometimes, we may have predictors that "interact". This means that the effect of independent variable `absences` will be different depending upon the value of independent variable `tutoring`.

We expect there may be an interaction between absences and tutoring as a significant predictors of Score increase ,and this model is only including "main effects"

What we want is a model that estimates this:

$$
\begin{aligned}
Scoreincrease \\
&= \beta_{Intercept} + \beta_{absences} \times absences + \beta_{tutoring1} \times tutoring1 \\
&\quad + \beta_{absences:tutoring1} \times absences \times tutoring1 + \epsilon
\end{aligned}
$$

Where tutoring1 is equal to tutoring

NHST approach

```
#Creating linear regression model for Score increase,Absences and tutoring as
independent variables along with the interaction within those
absences:tutoring1

m.intr.all <- lm(Score_Increase~ absences*tutoring, data = tutoring_data)
summary(m.intr.all)

##
## Call:
## lm(formula = Score_Increase ~ absences * tutoring, data = tutoring_data)
##
## 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
## absences           -0.13696    0.17517  -0.782   0.4352
## tutoring1           4.03560    1.79026   2.254   0.0253 *
## absences:tutoring1  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
```

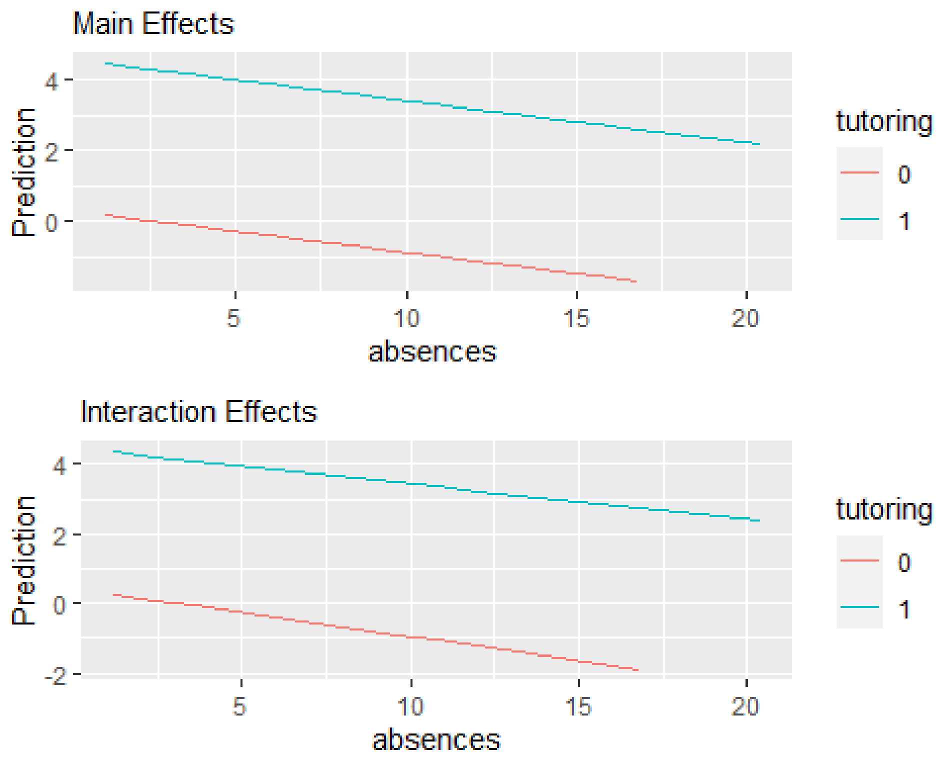
Estimation approach

```
#Calculating the confidence interval for our coefficients in our linear model
which try to explain the variance of Score increase by using
absences,tutoring,and their interaction as predictors.

cbind(coefficient=coef(m.intr.all), confint(m.intr.all))

##                    coefficient      2.5 %    97.5 %
## (Intercept)         0.42453076 -2.0440265 2.8930880
## absences           -0.13695899 -0.4824149 0.2084969
## tutoring1           4.03559915  0.5049560 7.5662423
## absences:tutoring1  0.03470584 -0.4432394 0.5126511
```

Using * instead of + tells the `lm()` function to generate main effects and interactions between the predictors.

Plotting main effects and interaction effects: Main effect and interaction effect are identical. Confirming our previous test which reflected that the effect of absences on the change in scores had no interaction with the effect of tutoring.

```
m.intr.main <- lm(Score_Increase~ absences+tutoring, data = tutoring_data)

p1 <- mutate(tutoring_data,
       main.hat = predict(m.intr.main, tutoring_data),
       intr.hat = predict(m.intr.all, tutoring_data)) %>%
ggplot() +
    geom_line(aes(absences, main.hat, colour = tutoring)) +
  labs(y = "Prediction", subtitle = "Main Effects")

p2 <- mutate(tutoring_data,
       main.hat = predict(m.intr.main, tutoring_data),
       intr.hat = predict(m.intr.all,tutoring_data)) %>%
  ggplot() +
    geom_line(aes(absences, intr.hat, colour = tutoring)) +
  labs(y = "Prediction", subtitle = "Interaction Effects")

grid.arrange(p1, p2)
```



Checking Multicollinearity

When we are using multiple predictors in a regression we can run into issues when we have correlated predictor variables

Let’s first run some models for Score increase a dependent variable which will have different predictors

$$Scoreincrease = \beta_{Intercept} + \epsilon$$

```
m.intr.solo <- lm(Score_Increase~ 1, data = tutoring_data)
summary(m.intr.solo)
```

```
##
## Call:
## lm(formula = Score_Increase ~ 1, data = tutoring_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.188  -4.133  -0.118   3.913  18.079
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.6628     0.4389   3.789 0.000201 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.207 on 199 degrees of freedom
```

$$Scoreincrease = \beta_{Intercept} + \beta_{absences} \times absences + \epsilon$$

```
m.collin1.absences <- lm(Score_Increase ~ absences, data = tutoring_data)
summary(m.collin1.absences)
```

```
##
## Call:
## lm(formula = Score_Increase ~ absences, data = tutoring_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.5013  -4.0944  -0.0555   3.9418  17.9472
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1575     0.9466   2.279   0.0237 *
## absences     -0.0755     0.1280  -0.590   0.5558
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.217 on 198 degrees of freedom
## Multiple R-squared:  0.001756,   Adjusted R-squared:  -0.003286
## F-statistic: 0.3482 on 1 and 198 DF,  p-value: 0.5558
```

$$Scoreincrease = \beta_{Intercept} + \beta_{tutoring1} \times tutoring1 + \epsilon$$

```
m.collin1.tutoring<- lm(Score_Increase ~ tutoring, data = tutoring_data)
summary(m.collin1.tutoring)

##
## Call:
## lm(formula = Score_Increase ~ tutoring, data = tutoring_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.0850  -3.7445  -0.0812   3.2597  15.9767
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.4400     0.5853  -0.752    0.453
## tutoring1     4.2056     0.8277   5.081 8.65e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.853 on 198 degrees of freedom
## Multiple R-squared:  0.1154, Adjusted R-squared:  0.1109
## F-statistic: 25.82 on 1 and 198 DF,  p-value: 8.652e-07
```

$$Scoreincrease = \beta_{Intercept} + \beta_{score.t1} \times score.t1 + \epsilon$$

```
m.collin1.score.t1 <- lm(Score_Increase ~ score.t1, data = tutoring_data)
summary(m.collin1.score.t1)

##
## Call:
## lm(formula = Score_Increase ~ score.t1, data = tutoring_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.2505  -4.0282  -0.0404   3.9775  17.9915
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.29048    1.91152  -0.152    0.879
## score.t1     0.03628    0.03455   1.050    0.295
##
## Residual standard error: 6.205 on 198 degrees of freedom
## Multiple R-squared:  0.005536,   Adjusted R-squared:  0.0005137
## F-statistic: 1.102 on 1 and 198 DF,  p-value: 0.295
```

$$Scoreincrease = \beta_{Intercept} + \beta_{score.t2} \times score.t2 + \epsilon$$

```
m.collin1.score.t2 <- lm(Score_Increase ~ score.t2, data = tutoring_data)
summary(m.collin1.score.t2)
```

```
## 
## Call:
## lm(formula = Score_Increase ~ score.t2, data = tutoring_data)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -12.233  -3.469  -0.621   3.816  14.727 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -9.94402    1.51315  -6.572 4.30e-10 ***
## score.t2     0.20910    0.02637   7.929 1.58e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 5.421 on 198 degrees of freedom
## Multiple R-squared:  0.241,  Adjusted R-squared:  0.2372 
## F-statistic: 62.87 on 1 and 198 DF,  p-value: 1.58e-13
```

$$Scoreincrease = \beta_{Intercept} + \beta_{absences} \times absences + \beta_{tutoring1} \times tutoring1 + \beta_{score.t1} \times score.t1 + \beta_{score.t2} \times score.t2 + \epsilon$$

```
m.collin1.all <- lm(Score_Increase~ absences + tutoring+score.t1+ score.t2, 
data = tutoring_data)
```

if we were to compare our following models:

- $Scoreincrease = \beta_{Intercept} + \epsilon$
- $Scoreincrease = \beta_{Intercept} + \beta_{absences} \times absences + \epsilon$

it can be done by using the anova() function as follows:

anova comparison technique tells us that absences does not add significant predictive accuracy to our model

```
anova(m.intr.solo,m.collin1.absences)

## Analysis of Variance Table
## 
## Model 1: Score_Increase ~ 1
## Model 2: Score_Increase ~ absences
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    199 7666.6                           
## 2    198 7653.1  1    13.459 0.3482 0.5558
```

We can use `rcorr` to see that they are significantly correlated with each other: The pearson correlation coefficient is calculated using for normally distributed variables:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})\,(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

```
# pearson correlation for continuos normally distributed data. As mentioned
before, there only significant predictor for Score increase are tutoring and
score at the end of the period according to the p values.

rcorr(as.matrix(tutoring_data))

##                tutoring absences score.t1 score.t2 Score_Increase
## tutoring           1.00     0.07     0.07     0.21           0.34
## absences           0.07     1.00    -0.34    -0.32          -0.04
## score.t1           0.07    -0.34     1.00     0.91           0.07
## score.t2           0.21    -0.32     0.91     1.00           0.49
## Score_Increase     0.34    -0.04     0.07     0.49           1.00
##
## n= 200
##
##
## P
##                tutoring absences score.t1 score.t2 Score_Increase
## tutoring                0.3257   0.2965   0.0029   0.0000
## absences       0.3257            0.0000   0.0000   0.5558
## score.t1       0.2965   0.0000            0.0000   0.2950
## score.t2       0.0029   0.0000   0.0000            0.0000
## Score_Increase 0.0000   0.5558   0.2950   0.0000

#The only two significant predictor for Score increase is tutoring with 0.34
of relationship, and score at the end of the academic year as a structural
correlation.
```

If we want to see if this is a problem in our data then we can use the Variance Inflation Factor (VIF): In this case we have a structural Multicollinearity for score.t2 and score.t1 and Score_ increase since our dependent variable (Score_Increase) was calculated by using these variables.

Generally, VIF scores of less than 5 don’t warrant any further action. When they are greater than 5 we have to consider whether it is justified keeping all of the predictors in the model

```
vif(m.collin1.all)

## absences tutoring score.t1 score.t2
## 1.148411 1.148459 6.070274 6.267240
```

This is a measure that takes into account each variable’s shared variance with all other variables, whereas a correlation table can only identify individual pairings of variables

An extreme example could be a case of using two different aptitude tests. If the scores are highly correlated then they are likely measuring the same thing and you only need to include one of them in your regression, or perhaps an average of the two scores

As we have seen before, adding absences does not add significant accuracy to our model.Now let’s check for tutoring, scoret.t1 and score.t2 by using `anova()` for a model comparison to see if a more complex model is more accurate overall, even when the beta coefficients and p-values may be difficult to interpret.

- Adding tutoring to Score_Increase~ beta(intercept) improves our model

This shows us that including the variable `tutoring` does add significant predictive accuracy to our model, so it is explaining a significant amount of variance in Score_Increase independent of that also explained by the variable `tutoring`.

- Model comparison shows that a regression model including tutoring results in a significantly better overall fit than a model only including the intercept $F(-1,199) = 25.8171, p = 0.0000$.”

```
anova(m.collin1.tutoring, m.intr.solo)

## Analysis of Variance Table
##
## Model 1: Score_Increase ~ tutoring
## Model 2: Score_Increase ~ 1
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    198 6782.3
## 2    199 7666.6 -1   -884.35 25.817 8.652e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Adding score.t1 to Score_Increase ~ beta(intercept) does not improve our model

Model comparison shows that a regression model including score at the beginning of the academic year has no significantly better overall fit than a model only including the intercept $F(1,199) = 1.1023, p > 0.05$.”

```
anova(m.intr.solo,m.collin1.score.t1)

## Analysis of Variance Table
##
## Model 1: Score_Increase ~ 1
## Model 2: Score_Increase ~ score.t1
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    199 7666.6
## 2    198 7624.2  1    42.444 1.1023  0.295
```

- Adding score.t2 to Score_Increase ~ beta(intercept) improves our model since there is a structural multicolinearity

$F(1,199) = 62.87, p < 0.000$

```
anova(m.intr.solo,m.collin1.score.t2)

## Analysis of Variance Table
##
```

```
## Model 1: Score_Increase ~ 1
## Model 2: Score_Increase ~ score.t2
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)    
## 1    199 7666.6                                  
## 2    198 5818.8  1    1847.8 62.874 1.58e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Section 2

Report

1 Check whether the students allocated to the tutored and non-tutored groups had similar or different average test scores before the tutoring scheme began.

To address this requirement a t test (NHST) along with Estimation approach techniques were developed to verify whether the students allocated to the tutored and non-tutored groups had similar or different average test scores before the tutoring scheme began.

Null hypothesis significance testing (NHST)

$$t = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{s_{\bar{x}_1 - \bar{x}_2}} = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

Symbol	Description
$\bar{x}_i$	The mean of the sample i
$s_{\bar{x}_1 - \bar{x}_2}$	The standard error of the difference in sample means
s_i	The standard deviation of sample i
n_i	The number of observations in the sample i

- Null Hypothesis : The difference between mean of score at the beginning of the academic year in tutoring (tutoring1) and non-tutoring (tutoring0) is zero.
- Alternate Hypothesis : The difference between mean of score at the beginning of the academic year in tutoring (tutoring1) and non-tutoring (tutoring0) is NOT zero.

```
## 
##  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 0 and group 
1 is not equal to 0
## 95 percent confidence interval:
##  -5.433861  1.665720
## sample estimates:
## mean in group 0 mean in group 1 
##        52.90345        54.78753
```

The students allocated to the tutored and non-tutored groups had similar average test scores before the tutoring scheme began.This mean difference for the score at the beginning of the academic year is no statistically significant by tutoring, $t(196.54) = -1.0467, p > 0.05$

- The t=test tells us that we should accept the null hypothesis and the data is not statistically significant

By creating the model $Score.t1 = \beta_{Intercept} + \beta_{tutoring} \times tutoring + \epsilon$

The summary() output contains a NHST of whether the coefficient is zero or not. Given the results we find that the effect of tutoring equal to 1 (TRUE) on the scores at the beginning of the academic year is zero. $t(198) = 1.047. p > .05$

```
## 
## Call:
## lm(formula = score.t1 ~ tutoring, data = tutoring_data)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -35.595  -7.414  -0.194   9.067  35.554 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   52.903      1.273  41.565   <2e-16 ***
## tutoring1      1.884      1.800   1.047    0.297    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 12.73 on 198 degrees of freedom
## Multiple R-squared:  0.005503,   Adjusted R-squared:  0.0004801 
## F-statistic: 1.096 on 1 and 198 DF,  p-value: 0.2965
```

Estimation approach

For the estimation approach we are calculating our 98% confidence interval:

- Since the confidence interval range is between 95% CI[-1.66-5.43].It tells us how confident we can be that our estimate coefficient for tutoring1 will fall into that range depending on how wide our interval is.Additionally, it is important to mention that besides being a wide interval, it also includes zero.Therefore,students allocated to the tutored and non-tutored groups had similar average test scores before the tutoring scheme began.

Estimation approach

Here we make a model object `m.scorest1.by.tutoring` and then use `emmeans` to get confidence intervals for the scores at the beginning of the year for tutored and non-tutored pupils and `pairs()` and `confint()` to *contrast* the tutoring and non-tutoring.

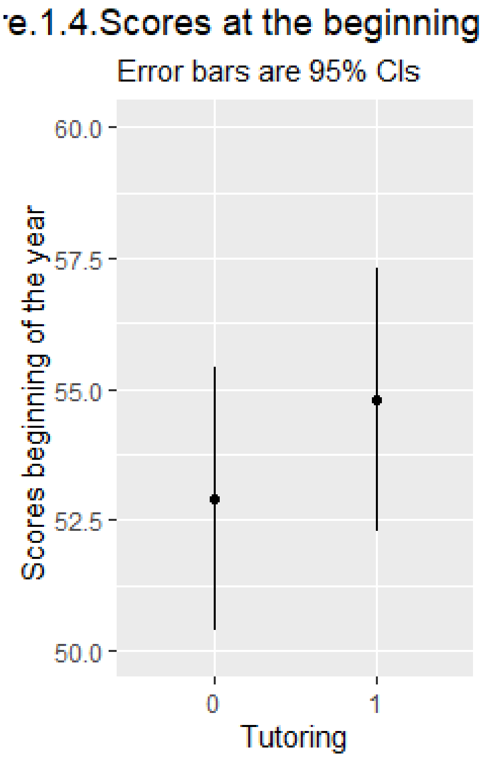
```
##  tutoring emmean   SE  df lower.CL upper.CL
##  0          52.9 1.27 198     50.4     55.4
##  1          54.8 1.27 198     52.3     57.3
##
## Confidence level used: 0.95
```

```
##  contrast              estimate  SE  df lower.CL upper.CL
##  tutoring0 - tutoring1    -1.88 1.8 198    -5.43     1.67
##
## Confidence level used: 0.95
```

```
##             coefficients     2.5 %    97.5 %
## (Intercept)    52.903455 50.393488 55.413421
## tutoring1       1.884071 -1.665557  5.433699
```

Reporting 95% confidence interval difference:

- The mean in scores at the beginning of the year for non-tutoring is 52.9, 95.The mean in scores at the beginning of the year for tutoring scheme is 54.8, 95. This difference in means is not significantly different from zero.


e.1.4.Scores at the beginning

Error bars are 95% CIs

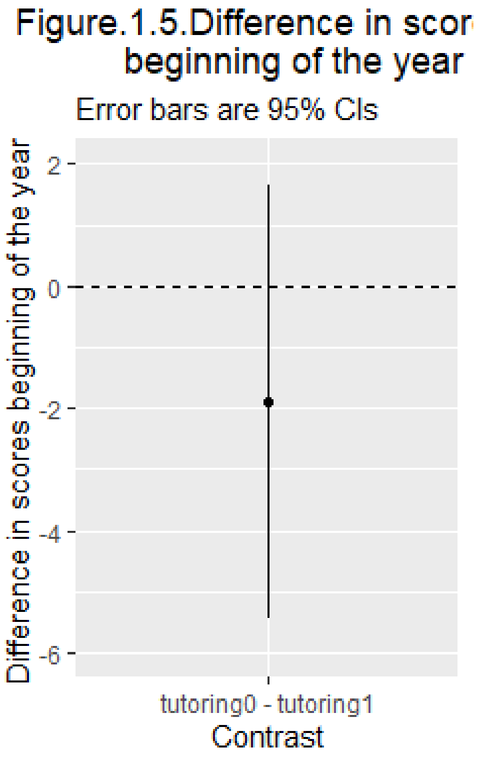


Figure.1.5.Difference in scor beginning of the year

Error bars are 95% CIs

__2__ Did the tutored and non-tutored students have similar or different rates of absences on average?

To address this requirement a t test (NHST) along with Estimation approach techniques were developed to verify whether the tutored and non-tutored students had similar or different rates of absences on average.

Null hypothesis significance testing (NHST)

- Null Hypothesis : The difference between mean of absences for tutored and non-tutored students is zero.
- Alternate Hypothesis : The difference between mean of absences for tutored and non-tutored students is NOT zero

```
##
##  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 0 and group
1 is not equal to 0
## 95 percent confidence interval:
##  -1.440721  0.480721
## sample estimates:
## mean in group 0 mean in group 1
##           6.312           6.792
```

The students allocated to the tutored and non-tutored groups had similar average absenteeism rate .This mean difference for absences rate is no statistically significant by tutored and non-tutored students, $t(197.6) = -0.98528, p > 0.05$

- The t=test tells us that we should accept the null hypothesis and the data is not statistically significant

By creating the model $absences = \beta_{Intercept} + \beta_{tutoring} \times tutoring + \epsilon$

The summary() output contains a NHST of whether the coefficient is zero or not. Given the results we find that the effect of tutoring scheme on the rate of absences is zero. $t(198) = 0.985. p > .05$

```
##
## Call:
## lm(formula = absences ~ tutoring, data = absence_vs_tutoring)
##
## 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 ***
## tutoring1     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
```

Estimation approach

For the estimation approach we are calculating our 98% confidence interval:

- Since the confidence interval range is between 95% CI[-0.480-1.440].It tells us how confident we can be that our estimate coefficient for tutoring1 will fall into that range depending on how wide our interval is.Additionally, it is important to mention that besides being a wide interval, it also includes zero.Therefore, tutored and non-tutored students have similar rates of absences on average.

Here we make a model object `m.absences.by.tutoring` and then use `emmeans` to get confidence intervals for the absences for tutored and non-tutored pupils and `pairs()` and `confint()` to *contrast* the tutoring and non-tutoring.

```
##  tutoring emmean    SE  df lower.CL upper.CL
##  0          6.31 0.344 198     5.63     6.99
##  1          6.79 0.344 198     6.11     7.47
##
## Confidence level used: 0.95
```

```
##  contrast              estimate    SE  df lower.CL upper.CL
##  tutoring0 - tutoring1    -0.48 0.487 198    -1.44    0.481
##
## Confidence level used: 0.95
```

```
##             coefficients      2.5 %   97.5 %
## (Intercept)        6.312  5.6326761 6.991324
## tutoring1          0.480 -0.4807091 1.440709
```

Reporting 95% confidence interval difference:

- The mean in absenteeism rate for non-tutoring students is 6.31, 95.The mean in absenteeism rate for tutored students is 6.79, 95. This difference in means is not significantly different from zero.

Figure.1.6.Absences to regular classes as a propo

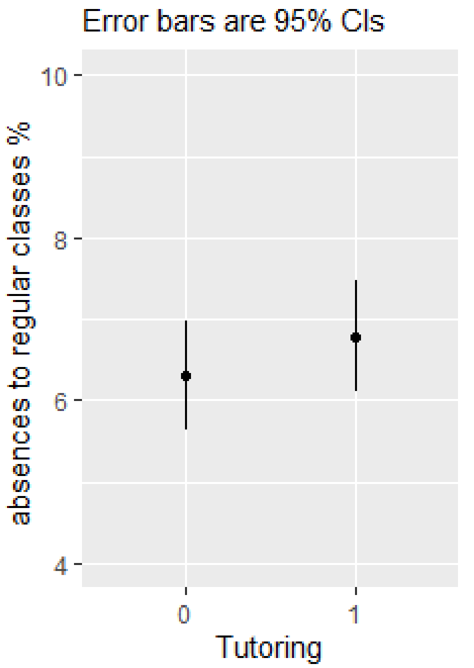
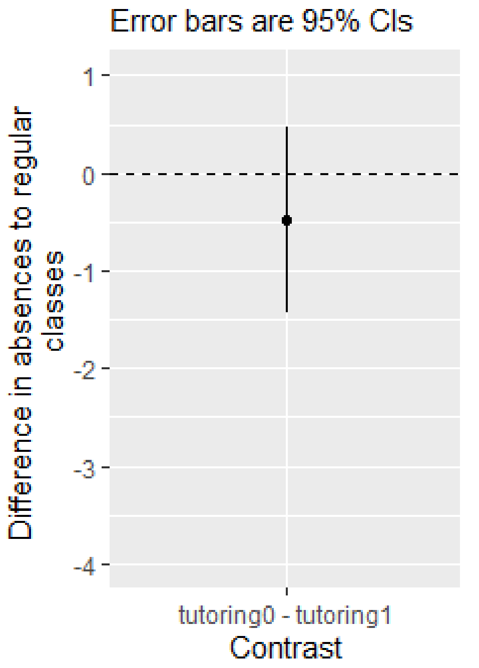


Figure.1.7.Difference in absen regular classes as a propo



***3** Did the tutored students show an increase in their scores compared to the students who did not receive tutoring?*

To answer this question, a new variable called Score_Increase was calculated as follows: This allow us to continue our analysis to check whether there were changes on the scores from the end of the academic year and the beginning:

$$Score_{I}ncrease = (score.t2 - score.t1)$$

Afterwards, to address this requirement a t test (NHST) along with Estimation approach techniques were developed to verify whether the tutored students showed an increase in their scores compared to the students who did not receive tutoring.

Null hypothesis significance testing (NHST)

- Null Hypothesis : The difference between mean of Score increase for tutored and non-tutored students is zero.
- Alternate Hypothesis : The difference between mean of Score increase for tutored and non-tutored students is NOT zero

```
##
##  Welch Two Sample t-test
##
## data:  Score_Increase by tutoring
## t = -5.0811, df = 194.29, p-value = 8.78e-07
## alternative hypothesis: true difference in means between group 0 and group
1 is not equal to 0
```

```
## 95 percent confidence interval:
##  -5.837998 -2.573164
## sample estimates:
## mean in group 0 mean in group 1
##      -0.4399544       3.7656265
```

The students who attended the ‘tutoring classes’ have a mean Score increase between the beginning and end of the academic year, increase of 3.76 . Individuals who did not take tutoring classes have a mean decrease of 0.4399. This mean difference for the score increase is significantly large by non-tutoring and tutored students, $t(194.29) = -5.0811$, $p < 0.000$, with a difference of 3.32.

- The t=test tells us that we should reject the null hypothesis and the data is statistically significant

By creating the model $Scoreincrease = \beta_{Intercept} + \beta_{tutoring1} \times tutoring1 + \epsilon$

The summary() output contains a NHST of whether the coefficient is zero or not. Given the results There are 4.2056 increase in Score increase for every extra attendance to tutoring classes.This increase is significantly different from zero. $t(198) = 5.081. p < .0001$

```
##
## Call:
## lm(formula = Score_Increase ~ tutoring, data = scoret1t2_vs_tutoring)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.0850  -3.7445  -0.0812   3.2597  15.9767
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.4400     0.5853  -0.752    0.453
## tutoring1     4.2056     0.8277   5.081 8.65e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.853 on 198 degrees of freedom
## Multiple R-squared:  0.1154, Adjusted R-squared:  0.1109
## F-statistic: 25.82 on 1 and 198 DF,  p-value: 8.652e-07
```

Estimation approach

For the estimation approach we are calculating our 98% confidence interval:

- Since the confidence interval range is between 95% CI[2.57-5.837].It tells us how confident we can be that our estimate coefficient for tutoring1 will fall into that range depending on how wide our interval is.Additionally, it is important to mention that besides being a fairly narrow interval, it also does not include zero.Therefore, the tutored students show an increase in their scores compared to the students who did not receive tutoring.

Here we make a model object `m.difference.by.tutoring` and then use `emmeans` to get confidence intervals for the difference in scores (t1 and t2) for tutored and non-tutored pupils and `pairs()` and `confint()` to *contrast* the tutoring and non-tutoring.

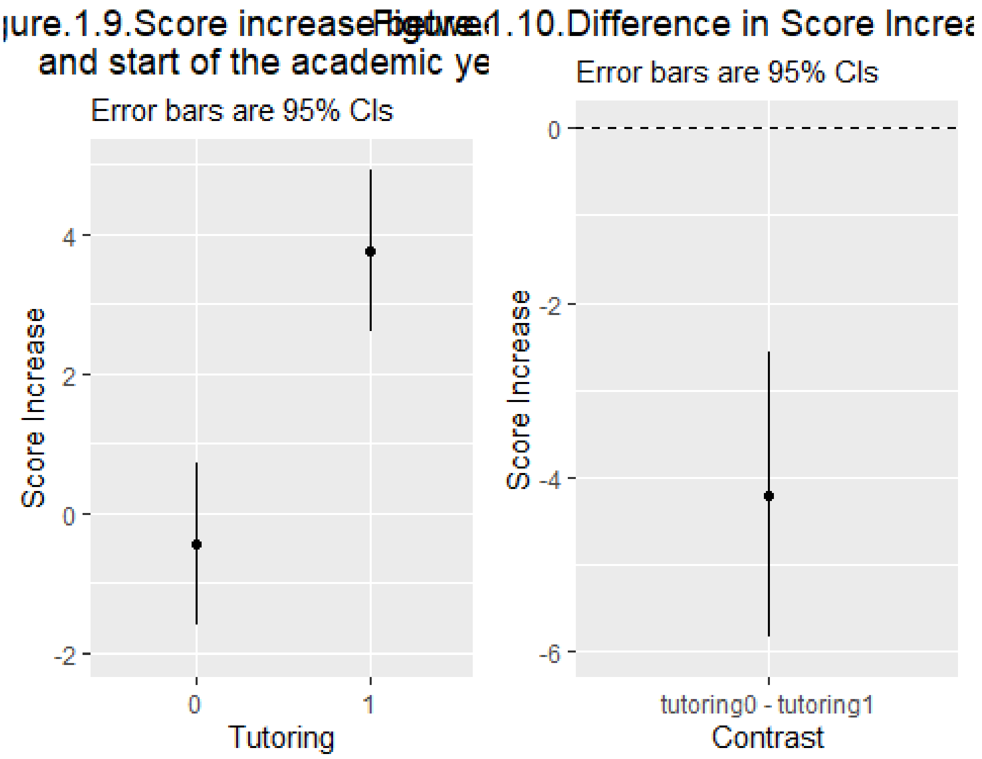
```
##  tutoring emmean    SE  df lower.CL upper.CL
##  0         -0.44 0.585 198    -1.59    0.714
##  1          3.77 0.585 198     2.61    4.920
## 
## Confidence level used: 0.95
```

```
##  contrast              estimate    SE  df lower.CL upper.CL
##  tutoring0 - tutoring1    -4.21 0.828 198    -5.84    -2.57
## 
## Confidence level used: 0.95
```

```
##             coefficients     2.5 %    97.5 %
## (Intercept)   -0.4399544 -1.594112 0.7142034
## tutoring1      4.2055809  2.573355 5.8378065
```

Reporting 95% confidence interval difference:

- The mean in Score increase for non-tutoring students is −0.44, 95.The mean in Score increase for tutored students is 3.77, 95. Students who attended to the tutorials increase their scores from period 1 to 2 significantly (3.77 avg) compared to those non-tutored whose scores reduced −0.44 on average. The score difference is 3.32 95.



4 *was there any effect of absences on the change in scores?*

To address this requirement NHST along with Estimation approach for coefficients techniques were developed to verify there was any effect of absences on the change in scores.

NHST approach

The regression coefficient tells us about the best-fitting straight line through the data. The slope tells us about the effect a unit change in our independent variable (on the x axis which is absences) has on our dependent variable (on the y axis which is the Score increase)

Creating a linear regression model for Score increase and absences as independent variable

`Score_Increase~absences` means predict the outcome variable on the left using the predictor(s) on the right. This produces a model object that we are saving here as `m.scoreincrease.by.absences`

$$Score_Increase = \beta_{Intercept} + \beta_{absences} \times absences + \epsilon$$

- In this case, the p value 0.5558 is telling us that is very likely that there is no a significant relationship between absenteeism and score difference. Meaning that , the effect of absences on the change in scores is likely to be zero (there is no statistically significant effect).We find that the effect of absences on the change in scores is zero. $t(198) = -0.590. p > .05$

```
##
## Call:
## lm(formula = Score_Increase ~ absences, data = tutoring_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.5013  -4.0944  -0.0555   3.9418  17.9472
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1575     0.9466   2.279   0.0237 *
## absences     -0.0755     0.1280  -0.590   0.5558
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.217 on 198 degrees of freedom
## Multiple R-squared:  0.001756,   Adjusted R-squared:  -0.003286
## F-statistic: 0.3482 on 1 and 198 DF,  p-value: 0.5558
```

Estimation approach to regression coefficients

For the estimation approach we are calculating our 98% confidence interval:

- Since the confidence interval range is between 95% $CI[-0.327 - 0.1768]$.It tells us how confident we can be that our estimate coefficient for absences will fall into that range depending on how wide our interval is.Additionally, this interval includes zero.Therefore, there was not effect of absences on the change in scores.

```
##              coefficients      2.5 %    97.5 %
## (Intercept)    2.15751104  0.2908417 4.0241803
## absences      -0.07549985 -0.3278114 0.1768117
```

Correlation

- The only two significant predictor for Score_Increase are tutoring with 0.34 of relationship,when there is a positive relationship between tutoring (when tutoring is TRUE the Score increase is higher).

- The relationship Score_Increase and score at the end of the academic year (0.49) is positive;however it is regarded to be a structural correlation since it was part of the calculation for Score_Increase.

```
##                 tutoring absences score.t1 score.t2 Score_Increase
## tutoring            1.00     0.07     0.07     0.21           0.34
## absences            0.07     1.00    -0.34    -0.32          -0.04
## score.t1            0.07    -0.34     1.00     0.91           0.07
## score.t2            0.21    -0.32     0.91     1.00           0.49
## Score_Increase      0.34    -0.04     0.07     0.49           1.00
##
## n= 200
##
##
## P
##                 tutoring absences score.t1 score.t2 Score_Increase
## tutoring                 0.3257   0.2965   0.0029   0.0000
## absences        0.3257            0.0000   0.0000   0.5558
## score.t1        0.2965   0.0000            0.0000   0.2950
## score.t2        0.0029   0.0000   0.0000            0.0000
## Score_Increase  0.0000   0.5558   0.2950   0.0000
```

5.*did the effect of absences on the change in scores have any interaction with the effect of tutoring?*

To address this requirement NHST along with Estimation approach for coefficients techniques for interaction terms (absences:tutoring1 (tutored)) were developed to verify whether the effect of absences on the change in scores had any interaction with the effect of tutoring.

Discrete predictor: interaction effect

The possible interaction between absences and tutoring was addressed by assuming that we may have predictors that "interact". This means that the effect of independent variable `absences` will be different depending upon the value of independent variable `tutoring`.

We expect there may be an interaction between absences and tutoring as a significant predictors of Score Increase ,and this model is only including “main effects”

What we want is a model that estimates this:

$$
\begin{aligned}
Scoreincrease \\
&= \beta_{Intercept} + \beta_{absences} \times absences + \beta_{tutoring1} \times tutoring1 \\
&\quad + \beta_{absences:tutoring1} \times absences \times tutoring1 + \epsilon
\end{aligned}
$$

Where tutoring1 is equal to tutoring

The beta coefficients for a discrete predictor show the difference in the outcome variable for that level/category compared to the reference or baseline category

Here, the reference category is “tutoring0” (non-tutored), so the coefficient shows us how much higher Score Increase is when tutoring is “1” (tutoring scheme)

Given the results we have the following analysis:

For the NHST approach we might say:

- The beta coefficient for absences:tutoring1 (tutored) shows us that there is no a significant interaction between absences and tutoring1 when predicting Score_Increase. $t(196) = 0.143, p > 0.05$. Meaning that, the effect of absences on the change in scores had no interaction with the effect of tutoring1.

- “The results of the regression show that there is NO a significant main effect of `absences` upon `Scoreincrease` ($b = -0.1370 $, $t(197) = -0.768, p = 0.4435$).

- However, there was a statistically significant effect of `tutoring1`(tutoring) upon `Scoreincrease` ($b = 4.035, t(197) = 4.03560, p = 0.0253$).For tutoring (tutoring1), Score Increase is higher by 4.035,

```
##
## Call:
## lm(formula = Score_Increase ~ absences * tutoring, data = tutoring_data)
##
## 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
## absences           -0.13696    0.17517  -0.782   0.4352
## tutoring1           4.03560    1.79026   2.254   0.0253 *
## absences:tutoring1  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
```

Estimation approach

- For the absences coefficient: The confidence intervals include zero (95 and this decrease is not significantly different from zero, $t(197) = -0.768, p = 0.443$"
- For tutoring (tutoring1). Score Increase is higher by 4.035, $95CI[0.5049560 - 7.56624]$ when tutoring is 1, TRUE. Meaning that we can define the variance of Score Increase by using tutoring as a predictor.Therefore, the effect of tutoring1 on Score Increase is statistically significant.
- Lastly, for the absences:tutoring1 coefficient: The confidence intervals include zero (95 and this decrease is not significantly different from zero, $t(196) = 0.143, p > 0.05., p = 0.443$". Then, absences did not have any interaction with the effect of tutoring on the change of scores.

```
##                      coefficient      2.5 %    97.5 %
## (Intercept)           0.42453076 -2.0440265 2.8930880
## absences             -0.13695899 -0.4824149 0.2084969
## tutoring1             4.03559915  0.5049560 7.5662423
## absences:tutoring1    0.03470584 -0.4432394 0.5126511
```

Question 2a

Section 1

```
#Reading the database

beers <- read_csv("Craft-Beer_data_set.csv")

summary(beers)

##      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
```

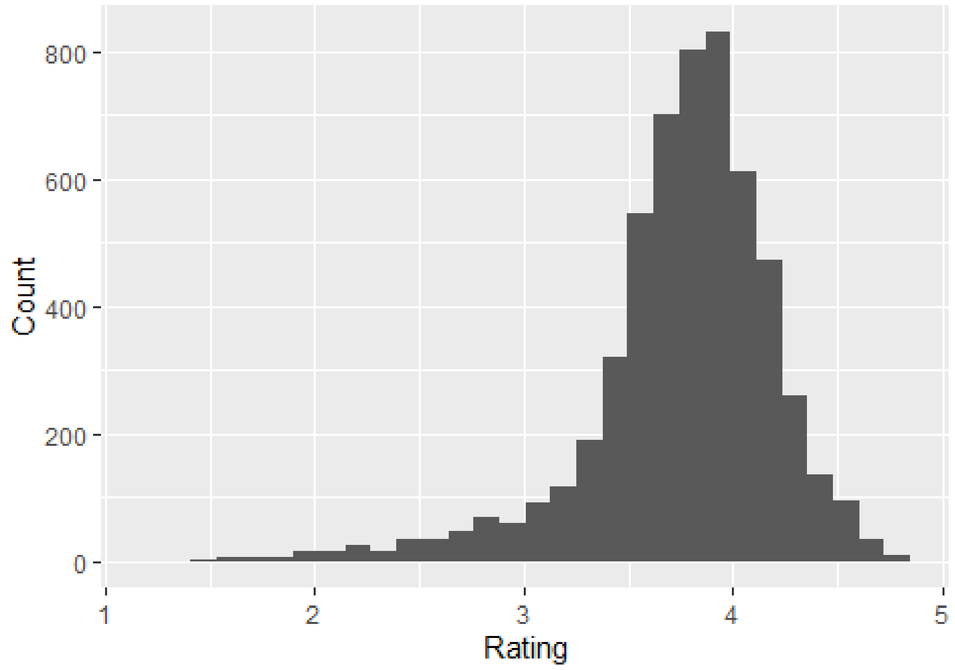
```
str(beers)
```

```
## spc_tbl_ [5,558 × 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." "Uerige ObergÃ¤rige Hausbrauerei" ...
##  $ 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
# Verifying our variable called rating  (a little Negative Skewed)

beers %>% ggplot() + geom_histogram(aes(rating))+labs(x="Rating",
y="Count",title="Figure 2.1. A histogram describing the
distribution of beers rating")+theme(plot.title = element_text(hjust = 0.5))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

Figure 2.1. A histogram describing the distribution of beers rating



```
# Checking categorical variables (style)
# we have total of 112 styles of beer
unique(beers$Style)

# each style of beer register between 16 to 50 records of our dataset

table(beers$Style)
```

Modifying the style variable to determine the category names depending on the given classification

```
#Creating a new vector with our category names

x <- c("IPA ", "Lager ","Porter ", "Stout ","Wheat","Pale","Pilsner ","Bock
")

# Splitting the column "style" into category and mark by using - slash symbol

beers <- separate(beers,Style,into=c("category","mark"),sep="-")

## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 1249 rows
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,
## 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].

# Deleting the column mark

beers$mark <- NULL
```

```
# Listing the names categories to check whether we have to make changes on
them

unique(beers$category)
```

```
##  [1] "Altbier"                             "Barleywine "
##  [3] "Bitter "                             "BiÃ¨re de Champagne / BiÃ¨re
Brut"
##  [5] "Blonde Ale "                         "Bock "
##  [7] "Braggot"                             "Brett Beer"
##  [9] "Brown Ale "                          "California Common / Steam Beer"
## [11] "Chile Beer"                          "Cream Ale"
## [13] "Dubbel"                              "Farmhouse Ale "
## [15] "Fruit and Field Beer"                "Gruit / Ancient Herbed Ale"
## [17] "Happoshu"                            "Herb and Spice Beer"
## [19] "IPA "                                "Kvass"
## [21] "KÃ¶lsch"                             "Lager "
## [23] "Lambic "                             "Low Alcohol Beer"
## [25] "Mild Ale "                           "Old Ale"
## [27] "Pale Ale "                           "Pilsner "
## [29] "Porter "                             "Pumpkin Beer"
## [31] "Quadrupel (Quad)"                    "Red Ale "
## [33] "Rye Beer "                           "Rye Beer"
## [35] "Scotch Ale / Wee Heavy"              "Scottish Ale"
## [37] "Smoked Beer"                         "Sour "
## [39] "Stout "                              "Strong Ale "
## [41] "Tripel"                              "Wheat Beer "
## [43] "Wild Ale"                            "Winter Warmer"
```

```
# Renaming "Wheat Beer " into "Wheat", and "Pale Ale " into "Pale"

beers$category[beers$category=="Wheat Beer "] <- "Wheat"

beers$category[beers$category=="Pale Ale "] <- "Pale"

#  relabeling the category which are not included in our vector for labels
into "Other"

beers$category[!(beers$category %in% x) ] <-"Other"

# Checking our new notation for the column category

unique(beers$category)
```

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

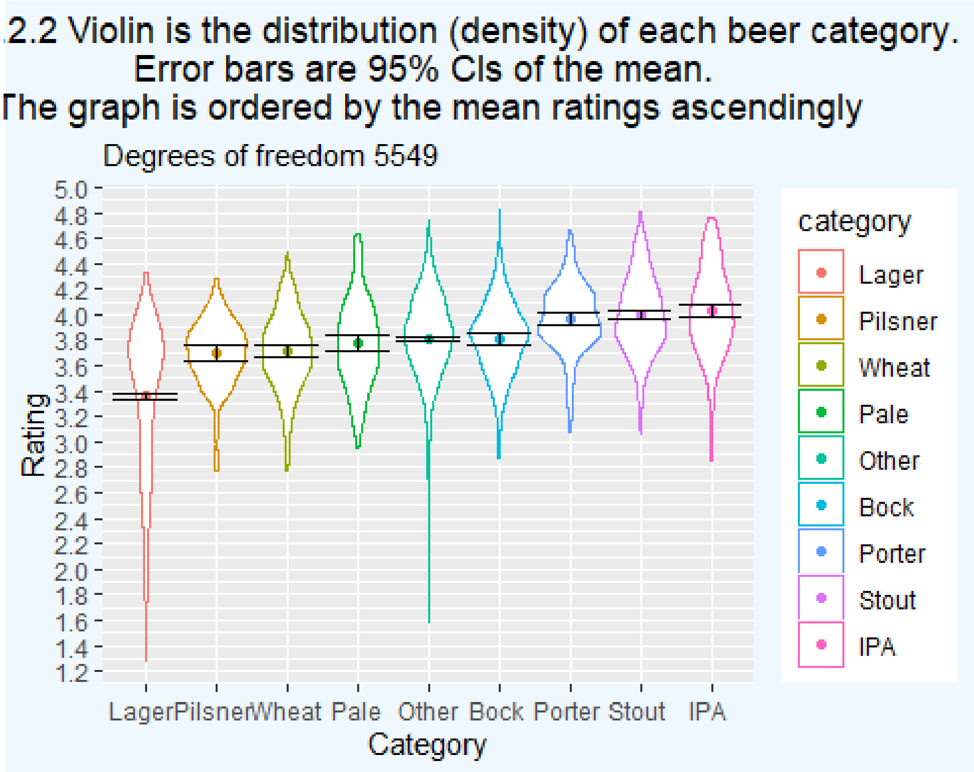
Calculating the mean rating and 95% confidence intervals of the rating within each category using a linear model: R output that shows the mean and 95% confidence intervals numerically.

```
#Creating a linear model to include rating as dependent variable of each
category beer

m.rating.by.category <- lm(rating~category,data=beers)

#Calculating the mean and 95% confidence interval for each beer category

(m.rating.by.category.emm <- emmeans(m.rating.by.category, ~category))

##  category emmean       SE   df lower.CL upper.CL
##  Bock      3.812 0.025116 5549    3.762    3.861
##  IPA       4.029 0.021227 5549    3.988    4.071
##  Lager     3.357 0.013237 5549    3.331    3.383
##  Other     3.806 0.007631 5549    3.791    3.821
##  Pale      3.779 0.032425 5549    3.716    3.843
##  Pilsner   3.690 0.032425 5549    3.627    3.754
##  Porter    3.967 0.022928 5549    3.922    4.012
##  Stout     3.999 0.019856 5549    3.960    4.038
##  Wheat     3.711 0.021227 5549    3.670    3.753
##
## Confidence level used: 0.95

NumericValues <- summary(m.rating.by.category.emm)
```

Violin plot displaying the requested measures

Creating a violin plot displaying the distribution of the ratings within each category, the mean ratings and 95% confidence intervals calculated above.

```
beers %>% mutate(category= fct_reorder(category,rating,.fun='mean'))%>%
ggplot(
aes(x=rating,y=category,color=category))+geom_violin()+geom_point(data=Numeri
cValues, aes(y=category,x=emmean)) +
geom_errorbar(data=NumericValues,aes(y=category,xmin=lower.CL,
xmax=upper.CL),inherit.aes = FALSE)+coord_flip()+
scale_x_continuous(breaks=seq(1,5,by=0.2))+labs(y="Category",
x="Rating",subtitle = "Degrees of freedom 5549") + theme(plot.title =
element_text(hjust = 0.5),plot.background = element_rect(fill =
"aliceblue"))+ggtitle("Figure.2.2 Violin is the distribution (density) of
each beer category.
Error bars are 95% CIs of the mean.
The graph is ordered by the mean ratings ascendingly")
```



Section 2

Report

For starting, description of our variables to be used to address the analysis for this section:

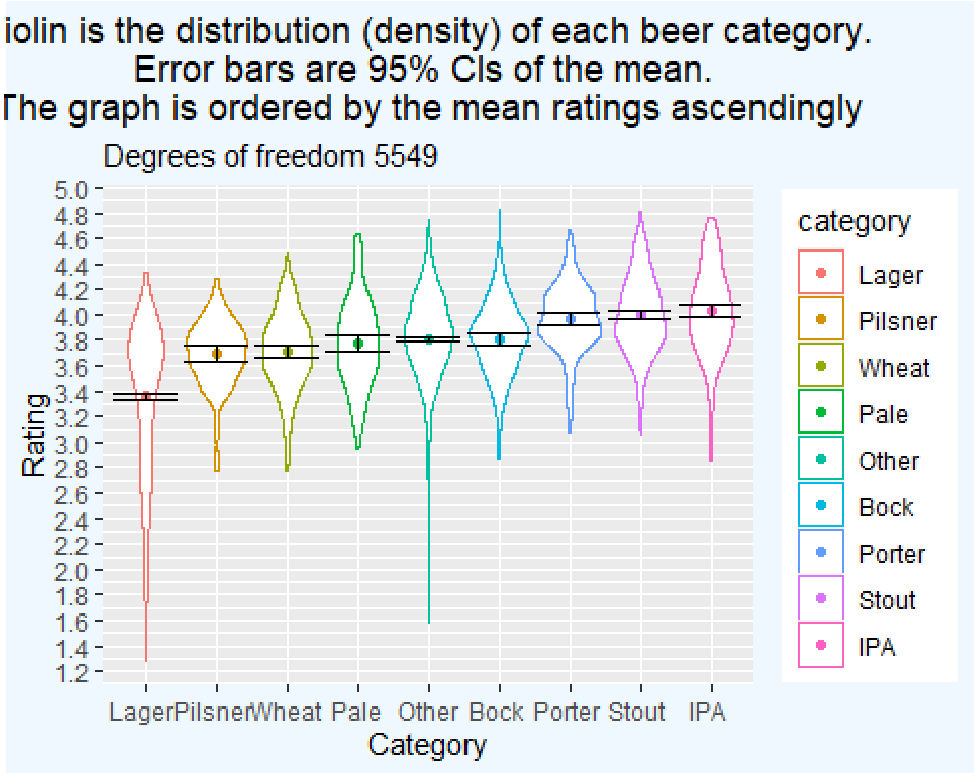
Variable	Description
category	General categories
rating	rating for each beer category

The potential tendency of types of beers to receive higher ratings than others was addressed, it was examined using a histogram to show the distribution of our given ratings variables, and spot any possible outliers or anomalies in our data. Next, we categorized the beers in the dataset by the following general categories:

Category
IPA
Lager
Porter
Stout
Wheat
Pale
Pilsner

Category
Bock
Others

Subsequently,by using a linear model, the mean rating and 95% confidence intervals of the rating within each category were calculated to plot the graph displayed below:



The graph visualized above illustrates that some beers tend to score higher ratings than others. Since the violin graph is ordered by the mean ratings ascendingly, we can appreciate that the larger beer category receives the lowest ratings overall compared to the rest types of beers. This sort of graph is also a useful aid to spot outliers more robustly. Further, it shows us that the lager category is occasionally rated extremely low accounted down to 1.27 approx. Regarding the error bars 95% CI of the mean for “larger” beer, this interval is narrow. Hence, we can be confident that the mean given is correct and falls into that range since a narrow confidence interval enables more precise population estimates.

Moving on through the positions, we have Pilsner, wheat, pale, and Bock categories whose rating average is fairly similar,3.6,3.7,3.72,3.78, and 3.8 respectively. From the distribution, the vast majority of beers that belong to these group classifications received a rating between 3.5 and 3.7 roughly. However, the error bars, 95% CI of the mean differ depending on each category. For Pale and Bock, we can be more confident that its average is a less precise estimate than for Pilsner and Wheat since the confidence intervals for the latest are more narrowed.

Finally, IPA, Sout, and Porter are the top 3 best-rating beverages on the list, IPA being the highest-rated category among all others with an outstanding average of around 4.1. Similarly, regarding the confidence intervals, since their respective CI for these three categories is narrowed, it indicates a more precise estimate of their average.

As a business conclusion, we can affirm that types of beers such as IPA, Sout, and Porter are prone to receive higher ratings compared to Lager, Pilsner, Wheat and others.

Question 2b

Section 1

Firstly, for this section, the variables contained on the beers dataset that will be included in our analysis are rating,ABV,Sweet,and Malty. Therefore, let's plot them and check their distributions and structure.

```
#Reading the database

beers <- read_csv("Craft-Beer_data_set.csv")

summary(beers)

##      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
```

```
str(beers)
```

```
## spc_tbl_ [5,558 × 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." "Uerige ObergÃ¤rige Hausbrauerei" ...
##  $ 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>
```

Data Dictionary

Variable	Description
Name	Name of the beer
category	category (IPA,Lager,Porter,Stout,Wheat,Pale,Pilsner,Bock,And Others)
Brewery	Place where beer is made commercially
ABV	Alcohol by volume
minIBU	minimum measuare bitterness
maxIBU	maximum measuare bitterness
Astringency	Flavor and aroma that comes from a few sources
Body	Sensation of palate fullness, the viscosity and feel of beer in the mouth
Alcohol	alcohol
Bitter	Bitterness flavour
Sweet	Sweetness flavour
Sour	Sourness flavour
Salty	Saltiness flavour
Fruits	Flavoured with fruit
Hoppy	Taste of hops (fruity,earthy,citric,floral,piney etc)
Spices	Spices flavour
Malty	Combination of flavours (sweet,nutty,caramel,or coffee)

```
# Verifying our variable called rating (negative skewed)
```

```
beers %>% ggplot() + geom_histogram(aes(rating))+labs(x="Rating",
y="Count",title="Figure 3.1. A histogram describing the
distribution of beers rating")+theme(plot.title = element_text(hjust = 0.5))

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

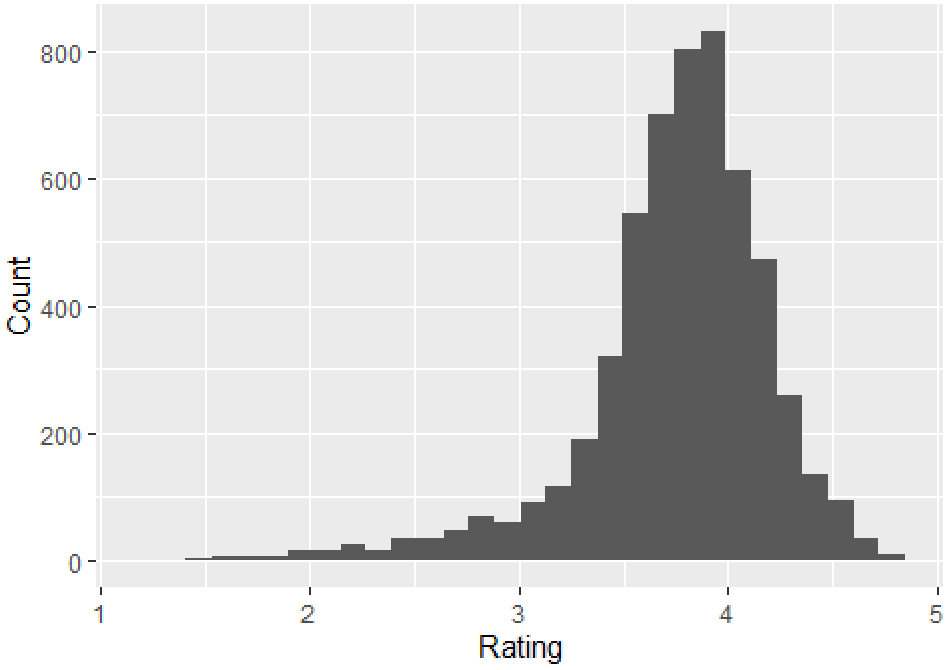
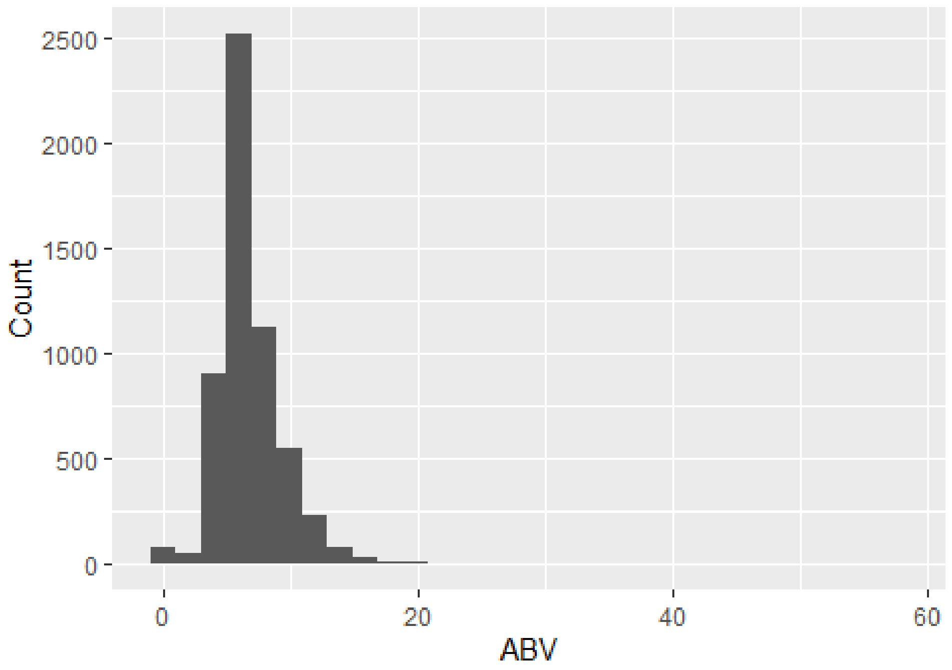


Figure 3.1. A histogram describing the distribution of beers rating

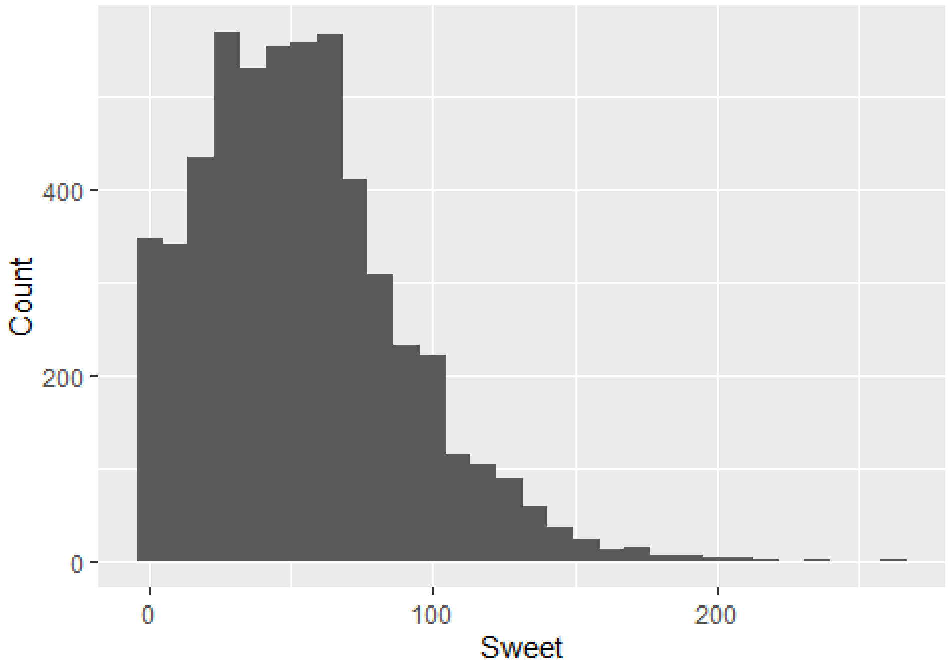
```
# Verifying our variable called ABV (alcohol by volume) (little positive
skewed since there are some beers that contain a high level of alcohol)

beers %>% ggplot() + geom_histogram(aes(ABV))+labs(x="ABV",
y="Count",title="Figure 3.2. A histogram describing the
distribution of beers level of ABV")+theme(plot.title = element_text(hjust =
0.5))

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

Figure 3.2. A histogram describing the distribution of beers level of ABV



```
# Verifying our variable called Sweet (positive skewed)

beers %>% ggplot() + geom_histogram(aes(Sweet))+labs(x="Sweet",
y="Count",title="Figure 3.3. A histogram describing the
distribution of beers Sweeteness")+theme(plot.title = element_text(hjust =
0.5))

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

Figure 3.3. A histogram describing the distribution of beers Sweeteness



```
# Verifying our variable called Malty (positive skewed)

beers %>% ggplot() + geom_histogram(aes(Malty))+labs(x="Malty",
y="Count",title="Figure 3.4. A histogram describing the
distribution of beers level of Malty")+theme(plot.title = element_text(hjust
= 0.5))

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

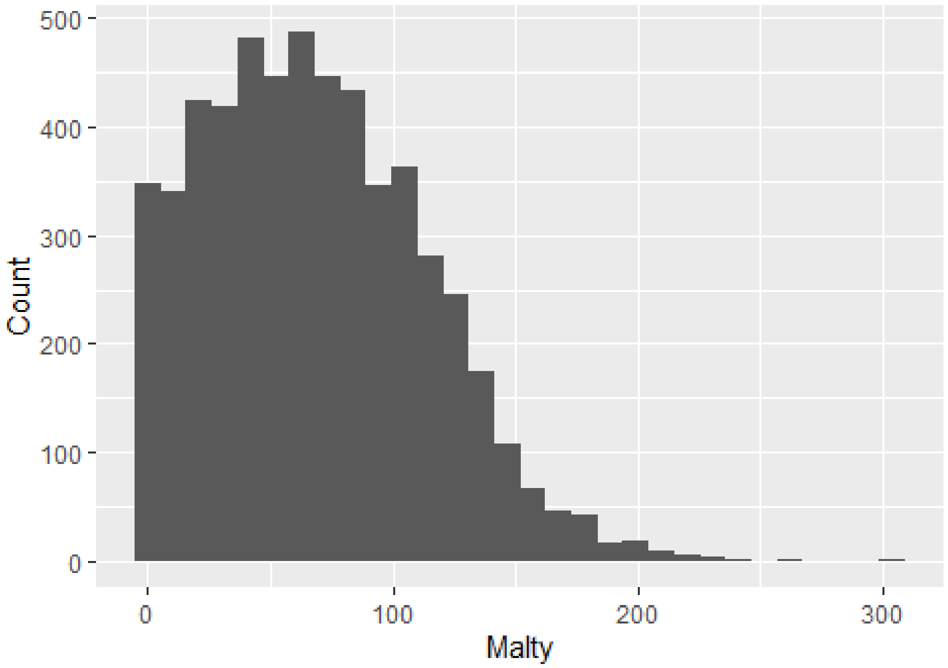


Figure 3.4. A histogram describing the distribution of beers level of Malty

Nevertheless, for this specific section we are only focus on the following data: (rating,Malty,Sweet,ABV)

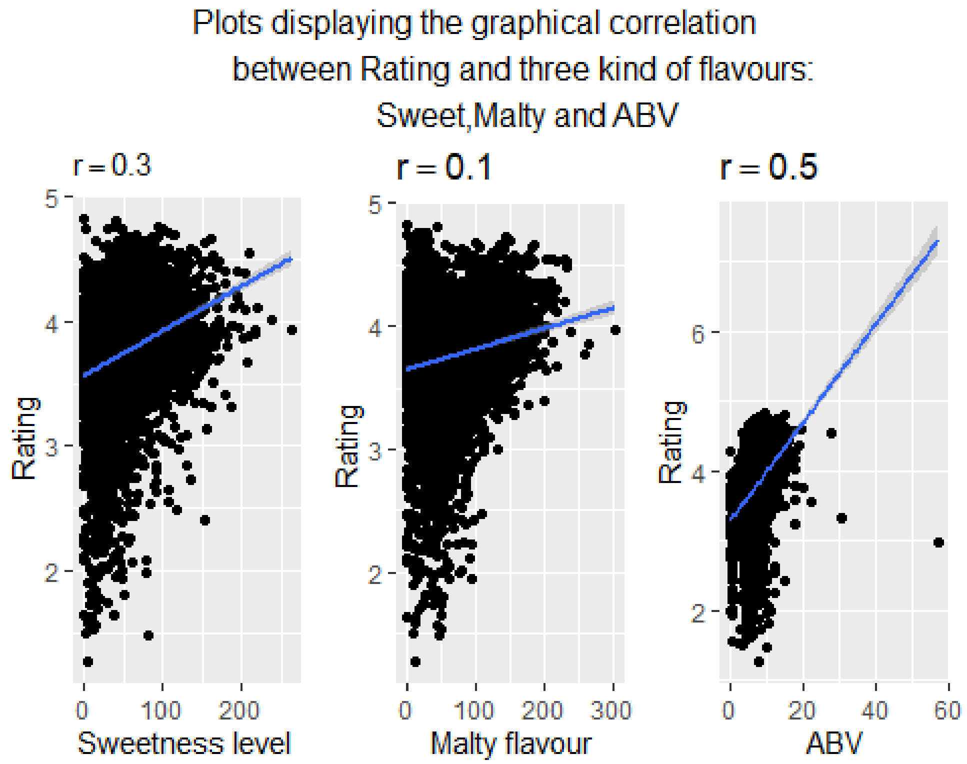
```
# Creating a new selection for our analysis for question 3

beerssection2 <- beers %>% select(rating,Malty,Sweet,ABV)
```

Correlation Patterns

Before running our test , we would like to see potential correlation patterns between our variables:

```
grid.arrange(
    ggplot(beerssection2, aes(y=rating, x=Sweet)) + geom_point() +
labs(y="Rating", x="Sweetness level ",subtitle=expression(r== 0.30)) +
geom_smooth(method=lm),
    ggplot(beerssection2, aes(y=rating, x=Malty)) + geom_point() +
labs(y="Rating ", x="Malty flavour", title=expression(r==0.10)) +
geom_smooth(method=lm),
    ggplot(beerssection2, aes(y=rating, x=ABV)) + geom_point() +
labs(y="Rating", x="ABV", title=expression(r==0.50)) +
geom_smooth(method=lm),
    ncol=3,top="Plots displaying the graphical correlation
                between Rating and three kind of flavours:
                Sweet,Malty and ABV")
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



```
#For non-normally distributed data we use spearman correlation.This uses the
rank of the values we are going to perform the correlation on, rather than
the continuous values.it uses the rank position of all numbers rather than
the absolute number

all.flavours <- beers %>% select(4:18)

rcorr(as.matrix(all.flavours), type="spearman")

##               ABV rating minIBU maxIBU Astringency  Body Alcohol Bitter
Sweet  Sour Salty Fruits
## ABV         1.00   0.50   0.40   0.48       -0.19  0.29    0.59   0.10
0.43  0.13 -0.20   0.26
## rating      0.50   1.00   0.28   0.28        0.07  0.30    0.27   0.15
0.30  0.25 -0.05   0.33
## minIBU      0.40   0.28   1.00   0.81        0.03  0.40    0.33   0.56
0.28  0.05 -0.03  0.10
## maxIBU      0.48   0.28   0.81   1.00       -0.02  0.36    0.38   0.51
0.31  0.09 -0.07   0.20
## Astringency -0.19   0.07   0.03  -0.02        1.00  0.17    0.10   0.33
0.19  0.61  0.42   0.50
## Body         0.29   0.30   0.40   0.36        0.17  1.00    0.53   0.62
0.67  0.12 -0.04   0.19
```

```
## Alcohol      0.59   0.27   0.33   0.38        0.10  0.53    1.00   0.29
0.64  0.34  0.00   0.44
## Bitter       0.10   0.15   0.56   0.51        0.33  0.62    0.29   1.00
0.33  0.15  0.12   0.14
## Sweet        0.43   0.30   0.28   0.31        0.19  0.67    0.64   0.33
1.00  0.42 -0.04   0.52
## Sour         0.13   0.25   0.05   0.09        0.61  0.12    0.34   0.15
0.42  1.00  0.25   0.90
## Salty       -0.20  -0.05  -0.03  -0.07        0.42 -0.04    0.00   0.12 -
0.04  0.25  1.00   0.19
## Fruits       0.26   0.33   0.10   0.20        0.50  0.19    0.44   0.14
0.52  0.90  0.19   1.00
## Hoppy       -0.05  -0.01   0.40   0.35        0.53  0.31    0.20   0.79
0.20  0.39  0.29   0.32
## Spices       0.30   0.26   0.14   0.20        0.15  0.48    0.53   0.21
0.41  0.24  0.05   0.33
## Malty        0.17   0.10   0.37   0.34        0.12  0.82    0.46   0.64
0.61 -0.04  0.01   0.02
##             Hoppy Spices Malty
## ABV         -0.05   0.30  0.17
## rating      -0.01   0.26  0.10
## minIBU       0.40   0.14  0.37
## maxIBU       0.35   0.20  0.34
## Astringency  0.53   0.15  0.12
## Body         0.31   0.48  0.82
## Alcohol      0.20   0.53  0.46
## Bitter       0.79   0.21  0.64
## Sweet        0.20   0.41  0.61
## Sour         0.39   0.24 -0.04
## Salty        0.29   0.05  0.01
## Fruits       0.32   0.33  0.02
## Hoppy        1.00   0.15  0.37
## Spices       0.15   1.00  0.41
## Malty        0.37   0.41  1.00
##
## n= 5558
##
##
## P
##             ABV    rating minIBU maxIBU Astringency Body   Alcohol Bitter
Sweet  Sour   Salty
## ABV                0.0000 0.0000 0.0000 0.0000      0.0000 0.0000  0.0000
0.0000 0.0000 0.0000
## rating      0.0000        0.0000 0.0000 0.0000      0.0000 0.0000  0.0000
0.0000 0.0000 0.0000
## minIBU      0.0000 0.0000        0.0000 0.0158      0.0000 0.0000  0.0000
0.0000 0.0004 0.0143
## maxIBU      0.0000 0.0000 0.0000        0.2432      0.0000 0.0000  0.0000
0.0000 0.0000 0.0000
## Astringency 0.0000 0.0000 0.0158 0.2432             0.0000 0.0000  0.0000
```

```
0.0000 0.0000 0.0000
## Body        0.0000 0.0000 0.0000 0.0000 0.0000             0.0000  0.0000
0.0000 0.0000 0.0028
## Alcohol     0.0000 0.0000 0.0000 0.0000 0.0000      0.0000         0.0000
0.0000 0.0000 0.8547
## Bitter      0.0000 0.0000 0.0000 0.0000 0.0000      0.0000 0.0000
0.0000 0.0000 0.0000
## Sweet       0.0000 0.0000 0.0000 0.0000 0.0000      0.0000 0.0000  0.0000
0.0000 0.0051
## Sour        0.0000 0.0000 0.0004 0.0000 0.0000      0.0000 0.0000  0.0000
0.0000        0.0000
## Salty       0.0000 0.0000 0.0143 0.0000 0.0000      0.0028 0.8547  0.0000
0.0051 0.0000
## Fruits      0.0000 0.0000 0.0000 0.0000 0.0000      0.0000 0.0000  0.0000
0.0000 0.0000 0.0000
## Hoppy       0.0004 0.4593 0.0000 0.0000 0.0000      0.0000 0.0000  0.0000
0.0000 0.0000 0.0000
## Spices      0.0000 0.0000 0.0000 0.0000 0.0000      0.0000 0.0000  0.0000
0.0000 0.0000 0.0000
## Malty       0.0000 0.0000 0.0000 0.0000 0.0000      0.0000 0.0000  0.0000
0.0000 0.0066 0.5047
##             Fruits Hoppy  Spices Malty
## ABV         0.0000 0.0004 0.0000 0.0000
## rating      0.0000 0.4593 0.0000 0.0000
## minIBU      0.0000 0.0000 0.0000 0.0000
## maxIBU      0.0000 0.0000 0.0000 0.0000
## Astringency 0.0000 0.0000 0.0000 0.0000
## Body        0.0000 0.0000 0.0000 0.0000
## Alcohol     0.0000 0.0000 0.0000 0.0000
## Bitter      0.0000 0.0000 0.0000 0.0000
## Sweet       0.0000 0.0000 0.0000 0.0000
## Sour        0.0000 0.0000 0.0000 0.0066
## Salty       0.0000 0.0000 0.0000 0.5047
## Fruits             0.0000 0.0000 0.1004
## Hoppy       0.0000        0.0000 0.0000
## Spices      0.0000 0.0000        0.0000
## Malty       0.1004 0.0000 0.0000
```

Properties that correlate significantly with the rating: ABV,BITTER,ALCOHOL AND FRUITS. All being the highest positive correlated with rating.

- 0.56 Bitter
- 0.50 ABV
- 0.33 FRUITS
- 0.33 Alcohol
- 0.30 BODY
- 0.28 minIBU
- 0.28 maxIBU

- 0.28 Sweet
- 0.26 Spices
- 0.25 SOUR
- 0.10 Malty

The Request

1 On average, does a beer receive a higher rating if it has a higher or lower ABV?

NHSTing regression coefficients

Creating linear regression model for ABV independent variable and rating as dependent

$$rating = \beta_{Intercept} + \beta_{ABV} \times ABV + \epsilon$$

```
m.rating.by.ABV <- lm(rating~ABV, data=beerssection2)
summary(m.rating.by.ABV)
```

```
## 
## Call:
## lm(formula = rating ~ ABV, data = beerssection2)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -4.3454 -0.1461  0.0522  0.2322  1.0090 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 3.296594   0.015343  214.85   <2e-16 ***
## ABV         0.069892   0.002162   32.33   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.4064 on 5556 degrees of freedom
## Multiple R-squared:  0.1583, Adjusted R-squared:  0.1582 
## F-statistic:  1045 on 1 and 5556 DF,  p-value: < 2.2e-16
```

The `summary()` output contains a NHST of whether the coefficient is zero or not.

We can also use `anova()` to compare a model of rating rates with and without ABV

- By adding ABV (Alcohol by volume) we can improve our model. $F(1{,}5556) = 1045$, $p < .0000$

```
anova(m.rating.by.ABV)
```

```
## Analysis of Variance Table
## 
## Response: rating
##             Df Sum Sq Mean Sq F value    Pr(>F)    
## ABV          1 172.61 172.611    1045 < 2.2e-16 ***
## Residuals 5556 917.70   0.165                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This test compares the fit of the model

$$rating = \beta_{Intercept} + \beta_{ABV} \times ABV + \epsilon$$

with the simpler model where $rating$ is unrelated to ABV

$$rating = \beta_{Intercept} + \epsilon$$

The F ratio is related to the t statistic: $F = t^2$—so they tell us exactly the same thing

This `anova()` based model approach is particularly useful for more comparing the performance of more complex models

By using the anova() function it will remove our additional predictor ABV from the model and perform a F test to check whether our model is a significant better model overall with ABV included vs the simpler version where there is no predictor.In this case, adding our ABV predictor does provide a significant better model overall.

Estimation approach

Calculating the confidence interval for our ABV coefficient in our linear model which tries to explain the variance of rating by using ABV as a predictor.

```
cbind(coefficients=coef(m.rating.by.ABV),confint(m.rating.by.ABV))
```

```
##             coefficients      2.5 %     97.5 %
## (Intercept)   3.29659406 3.26651489 3.32667323
## ABV           0.06989209 0.06565365 0.07413053
```

Note the difference between the NHST approach and the estimation approach

- The NHST approach tells us that the increase in rating is not zero
- The estimation approach tells us how much is the increase in rating

2 *having more or less Sweet or Malty elements in the flavour results in higher or lower ratings?*

NHSTing regression coefficients

Creating linear regression model for rating and sweet,malty and ABV as independent variables

$$rating = \beta_{Intercept} + \beta_{sweet} \times sweet + \beta_{malty} \times malty + \beta_{ABV} \times ABV + \epsilon$$

```
m.rating.by.Sweet.Malty.ABV <- lm(rating~Sweet+Malty+ABV, data=beerssection2)
summary(m.rating.by.Sweet.Malty.ABV )
```

```
## 
## Call:
## lm(formula = rating ~ Sweet + Malty + ABV, data = beerssection2)
## 
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -3.6982 -0.1560  0.0462  0.2331  1.0910 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 3.2576811  0.0162110 200.955   <2e-16 ***
## Sweet       0.0017667  0.0001943   9.094   <2e-16 ***
## Malty       0.0002375  0.0001457   1.630    0.103    
## ABV         0.0590198  0.0023340  25.287   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.4014 on 5554 degrees of freedom
## Multiple R-squared:  0.1793, Adjusted R-squared:  0.1789 
## F-statistic: 404.6 on 3 and 5554 DF,  p-value: < 2.2e-16
```

The summary() output contains a NHST of whether the coefficient is zero or not.

$$rating = \beta_{Intercept} + \beta_{sweet} \times sweet + \beta_{ABV} \times ABV + \epsilon$$

```
m.rating.by.Sweet.ABV <- lm(rating~Sweet+ABV, data=beerssection2)
summary(m.rating.by.Sweet.ABV)
```

```
## 
## Call:
## lm(formula = rating ~ Sweet + ABV, data = beerssection2)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -3.6966 -0.1555  0.0461  0.2330  1.0837 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 3.2660727  0.0153742  212.44   <2e-16 ***
## Sweet       0.0019367  0.0001639   11.81   <2e-16 ***
## ABV         0.0588357  0.0023316   25.23   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.4014 on 5555 degrees of freedom
## Multiple R-squared:  0.1789, Adjusted R-squared:  0.1786 
## F-statistic: 605.3 on 2 and 5555 DF,  p-value: < 2.2e-16
```

We can also use anova() to compare a model of rating rates with and without Sweet and Malty

This test compares the fit of the model

$$rating = \beta_{Intercept} + \beta_{Sweet} \times Sweet + \beta_{Malty} \times Malty + \beta_{ABV} \times ABV + \epsilon$$

with the simpler model where $rating$ is unrelated to ABV

$$rating = \beta_{Intercept} + \epsilon$$

The F ratio is related to the t statistic: $F = t^2$—so they tell us exactly the same thing

If we want to directly compare the fit of any two nested models we can use `anova()` to do so

By performing anova(). The following result was obtained:

```
## the 1 is telling r we want to predict our model with only the intercept

m.rating.baseline <- lm(rating~1, data = beerssection2)
summary(m.rating.baseline)

##
## Call:
## lm(formula = rating ~ 1, data = beerssection2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.49024 -0.17024  0.05976  0.27976  1.06976
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.760239   0.005942   632.9   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.443 on 5557 degrees of freedom

##comparing our two models

anova(m.rating.baseline, m.rating.by.Sweet.ABV)

## Analysis of Variance Table
##
## Model 1: rating ~ 1
## Model 2: rating ~ Sweet + ABV
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1   5557 1090.31
## 2   5555  895.21  2     195.1 605.34 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This shows us that including the variables `Sweet + ABV` do add significant predictive accuracy to our model, so it is explaining a significant amount of variance in y independent of that also explained by the variable `rating`.

- "Model comparison shows that a regression model including `Sweet + ABV` results in a significantly better overall fit than a model only including no predictor $F(2,5555) = 605.34, p < 0.000$."

Estimation approach

Calculating the 95% confidence interval for our coefficients (Sweet.Malty.ABV) in our linear model which try to explain the variance of rating by using Sweet.Malty.ABV as a predictors.

```
cbind(coefficients=coef(m.rating.by.Sweet.Malty.ABV
),confint(m.rating.by.Sweet.Malty.ABV ))

##              coefficients         2.5 %      97.5 %
## (Intercept) 3.2576810933  3.225901e+00 3.289460919
## Sweet       0.0017667114  1.385850e-03 0.002147572
## Malty       0.0002374678 -4.810545e-05 0.000523041
## ABV         0.0590198467  5.444427e-02 0.063595420
```

Note the difference between the NHST approach and the estimation approach

- The NHST approach tells us that the increase in rating is not zero
- The estimation approach tells us how much is the increase in rating given a change on Sweet and ABV.

*3 Do the results suggest that beers with higher or lower ABVs should have different flavours if the company is trying to maximise ratings_*

For this question, we should look at interation terms:

Reason why we had to include interaction terms:

Run multiple linear regression for the effect of ABV and Sweet on rating

$$Rating = \beta_{Intercept} + \beta_{Sweet} \times Sweet + \beta_{ABV} \times ABV + \epsilon$$

```
# NHST approach

m.sweet.ABV.main <- lm(rating ~ Sweet + ABV, data = beerssection2)
summary(m.sweet.ABV.main)

##
## Call:
## lm(formula = rating ~ Sweet + ABV, data = beerssection2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6966 -0.1555  0.0461  0.2330  1.0837
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.2660727  0.0153742  212.44   <2e-16 ***
## Sweet       0.0019367  0.0001639   11.81   <2e-16 ***
## ABV         0.0588357  0.0023316   25.23   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4014 on 5555 degrees of freedom
## Multiple R-squared:  0.1789, Adjusted R-squared:  0.1786
## F-statistic: 605.3 on 2 and 5555 DF,  p-value: < 2.2e-16
```

```r
#Estimation approach

cbind(coef(m.sweet.ABV.main), confint(m.sweet.ABV.main))
```

```
##                              2.5 %      97.5 %
## (Intercept) 3.266072731 3.23593325 3.296212208
## Sweet       0.001936738 0.00161537 0.002258107
## ABV         0.058835653 0.05426476 0.063406543
```

$$Rating = \beta_{Intercept} + \beta_{Sweet} \times Sweet + \beta_{ABV} \times ABV + \beta_{Sweet:ABV} \times Sweet \times ABV + \epsilon$$

```r
# NHST approach

m.sweet.ABV.intr <- lm(rating~Sweet*ABV, data=beerssection2)
summary(m.sweet.ABV.intr)
```

```
##
## Call:
## lm(formula = rating ~ Sweet * ABV, data = beerssection2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1763 -0.1567  0.0455  0.2331  1.0866
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.189e+00  2.433e-02 131.055  < 2e-16 ***
## Sweet        3.448e-03  4.033e-04   8.550  < 2e-16 ***
## ABV          7.009e-02  3.598e-03  19.478  < 2e-16 ***
## Sweet:ABV   -1.997e-04  4.869e-05  -4.101 4.18e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4009 on 5554 degrees of freedom
## Multiple R-squared:  0.1814, Adjusted R-squared:  0.181
## F-statistic: 410.3 on 3 and 5554 DF,  p-value: < 2.2e-16
```

```r
#Estimation approach

cbind(coef(m.sweet.ABV.intr), confint(m.sweet.ABV.intr))
```

```
##                                    2.5 %        97.5 %
## (Intercept)  3.1886692461  3.1409715134  3.2363669788
## Sweet        0.0034480150  0.0026574678  0.0042385622
## ABV          0.0700855819  0.0630316312  0.0771395326
## Sweet:ABV   -0.0001996505 -0.0002950955 -0.0001042054
```

Testing if adding the interaction terms to our model improved its performance:

A model comparison shows that the model fit is significantly improved by the inclusion of the interaction term Sweet * ABV (F(1,5554) = 16.816 , p < 0.000)

```
## Analysis of Variance Table
##
## Model 1: rating ~ Sweet + ABV
## Model 2: rating ~ Sweet * ABV
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   5555 895.21
## 2   5554 892.51  1    2.7022 16.816 4.178e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Checking Multicolinearity

For the model considering only the main effects there is no evidence of multicolinearity since it is not above 5 whereas for the model with interactions there is a sign of multicolinearity for Sweet and Sweet:ABV .

```
vif(m.sweet.ABV.main)

##    Sweet      ABV
## 1.192036 1.192036

vif(m.sweet.ABV.intr)

## there are higher-order terms (interactions) in this model
## consider setting type = 'predictor'; see ?vif

##     Sweet       ABV Sweet:ABV
##  7.233917  2.847000 11.235226
```

Therefore, the high VIF scores for the interaction model are due to structural multicollinearity that come about from calculating the interaction term using the other two variables.

One of the ways we can see that structural multicollinearity is not a concern in the same way, is by re-centering the variables

```
beerssection2 <- mutate(beerssection2,
                      centred.Sweet = Sweet - mean(Sweet),
                      centred.Malty = Malty - mean(Malty),
                      centred.ABV = ABV - mean(ABV))
```

```
m.rating.intr.centred.sweet.abv <- lm(rating ~ centred.Sweet*centred.ABV,
data = beerssection2)
summary(m.rating.intr.centred.sweet.abv)
```

```
##
## Call:
## lm(formula = rating ~ centred.Sweet * centred.ABV, data = beerssection2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1763 -0.1567  0.0455  0.2331  1.0866
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 3.767e+00  5.660e-03 665.639  < 2e-16 ***
## centred.Sweet               2.124e-03  1.699e-04  12.497  < 2e-16 ***
## centred.ABV                 5.938e-02  2.332e-03  25.462  < 2e-16 ***
## centred.Sweet:centred.ABV -1.997e-04  4.869e-05  -4.101 4.18e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4009 on 5554 degrees of freedom
## Multiple R-squared:  0.1814, Adjusted R-squared:  0.181
## F-statistic: 410.3 on 3 and 5554 DF,  p-value: < 2.2e-16
```

```
cbind(coef(m.rating.intr.centred.sweet.abv),
confint(m.rating.intr.centred.sweet.abv))
```

```
##                                                   2.5 %        97.5 %
## (Intercept)                 3.7674854804  3.7563897597  3.7785812010
## centred.Sweet               0.0021235878  0.0017904760  0.0024566995
## centred.ABV                 0.0593783827  0.0548066175  0.0639501479
## centred.Sweet:centred.ABV -0.0001996505 -0.0002950955 -0.0001042054
```

```
vif(m.rating.intr.centred.sweet.abv)
```

```
## there are higher-order terms (interactions) in this model
## consider setting type = 'predictor'; see ?vif
```

```
##             centred.Sweet               centred.ABV
centred.Sweet:centred.ABV
##                  1.284393                  1.195888
1.111349
```

The full model shows high VIF scores for both main effects and the interaction term. However, VIF scores for the main effects model are all below 2, showing that there is little multicollinearity between the predictor variables and the high VIF scores for the full model are due to the structuralmulticollinearity caused by the interaction term.

Run multiple linear regression for the effect of ABV and Malty on rating

$$Rating = \beta_{Intercept} + \beta_{Malty} \times Malty + +\beta_{ABV} \times ABV + \epsilon$$

```
# NHST approach

m.Malty.ABV.main <- lm(rating ~ Malty + ABV, data = beerssection2)
summary(m.Malty.ABV.main)

##
## Call:
## lm(formula = rating ~ Malty + ABV, data = beerssection2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1243 -0.1523  0.0413  0.2236  1.0658
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.2523665  0.0163191 199.298  < 2e-16 ***
## Malty       0.0009487  0.0001238   7.663 2.13e-14 ***
## ABV         0.0667503  0.0021896  30.485  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4043 on 5555 degrees of freedom
## Multiple R-squared:  0.1671, Adjusted R-squared:  0.1668
## F-statistic: 557.3 on 2 and 5555 DF,  p-value: < 2.2e-16

# Estimation approach

cbind(coef(m.Malty.ABV.main), confint(m.Malty.ABV.main))

##                                   2.5 %      97.5 %
## (Intercept) 3.2523665091 3.2203746585 3.284358360
## Malty       0.0009486502 0.0007059559 0.001191345
## ABV         0.0667503129 0.0624577980 0.071042828
```

$$Rating = \beta_{Intercept} + \beta_{Malty} \times Malty + \beta_{ABV} \times ABV \beta_{Malty:ABV} \times Malty \times ABV + \epsilon$$

```
# NHST approach

m.Malty.ABV.intr <- lm(rating~Malty*ABV, data=beerssection2)
summary(m.Malty.ABV.intr)

##
## Call:
## lm(formula = rating ~ Malty * ABV, data = beerssection2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4392 -0.1513  0.0429  0.2279  1.0478
##
```

```
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.351e+00  2.547e-02 131.601  < 2e-16 ***
## Malty       -5.863e-04  3.281e-04  -1.787    0.074 .
## ABV          5.237e-02  3.590e-03  14.587  < 2e-16 ***
## Malty:ABV    2.139e-04  4.236e-05   5.049 4.57e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4034 on 5554 degrees of freedom
## Multiple R-squared:  0.1709, Adjusted R-squared:  0.1705
## F-statistic: 381.7 on 3 and 5554 DF,  p-value: < 2.2e-16

# Estimation approach

cbind(coef(m.Malty.ABV.intr), confint(m.Malty.ABV.intr))

##                                   2.5 %       97.5 %
## (Intercept)  3.3512296311  3.3013079361 3.401151e+00
## Malty       -0.0005862561 -0.0012294861 5.697383e-05
## ABV          0.0523664944  0.0453287900 5.940420e-02
## Malty:ABV    0.0002139100  0.0001308626 2.969575e-04
```

A model comparison shows that the model fit is significantly improved by the inclusion of the interaction term Malty * ABV (F(1,5554) = 25.497 , p < 0.000)

```
anova(m.Malty.ABV.main, m.Malty.ABV.intr)

## Analysis of Variance Table
##
## Model 1: rating ~ Malty + ABV
## Model 2: rating ~ Malty * ABV
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   5555 908.10
## 2   5554 903.95  1    4.1499 25.497 4.571e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Checking Multicolinearity

For the model considering only the main effects there is no evidence of multicolinearity since it is not above 5 whereas for the model with interactions there is a sign of multicolinearity for Malty and Malty:ABV.

```
vif(m.Malty.ABV.main)

##    Malty      ABV
## 1.036336 1.036336

vif(m.Malty.ABV.intr)
```

```
## there are higher-order terms (interactions) in this model
## consider setting type = 'predictor'; see ?vif

##     Malty       ABV Malty:ABV
##  7.311804  2.798014 10.282331
```

Therefore, the high VIF scores for the interaction model are due to structural multicollinearity that come about from calculating the interaction term using the other two variables.

One of the ways we can see that structural multicollinearity is not a concern in the same way, is by re-centering the variables

```
m.rating.intr.centred.malty.abv <- lm(rating ~ centred.Malty*centred.ABV,
data = beerssection2)
summary(m.rating.intr.centred.malty.abv)

##
## Call:
## lm(formula = rating ~ centred.Malty * centred.ABV, data = beerssection2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4392 -0.1513  0.0429  0.2279  1.0478
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               3.756e+00  5.484e-03 684.799  < 2e-16 ***
## centred.Malty             8.328e-04  1.256e-04   6.628 3.72e-11 ***
## centred.ABV               6.704e-02  2.186e-03  30.674  < 2e-16 ***
## centred.Malty:centred.ABV 2.139e-04  4.236e-05   5.049 4.57e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4034 on 5554 degrees of freedom
## Multiple R-squared:  0.1709, Adjusted R-squared:  0.1705
## F-statistic: 381.7 on 3 and 5554 DF,  p-value: < 2.2e-16

cbind(coef(m.rating.intr.centred.malty.abv),
confint(m.rating.intr.centred.malty.abv))

##                                                2.5 %       97.5 %
## (Intercept)               3.7557353270 3.7449836930 3.7664869610
## centred.Malty             0.0008327651 0.0005864602 0.0010790700
## centred.ABV               0.0670388816 0.0627543352 0.0713234280
## centred.Malty:centred.ABV 0.0002139100 0.0001308626 0.0002969575

vif(m.rating.intr.centred.malty.abv)

## there are higher-order terms (interactions) in this model
## consider setting type = 'predictor'; see ?vif
##             centred.Malty                centred.ABV 
centred.Malty:centred.ABV 
##                  1.072107                   1.037045 
1.034595
```

The full model shows high VIF scores for both main effects and the interaction term. However, VIF scores for the main effects model are all below 2, showing that there is little multicollinearity between the predictor variables and the high VIF scores for the full model are due to the structuralmulticollinearity caused by the interaction term.

Contrast for the prediction of rating with different levels of Malty and Sweet with ABV on the x axis

```
#For Sweet

#c(0.1, 0.5, 0.9) is where we want to predict rating based on 3 different 
values of ABV, a low, a medium and a high ABV. So, we used the quantile 
function to take the beerssection2$ABV, so 0.1 is the 10% quantile of the 
ABV, 0.5 is the 50% quantile of the ABV and 0.9 is the 90% quantile of ABV.


preds.rating.intr <- tibble(ABV = rep(quantile(beerssection2$ABV, c(0.1, 0.5, 
0.9)), 2),
                    Sweet = c(rep(min(beerssection2$Sweet), 3), 
rep(max(beerssection2$Sweet), 3)))

preds.rating.intr <- mutate(preds.rating.intr, rating.hat = 
predict(m.sweet.ABV.intr , preds.rating.intr),
                            ABV = factor(ABV))

p1<- ggplot(preds.rating.intr) + geom_line(aes(x = Sweet, y = rating.hat, 
colour = ABV)) + ylab("Predicted Rating")+labs(title="Figure 3.5. Figure 
demonstrating the interactions between ABV and Sweet 
when predicting rating")+xlab("Sweet")+scale_colour_manual(labels = c("Low", 
"Medium" , "High"), values = c("red", "green" , "blue"))


# For Malty

#c(0.1, 0.5, 0.9) is where we want to predict rating based on 3 different 
values of ABV, a low, a medium and a high ABV. So, we used the quantile 
function to take the beerssection2$ABV, so 0.1 is the 10% quantile of the 
ABV, 0.5 is the 50% quantile of the ABV and 0.9 is the 90% quantile of ABV.

preds.rating.intr2 <- tibble(ABV = rep(quantile(beerssection2$ABV, c(0.1, 
0.5, 0.9)), 2),
                    Malty = c(rep(min(beerssection2$Malty), 3), 
rep(max(beerssection2$Malty), 3)))
```

```
preds.rating.intr2 <- mutate(preds.rating.intr2, rating.hat =
predict(m.Malty.ABV.intr, preds.rating.intr2),
                              ABV = factor(ABV))

p2 <- ggplot(preds.rating.intr2) + geom_line(aes(x = Malty, y = rating.hat,
colour = ABV)) + ylab("Predicted Rating")+labs(title="Figure 3.6. Figure
demonstrating the interactions between ABV and Malty
when predicting rating")+xlab("Malty")+scale_colour_manual(labels = c("Low",
"Medium" , "High"), values = c("red", "green" , "blue"))

grid.arrange(p1,p2)
```

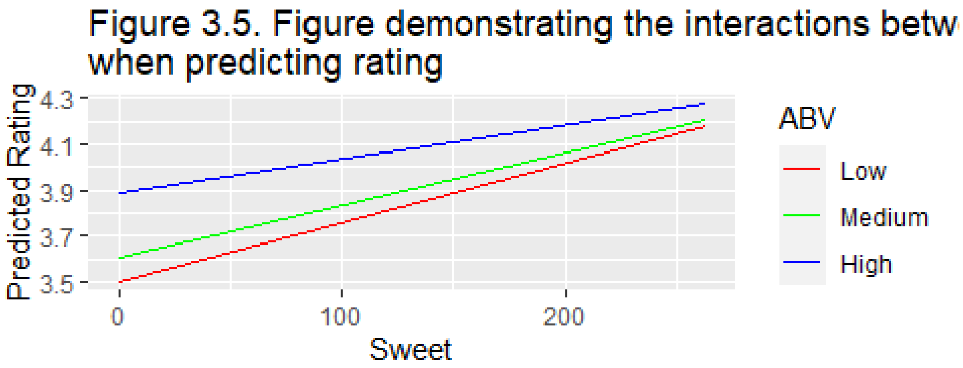


Figure 3.5. Figure demonstrating the interactions betw
when predicting rating

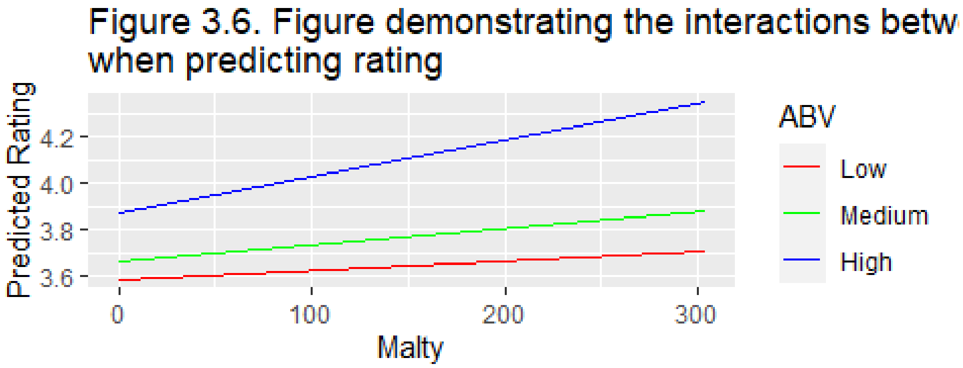


Figure 3.6. Figure demonstrating the interactions betw
when predicting rating

A few example values of Sweet and Malty, and then plot the slope of ABV for each of them

```
# For sweet

preds.rating.intr.main <- tibble(ABV = rep(quantile(beerssection2$ABV, c(0.1,
0.5, 0.9)), 2),
                    Sweet = c(rep(min(beerssection2$Sweet), 3),
rep(max(beerssection2$Sweet), 3)))

preds.rating.intr.main <- mutate(preds.rating.intr.main, intr.hat =
predict(m.sweet.ABV.intr , preds.rating.intr.main),main.hat =
predict(m.sweet.ABV.main, preds.rating.intr.main))
```

```r
sweet_interation <- filter(preds.rating.intr.main, ABV %in% c(4.5 ,6,10)) %>%
  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) +
  ylab("Prediction rating")+labs(title="Figure 3.7. Figure contrasting main
effects (solid line) and interaction (dashed-line)
between ABV and Sweet when predicting rating
")+xlab("Sweet")+scale_colour_manual(labels = c("Low", "Medium" , "High"),
values = c("red", "green" , "blue"))
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
generated.
```

```r
# For Malty

preds.rating.intr.main2 <- tibble(ABV = rep(quantile(beerssection2$ABV,
c(0.1, 0.5, 0.9)), 2),
                    Malty = c(rep(min(beerssection2$Malty), 3),
rep(max(beerssection2$Malty), 3)))

preds.rating.intr.main2 <- mutate(preds.rating.intr.main2, intr.hat =
predict(m.Malty.ABV.intr, preds.rating.intr.main2),main.hat =
predict(m.Malty.ABV.main, preds.rating.intr.main2))

malty_interation <- filter(preds.rating.intr.main2, ABV %in% c(4.5,6,10)) %>%
  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) +
  ylab("Prediction rating")+labs(title="Figure 3.8. Figure contrasting main
effects (solid line) and interaction (dashed-line)
between ABV and Malty when predicting rating
")+xlab("Malty")+scale_colour_manual(labels = c("Low", "Medium" , "High"),
values = c("red", "green" , "blue"))

grid.arrange(sweet_interation,malty_interation)
```

Figure 3.7. Figure contrasting main effects (solid line) a
between ABV and Sweet when predicting rating

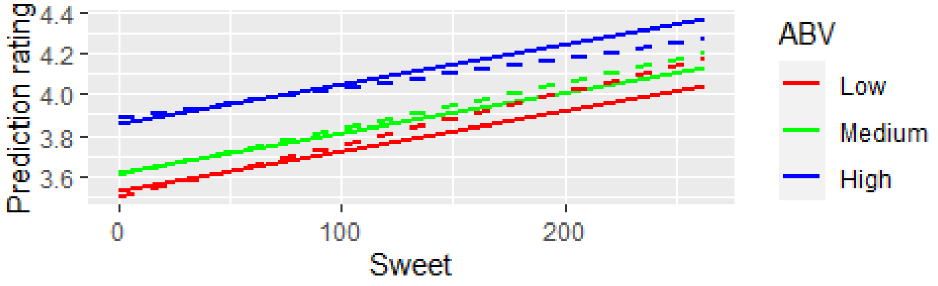
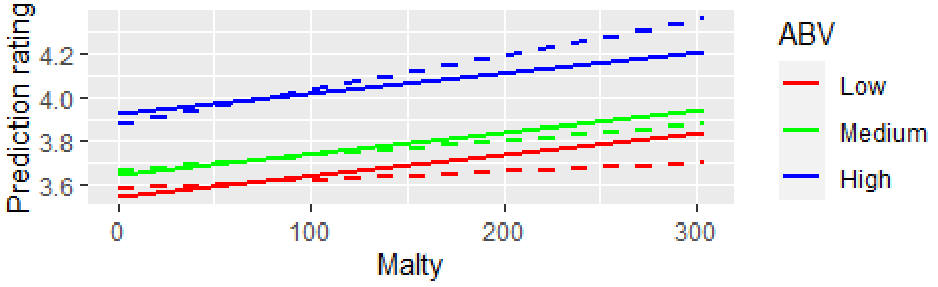


Figure 3.8. Figure contrasting main effects (solid line) a
between ABV and Malty when predicting rating



Section 2

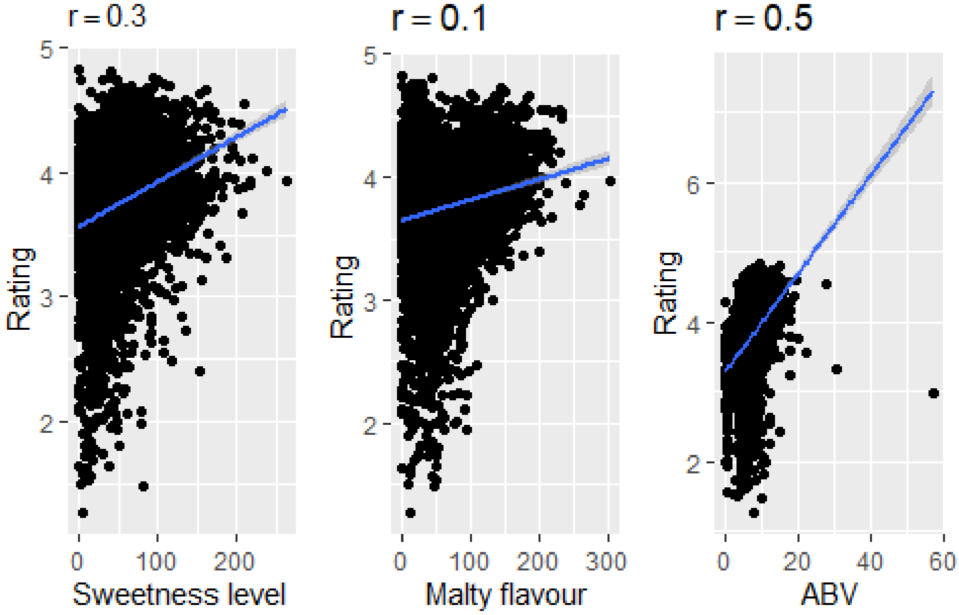
Report

- Correlation patterns

ABV and rating have a fairly strong correlation , almost close to 1. While the level of ABV goes up, the rating as well. Followed by sweet with a lower correlation of 0.3. Whereas, for Malty flavour we would not say that we can predict much of the variance on rating from Malty flavour.

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```

Plots displaying the graphical correlation between Rating and three kind of flavours: Sweet,Malty and ABV



1 On average, does a beer receive a higher rating if it has a higher or lower ABV?

NHSTing regression coefficients

Creating linear regression model for Score at the beginning of the academic year and tutoring as independent variable

$$rating = \beta_{Intercept} + \beta_{ABV} \times ABV + \epsilon$$

- On average a beer receives a higher rating if it has a higher ABV (alcohol by volume). There are 0.070 increase in rating for every extra level alcohol by volume (ABV) .This increase is significantly different from zero. $t(5556) = 32.33. p < .0001$

```
##
## Call:
## lm(formula = rating ~ ABV, data = beerssection2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3454 -0.1461  0.0522  0.2322  1.0090
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.296594   0.015343  214.85   <2e-16 ***
## ABV         0.069892   0.002162   32.33   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.4064 on 5556 degrees of freedom
## Multiple R-squared:  0.1583, Adjusted R-squared:  0.1582
## F-statistic:  1045 on 1 and 5556 DF,  p-value: < 2.2e-16
```

Estimation approach

- "There is a 0.070 increase in rating expected 95% CI[0.066-0.074] for every extra alcohol level by volume"

```
##             coefficients      2.5 %     97.5 %
## (Intercept)   3.29659406 3.26651489 3.32667323
## ABV           0.06989209 0.06565365 0.07413053
```

2 *having more or less Sweet or Malty elements in the flavour results in higher or lower ratings?*

To address this question NHSTing regression coefficients and estimation approaches for coefficients were developed to verify whether having more or less Sweet or Malty elements in the flavour results in higher or lower ratings.

NHSTing regression coefficients

Creating linear regression model for Score at the beginning of the academic year and tutoring as independent variable

$$rating = \beta_{Intercept} + \beta_{sweet} \times sweet + \beta_{malty} \times malty + \beta_{ABV} \times ABV + \epsilon$$

*Having more Sweet elements in the flavor results in higher ratings. "There is 0.0017 increase in rating for every extra sweetness. This increase is statistically significantly different from zero, $t(5554) = 9.094, p < .0001$", Then, there is a significant positive relationship between rating and sweetness.

- Additionally, "There is 0.059 increase in rating for every extra alcohol by volume level. This increase is statistically significantly different from zero, $t(5554) = 25.287, p < .0001$"Then, there is a significant positive relationship between rating and ABV.

- However, regarding Malty flavor we can say that this effect of Malty flavor on rating is not statistically significant, $t(5554)=1.630 $, $p > 0.05$"Then, there is no a significant relationship between rating and malty flavour.

```
##
## Call:
## lm(formula = rating ~ Sweet + Malty + ABV, data = beerssection2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6982 -0.1560  0.0462  0.2331  1.0910
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.2576811  0.0162110 200.955   <2e-16 ***
## Sweet       0.0017667  0.0001943   9.094   <2e-16 ***
## Malty       0.0002375  0.0001457   1.630    0.103    
## ABV         0.0590198  0.0023340  25.287   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.4014 on 5554 degrees of freedom
## Multiple R-squared:  0.1793, Adjusted R-squared:  0.1789 
## F-statistic: 404.6 on 3 and 5554 DF,  p-value: < 2.2e-16
```

Estimation approach

Calculating the 95% confidence interval for our coefficients (Sweet.Malty.ABV) in our linear model which try to explain the variance of rating by using Sweet.Malty.ABV as a predictors.

```
##             coefficients         2.5 %      97.5 %
## (Intercept) 3.2576810933  3.225901e+00 3.289460919
## Sweet       0.0017667114  1.385850e-03 0.002147572
## Malty       0.0002374678 -4.810545e-05 0.000523041
## ABV         0.0590198467  5.444427e-02 0.063595420
```

- “There is 0.05901 increase in rating expected 95% CI[0.0544-0.0635] for every extra alcohol level by volume” AND “There is 0.00177 increase in rating expected 95% CI[0.001385-0.00215] for every extra sweetness level”

*3 Do the results suggest that beers with higher or lower ABVs should have different flavours if the company is trying to maximise ratings_*

To address this question, it is necessary to include interaction terms since we want to reveal whether the effect of ABV on the rating variable changes, depending on the value of different flavors, this analysis focuses on Sweet and Malty.Hence, we will be able to test if beers with higher or lower ABVs should have different flavours if the company is trying to maximize ratings.

Reason why we had to include interaction terms:

Run multiple linear regression for the effect of ABV and Sweet on rating

- When including only the main effects:

$$Rating = \beta_{Intercept} + \beta_{Sweet} \times Sweet + \beta_{ABV} \times ABV + \epsilon$$

we see a significant effect of Sweet on the rating beers. A 1 percent increase of sweet rates is related to an increment in average rating of $0.0019 (t(5555) = 11.81, p < 0.000, 95% CI [0.0016, 0.00225]). Additionally,there was a significant effect of ABV on the rating beers. A 1 percent increase of ABV rates is related to an increment in average rating of $0.05883 (t(5555) = 25.23, p < 0.000, 95% CI [0.0542, 0.0634]).

```
## 
## Call:
## lm(formula = rating ~ Sweet + ABV, data = beerssection2)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6966 -0.1555  0.0461  0.2330  1.0837
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.2660727  0.0153742  212.44   <2e-16 ***
## Sweet       0.0019367  0.0001639   11.81   <2e-16 ***
## ABV         0.0588357  0.0023316   25.23   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4014 on 5555 degrees of freedom
## Multiple R-squared:  0.1789, Adjusted R-squared:  0.1786
## F-statistic: 605.3 on 2 and 5555 DF,  p-value: < 2.2e-16

##                               2.5 %      97.5 %
## (Intercept) 3.266072731 3.23593325 3.296212208
## Sweet       0.001936738 0.00161537 0.002258107
## ABV         0.058835653 0.05426476 0.063406543
```

- For a model including interaction terms

$$Rating = \beta_{Intercept} + \beta_{Sweet} \times Sweet + \beta_{ABV} \times ABV + \beta_{Sweet:ABV} \times Sweet \times ABV + \epsilon$$

There is a significant main effect of Sweet with an increase in rating of $0.003448 for every additional sweet level (t(5554) = 8.55, p < 0.000, 95% CI [0.002657,0.004238]).

There is a significant main effect of ABV with an increase in rating of $0.07008 for every additional ABV level (t(5554) = 19.47 p < 0.000, 95% CI [0.06303, 0.0771395]).

There is a significant negative interaction between Sweet and Malty of -0.000199 (t(5554) = -4.101 p < 0.0000 , 95% CI [-0.00029, -0.000104]).

```
##
## Call:
## lm(formula = rating ~ Sweet * ABV, data = beerssection2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1763 -0.1567  0.0455  0.2331  1.0866
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.189e+00  2.433e-02 131.055  < 2e-16 ***
## Sweet        3.448e-03  4.033e-04   8.550  < 2e-16 ***
## ABV          7.009e-02  3.598e-03  19.478  < 2e-16 ***
## Sweet:ABV   -1.997e-04  4.869e-05  -4.101 4.18e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.4009 on 5554 degrees of freedom
## Multiple R-squared:  0.1814, Adjusted R-squared:  0.181
## F-statistic: 410.3 on 3 and 5554 DF,  p-value: < 2.2e-16
```

```
##                                  2.5 %        97.5 %
## (Intercept)  3.1886692461  3.1409715134  3.2363669788
## Sweet        0.0034480150  0.0026574678  0.0042385622
## ABV          0.0700855819  0.0630316312  0.0771395326
## Sweet:ABV   -0.0001996505 -0.0002950955 -0.0001042054
```

Run multiple linear regression for the effect of ABV and Malty on rating

- When including only the main effects:

$$Rating = \beta_{Intercept} + \beta_{Malty} \times Malty + +\beta_{ABV} \times ABV + \epsilon$$

we see a significant effect of Malty on the rating beers. A 1 percent increase of Malty rates is related to an increment in average rating of $0.00094 (t(5555) = 7.663, p < 0.000, 95% CI [0.000705, 0.00119]). Additionally,there was a significant effect of ABV on the rating beers. A 1 percent increase of ABV rates is related to an increment in average rating of $0.0667 (t(5555) = 30.48, p < 0.000, 95% CI [0.062457, 0.071042]).

```
##
## Call:
## lm(formula = rating ~ Malty + ABV, data = beerssection2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1243 -0.1523  0.0413  0.2236  1.0658
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.2523665  0.0163191 199.298  < 2e-16 ***
## Malty       0.0009487  0.0001238   7.663 2.13e-14 ***
## ABV         0.0667503  0.0021896  30.485  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4043 on 5555 degrees of freedom
## Multiple R-squared:  0.1671, Adjusted R-squared:  0.1668
## F-statistic: 557.3 on 2 and 5555 DF,  p-value: < 2.2e-16
```

```
##                                2.5 %      97.5 %
## (Intercept) 3.2523665091 3.2203746585 3.284358360
## Malty       0.0009486502 0.0007059559 0.001191345
## ABV         0.0667503129 0.0624577980 0.071042828
```

- For a model including interaction terms

$$Rating = \beta_{Intercept} + \beta_{Malty} \times Malty + \beta_{ABV} \times ABV \beta_{Malty:ABV} \times Malty \times ABV + \epsilon$$

There is a significant main effect of ABV with an increase in rating of $0.052366 for every additional ABV level (t(5554) = 14.587 p < 0.000, 95% CI [0.045328, 0.059404]).

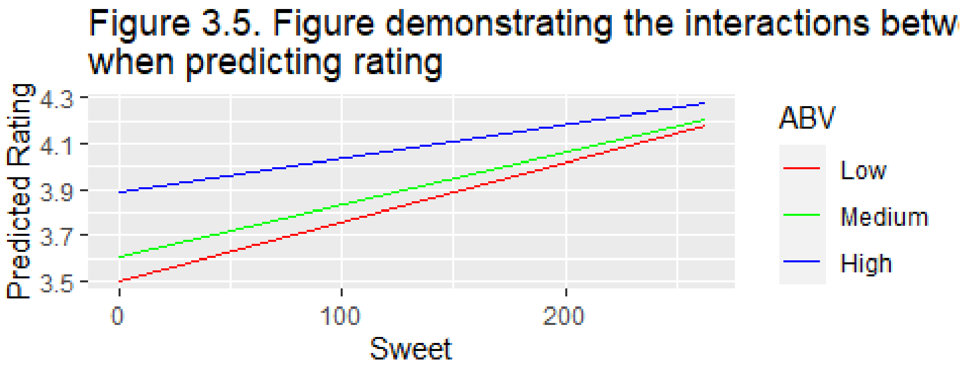
There is NO a significant main effect of Malty on rating (t(5554) = -1.787 p= 0.074 , 95% CI [0.045328, 0.059404]).

There is a significant positive interaction between ABV and Malty of 0.000213910 (t(5554) = 5.049 p < 0.0000 , 95% CI [0.00013, 0.00029695]).

```
## 
## Call:
## lm(formula = rating ~ Malty * ABV, data = beerssection2)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -3.4392 -0.1513  0.0429  0.2279  1.0478 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  3.351e+00  2.547e-02 131.601  < 2e-16 ***
## Malty       -5.863e-04  3.281e-04  -1.787    0.074 .  
## ABV          5.237e-02  3.590e-03  14.587  < 2e-16 ***
## Malty:ABV    2.139e-04  4.236e-05   5.049 4.57e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.4034 on 5554 degrees of freedom
## Multiple R-squared:  0.1709, Adjusted R-squared:  0.1705 
## F-statistic: 381.7 on 3 and 5554 DF,  p-value: < 2.2e-16
```

```
##                                   2.5 %       97.5 %
## (Intercept)  3.3512296311  3.3013079361 3.401151e+00
## Malty       -0.0005862561 -0.0012294861 5.697383e-05
## ABV          0.0523664944  0.0453287900 5.940420e-02
## Malty:ABV    0.0002139100  0.0001308626 2.969575e-04
```

Contrast for the prediction of rating with different levels of Malty and Sweet with ABV on the x axis



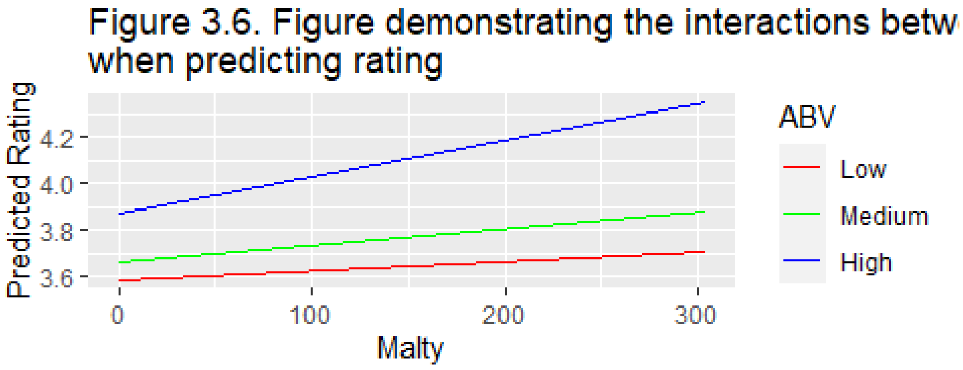


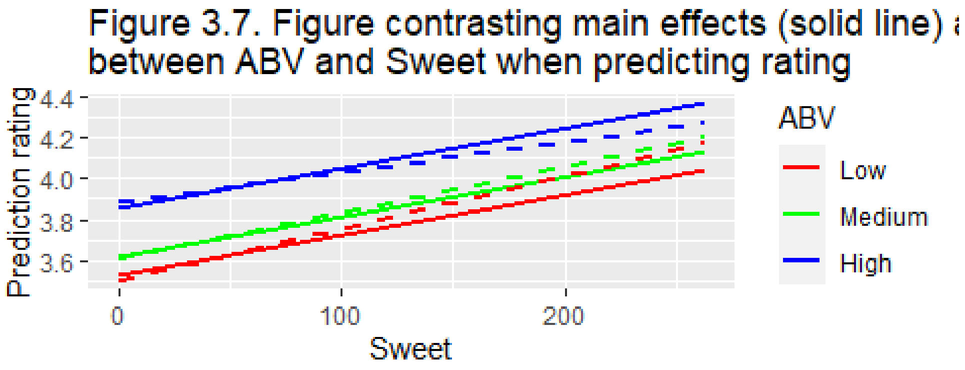
Figure 3.5. Figure demonstrating the interactions between ABV and Sweet when predicting rating

- The results of the full model indicate that there is a significant interaction between Sweet and ABV when predicting rating. The figure helps to demonstrate that when ABV is high, increasing Sweet predict lower rating, but when ABV is low, increasing sweet predict higher rating (it is steeper).

Figure 3.6. Figure demonstrating the interactions between ABV and Malty when predicting rating

- The results of the full model indicate that there is a significant interaction between Malty and ABV when predicting rating. The figure helps to demonstrate that when ABV is high, increasing Malty predict higher ratings (it is stepper), but when ABV is low, increasing Malty predict lower ratings.

A few example values of Sweet and Malty, and then plot the slope of ABV for each of them



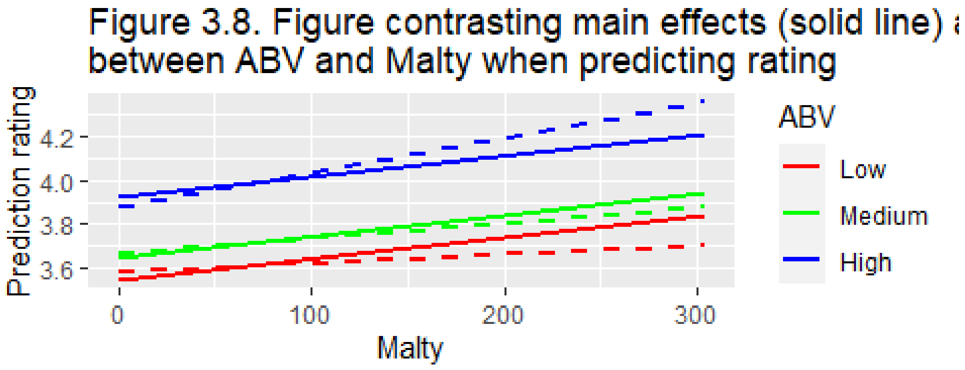


Figure 3.7. Figure contrasting main effects (solid line) and interaction (dashed-line) between ABV and Sweet when predicting rating

- In the main effects model, the slope of Sweet against Prediction rating is always parallel for all different values of ABV.
- In the interaction model, the slope of Sweet against Prediction rating is shallower for high values of ABV and steeper for low values of ABV.

Figure 3.8. Figure contrasting main effects (solid line) and interaction (dashed-line) between ABV and Malty when predicting rating

- In the main effects model, the slope of Malty against Prediction rating is always parallel for all different values of ABV
- In the interaction model, the slope of Malty against Prediction rating is shallower for low values of ABV and steeper for high values of ABV.

Having provided the relevant information to justify the use of interaction terms as for the results listed above. It is notable that in order to provide the appropriate answer to the question required, it is advisable to include interaction terms.

4 What flavourings should the company use more/less of if they are creating a high ABV beer?

When a beer contains a high ABV, there is a negative relationship between Sweetness and rating.Therefore, if a beer has high level of alcohol by volume, the sweetness level should be reduced so that the beverage will get higher rating patterns.Whereas the level of Malty should be higher for those beers with high ABV, then and the return to adding sweet is diminishing.

5 *What flavourings should the company use more/less of if they are creating a low ABV beer?*

On the other hand, if the company is creating a beer with low ABV the level of Sweetness should be increased while the malty flavour lowered.

Main Business Conclusion For Part 2b:

To produce a high ABV beer, using too much sweetness should be avoided. The interaction term is telling us that when ABV is high, increasing sweetness predicts significantly lower rating'. Nevertheless, adding more Malty flavors to a high ABV will be beneficial for ratings: The interaction term is telling us that when ABV is high, increasing malty predicts significantly higher rating'.

Concluding that, as has been mentioned above, if the company is creating a high ABV beer, then they should use more Malty flavours and less Sweetness whereas for the production of a low ABV they should use more Sweetness flavours and less Malty.