141 final_EDA

Ang Li

2023-05-20

Load the data set

```
data = read.csv("final.csv")
#View(data)
```

```
library("ggplot2")
library("dplyr")
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library("tm")
```

```
## Loading required package: NLP
```

```
##
## Attaching package: 'NLP'
```

```
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
library("SnowballC")
library("wordcloud")
```

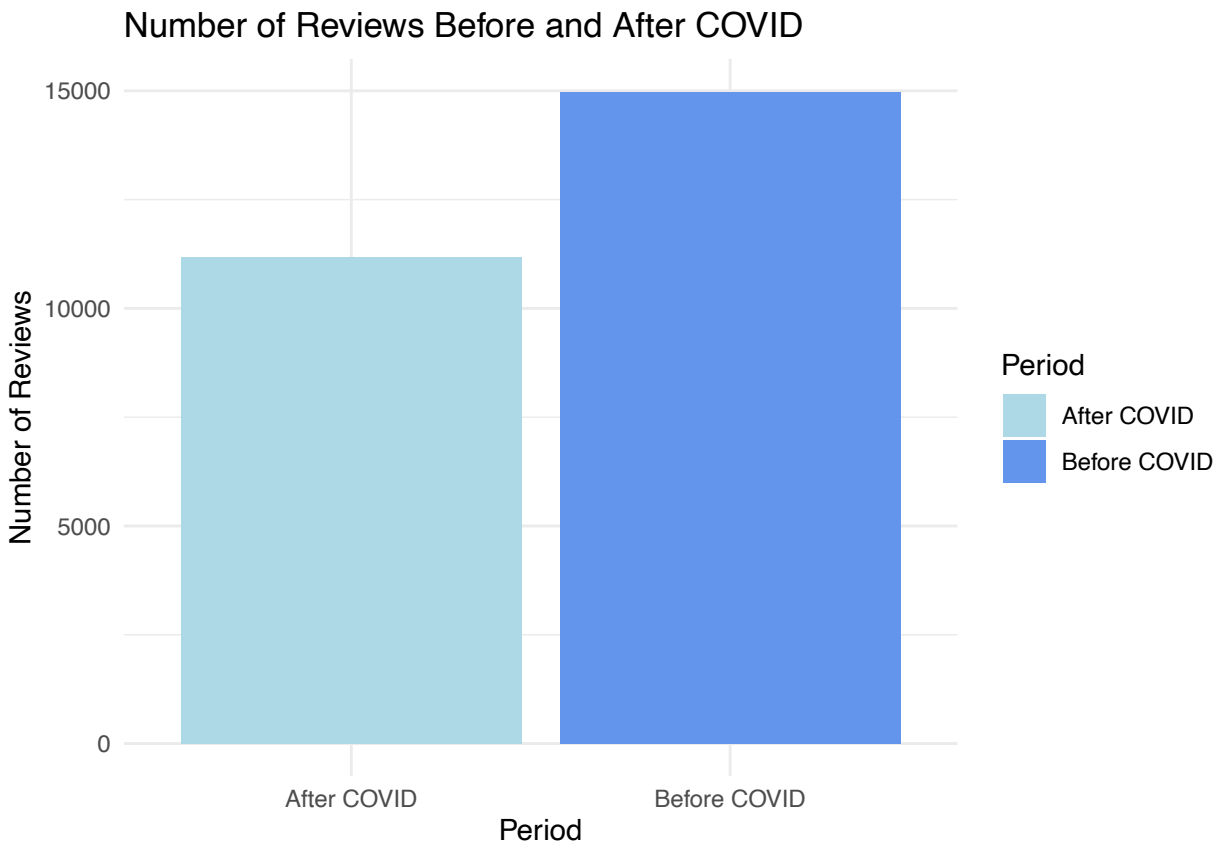
```
## Loading required package: RColorBrewer
```

```
library("RColorBrewer")
library("syuzhet")
library("ggplot2")
```

EDA

The number of reviews before and after covid

```
library(ggplot2)
data$COVID <- ifelse(data$Year < 2020, "Before COVID", "After COVID")
review_counts <- table(data$COVID)
plot_data <- data.frame(Period = names(review_counts), Count = as.numeric(review_counts))
# Plotting the graph
ggplot(plot_data, aes(x = Period, y = Count, fill = Period)) +
  geom_bar(stat = "identity") +
  labs(title = "Number of Reviews Before and After COVID",
       x = "Period",
       y = "Number of Reviews") +
  scale_fill_manual(values = c("lightblue", "cornflowerblue")) +
  theme_minimal()
```



Course rating based on south and north campus

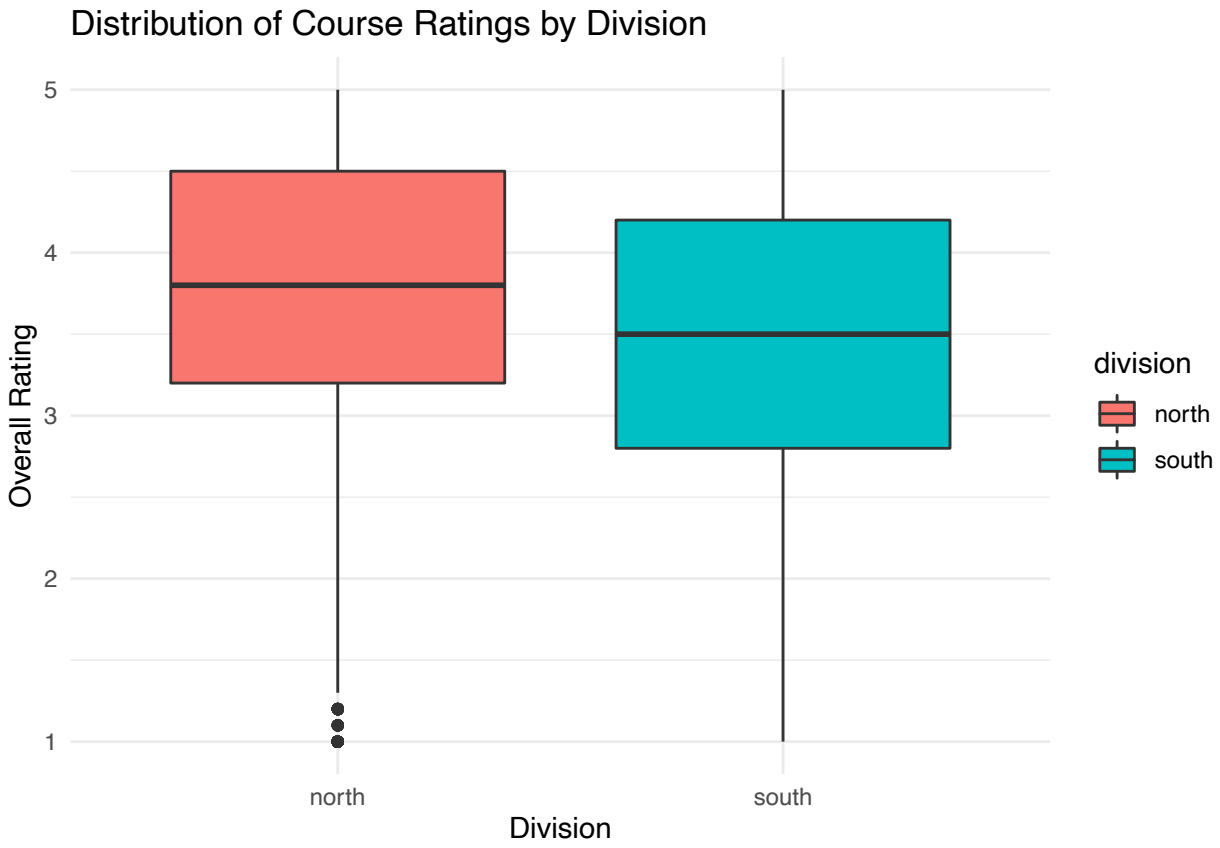
```
summary_stats <- data %>%
  group_by(division) %>%
  summarise(mean_overall = mean(Overall, na.rm = TRUE),
            median_overall = median(Overall, na.rm = TRUE),
            min_overall = min(Overall, na.rm = TRUE),
            max_overall = max(Overall, na.rm = TRUE),
            mean_easiness = mean(Easiness, na.rm = TRUE),
            median_easiness = median(Easiness, na.rm = TRUE),
            min_easiness = min(Easiness, na.rm = TRUE),
            max_easiness = max(Easiness, na.rm = TRUE),
            mean_clarity = mean(Clarity, na.rm = TRUE),
            median_clarity = median(Clarity, na.rm = TRUE),
            min_clarity = min(Clarity, na.rm = TRUE),
            max_clarity = max(Clarity, na.rm = TRUE),
            mean_workload = mean(Workload, na.rm = TRUE),
            median_workload = median(Workload, na.rm = TRUE),
            min_workload = min(Workload, na.rm = TRUE),
            max_workload = max(Workload, na.rm = TRUE),
            mean_helpfulness = mean(Helpfulness, na.rm = TRUE),
            median_helpfulness = median(Helpfulness, na.rm = TRUE),
            min_helpfulness = min(Helpfulness, na.rm = TRUE),
            max_helpfulness = max(Helpfulness, na.rm = TRUE))

# Print the summary statistics
transposed_stats <- t(summary_stats)

# Convert the transposed data frame to an xtable object
table_obj <- xtable::xtable(transposed_stats)

# Print the table
#print(table_obj)
```

% latex table generated in R 4.2.1 by xtable 1.8-4 package % Fri Jun 2 21:25:29 2023

Overall

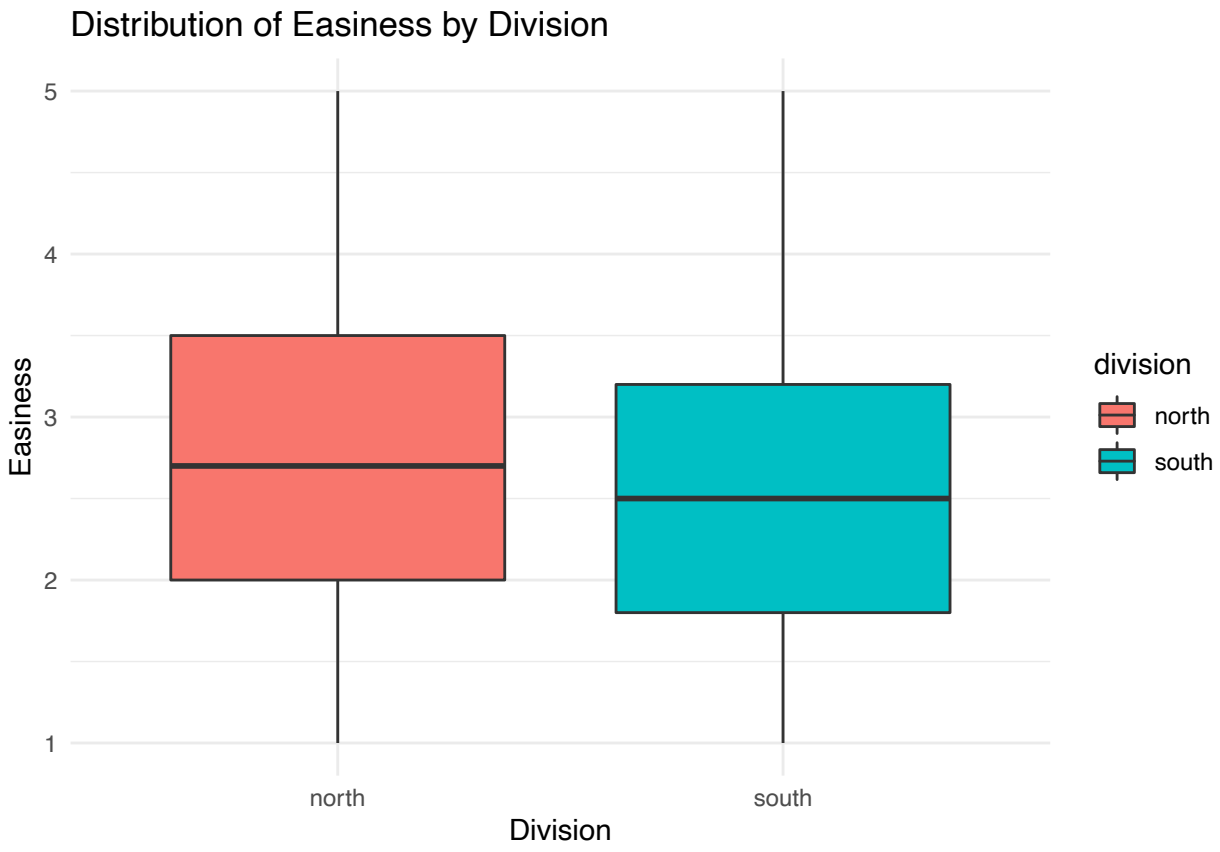
```
data = na.omit(data)
ggplot(data, aes(x = division, y = Overall, fill = division)) +
  geom_boxplot() +
  labs(title = "Distribution of Course Ratings by Division",
       x = "Division", y = "Overall Rating") +
  theme_minimal()
```

	1	2
division	north	south
mean_overall	3.752039	3.471436
median_overall	3.8	3.5
min_overall	1	1
max_overall	5	5
mean_easiness	2.781739	2.546778
median_easiness	2.7	2.5
min_easiness	1	1
max_easiness	5	5
mean_clarity	3.681305	3.399485
median_clarity	3.8	3.5
min_clarity	1	1
max_clarity	5	5
mean_workload	2.706349	2.586271
median_workload	2.6	2.5
min_workload	1	1
max_workload	5	5
mean_helpfulness	3.743536	3.544944
median_helpfulness	3.8	3.7
min_helpfulness	1	1
max_helpfulness	5	5



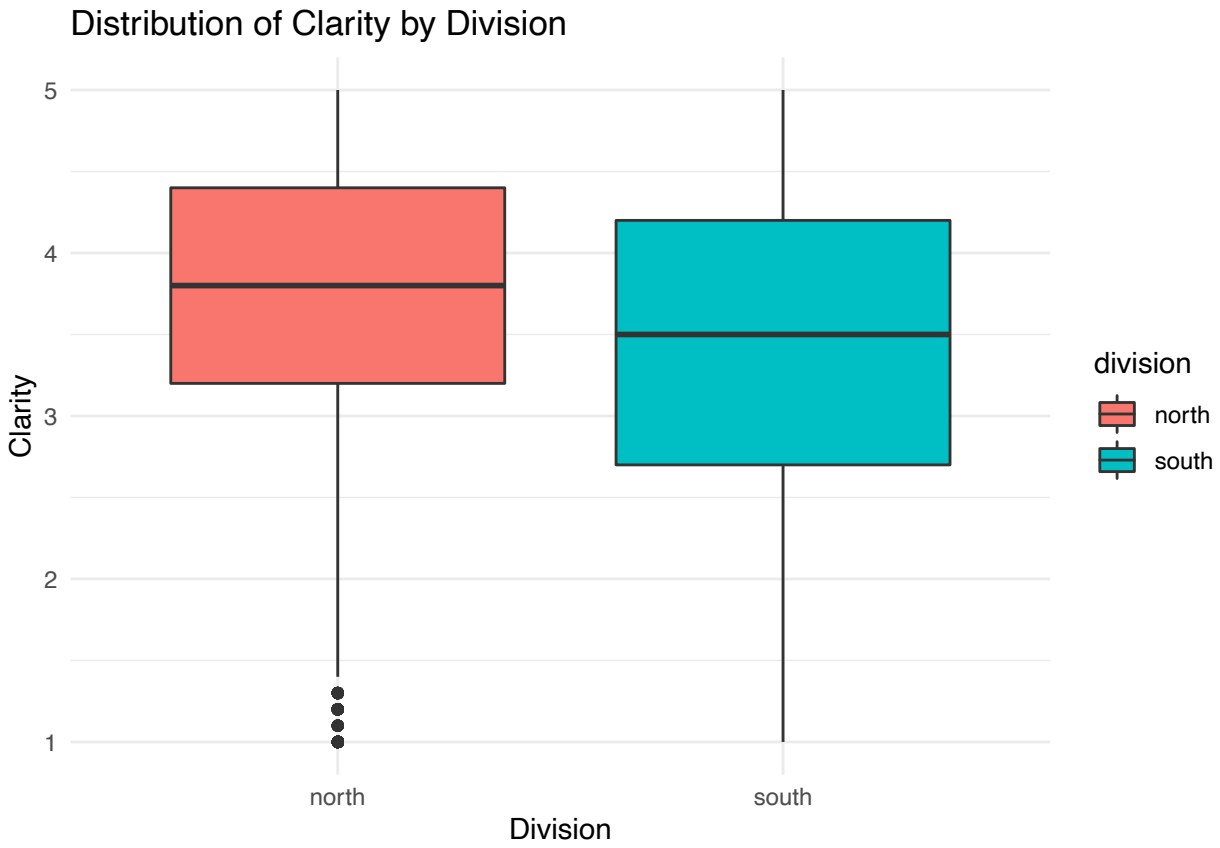
Easiness

```
ggplot(data, aes(x = division, y = Easiness, fill = division)) +
  geom_boxplot() +
  labs(title = "Distribution of Easiness by Division",
       x = "Division", y = "Easiness") +
  theme_minimal()
```



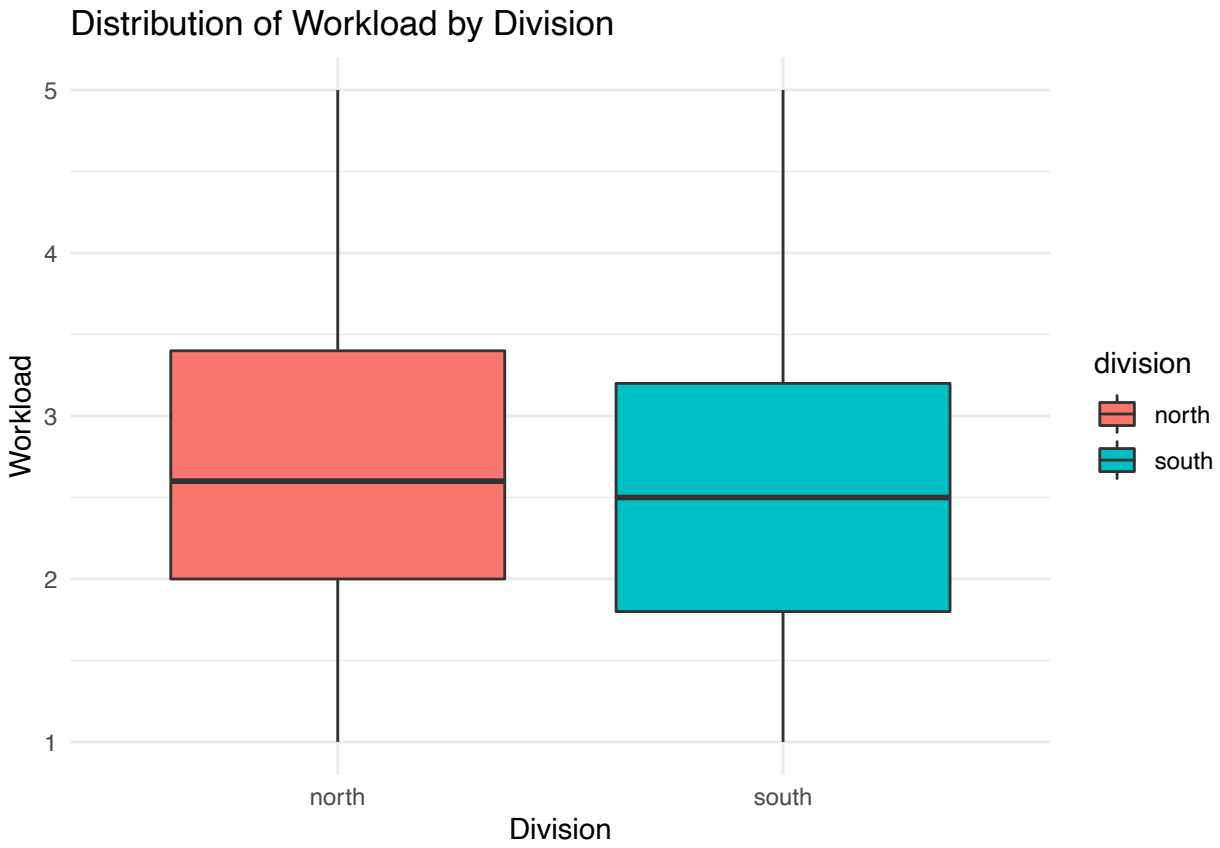
Clarity

```
ggplot(data, aes(x = division, y = Clarity, fill = division)) +
  geom_boxplot() +
  labs(title = "Distribution of Clarity by Division",
       x = "Division", y = "Clarity") +
  theme_minimal()
```



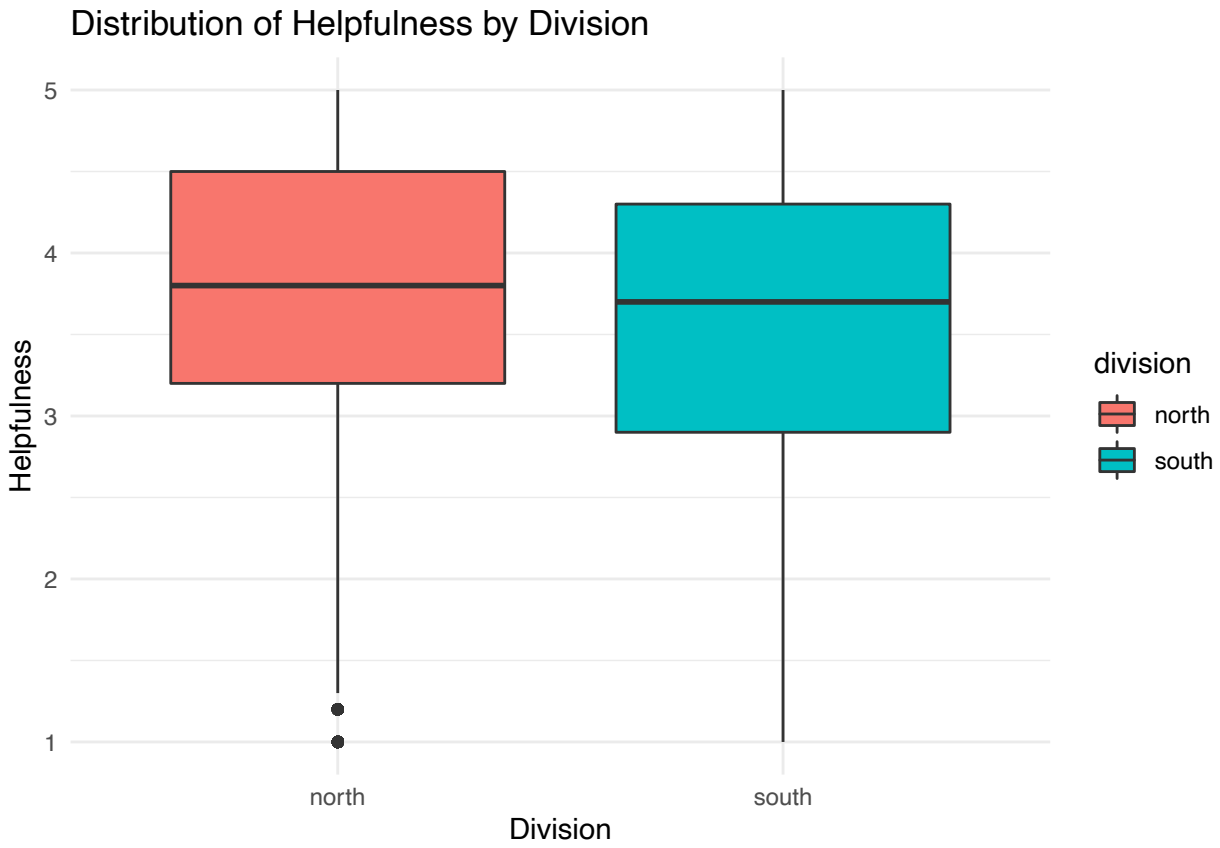
Workload

```
ggplot(data, aes(x = division, y = Workload, fill = division)) +
  geom_boxplot() +
  labs(title = "Distribution of Workload by Division",
       x = "Division", y = "Workload") +
  theme_minimal()
```



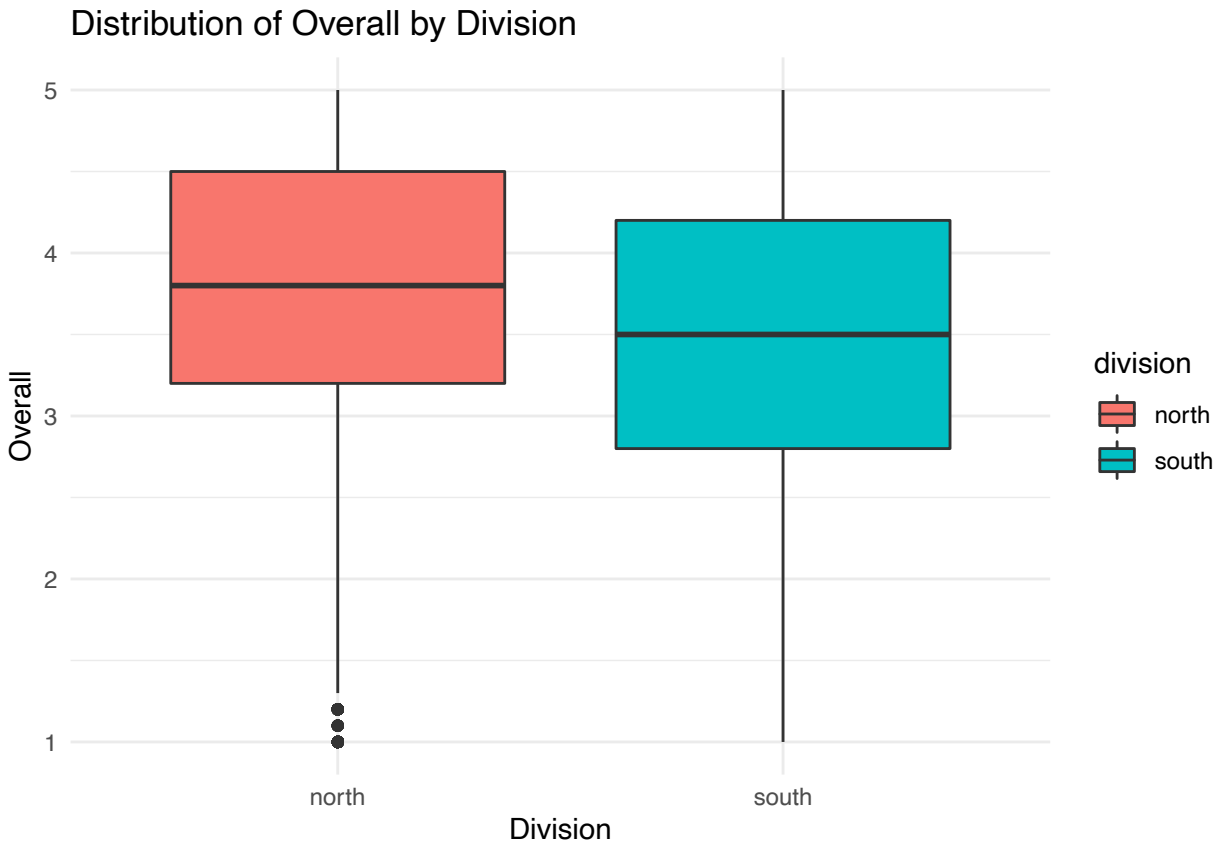
Helpfulness

```
ggplot(data, aes(x = division, y = Helpfulness, fill = division)) +
  geom_boxplot() +
  labs(title = "Distribution of Helpfulness by Division",
       x = "Division", y = "Helpfulness") +
  theme_minimal()
```



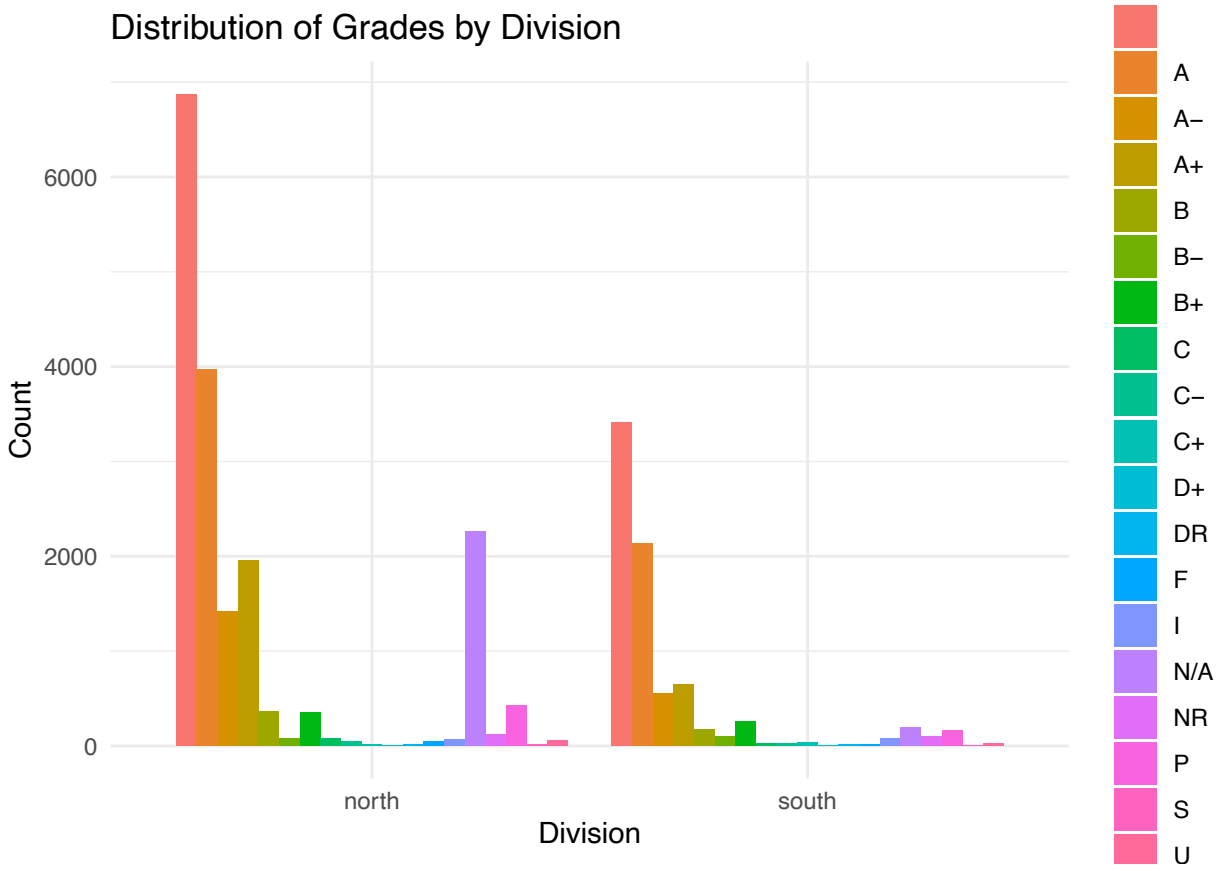
Overall

```
ggplot(data, aes(x = division, y = Overall, fill = division)) +
  geom_boxplot() +
  labs(title = "Distribution of Overall by Division",
       x = "Division", y = "Overall") +
  theme_minimal()
```



Grade Vs. Division

```
course_data_clean <- na.omit(data)

# Create a table of counts for each grade and division combination
grade_counts <- table(course_data_clean$division, course_data_clean$Grade)

# Convert the table into a data frame
grade_data <- as.data.frame(grade_counts)

# Create a grouped bar plot
ggplot(grade_data, aes(x = Var1, y = Freq, fill = Var2)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Distribution of Grades by Division",
       x = "Division", y = "Count") +
  theme_minimal()
```



"Review" Text Mining

```
# choose the upper division course
course_upper <- data %>%
  filter(as.numeric(gsub("[^0-9]", "", Course.Code)) > 100)

# division = south
south_course_upper <- course_upper %>%
  filter(division == "south")

north_course_upper <- course_upper %>%
  filter(division == "north")
```

For division == "south"

```
review_south = south_course_upper$Review.Text
TextDoc <- Corpus(VectorSource(review_south))
#Replacing "/", "@" and "|" with space
toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
TextDoc <- tm_map(TextDoc, toSpace, "/")
```

```
## Warning in tm_map.SimpleCorpus(TextDoc, toSpace, "/"): transformation drops
```

The scale for sentiment scores using the syuzhet method is decimal and ranges from -1(indicating most negative) to +1(indicating most positive). Note that the summary statistics of the suyzhet vector of south and north show a median value of 3.15 and 3.85, which are above zero and can be interpreted as the overall average sentiment across all the responses are positive.

Which variables are most significant affecting overall rating

Method 1

```
## Ordinal regression
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
## remove rows with no grade
data <- data[!is.na(data$Grade) & data$Grade != "", ]
data$Overall_factor <- factor(data$Overall)
## keep grades into whole categories
grade_ranges <- list(A = c("A+", "A", "A-"),
                     B = c("B+", "B", "B-"),
                     C = c("C+", "C", "C-"),
                     D = c("D+", "D", "D-"))
data$Standing <- data$Grade
for (category in names(grade_ranges)) {
  grade_range <- grade_ranges[[category]]
  data$Standing[data$Grade %in% grade_range] <- category
}
ordinal_reg <- polr(Overall_factor ~ Standing + Easiness + Clarity + Workload + Helpfulness + division,
summary(ordinal_reg)
```

```
##
## Re-fitting to get Hessian

## Call:
## polr(formula = Overall_factor ~ Standing + Easiness + Clarity +
##     Workload + Helpfulness + division, data = data)
##
## Coefficients:
##                 Value Std. Error t value
## StandingB      0.1091    0.05165  2.1121
## StandingC     -0.3834    0.11012 -3.4812
## StandingD     -0.5371    1.00928 -0.5322
## StandingDR    -0.9943    0.36386 -2.7325
## StandingF     -0.3209    0.18958 -1.6927
```

```
## StandingI     -0.9735    0.14071 -6.9181
## StandingN/A    0.2253    0.04534  4.9689
## StandingNR    -0.5129    0.12317 -4.1642
## StandingP     -0.1329    0.07559 -1.7580
## StandingS      0.7638    0.46071  1.6578
## StandingU     -1.4662    0.17117 -8.5655
## Easiness       0.5455    0.03257 16.7489
## Clarity        2.6548    0.04115 64.5107
## Workload       0.2387    0.03448  6.9243
## Helpfulness    2.4477    0.04155 58.9022
## divisionsouth -0.2213    0.03192 -6.9345
##
## Intercepts:
##         Value    Std. Error t value
## 1|1.2     7.0521   0.1619    43.5620
## 1.2|1.3   7.6320   0.1504    50.7523
## 1.3|1.4   8.0085   0.1429    56.0515
## 1.4|1.5   8.7984   0.1337    65.7877
## 1.5|1.6   9.3075   0.1304    71.3585
## 1.6|1.7   9.5486   0.1290    73.9944
## 1.7|1.8   9.9996   0.1277    78.3179
## 1.8|1.9  10.2202   0.1273    80.2771
## 1.9|2    10.4269   0.1270    82.0882
## 2|2.1    11.2497   0.1259    89.3724
## 2.1|2.2  11.5027   0.1256    91.5813
## 2.2|2.3  11.9328   0.1255    95.0740
## 2.3|2.4  13.0191   0.1264   103.0292
## 2.4|2.5  13.5983   0.1278   106.3857
## 2.5|2.6  14.0774   0.1294   108.8131
## 2.6|2.7  14.4618   0.1309   110.4995
## 2.7|2.8  15.1034   0.1340   112.7377
## 2.8|2.9  15.6903   0.1374   114.1686
## 2.9|3    16.1237   0.1401   115.0523
## 3|3.1    17.1679   0.1469   116.9031
## 3.1|3.2  17.7782   0.1505   118.1301
## 3.2|3.3  18.2446   0.1533   119.0196
## 3.3|3.4  18.8503   0.1567   120.2593
## 3.4|3.5  19.2014   0.1585   121.1099
## 3.5|3.6  19.7111   0.1611   122.3495
## 3.6|3.7  20.3257   0.1643   123.7033
## 3.7|3.8  20.9407   0.1678   124.7998
## 3.8|3.9  21.7827   0.1733   125.6695
## 3.9|4    22.0835   0.1755   125.8173
## 4|4.1    22.9708   0.1821   126.1538
## 4.1|4.2  23.3248   0.1845   126.4221
## 4.2|4.3  23.9773   0.1887   127.0704
## 4.3|4.4  24.5363   0.1919   127.8623
## 4.4|4.5  25.0664   0.1947   128.7389
## 4.5|4.6  25.8738   0.1992   129.9045
## 4.6|4.7  26.4845   0.2026   130.7238
## 4.7|4.8  27.1077   0.2062   131.4509
## 4.8|4.9  27.7713   0.2101   132.1967
## 4.9|5    28.1940   0.2123   132.8312
##
```

```
## Residual Deviance: 74681.98
## AIC: 74791.98
```

```
## calculate the Z and P-value
coeffs <- coef(summary(ordinal_reg))
```

```
##
## Re-fitting to get Hessian
```

```
p <- pnorm(abs(coeffs[, "t value"]), lower.tail = FALSE) * 2
cbind(coeffs, "p value" = round(p,3))
```

```
##                    Value Std. Error     t value p value
## StandingB      0.1090854 0.05164787   2.1120996   0.035
## StandingC     -0.3833627 0.11012353  -3.4812063   0.000
## StandingD     -0.5371044 1.00928478  -0.5321634   0.595
## StandingDR    -0.9942517 0.36386360  -2.7324847   0.006
## StandingF     -0.3208868 0.18957545  -1.6926601   0.091
## StandingI     -0.9734506 0.14071008  -6.9181299   0.000
## StandingN/A    0.2253098 0.04534376   4.9689255   0.000
## StandingNR    -0.5129170 0.12317309  -4.1641967   0.000
## StandingP     -0.1328912 0.07559218  -1.7580025   0.079
## StandingS      0.7637772 0.46070798   1.6578337   0.097
## StandingU     -1.4661787 0.17117338  -8.5654596   0.000
## Easiness       0.5454854 0.03256844  16.7488952   0.000
## Clarity        2.6548224 0.04115321  64.5107024   0.000
## Workload       0.2387385 0.03447839   6.9242934   0.000
## Helpfulness    2.4476735 0.04155488  58.9021893   0.000
## divisionsouth -0.2213296 0.03191701  -6.9345342   0.000
## 1|1.2          7.0521225 0.16188702  43.5620020   0.000
## 1.2|1.3        7.6320291 0.15037813  50.7522532   0.000
## 1.3|1.4        8.0084909 0.14287748  56.0514578   0.000
## 1.4|1.5        8.7984361 0.13373984  65.7876966   0.000
## 1.5|1.6        9.3074939 0.13043292  71.3584743   0.000
## 1.6|1.7        9.5486383 0.12904547  73.9943696   0.000
## 1.7|1.8        9.9995561 0.12767908  78.3178880   0.000
## 1.8|1.9       10.2202435 0.12731201  80.2771329   0.000
## 1.9|2         10.4269060 0.12702069  82.0882474   0.000
## 2|2.1         11.2496728 0.12587410  89.3724170   0.000
## 2.1|2.2       11.5026913 0.12560089  91.5812855   0.000
## 2.2|2.3       11.9328237 0.12551094  95.0739760   0.000
## 2.3|2.4       13.0191022 0.12636325 103.0291850   0.000
## 2.4|2.5       13.5983218 0.12782094 106.3857099   0.000
## 2.5|2.6       14.0774362 0.12937263 108.8130938   0.000
## 2.6|2.7       14.4618297 0.13087693 110.4994602   0.000
## 2.7|2.8       15.1034333 0.13396972 112.7376650   0.000
## 2.8|2.9       15.6902536 0.13743051 114.1686343   0.000
## 2.9|3         16.1236807 0.14014225 115.0522500   0.000
## 3|3.1         17.1679423 0.14685620 116.9030799   0.000
## 3.1|3.2       17.7781642 0.15049649 118.1300908   0.000
## 3.2|3.3       18.2446461 0.15329111 119.0195933   0.000
## 3.3|3.4       18.8503348 0.15674746 120.2592703   0.000
## 3.4|3.5       19.2013593 0.15854492 121.1098993   0.000
```

```
## 3.5|3.6       19.7111412 0.16110525 122.3494670   0.000
## 3.6|3.7       20.3257041 0.16431012 123.7033018   0.000
## 3.7|3.8       20.9407001 0.16779435 124.7997949   0.000
## 3.8|3.9       21.7827379 0.17333350 125.6695241   0.000
## 3.9|4         22.0834879 0.17552034 125.8172549   0.000
## 4|4.1         22.9708141 0.18208579 126.1538006   0.000
## 4.1|4.2       23.3248235 0.18449962 126.4220690   0.000
## 4.2|4.3       23.9773443 0.18869333 127.0704418   0.000
## 4.3|4.4       24.5362919 0.19189615 127.8623477   0.000
## 4.4|4.5       25.0663627 0.19470701 128.7388821   0.000
## 4.5|4.6       25.8737719 0.19917539 129.9044625   0.000
## 4.6|4.7       26.4845301 0.20259907 130.7238455   0.000
## 4.7|4.8       27.1077186 0.20621936 131.4508919   0.000
## 4.8|4.9       27.7712748 0.21007536 132.1967273   0.000
## 4.9|5         28.1939901 0.21225425 132.8312176   0.000
```

```
## Proportional odds ratios
exp(coef(ordinal_reg))
```

```
##     StandingB     StandingC     StandingD    StandingDR     StandingF
##     1.1152576     0.6815656     0.5844381     0.3700002     0.7255054
##     StandingI   StandingN/A    StandingNR     StandingP     StandingS
##     0.3777772     1.2527107     0.5987465     0.8755603     2.1463682
##     StandingU      Easiness       Clarity      Workload   Helpfulness
##     0.2308058     1.7254458    14.2224597     1.2696464    11.5614173
## divisionsouth
##     0.8014525
```

Clarity and Helpfulness are two most significant variables affecting overall rating.

Method 2

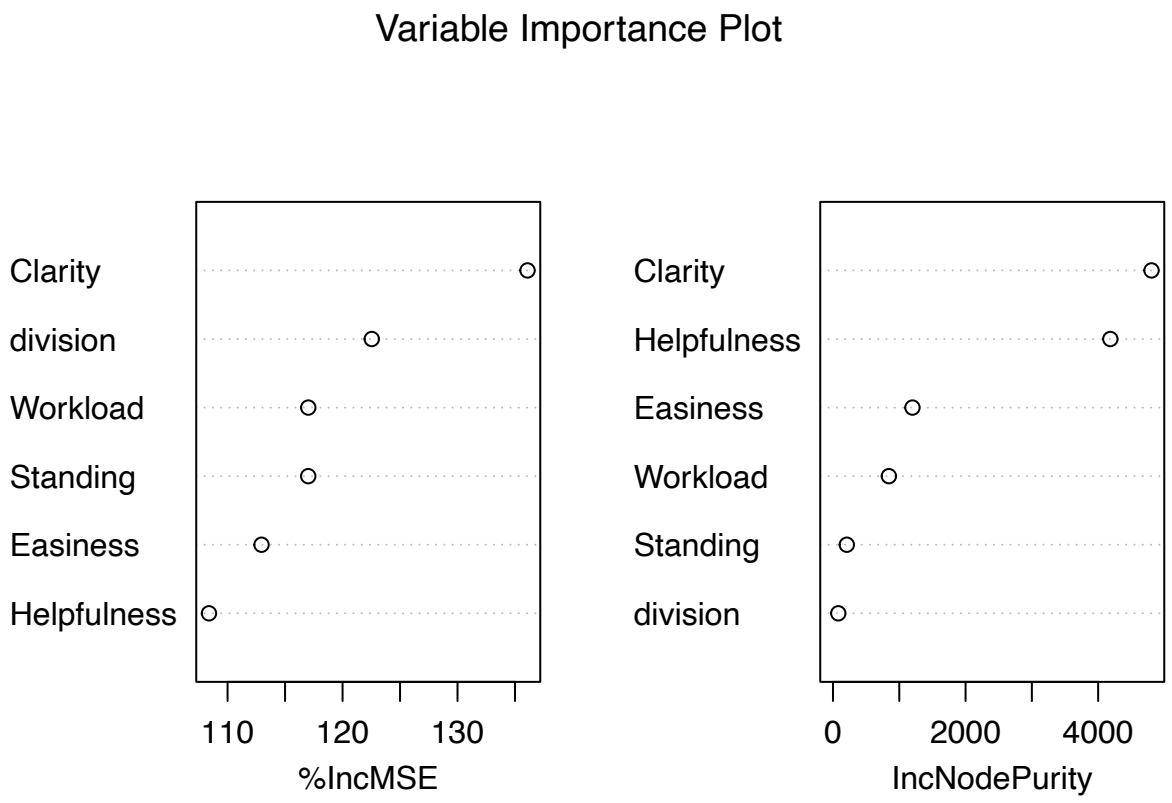
```
library(randomForest)
```

```
## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

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

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
#View(data_complete)
data_complete <- na.omit(data[, c("Overall_factor", "Standing", "Easiness", "Clarity", "Workload", "Help
data_complete <- data_complete[!is.na(data_complete$Overall_factor) & data_complete$Overall_factor != "
dependent_var <- data_complete$Overall
independent_vars <- data_complete[, c("Standing", "Easiness", "Clarity", "Workload", "Helpfulness", "di

rf1 <- randomForest(x = independent_vars, y = dependent_var, ntree = 1000, importance = TRUE)
print(importance(rf1))
```

```
##              %IncMSE IncNodePurity
## Standing    117.0153     208.41791
## Easiness    112.9514    1198.80067
## Clarity     136.0880    4804.74507
## Workload    117.0198     843.17816
## Helpfulness 108.3818    4183.62698
## division    122.5397      80.25874
```

```r
varImpPlot(rf1, main = "Variable Importance Plot")
```



Question2

```r
## Set higher than 4.0, the Overall is Good
#data$Overall = as.numeric(data$Overall)
data$Overall_category <- ifelse(data$Overall > 4.0, "Good", "Bad")
## choose the helpfulness and clarity are lower than 2.0
```

```r
selected_rows <- data[data$Helpfulness <= 2.0 | data$Clarity <= 2.0, ]
#View(selected_rows)
data$Overall
```

```
##   [1] 5.0 5.0 5.0 5.0 5.0 2.3 2.3 2.3 2.3 2.3 2.3 2.3 5.0 5.0 5.0 5.0 4.8 4.8
##  [19] 4.8 4.8 4.8 5.0 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2 3.7 3.7 3.7 3.7
##  [37] 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7
##  [55] 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7
##  [73] 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 4.2 4.2 4.2 4.2 4.2 4.2
##  [91] 4.2 4.2 4.5 4.5 3.8 3.8 3.8 3.8 3.8 3.8 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7
## [109] 3.7 3.7 3.7 3.7 3.7 3.7 3.7 2.0 2.0 2.0 2.0 2.0 1.0 3.0 3.0 1.0 3.6 3.6
## [127] 3.6 3.6 3.6 3.6 3.6 3.6 3.6 3.6 3.6 3.6 3.6 3.6 3.6 3.6 3.6 3.6 3.6 3.6
## [145] 3.6 3.6 3.6 3.6 3.6 3.6 3.6 3.6 3.6 3.6 3.6 3.6 3.6 3.6 3.6 3.6 3.6 3.6
## [163] 3.6 3.6 3.6 3.6 3.6 3.6 3.6 3.6 3.6 3.6 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0
## [181] 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 4.5 4.5 4.6 4.6 4.6 4.6 4.6 4.6 4.6
## [199] 4.6 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1
## [217] 4.1 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2
## [235] 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2
## [253] 4.2 4.2 4.2 4.2 4.2 4.0 4.0 4.0 4.0 4.0 4.0 4.3 4.3 4.3 4.0 4.0 4.0 5.0
## [271] 5.0 2.0 3.8 3.8 3.8 3.8 3.8 3.8 3.8 3.8 3.8 3.8 3.8 4.0 4.0 4.0 3.4 3.4
## [289] 3.4 3.4 3.4 3.4 3.4 3.4 3.4 3.4 3.4 3.5 3.5 3.3 3.3 3.3 3.3 3.3 3.3 3.3
## [307] 3.4 3.4 3.4 3.4 3.4 3.4 3.4 3.4 3.4 3.4 3.4 3.4 3.4 3.4 3.4 2.3 2.3 2.3 2.3
## [325] 2.3 2.3 2.3 2.3 2.3 2.3 3.8 3.8 3.8 3.8 3.8 3.8 3.8 3.8 3.8 3.8 3.8 3.8
## [343] 3.8 3.4 3.4 3.4 3.4 3.4 3.4 3.4 3.4 3.4 3.4 3.4 3.2 3.2 3.2 3.2 3.2 3.2 3.2
## [361] 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2 5.0 3.1 3.1 3.1 3.1 3.1 3.1 3.1 3.1 3.1
## [379] 3.0 3.0 3.0 3.0 3.0 3.0 3.0 2.7 2.7 2.7 2.7 2.7 2.7 2.7 2.7 2.7 2.7 2.5 2.5
## [397] 2.5 2.5 1.0 1.0 2.4 2.4 2.4 2.4 2.4 2.4 2.4 2.4 2.4 2.4 2.4 2.4 2.4 2.4 4.5
## [415] 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0
## [433] 5.0 3.7 3.7 3.7 5.0 5.0 4.7 4.7 4.7 4.7 4.7 4.7 5.0 3.9 3.9 3.9 3.9 3.9
## [451] 3.9 3.9 3.9 3.9 4.0 3.8 3.8 3.8 3.8 3.8 5.0 1.0 3.2 3.2 3.2 3.2 3.2 3.2
## [469] 3.2 3.2 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0
## [487] 5.0 5.0 5.0 4.8 4.8 4.8 4.8 4.8 4.8 5.0 4.8 4.8 4.8 4.8 4.8 5.0 5.0 5.0
## [505] 5.0 5.0 4.4 4.4 4.4 4.4 4.4 4.4 4.4 4.4 4.4 4.4 4.4 4.4 4.4 4.4 4.4 4.4 4.4
## [523] 4.4 4.4 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5
## [541] 4.5 4.5 4.5 4.5 4.6 4.6 4.6 4.6 4.6 3.3 3.3 3.3 3.3 3.3 3.3 3.3 3.3 3.3 3.3
## [559] 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.2 4.1 4.1 4.1 4.1 4.1 4.1
## [577] 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.8 4.8 4.8 4.8 4.8
## [595] 4.8 4.8 4.8 4.8 4.8 4.8 4.8 4.8 4.8 4.8 4.8 4.8 4.8 4.8 4.0 3.5 3.5 3.5
## [613] 3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.8 3.8 3.8 3.8
## [631] 3.8 3.8 3.8 3.8 3.8 4.7 4.7 4.7 3.3 3.3 3.3 3.0 3.0 2.9 2.9 2.9 2.9 2.9
## [649] 2.9 2.9 2.9 2.9 2.9 2.9 2.9 2.9 2.9 2.9 2.9 2.9 2.9 2.9 2.9 2.9 2.9 2.9
## [667] 2.9 2.9 2.9 2.9 2.9 2.9 2.9 2.9 2.9 2.9 2.9 2.9 2.9 2.9 2.9 2.9 2.5 2.5
## [685] 2.5 2.5 2.5 2.5 2.8 2.8 2.8 2.8 2.8 2.9 2.9 2.9 2.9 2.9 2.9 2.9 2.9 2.9
## [703] 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 3.0 2.0 2.0 5.0 5.0 3.8 3.8 3.8 3.8
## [721] 3.8 3.8 3.8 3.8 3.8 3.8 3.8 3.8 3.8 3.8 3.8 3.8 3.8 3.8 3.8 3.8 5.0 5.0
## [739] 1.0 4.4 4.4 4.4 4.4 4.4 4.4 4.4 4.4 4.4 4.4 4.4 4.9 4.9 4.9 4.9 4.9 4.9 4.9 4.5
## [757] 4.5 4.0 4.0 3.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0
## [775] 5.0 5.0 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 5.0 5.0 5.0 5.0 5.0 5.0 5.0
## [793] 5.0 4.2 4.2 4.2 4.2 4.2 4.2 3.3 3.3 3.3 3.3 3.3 3.3 3.3 3.3 3.3 3.3 3.5 3.5
## [811] 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 3.3 3.3 3.3 3.3 3.3 3.3 3.3 4.1 4.1 4.1
## [829] 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1
## [847] 4.1 4.1 4.1 2.0 5.0 3.0 3.0 3.0 1.0 2.0 4.0 5.0 1.0 1.0 1.0 1.0 1.0 5.0
## [865] 4.4 4.4 4.4 4.4 4.4 4.4 4.4 4.4 4.4 4.4 2.0 4.0 4.0 4.0 2.8 2.8 2.8 2.8 2.8
```

```
## [15463] 4.0 5.0 2.0 4.6 4.6 4.6 4.6 4.6 4.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0
## [15481] 5.0 5.0 5.0 5.0 4.9 4.9 4.9 4.9 4.9 4.9 4.9 4.9 4.9 4.9 4.9 4.9 3.0 4.4
## [15499] 4.4 4.4 4.4 4.4 4.4 4.4 4.4 4.4 4.4 3.0 3.0 3.0 3.0 3.7 3.7 3.7 1.0 3.8 3.8
## [15517] 3.8 3.8 5.0 5.0 2.5 2.5 1.0 1.0 1.0 4.0 3.0 3.0 3.5 3.5 5.0 2.0 4.0 5.0
## [15535] 5.0 4.8 4.8 4.8 4.8 4.8 3.7 3.7 3.7 4.9 4.9 4.9 4.9 4.9 4.9 4.9 4.9 5.0
## [15553] 5.0 4.0 4.0 4.0 2.0 3.8 3.8 3.8 3.8 3.8 3.8 5.0 1.0 1.0 5.0 5.0 5.0 5.0
## [15571] 5.0 5.0 5.0 1.0 4.7 4.7 4.7 1.0 3.5 3.5 3.5 3.5 3.7 3.7 3.7 4.2 4.2 4.2
## [15589] 4.2 4.0 4.0 4.0 4.0 4.0 5.0 5.0 3.5 3.5 3.7 3.7 3.7 5.0 4.3 4.3 4.3 3.0
## [15607] 1.0 3.0 3.0 3.0 3.0 2.5 2.5 2.5 2.5 5.0 4.8 4.8 4.8 4.8 5.0 5.0 1.0 5.0
## [15625] 4.5 4.5 4.5 4.5 4.8 4.8 4.8 4.8 4.8 4.8 4.8 4.8 4.8 4.8 4.8 4.8 4.8 4.8
## [15643] 4.8 4.8 4.8 4.8 4.8 5.0 5.0 5.0 5.0 4.7 4.7 4.7 4.7 4.7 4.7 4.7 4.7 4.7
## [15661] 4.7 4.7 4.7 4.7 4.7 4.7 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 4.8 4.8 4.8 4.8
## [15679] 4.8 4.3 4.3 4.3 4.3 4.3 4.3 4.3 4.3 4.3 4.3 4.3 4.3 4.3 4.3 4.3 4.3 4.3
## [15697] 4.3 4.3 4.3 4.3 4.3 4.3 4.3 4.3 4.3 4.3 4.3 4.3 4.3 4.3 4.3 4.3 4.3 3.3
## [15715] 3.3 3.3 3.3 3.3 3.3 3.3 3.3 3.3 3.3 3.3 3.3 3.3 3.3 3.3 3.3 3.3 3.3 3.3
## [15733] 3.3 3.3 3.3 3.3 3.3 3.3 3.3 3.3 3.3 3.3 3.0 4.5 4.5 4.5 4.5 4.5 4.5 4.5
## [15751] 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5 3.5
## [15769] 3.5 3.5 3.5 3.5 3.5 4.0 4.0 4.0 4.0 4.0 4.0 1.7 1.7 1.7 1.5 1.5 3.0 3.0
## [15787] 3.0 3.0 3.0 4.0 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 4.6 4.6 4.6 4.6 4.6
## [15805] 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6
## [15823] 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 5.0 3.0 5.0 3.7 3.7 3.7
## [15841] 5.0 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 5.0 5.0 3.0 3.0 2.0 2.0 5.0 5.0
## [15859] 5.0 5.0 5.0 5.0 5.0 5.0
```

```
## use the logistic regression
library(stats)
library(sjPlot)
# Recode "Good" as 1 and "Bad" as 0
selected_rows$Overall_category <- ifelse(selected_rows$Overall_category == "Good", 1, 0)
#class(selected_rows$Overall_category)
## logistic regression
lg <- glm(Overall_category ~ Standing + Easiness + Clarity + Workload + Helpfulness + division, data = s
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(lg)
```

```
##
## Call:
## glm(formula = Overall_category ~ Standing + Easiness + Clarity +
##     Workload + Helpfulness + division, family = binomial, data = selected_rows)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.21801  -0.02595  -0.00001   0.00000   2.98452
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -19.5270     7.9122  -2.468  0.01359 *
## StandingB       -16.3545  4842.2195  -0.003  0.99731
## StandingC       -14.8778  9147.6096  -0.002  0.99870
## StandingDR      -13.4733 16206.0966  -0.001  0.99934
## StandingF       -13.7320 56188.1226   0.000  0.99981
```

```
## StandingI       -12.9821 32032.0166   0.000  0.99968
## StandingN/A     -16.5609  3800.5430  -0.004  0.99652
## StandingNR      -14.2042 13586.9598  -0.001  0.99917
## StandingP       -18.9046 10501.6520  -0.002  0.99856
## StandingS       -10.8232 38930.1356   0.000  0.99978
## StandingU        -8.8197 79462.0051   0.000  0.99991
## Easiness          1.2051     0.6903   1.746  0.08083 .
## Clarity           0.4621     2.7280   0.169  0.86550
## Workload          0.5633     0.6864   0.821  0.41179
## Helpfulness       2.5501     0.7905   3.226  0.00126 **
## divisionsouth     1.1950     1.5282   0.782  0.43424
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 73.816  on 1038  degrees of freedom
## Residual deviance: 24.575  on 1023  degrees of freedom
## AIC: 56.575
##
## Number of Fisher Scoring iterations: 22
```

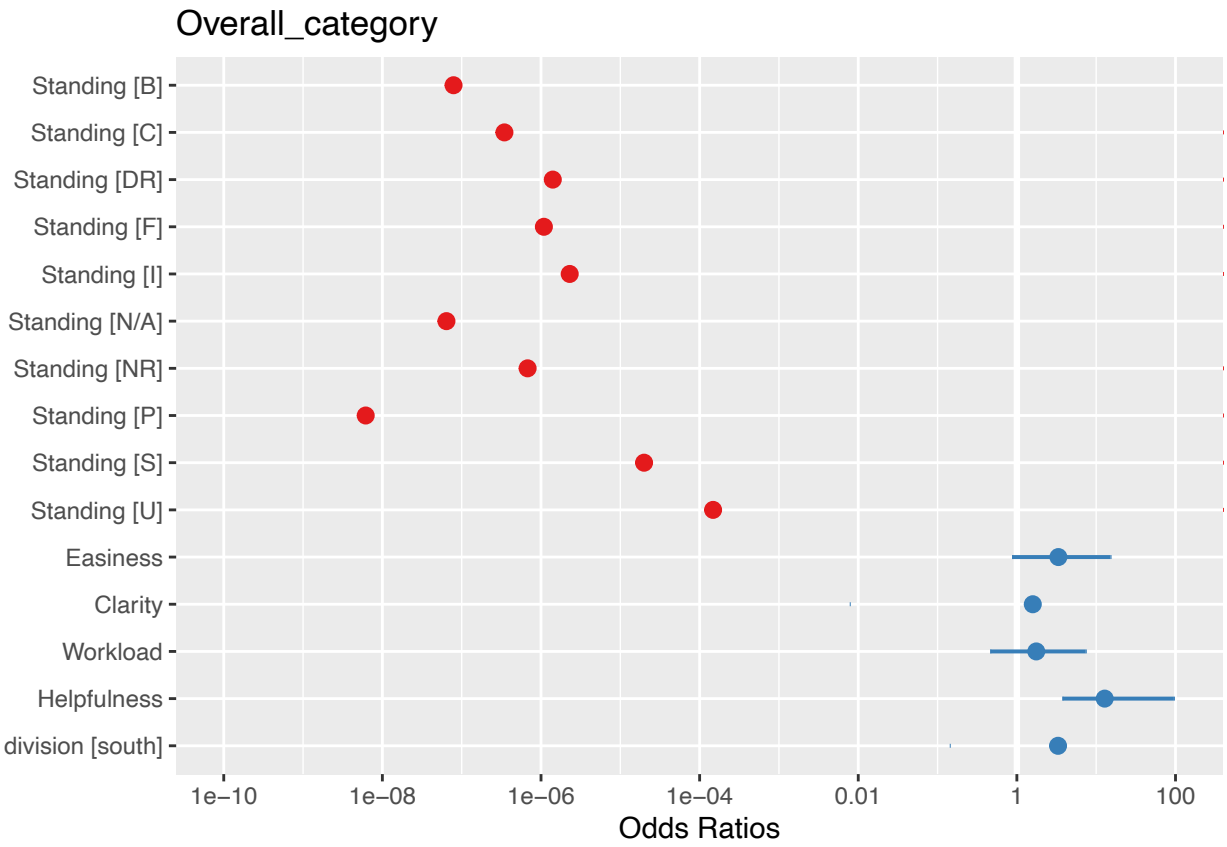
```
odds_ratio = exp(coef(lg))
print(odds_ratio)
```

```
##   (Intercept)     StandingB     StandingC    StandingDR     StandingF
##  3.307695e-09  7.894519e-08  3.456601e-07  1.408015e-06  1.087076e-06
##     StandingI   StandingN/A    StandingNR     StandingP     StandingS
##  2.301074e-06  6.422525e-08  6.779448e-07  6.163376e-09  1.993166e-05
##     StandingU      Easiness       Clarity      Workload   Helpfulness
##  1.477892e-04  3.337162e+00  1.587376e+00  1.756503e+00  1.280893e+01
## divisionsouth
##  3.303507e+00
```

```
plot_model(lg)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```



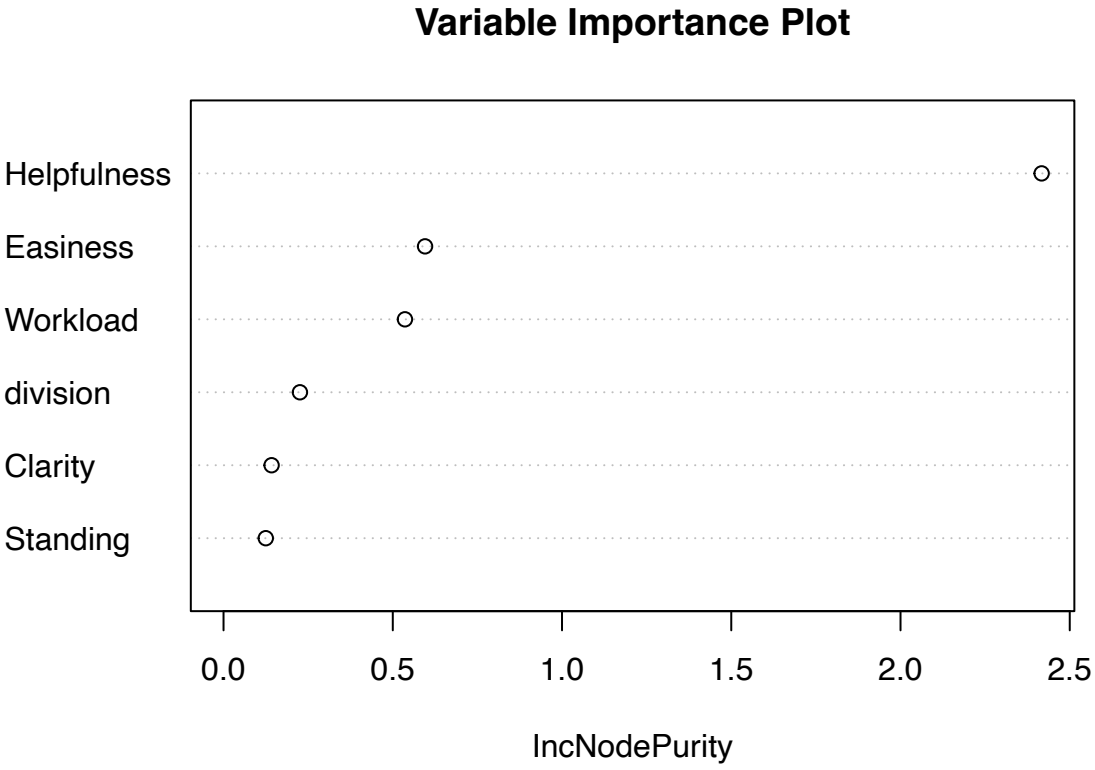

```
rf_model <- randomForest(Overall_category ~ Standing + Easiness + Clarity + Workload + Helpfulness + di
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values.  Are you sure you want to do regression?
```

```
print(importance(rf_model))
```

```
##             IncNodePurity
## Standing        0.1247880
## Easiness        0.5953282
## Clarity         0.1419522
## Workload        0.5359244
## Helpfulness     2.4169149
## division        0.2257458
```

```
varImpPlot(rf_model, main = "Variable Importance Plot")
```


Variable Importance Plot
Helpfulness
Easiness
Workload
division
Clarity
Standing
0.0
0.5
1.0
1.5
2.0
2.5
IncNodePurity