EDA With Grade Averages Dataset

Suraj Rajan (UID: 206066871)

2024-06-07

```
grade_avg <- read.csv("grade_cleaned3_with_averages.csv")
```

```
#head(grade_avg)

dim(grade_avg)
```

```
## [1] 12306    28
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
##
## 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
```

```
grade_avg_filt <- grade_avg %>%
   filter(!is.na(Overall_Rating) & !is.na(Easiness) & !is.na(Clarity) & !is.na(Workload) & !is.na(Helpfu
            !is.na(A) & !is.na(A.Minus) & !is.na(B.Plus) & !is.na(B) & !is.na(B.Minus) & !is.na(C.Plus) &

dim(grade_avg_filt)
```

```
## [1] 7861   28
```

```
summary(grade_avg_filt)
```

```
##   Professor          Class_Code            Dept            Dept.Type
##  Length:7861        Length:7861        Length:7861        Length:7861
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
```

```
##
##
##   Class_Name        Overall_Rating     Easiness        Clarity
##  Length:7861        Min.   :1.000   Min.   :1.000   Min.   :1.000
##  Class :character   1st Qu.:3.000   1st Qu.:2.100   1st Qu.:2.800
##  Mode  :character   Median :3.700   Median :2.800   Median :3.700
##                     Mean   :3.637   Mean   :2.791   Mean   :3.525
##                     3rd Qu.:4.300   3rd Qu.:3.400   3rd Qu.:4.300
##                     Max.   :5.000   Max.   :5.000   Max.   :5.000
##     Workload      Helpfulness        A.Plus              A
##  Min.   :1.000   Min.   :1.000   Min.   :0.00000   Min.   :0.000
##  1st Qu.:2.300   1st Qu.:3.000   1st Qu.:0.00700   1st Qu.:0.151
##  Median :3.000   Median :3.800   Median :0.04400   Median :0.233
##  Mean   :2.907   Mean   :3.685   Mean   :0.08415   Mean   :0.287
##  3rd Qu.:3.500   3rd Qu.:4.500   3rd Qu.:0.10300   3rd Qu.:0.370
##  Max.   :5.000   Max.   :5.000   Max.   :1.00000   Max.   :1.000
##     A.Minus           B.Plus             B             B.Minus
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.0850   1st Qu.:0.0690   1st Qu.:0.0750   1st Qu.:0.01900
##  Median :0.1320   Median :0.1120   Median :0.1250   Median :0.06100
##  Mean   :0.1441   Mean   :0.1176   Mean   :0.1309   Mean   :0.06646
##  3rd Qu.:0.1900   3rd Qu.:0.1580   3rd Qu.:0.1790   3rd Qu.:0.10300
##  Max.   :0.8000   Max.   :0.5790   Max.   :0.6350   Max.   :0.50000
##      C.Plus              C              C.Minus            D.Plus
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.000000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.000000
##  Median :0.03800   Median :0.04100   Median :0.01100   Median :0.000000
##  Mean   :0.04801   Mean   :0.05635   Mean   :0.02635   Mean   :0.006514
##  3rd Qu.:0.07900   3rd Qu.:0.08700   3rd Qu.:0.04300   3rd Qu.:0.005000
##  Max.   :0.33300   Max.   :0.45500   Max.   :0.37900   Max.   :0.200000
##        D              D.Minus               F           Grade.Quarter
##  Min.   :0.00000   Min.   :0.000000   Min.   :0.00000   Length:7861
##  1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:0.00000   Class :character
##  Median :0.00000   Median :0.000000   Median :0.00500   Mode  :character
##  Mean   :0.01281   Mean   :0.003303   Mean   :0.01646
##  3rd Qu.:0.01900   3rd Qu.:0.000000   3rd Qu.:0.02300
##  Max.   :0.21100   Max.   :0.160000   Max.   :0.35700
##     Review          Review.Quarter     Reviewer.Grade      Average_GPA
##  Length:7861        Length:7861        Length:7861        Min.   :1.403
##  Class :character   Class :character   Class :character   1st Qu.:2.956
##  Mode  :character   Mode  :character   Mode  :character   Median :3.268
##                                                           Mean   :3.270
##                                                           3rd Qu.:3.585
##                                                           Max.   :4.004
```

```
table(grade_avg_filt$Dept.Type)
```

```
##
## non-stem     stem
##     3822     4039
```

```
correlation_data <- grade_avg_filt %>%
  select(Overall_Rating, Easiness, A.Plus, A, A.Minus, B.Plus, B, B.Minus, C.Plus, C, C.Minus, D.Plus, D
```

```r
cor_matrix <- cor(correlation_data, use = "complete.obs", method = "pearson")
print(cor_matrix)
```

```
##                Overall_Rating    Easiness      A.Plus           A     A.Minus
## Overall_Rating     1.00000000  0.51371599  0.11971367  0.22872833  0.01241743
## Easiness           0.51371599  1.00000000  0.21908846  0.31226044  0.03616113
## A.Plus             0.11971367  0.21908846  1.00000000  0.01359137 -0.15418707
## A                  0.22872833  0.31226044  0.01359137  1.00000000 -0.12034749
## A.Minus            0.01241743  0.03616113 -0.15418707 -0.12034749  1.00000000
## B.Plus            -0.07161368 -0.11661164 -0.25935763 -0.39688471  0.27191961
## B                 -0.11213243 -0.22481244 -0.36614862 -0.43457591 -0.15705280
## B.Minus           -0.19739113 -0.27415955 -0.28138869 -0.55911925 -0.06466001
## C.Plus            -0.17111102 -0.24851725 -0.25836001 -0.53823186 -0.18757003
## C                 -0.15658445 -0.25141814 -0.27825608 -0.42422678 -0.35306593
## C.Minus           -0.16654056 -0.23418978 -0.21447715 -0.42245768 -0.22904750
## D.Plus            -0.08513194 -0.09141107 -0.10360813 -0.26901194 -0.10937862
## D                 -0.15290213 -0.21118524 -0.16482312 -0.29621992 -0.24671168
## D.Minus           -0.06852443 -0.08315496 -0.07834846 -0.18109094 -0.09997361
## F                 -0.18059821 -0.17679880 -0.10924053 -0.24292573 -0.17798109
##                      B.Plus           B     B.Minus     C.Plus           C
## Overall_Rating -0.071613677 -0.112132425 -0.19739113 -0.1711110 -0.1565844
## Easiness       -0.116611640 -0.224812438 -0.27415955 -0.2485172 -0.2514181
## A.Plus         -0.259357626 -0.366148616 -0.28138869 -0.2583600 -0.2782561
## A              -0.396884714 -0.434575908 -0.55911925 -0.5382319 -0.4242268
## A.Minus         0.271919607 -0.157052798 -0.06466001 -0.1875700 -0.3530659
## B.Plus          1.000000000  0.111205974  0.19110526  0.1112250 -0.1325086
## B               0.111205974  1.000000000  0.21897207  0.1834215  0.3405001
## B.Minus         0.191105256  0.218972067  1.00000000  0.4499019  0.2581376
## C.Plus          0.111225004  0.183421504  0.44990188  1.0000000  0.3843672
## C              -0.132508577  0.340500116  0.25813757  0.3843672  1.0000000
## C.Minus        -0.051465749  0.109718204  0.31963813  0.4439774  0.4170065
## D.Plus         -0.004881909  0.017366004  0.22012106  0.2818937  0.1768616
## D              -0.111671486  0.131560346  0.16786799  0.2636784  0.4071986
## D.Minus        -0.040955066 -0.001203377  0.13160336  0.1518197  0.1391590
## F              -0.073561093  0.047766611  0.10916306  0.1630782  0.2496862
##                    C.Minus       D.Plus          D      D.Minus           F
## Overall_Rating -0.16654056 -0.085131942 -0.1529021 -0.068524430 -0.18059821
## Easiness       -0.23418978 -0.091411075 -0.2111852 -0.083154960 -0.17679880
## A.Plus         -0.21447715 -0.103608128 -0.1648231 -0.078348460 -0.10924053
## A              -0.42245768 -0.269011937 -0.2962199 -0.181090937 -0.24292573
## A.Minus        -0.22904750 -0.109378616 -0.2467117 -0.099973611 -0.17798109
## B.Plus         -0.05146575 -0.004881909 -0.1116715 -0.040955066 -0.07356109
## B               0.10971820  0.017366004  0.1315603 -0.001203377  0.04776661
## B.Minus         0.31963813  0.220121064  0.1678680  0.131603361  0.10916306
## C.Plus          0.44397739  0.281893730  0.2636784  0.151819696  0.16307821
## C               0.41700650  0.176861558  0.4071986  0.139159014  0.24968615
## C.Minus         1.00000000  0.284708894  0.3746495  0.240208021  0.26498628
## D.Plus          0.28470889  1.000000000  0.1945437  0.301614492  0.14531416
## D               0.37464947  0.194543656  1.0000000  0.214395607  0.31956902
## D.Minus         0.24020802  0.301614492  0.2143956  1.000000000  0.18684918
## F               0.26498628  0.145314162  0.3195690  0.186849175  1.00000000
```

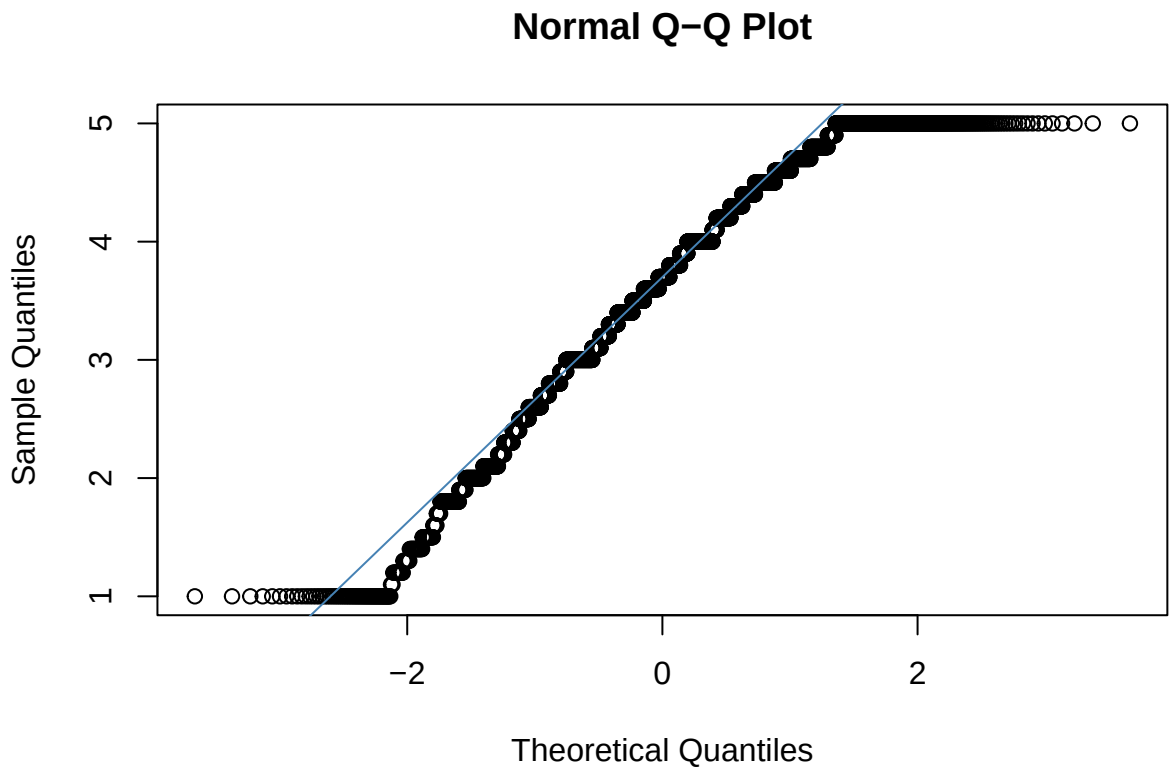
```
shapiro.test(grade_avg_filt$Overall_Rating[grade_avg_filt$Dept.Type == "stem"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  grade_avg_filt$Overall_Rating[grade_avg_filt$Dept.Type == "stem"]
## W = 0.95858, p-value < 2.2e-16
```

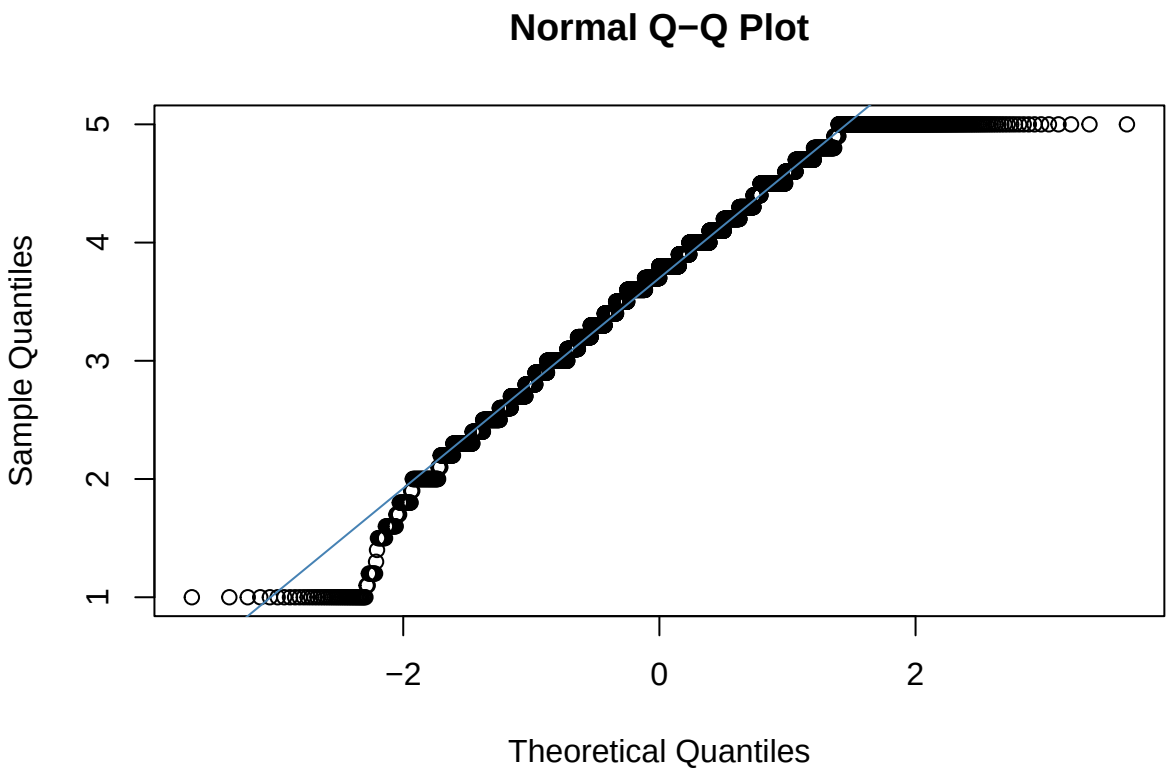
```
shapiro.test(grade_avg_filt$Overall_Rating[grade_avg_filt$Dept.Type == "non-stem"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  grade_avg_filt$Overall_Rating[grade_avg_filt$Dept.Type == "non-stem"]
## W = 0.96903, p-value < 2.2e-16
```

```
qqnorm(grade_avg_filt$Overall_Rating[grade_avg_filt$Dept.Type == "stem"])
qqline(grade_avg_filt$Overall_Rating[grade_avg_filt$Dept.Type == "stem"], col = "steelblue")
```




```
qqnorm(grade_avg_filt$Overall_Rating[grade_avg_filt$Dept.Type == "non-stem"])
qqline(grade_avg_filt$Overall_Rating[grade_avg_filt$Dept.Type == "non-stem"], col = "steelblue")
```




```
stem_ratings <- grade_avg_filt %>%
  filter(Dept.Type == "stem") %>%
  pull(Overall_Rating)

non_stem_ratings <- grade_avg_filt %>%
  filter(Dept.Type == "non-stem") %>%
  pull(Overall_Rating)

test_result <- wilcox.test(stem_ratings, non_stem_ratings, alternative = "two.sided")


print(test_result)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  stem_ratings and non_stem_ratings
## W = 7508170, p-value = 0.03628
## alternative hypothesis: true location shift is not equal to 0
```

```
summary(grade_avg_filt)
```

```
##    Professor          Class_Code            Dept            Dept.Type
##  Length:7861        Length:7861        Length:7861        Length:7861
```

```
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##   Class_Name        Overall_Rating     Easiness        Clarity
##  Length:7861        Min.   :1.000   Min.   :1.000   Min.   :1.000
##  Class :character   1st Qu.:3.000   1st Qu.:2.100   1st Qu.:2.800
##  Mode  :character   Median :3.700   Median :2.800   Median :3.700
##                     Mean   :3.637   Mean   :2.791   Mean   :3.525
##                     3rd Qu.:4.300   3rd Qu.:3.400   3rd Qu.:4.300
##                     Max.   :5.000   Max.   :5.000   Max.   :5.000
##     Workload      Helpfulness        A.Plus              A
##  Min.   :1.000   Min.   :1.000   Min.   :0.00000   Min.   :0.000
##  1st Qu.:2.300   1st Qu.:3.000   1st Qu.:0.00700   1st Qu.:0.151
##  Median :3.000   Median :3.800   Median :0.04400   Median :0.233
##  Mean   :2.907   Mean   :3.685   Mean   :0.08415   Mean   :0.287
##  3rd Qu.:3.500   3rd Qu.:4.500   3rd Qu.:0.10300   3rd Qu.:0.370
##  Max.   :5.000   Max.   :5.000   Max.   :1.00000   Max.   :1.000
##     A.Minus           B.Plus             B            B.Minus
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.0850   1st Qu.:0.0690   1st Qu.:0.0750   1st Qu.:0.01900
##  Median :0.1320   Median :0.1120   Median :0.1250   Median :0.06100
##  Mean   :0.1441   Mean   :0.1176   Mean   :0.1309   Mean   :0.06646
##  3rd Qu.:0.1900   3rd Qu.:0.1580   3rd Qu.:0.1790   3rd Qu.:0.10300
##  Max.   :0.8000   Max.   :0.5790   Max.   :0.6350   Max.   :0.50000
##      C.Plus              C              C.Minus           D.Plus
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.000000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.000000
##  Median :0.03800   Median :0.04100   Median :0.01100   Median :0.000000
##  Mean   :0.04801   Mean   :0.05635   Mean   :0.02635   Mean   :0.006514
##  3rd Qu.:0.07900   3rd Qu.:0.08700   3rd Qu.:0.04300   3rd Qu.:0.005000
##  Max.   :0.33300   Max.   :0.45500   Max.   :0.37900   Max.   :0.200000
##        D              D.Minus              F           Grade.Quarter
##  Min.   :0.00000   Min.   :0.000000   Min.   :0.00000   Length:7861
##  1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:0.00000   Class :character
##  Median :0.00000   Median :0.000000   Median :0.00500   Mode  :character
##  Mean   :0.01281   Mean   :0.003303   Mean   :0.01646
##  3rd Qu.:0.01900   3rd Qu.:0.000000   3rd Qu.:0.02300
##  Max.   :0.21100   Max.   :0.160000   Max.   :0.35700
##     Review          Review.Quarter     Reviewer.Grade      Average_GPA
##  Length:7861        Length:7861        Length:7861        Min.   :1.403
##  Class :character   Class :character   Class :character   1st Qu.:2.956
##  Mode  :character   Mode  :character   Mode  :character   Median :3.268
##                                                           Mean   :3.270
##                                                           3rd Qu.:3.585
##                                                           Max.   :4.004
```

-Departments, Visualize

```
write.csv(grade_avg_filt, "grade_cleaned_with_averages", row.names = FALSE)
```

```r
table(grade_avg_filt$Dept)
```

```
##
##    CHEM    COMM    ECON    MATH PHYSICS POL SCI    PSYCH    STATS
##    1058     787     727    1528     913     863     1445     540
```

```r
#tail(grade_avg_filt)
```

```r
prof <- grade_avg_filt %>%
  filter(Professor == "Guani Wu") %>%
  mutate(Avg_overall = mean(Average_GPA))

#prof
```

```r
average_grade_distribution <- grade_avg_filt %>%
  group_by(Professor, Class_Name) %>%
  summarise(
    Average_A_Plus = mean(A.Plus),
    Average_A = mean(A),
    Average_A_Minus = mean(A.Minus),
    Average_B_Plus = mean(B.Plus),
    Average_B = mean(B),
    Average_B_Minus = mean(B.Minus),
    Average_C_Plus = mean(C.Plus),
    Average_C = mean(C),
    Average_C_Minus = mean(C.Minus),
    Average_D_Plus = mean(D.Plus),
    Average_D = mean(D),
    Average_D_Minus = mean(D.Minus),
    Average_F = mean(F, na.rm = TRUE),
    .groups = 'drop'
  ) %>%
  mutate(Total_Average = Average_A_Plus + Average_A + Average_A_Minus +
           Average_B_Plus + Average_B + Average_B_Minus +
           Average_C_Plus + Average_C + Average_C_Minus +
           Average_D_Plus + Average_D + Average_D_Minus +
           Average_F) %>%
  mutate(
    GPA_A_Plus = Average_A_Plus * 4.0 * 100,
    GPA_A = Average_A * 4.0 * 100,
    GPA_A_Minus = Average_A_Minus * 3.7 * 100,
    GPA_B_Plus = Average_B_Plus * 3.3 * 100,
    GPA_B = Average_B * 3.0 * 100,
    GPA_B_Minus = Average_B_Minus * 2.7 * 100,
    GPA_C_Plus = Average_C_Plus * 2.3 * 100,
    GPA_C = Average_C * 2.0 * 100,
    GPA_C_Minus = Average_C_Minus * 1.7 * 100,
    GPA_D_Plus = Average_D_Plus * 1.3 * 100,
    GPA_D = Average_D * 1.0 * 100,
    GPA_D_Minus = Average_D_Minus * 0.7 * 100,
    GPA_F = Average_F * 0.0 * 100,
    Avg_GPA = (GPA_A_Plus + GPA_A + GPA_A_Minus +
```

```
                  GPA_B_Plus + GPA_B + GPA_B_Minus +
                  GPA_C_Plus + GPA_C + GPA_C_Minus +
                  GPA_D_Plus + GPA_D + GPA_D_Minus +
                  GPA_F) / 100
  )



#head(average_grade_distribution)
dim(average_grade_distribution)
```

```
## [1] 2366   30
```

```
enhanced_data <- grade_avg_filt %>%
  left_join(average_grade_distribution, by = c("Professor", "Class_Name")) %>%
  distinct(Professor, Class_Code, Dept, Dept.Type, Class_Name, Overall_Rating, Easiness, Clarity, Workl
  select(
    Professor, Class_Code,Class_Name, Dept, Dept.Type,
    Overall_Rating, Easiness, Clarity, Workload, Helpfulness,
    Average_A_Plus, Average_A, Average_A_Minus,
    Average_B_Plus, Average_B, Average_B_Minus,
    Average_C_Plus, Average_C, Average_C_Minus,
    Average_D_Plus, Average_D, Average_D_Minus, Average_F,
    Avg_GPA,
    Review, Review.Quarter, Reviewer.Grade
  )


#head(enhanced_data)

dim(enhanced_data)
```

```
## [1] 2474   27
```

```
table(enhanced_data$Dept)
```

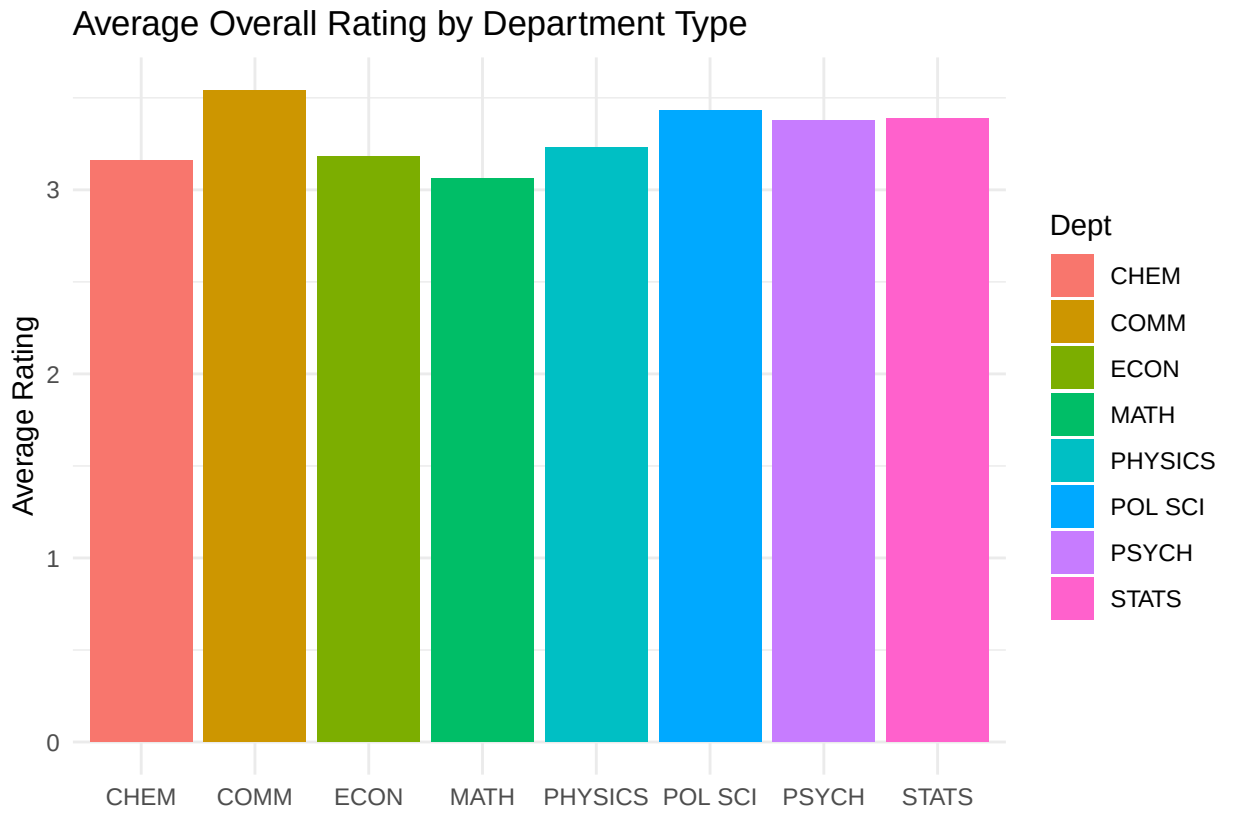
```
##
##    CHEM    COMM    ECON    MATH PHYSICS POL SCI   PSYCH   STATS
##     260     132     187     790     348     293     327     137
```

```
colnames(enhanced_data)
```

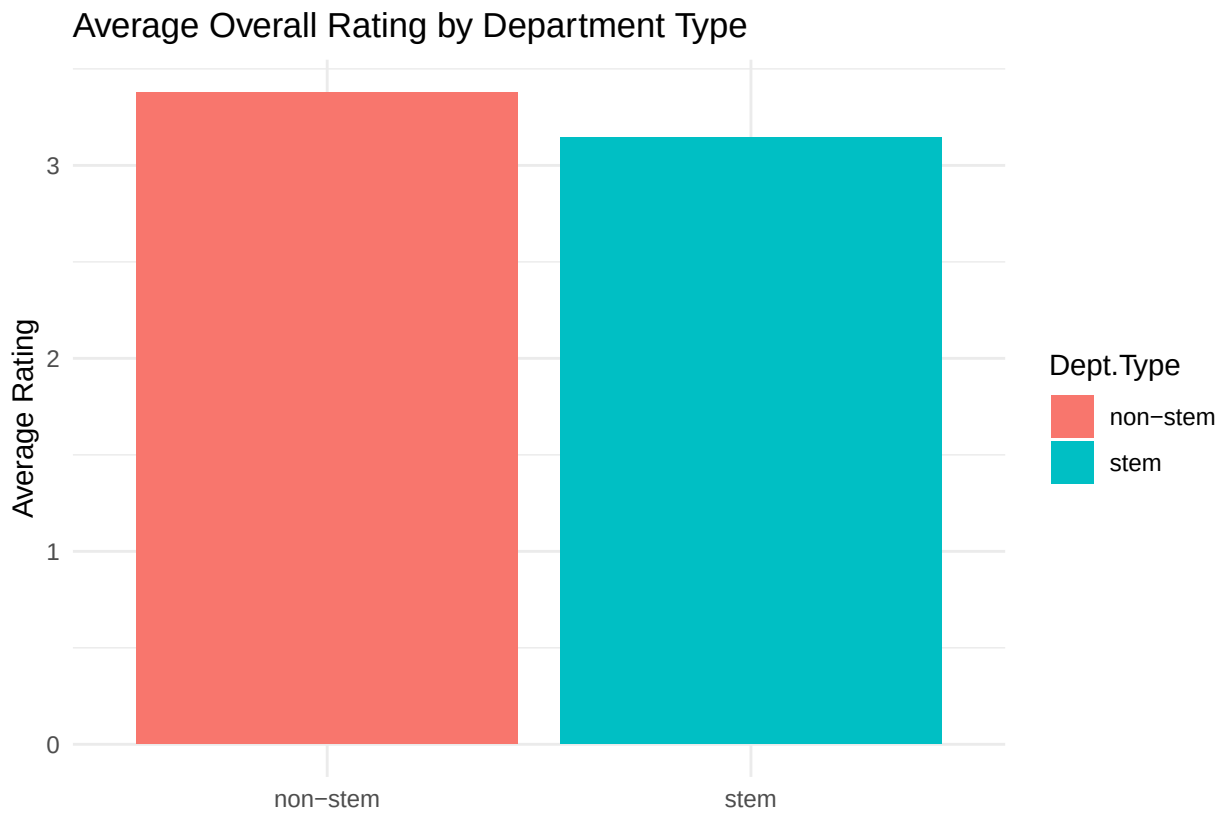
```
##  [1] "Professor"       "Class_Code"      "Class_Name"      "Dept"
##  [5] "Dept.Type"       "Overall_Rating"  "Easiness"        "Clarity"
##  [9] "Workload"        "Helpfulness"     "Average_A_Plus"  "Average_A"
## [13] "Average_A_Minus" "Average_B_Plus"  "Average_B"       "Average_B_Minus"
## [17] "Average_C_Plus"  "Average_C"       "Average_C_Minus" "Average_D_Plus"
## [21] "Average_D"       "Average_D_Minus" "Average_F"       "Avg_GPA"
## [25] "Review"          "Review.Quarter"  "Reviewer.Grade"
```

```
library(ggplot2)

ggplot(enhanced_data, aes(x = Dept, y = Avg_GPA, fill = Dept)) +
  geom_bar(stat = "summary", fun = "mean", position = "dodge") +
  labs(title = "Average Overall Rating by Department Type", y = "Average Rating", x = "") +
  theme_minimal()
```




```
ggplot(enhanced_data, aes(x = Dept.Type, y = Avg_GPA, fill = Dept.Type)) +
  geom_bar(stat = "summary", fun = "mean", position = "dodge") +
  labs(title = "Average Overall Rating by Department Type", y = "Average Rating", x = "") +
  theme_minimal()
```




```
gpa_rating_correlation <- cor(enhanced_data$Avg_GPA, enhanced_data$Easiness)

gpa_rating_correlation
```

```
## [1] 0.4002502
```

```
table(grade_avg_filt$Dept)
```

```
##
##    CHEM    COMM    ECON    MATH PHYSICS POL SCI    PSYCH    STATS
##    1058     787     727    1528     913     863    1445     540
```

```
table(enhanced_data$Dept)
```

```
##
##    CHEM    COMM    ECON    MATH PHYSICS POL SCI    PSYCH    STATS
##     260     132     187     790     348     293     327     137
```

```
#tail(enhanced_data)
```

```
stats_summary <- enhanced_data %>%
  group_by(Dept.Type) %>%
  summarise(
```

```
    Average_Rating = mean(Overall_Rating, na.rm = TRUE),
    Median_Rating = median(Overall_Rating, na.rm = TRUE),
    Average_Easiness = mean(Easiness, na.rm = TRUE),
    Average_Clarity = mean(Clarity, na.rm = TRUE),
    Average_Workload = mean(Workload, na.rm = TRUE),
    Average_Helpfulness = mean(Helpfulness, na.rm = TRUE)
  )


print(stats_summary)
```

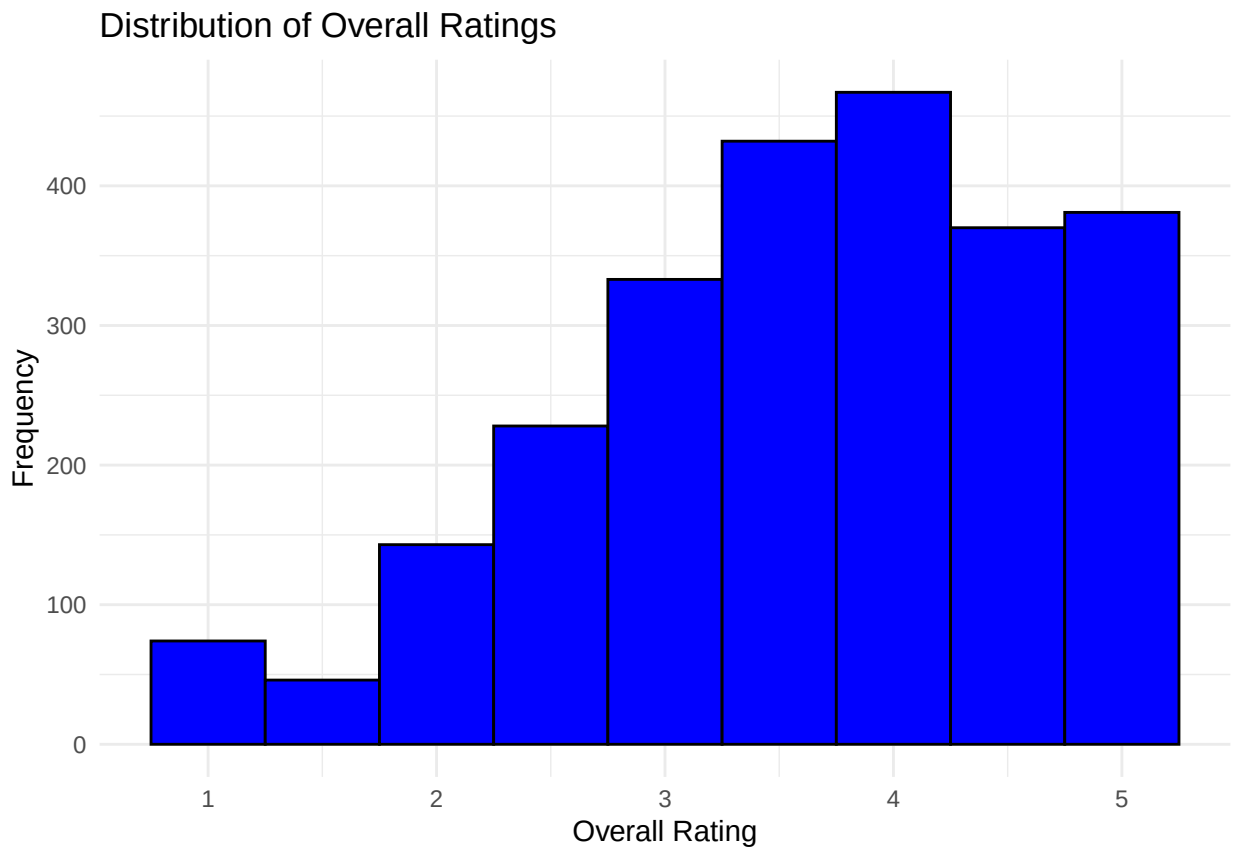
```
## # A tibble: 2 x 7
##   Dept.Type Average_Rating Median_Rating Average_Easiness Average_Clarity
##   <chr>              <dbl>         <dbl>            <dbl>           <dbl>
## 1 non-stem            3.63           3.7             2.82            3.53
## 2 stem                3.61           3.7             2.67            3.48
## # i 2 more variables: Average_Workload <dbl>, Average_Helpfulness <dbl>
```

```
table(enhanced_data$Dept.Type)
```

```
##
## non-stem     stem
##      939     1535
```

```
write.csv(enhanced_data, "grade_aggregated_data.csv", row.names = FALSE)
```

```
ggplot(enhanced_data, aes(x = Overall_Rating)) +
  geom_histogram(binwidth = 0.5, fill = "blue", color = "black") +
  labs(title = "Distribution of Overall Ratings", x = "Overall Rating", y = "Frequency") + theme_minimal
```




```
enhanced_data$Dept.Type <- factor(enhanced_data$Dept.Type)

levels(enhanced_data$Dept.Type)
```

```
## [1] "non-stem" "stem"
```

```
enhanced_data$Dept.Type <- relevel(enhanced_data$Dept.Type, ref = "non-stem")

levels(enhanced_data$Dept.Type)
```

```
## [1] "non-stem" "stem"
```

```
model.full <- glm(Dept.Type ~ Avg_GPA + Overall_Rating + Easiness + Helpfulness + Workload + Clarity, da

summary(model.full)
```

```
##
## Call:
## glm(formula = Dept.Type ~ Avg_GPA + Overall_Rating + Easiness +
##     Helpfulness + Workload + Clarity, family = binomial, data = enhanced_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2073  -1.1552   0.6661   0.9334   1.8450
```

```
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     6.02033    0.41746  14.421  < 2e-16 ***
## Avg_GPA        -1.98681    0.13621 -14.587  < 2e-16 ***
## Overall_Rating  0.10682    0.11170   0.956   0.3389
## Easiness       -0.11034    0.07730  -1.428   0.1534
## Helpfulness     0.45713    0.09947   4.596 4.31e-06 ***
## Workload        0.23161    0.07337   3.157   0.0016 **
## Clarity        -0.42092    0.09891  -4.256 2.08e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3284.7  on 2473  degrees of freedom
## Residual deviance: 2999.9  on 2467  degrees of freedom
## AIC: 3013.9
##
## Number of Fisher Scoring iterations: 4
```

Relevant variables: avg_gpa, helpfulness, workload, clarity

Notes On Model and Relevant Variables:

`Avg_GPA`: Each unit increase in average GPA decreases the likelihood of a department being categorized as stem, with statistical significance. The negative coefficient means that higher GPA is associated with non-stem departments.

`Helpfulness`: For each one-unit increase in Helpfulness, the likelihood of the dept being non-stem increases with statistical significance.

`Workload`: Significant and positive, departments with higher workload are more likely to be non-stem.

`Clarity`: Significant and negative, higher clarity in course presentation is associated with stem departments.

```
model.reduced <- glm(Dept.Type ~ Avg_GPA + Helpfulness + Workload + Clarity, data = enhanced_data, fami

summary(model.reduced)
```

```
##
## Call:
## glm(formula = Dept.Type ~ Avg_GPA + Helpfulness + Workload +
##     Clarity, family = binomial, data = enhanced_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1989  -1.1569   0.6648   0.9323   1.8274
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  6.11613    0.41109  14.878  < 2e-16 ***
## Avg_GPA     -2.01336    0.13308 -15.129  < 2e-16 ***
## Helpfulness  0.49766    0.08745   5.691 1.26e-08 ***
## Workload     0.16177    0.05065   3.194   0.0014 **
## Clarity     -0.38427    0.08499  -4.521 6.15e-06 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3284.7  on 2473  degrees of freedom
## Residual deviance: 3002.6  on 2469  degrees of freedom
## AIC: 3012.6
##
## Number of Fisher Scoring iterations: 4
```

We see that the reduced model seems to show improved statistical significance for the predictors.

```
anova(model.reduced, model.full, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: Dept.Type ~ Avg_GPA + Helpfulness + Workload + Clarity
## Model 2: Dept.Type ~ Avg_GPA + Overall_Rating + Easiness + Helpfulness +
##     Workload + Clarity
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      2469     3002.6
## 2      2467     2999.9  2   2.6338    0.268
```

The above tests whether the additional predictors in the full model significantly improve the fit of the model compared to that of the reduced model.

P-Value indicates that the decrease in deviance is not statistically significant. So, adding overall_rating and easiness does not significantly improve the model's ability to predict department type.

```
exp(coef(model.reduced))
```

```
## (Intercept)     Avg_GPA Helpfulness    Workload     Clarity
##  453.106864    0.133539    1.644871    1.175595    0.680946
```

The above exponentiated coefficients for the reduced model make interpretation easier.

(Intercept) = 453.106864

This extremely high odds ratio indicates that when all predictor variables are at zero, the odds of being in the "stem" category (relative to "non-stem") are very high. This can indicate specific baseline characteristics inherent to "stem" departments when no other influences are present.

Avg_GPA = 0.133539

An odds ratio less than 1 (approximately 0.13) for GPA now means that with every unit increase in GPA, the odds of the department being "stem" (as opposed to "non-stem") decrease significantly. Essentially, higher GPAs are strongly associated with remaining in or being classified as "non-stem".

Helpfulness = 1.644871

With an odds ratio above 1, a unit increase in helpfulness scores increases the likelihood of a department being "stem" by about 64.5%. This suggests that departments with higher helpfulness ratings are more likely to be "stem" compared to "non-stem".

Workload = 1.175595

This odds ratio indicates that a unit increase in workload results in a 17.6% increase in the likelihood of a department being "stem". This suggests that higher workloads are slightly more common in "stem" departments than in "non-stem" departments.

Clarity = 0.680946

An odds ratio below 1 implies that higher clarity decreases the odds of being "stem" by about 31.9%. Higher clarity is thus more associated with "non-stem" departments, indicating that clearer communication or expectations are more characteristic of "non-stem" departments.

Before we use our model to make predictions uisng a train test split, let's validate common assumptions with logistic regression.

Correlations of Indep variables

```
independent_vars <- enhanced_data[, c("Avg_GPA", "Helpfulness", "Workload", "Clarity")]


cor_matrix <- cor(independent_vars, use = "complete.obs")  # 'complete.obs' handles missing values by e

print(cor_matrix)
```

```
##                Avg_GPA Helpfulness  Workload   Clarity
## Avg_GPA     1.0000000   0.2369918 0.3273699 0.2128985
## Helpfulness 0.2369918   1.0000000 0.4152930 0.8601354
## Workload    0.3273699   0.4152930 1.0000000 0.3999387
## Clarity     0.2128985   0.8601354 0.3999387 1.0000000
```

Helpfulness and Clarity look to be very correlated.

```
library(car)
```

```
## Loading required package: carData

##
## Attaching package: 'car'

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

```
model_interaction <- glm(Dept.Type ~ Avg_GPA + Helpfulness + Workload + Clarity + Helpfulness * Clarity
                         data = enhanced_data,
                         family = binomial())

summary(model_interaction)
```

```
##
## Call:
## glm(formula = Dept.Type ~ Avg_GPA + Helpfulness + Workload +
```

```
##     Clarity + Helpfulness * Clarity, family = binomial(), data = enhanced_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2499  -1.1587   0.6553   0.9265   1.8918
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          7.73877    0.61196  12.646  < 2e-16 ***
## Avg_GPA             -2.08060    0.13513 -15.397  < 2e-16 ***
## Helpfulness          0.09216    0.13989   0.659 0.509998
## Workload             0.14039    0.05115   2.745 0.006058 **
## Clarity             -0.89057    0.16174  -5.506 3.67e-08 ***
## Helpfulness:Clarity  0.13937    0.03776   3.691 0.000223 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3284.7  on 2473  degrees of freedom
## Residual deviance: 2988.7  on 2468  degrees of freedom
## AIC: 3000.7
##
## Number of Fisher Scoring iterations: 4
```

```
vif_val <- vif(model_interaction)
```

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

```
print(vif_val)
```

```
##             Avg_GPA         Helpfulness            Workload             Clarity
##            1.207516           10.591057            1.342195           14.584367
## Helpfulness:Clarity
##           33.502276
```

```
library(car)
```

```
library(car)
```

```
model.interaction <- glm(formula = Dept.Type ~ Avg_GPA + Workload +
    Helpfulness:Clarity, family = binomial, data = enhanced_data)

# Calculate VIF
vif_values <- vif(model.interaction, type = 'terms')
```

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

```
print(vif_values)
```

```
##             Avg_GPA            Workload Helpfulness:Clarity
##            1.172166            1.339206            1.269514
```

```
summary(model.interaction)
```

```
##
## Call:
## glm(formula = Dept.Type ~ Avg_GPA + Workload + Helpfulness:Clarity,
##     family = binomial, data = enhanced_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1721  -1.1706   0.6885   0.9347   1.7350
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           6.245556   0.404130  15.454  < 2e-16 ***
## Avg_GPA              -1.978427   0.132087 -14.978  < 2e-16 ***
## Workload              0.153622   0.050506   3.042  0.00235 **
## Helpfulness:Clarity   0.018469   0.007231   2.554  0.01065 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3284.7  on 2473  degrees of freedom
## Residual deviance: 3029.8  on 2470  degrees of freedom
## AIC: 3037.8
##
## Number of Fisher Scoring iterations: 4
```

```
model.helpfulness <-  glm(formula = Dept.Type ~ Avg_GPA + Workload +
   Helpfulness, family = binomial, data = enhanced_data)

vif_values <- vif(model.helpfulness)
print(vif_values)
```

```
##     Avg_GPA    Workload Helpfulness
##    1.174618    1.311151    1.238045
```

```
summary(model.helpfulness)
```

```
##
## Call:
## glm(formula = Dept.Type ~ Avg_GPA + Workload + Helpfulness, family = binomial,
##     data = enhanced_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -2.1698  -1.1658   0.6817   0.9321   1.8203
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.97878    0.40760  14.668  < 2e-16 ***
## Avg_GPA     -1.99347    0.13231 -15.067  < 2e-16 ***
## Workload     0.13919    0.05004   2.781 0.005413 **
## Helpfulness  0.16695    0.04697   3.554 0.000379 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3284.7  on 2473  degrees of freedom
## Residual deviance: 3023.7  on 2470  degrees of freedom
## AIC: 3031.7
##
## Number of Fisher Scoring iterations: 4
```

```
summary(model.reduced)
```

```
##
## Call:
## glm(formula = Dept.Type ~ Avg_GPA + Helpfulness + Workload +
##     Clarity, family = binomial, data = enhanced_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1989  -1.1569   0.6648   0.9323   1.8274
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  6.11613    0.41109  14.878  < 2e-16 ***
## Avg_GPA     -2.01336    0.13308 -15.129  < 2e-16 ***
## Helpfulness  0.49766    0.08745   5.691 1.26e-08 ***
## Workload     0.16177    0.05065   3.194   0.0014 **
## Clarity     -0.38427    0.08499  -4.521 6.15e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3284.7  on 2473  degrees of freedom
## Residual deviance: 3002.6  on 2469  degrees of freedom
## AIC: 3012.6
##
## Number of Fisher Scoring iterations: 4
```

```
vif.reduced <- vif(model.reduced)

print(vif.reduced)
```

```
##     Avg_GPA Helpfulness    Workload     Clarity
##    1.180365    4.267084    1.329793    4.146905
```

Model Validation Through Cross Validation

```
library(boot)
```

```
##
## Attaching package: 'boot'
```

```
## The following object is masked from 'package:car':
##
##     logit
```

```
cv_help <- cv.glm(data = enhanced_data, glmfit = model.full, K = 10)

print(cv_help$delta)
```

```
## [1] 0.2110513 0.2109746
```

```
cv_reduced <- cv.glm(data = enhanced_data, glmfit = model.reduced, K = 10)

print(cv_reduced$delta)
```

```
## [1] 0.2108667 0.2108131
```

Confusion Matrix Testing

```
library(ROSE)
```

```
## Loaded ROSE 0.0-4
```

```
balanced_data <- ovun.sample(Dept.Type ~ ., data = enhanced_data, method = "over", N = 2000)$data

table(balanced_data$Dept.Type)
```

```
##
##     stem non-stem
##      974     1026
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'lattice'
```

```
## The following object is masked from 'package:boot':
##
##     melanoma
```

```r
set.seed(123)


trainIndex <- createDataPartition(balanced_data$Dept.Type, p = 0.7, list = FALSE)
trainData <- balanced_data[trainIndex, ]
testData <- balanced_data[-trainIndex, ]

model.reduced <- glm(formula = Dept.Type ~ Avg_GPA + Helpfulness + Workload + Clarity,
                     family = binomial(), data = trainData)




probabilities <- predict(model.reduced, newdata = testData, type = "response")


predicted_classes <- ifelse(probabilities > 0.4, "stem", "non-stem")
actual_classes <- testData$Dept.Type


conf_matrix <- confusionMatrix(as.factor(predicted_classes), as.factor(actual_classes))
```

```
## Warning in confusionMatrix.default(as.factor(predicted_classes),
## as.factor(actual_classes)): Levels are not in the same order for reference and
## data. Refactoring data to match.
```

```r
print(conf_matrix$table)
```

```
##           Reference
## Prediction stem non-stem
##   stem      169      265
##   non-stem  123       42
```

```r
print(conf_matrix$overall)
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##   3.522538e-01  -2.810857e-01   3.139759e-01   3.919967e-01   5.125209e-01
## AccuracyPValue  McnemarPValue
##   1.000000e+00   8.174893e-13
```

```r
print(conf_matrix$byClass)
```

```
##          Sensitivity          Specificity       Pos Pred Value
##            0.5787671            0.1368078            0.3894009
##       Neg Pred Value            Precision               Recall
##            0.2545455            0.3894009            0.5787671
##                   F1           Prevalence       Detection Rate
##            0.4655647            0.4874791            0.2821369
## Detection Prevalence    Balanced Accuracy
##            0.7245409            0.3577875
```

```r
library(caret)


set.seed(123)


trainIndex <- createDataPartition(enhanced_data$Dept.Type, p = 0.7, list = FALSE, times = 1)
trainData <- enhanced_data[trainIndex, ]
testData <- enhanced_data[-trainIndex, ]


model.reduced2 <- glm(formula = Dept.Type ~ Avg_GPA + Clarity + Helpfulness + Workload,
                  family = binomial, data = trainData)


probabilities <- predict(model.reduced2, newdata = testData, type = "response")
predicted_classes <- ifelse(probabilities > 0.5, "stem", "non-stem")
actual_classes <- testData$Dept.Type


conf_matrix <- confusionMatrix(as.factor(predicted_classes), as.factor(actual_classes))


print(conf_matrix$table)
```

```
##           Reference
## Prediction non-stem stem
##   non-stem      107   77
##   stem          174  383
```

```r
print(conf_matrix$overall)
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##   6.612686e-01   2.287555e-01   6.259270e-01   6.953201e-01   6.207827e-01
## AccuracyPValue  McnemarPValue
##   1.231932e-02   1.365740e-09
```

```r
print(conf_matrix$byClass)
```

```
##          Sensitivity          Specificity       Pos Pred Value
##            0.3807829            0.8326087            0.5815217
##       Neg Pred Value            Precision               Recall
##            0.6876122            0.5815217            0.3807829
##                   F1           Prevalence       Detection Rate
##            0.4602151            0.3792173            0.1443995
## Detection Prevalence    Balanced Accuracy
##            0.2483131            0.6066958
```

```r
summary(model.reduced2)
```

```
##
```

```
## Call:
## glm(formula = Dept.Type ~ Avg_GPA + Clarity + Helpfulness + Workload,
##     family = binomial, data = trainData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1831  -1.1743   0.6842   0.9328   1.8451
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.73349    0.48462  11.831  < 2e-16 ***
## Avg_GPA     -1.93816    0.15768 -12.292  < 2e-16 ***
## Clarity     -0.38569    0.10350  -3.726 0.000194 ***
## Helpfulness  0.52285    0.10707   4.883 1.04e-06 ***
## Workload     0.17538    0.06081   2.884 0.003926 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2301.1  on 1732  degrees of freedom
## Residual deviance: 2113.9  on 1728  degrees of freedom
## AIC: 2123.9
##
## Number of Fisher Scoring iterations: 4
```

```
library(caret)


set.seed(123)


trainIndex <- createDataPartition(enhanced_data$Dept.Type, p = 0.7, list = FALSE, times = 1)
trainData <- enhanced_data[trainIndex, ]
testData <- enhanced_data[-trainIndex, ]


model.reduced2 <- glm(formula = Dept.Type ~ Avg_GPA + Clarity + Overall_Rating,
                  family = binomial, data = trainData)


probabilities <- predict(model.reduced2, newdata = testData, type = "response")
predicted_classes <- ifelse(probabilities > 0.503, "stem", "non-stem")
actual_classes <- testData$Dept.Type


conf_matrix <- confusionMatrix(as.factor(predicted_classes), as.factor(actual_classes))


print(conf_matrix$table)
```

```
##           Reference
## Prediction non-stem stem
##   non-stem      112   71
```

```
##   stem          169  389
```

```
print(conf_matrix$overall)
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##   6.761134e-01   2.620073e-01   6.410940e-01   7.097235e-01   6.207827e-01
## AccuracyPValue  McnemarPValue
##   9.814208e-04   3.817247e-10
```

```
print(conf_matrix$byClass)
```

```
##          Sensitivity          Specificity       Pos Pred Value
##            0.3985765            0.8456522            0.6120219
##       Neg Pred Value            Precision               Recall
##            0.6971326            0.6120219            0.3985765
##                   F1           Prevalence       Detection Rate
##            0.4827586            0.3792173            0.1511471
## Detection Prevalence    Balanced Accuracy
##            0.2469636            0.6221143
```

```
summary(model.reduced2)
```

```
##
## Call:
## glm(formula = Dept.Type ~ Avg_GPA + Clarity + Overall_Rating,
##     family = binomial, data = trainData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0408  -1.1887   0.7029   0.9415   1.6903
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)      5.7036     0.4792  11.902  < 2e-16 ***
## Avg_GPA         -1.8000     0.1507 -11.946  < 2e-16 ***
## Clarity         -0.2134     0.1078  -1.979 0.047848 *
## Overall_Rating   0.3854     0.1139   3.383 0.000717 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2301.1  on 1732  degrees of freedom
## Residual deviance: 2140.0  on 1729  degrees of freedom
## AIC: 2148
##
## Number of Fisher Scoring iterations: 4
```

```
vif_p <- vif(model.reduced2)

print(vif_p)
```

```
##         Avg_GPA         Clarity Overall_Rating
##        1.096034        4.782886        4.941841
```

```
x <- model.matrix(Dept.Type ~ Avg_GPA + Clarity + Overall_Rating  - 1, data = enhanced_data) # -1 to ex
y <- as.factor(enhanced_data$Dept.Type)
```

```
library(glmnet)
```

```
## Loading required package: Matrix
```

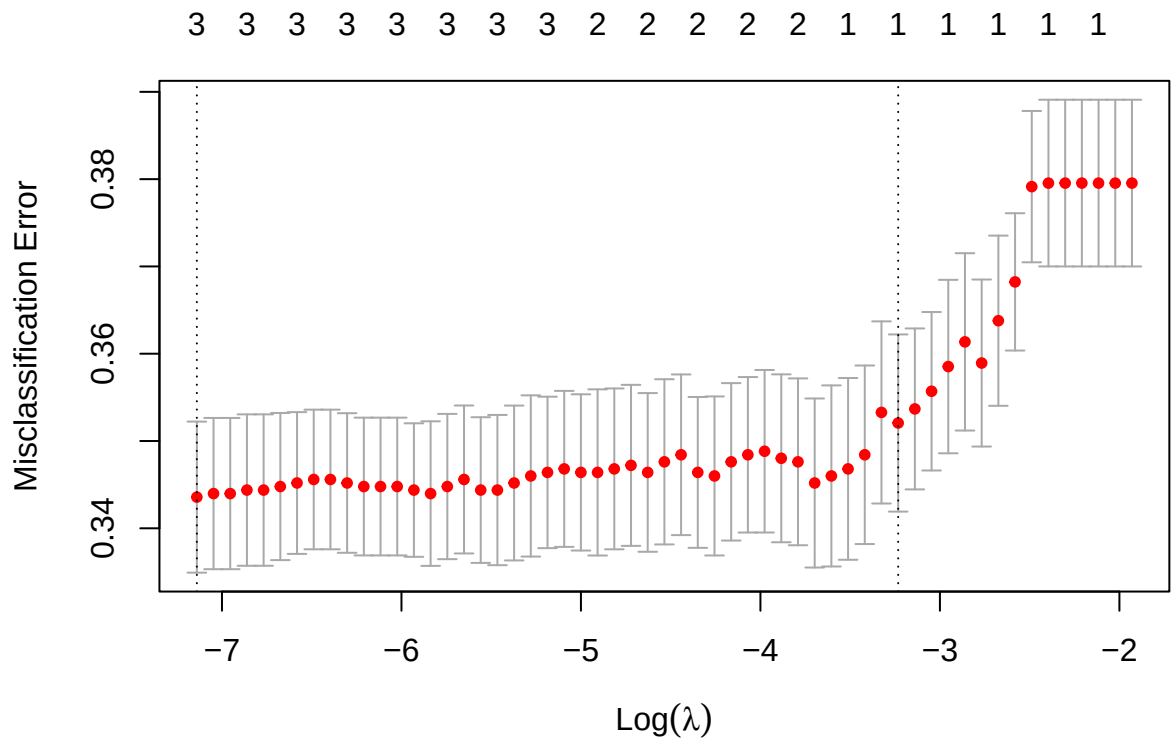
```
## Loaded glmnet 4.1-8
```

```
# Fit a logistic regression model with L1 regularization
fit <- glmnet(x, y, family = "binomial", alpha = 1)

# Fit with L2 regularization
fit_ridge <- glmnet(x, y, family = "binomial", alpha = 0)


fit_elastic <- glmnet(x, y, family = "binomial", alpha = 0.5)
```

```
cv_fit <- cv.glmnet(x, y, family = "binomial", type.measure = "class", alpha = 1)
plot(cv_fit)
```



```r
best_lambda <- cv_fit$lambda.min


final_model <- glmnet(x, y, family = "binomial", alpha = 1, lambda = best_lambda)
```

```r
set.seed(123)

enhanced_data2 <- na.omit(enhanced_data)

trainIndex <- createDataPartition(enhanced_data2$Dept.Type, p = 0.75, list = FALSE)
trainData <- enhanced_data2[trainIndex,]
testData <- enhanced_data2[-trainIndex,]


grid <- expand.grid(
  .alpha = seq(0, 1, by = 0.1),
  .lambda = 10^seq(-3, 1, by = 0.5)
)


trainControl <- trainControl(
  method = "cv",
  number = 10,
  savePredictions = "final",
  verboseIter = TRUE
)


set.seed(123)
model <- train(
  Dept.Type ~ Avg_GPA + Clarity + Overall_Rating + Helpfulness + Workload,
  data = trainData,
  method = "glmnet",
  tuneGrid = grid,
  trControl = trainControl
)
```

```
## + Fold01: alpha=0.0, lambda=10
## - Fold01: alpha=0.0, lambda=10
## + Fold01: alpha=0.1, lambda=10
## - Fold01: alpha=0.1, lambda=10
## + Fold01: alpha=0.2, lambda=10
## - Fold01: alpha=0.2, lambda=10
## + Fold01: alpha=0.3, lambda=10
## - Fold01: alpha=0.3, lambda=10
## + Fold01: alpha=0.4, lambda=10
## - Fold01: alpha=0.4, lambda=10
## + Fold01: alpha=0.5, lambda=10
## - Fold01: alpha=0.5, lambda=10
## + Fold01: alpha=0.6, lambda=10
## - Fold01: alpha=0.6, lambda=10
## + Fold01: alpha=0.7, lambda=10
## - Fold01: alpha=0.7, lambda=10
```

```
## + Fold01: alpha=0.8, lambda=10
## - Fold01: alpha=0.8, lambda=10
## + Fold01: alpha=0.9, lambda=10
## - Fold01: alpha=0.9, lambda=10
## + Fold01: alpha=1.0, lambda=10
## - Fold01: alpha=1.0, lambda=10
## + Fold02: alpha=0.0, lambda=10
## - Fold02: alpha=0.0, lambda=10
## + Fold02: alpha=0.1, lambda=10
## - Fold02: alpha=0.1, lambda=10
## + Fold02: alpha=0.2, lambda=10
## - Fold02: alpha=0.2, lambda=10
## + Fold02: alpha=0.3, lambda=10
## - Fold02: alpha=0.3, lambda=10
## + Fold02: alpha=0.4, lambda=10
## - Fold02: alpha=0.4, lambda=10
## + Fold02: alpha=0.5, lambda=10
## - Fold02: alpha=0.5, lambda=10
## + Fold02: alpha=0.6, lambda=10
## - Fold02: alpha=0.6, lambda=10
## + Fold02: alpha=0.7, lambda=10
## - Fold02: alpha=0.7, lambda=10
## + Fold02: alpha=0.8, lambda=10
## - Fold02: alpha=0.8, lambda=10
## + Fold02: alpha=0.9, lambda=10
## - Fold02: alpha=0.9, lambda=10
## + Fold02: alpha=1.0, lambda=10
## - Fold02: alpha=1.0, lambda=10
## + Fold03: alpha=0.0, lambda=10
## - Fold03: alpha=0.0, lambda=10
## + Fold03: alpha=0.1, lambda=10
## - Fold03: alpha=0.1, lambda=10
## + Fold03: alpha=0.2, lambda=10
## - Fold03: alpha=0.2, lambda=10
## + Fold03: alpha=0.3, lambda=10
## - Fold03: alpha=0.3, lambda=10
## + Fold03: alpha=0.4, lambda=10
## - Fold03: alpha=0.4, lambda=10
## + Fold03: alpha=0.5, lambda=10
## - Fold03: alpha=0.5, lambda=10
## + Fold03: alpha=0.6, lambda=10
## - Fold03: alpha=0.6, lambda=10
## + Fold03: alpha=0.7, lambda=10
## - Fold03: alpha=0.7, lambda=10
## + Fold03: alpha=0.8, lambda=10
## - Fold03: alpha=0.8, lambda=10
## + Fold03: alpha=0.9, lambda=10
## - Fold03: alpha=0.9, lambda=10
## + Fold03: alpha=1.0, lambda=10
## - Fold03: alpha=1.0, lambda=10
## + Fold04: alpha=0.0, lambda=10
## - Fold04: alpha=0.0, lambda=10
## + Fold04: alpha=0.1, lambda=10
## - Fold04: alpha=0.1, lambda=10
```

```
## + Fold04: alpha=0.2, lambda=10
## - Fold04: alpha=0.2, lambda=10
## + Fold04: alpha=0.3, lambda=10
## - Fold04: alpha=0.3, lambda=10
## + Fold04: alpha=0.4, lambda=10
## - Fold04: alpha=0.4, lambda=10
## + Fold04: alpha=0.5, lambda=10
## - Fold04: alpha=0.5, lambda=10
## + Fold04: alpha=0.6, lambda=10
## - Fold04: alpha=0.6, lambda=10
## + Fold04: alpha=0.7, lambda=10
## - Fold04: alpha=0.7, lambda=10
## + Fold04: alpha=0.8, lambda=10
## - Fold04: alpha=0.8, lambda=10
## + Fold04: alpha=0.9, lambda=10
## - Fold04: alpha=0.9, lambda=10
## + Fold04: alpha=1.0, lambda=10
## - Fold04: alpha=1.0, lambda=10
## + Fold05: alpha=0.0, lambda=10
## - Fold05: alpha=0.0, lambda=10
## + Fold05: alpha=0.1, lambda=10
## - Fold05: alpha=0.1, lambda=10
## + Fold05: alpha=0.2, lambda=10
## - Fold05: alpha=0.2, lambda=10
## + Fold05: alpha=0.3, lambda=10
## - Fold05: alpha=0.3, lambda=10
## + Fold05: alpha=0.4, lambda=10
## - Fold05: alpha=0.4, lambda=10
## + Fold05: alpha=0.5, lambda=10
## - Fold05: alpha=0.5, lambda=10
## + Fold05: alpha=0.6, lambda=10
## - Fold05: alpha=0.6, lambda=10
## + Fold05: alpha=0.7, lambda=10
## - Fold05: alpha=0.7, lambda=10
## + Fold05: alpha=0.8, lambda=10
## - Fold05: alpha=0.8, lambda=10
## + Fold05: alpha=0.9, lambda=10
## - Fold05: alpha=0.9, lambda=10
## + Fold05: alpha=1.0, lambda=10
## - Fold05: alpha=1.0, lambda=10
## + Fold06: alpha=0.0, lambda=10
## - Fold06: alpha=0.0, lambda=10
## + Fold06: alpha=0.1, lambda=10
## - Fold06: alpha=0.1, lambda=10
## + Fold06: alpha=0.2, lambda=10
## - Fold06: alpha=0.2, lambda=10
## + Fold06: alpha=0.3, lambda=10
## - Fold06: alpha=0.3, lambda=10
## + Fold06: alpha=0.4, lambda=10
## - Fold06: alpha=0.4, lambda=10
## + Fold06: alpha=0.5, lambda=10
## - Fold06: alpha=0.5, lambda=10
## + Fold06: alpha=0.6, lambda=10
## - Fold06: alpha=0.6, lambda=10
```

```
## + Fold06: alpha=0.7, lambda=10
## - Fold06: alpha=0.7, lambda=10
## + Fold06: alpha=0.8, lambda=10
## - Fold06: alpha=0.8, lambda=10
## + Fold06: alpha=0.9, lambda=10
## - Fold06: alpha=0.9, lambda=10
## + Fold06: alpha=1.0, lambda=10
## - Fold06: alpha=1.0, lambda=10
## + Fold07: alpha=0.0, lambda=10
## - Fold07: alpha=0.0, lambda=10
## + Fold07: alpha=0.1, lambda=10
## - Fold07: alpha=0.1, lambda=10
## + Fold07: alpha=0.2, lambda=10
## - Fold07: alpha=0.2, lambda=10
## + Fold07: alpha=0.3, lambda=10
## - Fold07: alpha=0.3, lambda=10
## + Fold07: alpha=0.4, lambda=10
## - Fold07: alpha=0.4, lambda=10
## + Fold07: alpha=0.5, lambda=10
## - Fold07: alpha=0.5, lambda=10
## + Fold07: alpha=0.6, lambda=10
## - Fold07: alpha=0.6, lambda=10
## + Fold07: alpha=0.7, lambda=10
## - Fold07: alpha=0.7, lambda=10
## + Fold07: alpha=0.8, lambda=10
## - Fold07: alpha=0.8, lambda=10
## + Fold07: alpha=0.9, lambda=10
## - Fold07: alpha=0.9, lambda=10
## + Fold07: alpha=1.0, lambda=10
## - Fold07: alpha=1.0, lambda=10
## + Fold08: alpha=0.0, lambda=10
## - Fold08: alpha=0.0, lambda=10
## + Fold08: alpha=0.1, lambda=10
## - Fold08: alpha=0.1, lambda=10
## + Fold08: alpha=0.2, lambda=10
## - Fold08: alpha=0.2, lambda=10
## + Fold08: alpha=0.3, lambda=10
## - Fold08: alpha=0.3, lambda=10
## + Fold08: alpha=0.4, lambda=10
## - Fold08: alpha=0.4, lambda=10
## + Fold08: alpha=0.5, lambda=10
## - Fold08: alpha=0.5, lambda=10
## + Fold08: alpha=0.6, lambda=10
## - Fold08: alpha=0.6, lambda=10
## + Fold08: alpha=0.7, lambda=10
## - Fold08: alpha=0.7, lambda=10
## + Fold08: alpha=0.8, lambda=10
## - Fold08: alpha=0.8, lambda=10
## + Fold08: alpha=0.9, lambda=10
## - Fold08: alpha=0.9, lambda=10
## + Fold08: alpha=1.0, lambda=10
## - Fold08: alpha=1.0, lambda=10
## + Fold09: alpha=0.0, lambda=10
## - Fold09: alpha=0.0, lambda=10
```

```
## + Fold09: alpha=0.1, lambda=10
## - Fold09: alpha=0.1, lambda=10
## + Fold09: alpha=0.2, lambda=10
## - Fold09: alpha=0.2, lambda=10
## + Fold09: alpha=0.3, lambda=10
## - Fold09: alpha=0.3, lambda=10
## + Fold09: alpha=0.4, lambda=10
## - Fold09: alpha=0.4, lambda=10
## + Fold09: alpha=0.5, lambda=10
## - Fold09: alpha=0.5, lambda=10
## + Fold09: alpha=0.6, lambda=10
## - Fold09: alpha=0.6, lambda=10
## + Fold09: alpha=0.7, lambda=10
## - Fold09: alpha=0.7, lambda=10
## + Fold09: alpha=0.8, lambda=10
## - Fold09: alpha=0.8, lambda=10
## + Fold09: alpha=0.9, lambda=10
## - Fold09: alpha=0.9, lambda=10
## + Fold09: alpha=1.0, lambda=10
## - Fold09: alpha=1.0, lambda=10
## + Fold10: alpha=0.0, lambda=10
## - Fold10: alpha=0.0, lambda=10
## + Fold10: alpha=0.1, lambda=10
## - Fold10: alpha=0.1, lambda=10
## + Fold10: alpha=0.2, lambda=10
## - Fold10: alpha=0.2, lambda=10
## + Fold10: alpha=0.3, lambda=10
## - Fold10: alpha=0.3, lambda=10
## + Fold10: alpha=0.4, lambda=10
## - Fold10: alpha=0.4, lambda=10
## + Fold10: alpha=0.5, lambda=10
## - Fold10: alpha=0.5, lambda=10
## + Fold10: alpha=0.6, lambda=10
## - Fold10: alpha=0.6, lambda=10
## + Fold10: alpha=0.7, lambda=10
## - Fold10: alpha=0.7, lambda=10
## + Fold10: alpha=0.8, lambda=10
## - Fold10: alpha=0.8, lambda=10
## + Fold10: alpha=0.9, lambda=10
## - Fold10: alpha=0.9, lambda=10
## + Fold10: alpha=1.0, lambda=10
## - Fold10: alpha=1.0, lambda=10
## Aggregating results
## Selecting tuning parameters
## Fitting alpha = 1, lambda = 0.00316 on full training set
```

```
print(model)
```

```
## glmnet
##
## 1172 samples
##    5 predictor
##    2 classes: 'non-stem', 'stem'
##
```

```
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1054, 1055, 1055, 1055, 1055, 1055, ...
## Resampling results across tuning parameters:
##
##   alpha  lambda        Accuracy   Kappa
##   0.0     0.001000000  0.6535709  0.19892175
##   0.0     0.003162278  0.6535709  0.19892175
##   0.0     0.010000000  0.6535709  0.19892175
##   0.0     0.031622777  0.6492974  0.17776624
##   0.0     0.100000000  0.6450167  0.14144740
##   0.0     0.316227766  0.6331233  0.03718374
##   0.0     1.000000000  0.6237216  0.00000000
##   0.0     3.162277660  0.6237216  0.00000000
##   0.0    10.000000000  0.6237216  0.00000000
##   0.1     0.001000000  0.6569752  0.21270876
##   0.1     0.003162278  0.6544111  0.20563810
##   0.1     0.010000000  0.6552731  0.20491139
##   0.1     0.031622777  0.6501449  0.17781050
##   0.1     0.100000000  0.6390410  0.11989272
##   0.1     0.316227766  0.6237216  0.00000000
##   0.1     1.000000000  0.6237216  0.00000000
##   0.1     3.162277660  0.6237216  0.00000000
##   0.1    10.000000000  0.6237216  0.00000000
##   0.2     0.001000000  0.6569752  0.21270876
##   0.2     0.003162278  0.6544111  0.20563810
##   0.2     0.010000000  0.6586846  0.21282150
##   0.2     0.031622777  0.6467261  0.16982515
##   0.2     0.100000000  0.6381863  0.11121605
##   0.2     0.316227766  0.6237216  0.00000000
##   0.2     1.000000000  0.6237216  0.00000000
##   0.2     3.162277660  0.6237216  0.00000000
##   0.2    10.000000000  0.6237216  0.00000000
##   0.3     0.001000000  0.6569752  0.21270876
##   0.3     0.003162278  0.6544111  0.20622551
##   0.3     0.010000000  0.6586846  0.21376033
##   0.3     0.031622777  0.6441620  0.16348590
##   0.3     0.100000000  0.6390410  0.10303252
##   0.3     0.316227766  0.6237216  0.00000000
##   0.3     1.000000000  0.6237216  0.00000000
##   0.3     3.162277660  0.6237216  0.00000000
##   0.3    10.000000000  0.6237216  0.00000000
##   0.4     0.001000000  0.6569752  0.21270876
##   0.4     0.003162278  0.6552586  0.20862876
##   0.4     0.010000000  0.6561278  0.20610692
##   0.4     0.031622777  0.6415906  0.15693616
##   0.4     0.100000000  0.6424453  0.10513678
##   0.4     0.316227766  0.6237216  0.00000000
##   0.4     1.000000000  0.6237216  0.00000000
##   0.4     3.162277660  0.6237216  0.00000000
##   0.4    10.000000000  0.6237216  0.00000000
##   0.5     0.001000000  0.6569752  0.21270876
##   0.5     0.003162278  0.6569607  0.21178068
##   0.5     0.010000000  0.6561278  0.20471387
```

```
##   0.5     0.031622777  0.6407359  0.15423739
##   0.5     0.100000000  0.6390627  0.08722695
##   0.5     0.316227766  0.6237216  0.00000000
##   0.5     1.000000000  0.6237216  0.00000000
##   0.5     3.162277660  0.6237216  0.00000000
##   0.5    10.000000000  0.6237216  0.00000000
##   0.6     0.001000000  0.6569752  0.21270876
##   0.6     0.003162278  0.6578154  0.21336565
##   0.6     0.010000000  0.6552731  0.20245295
##   0.6     0.031622777  0.6398812  0.15238886
##   0.6     0.100000000  0.6356584  0.06626350
##   0.6     0.316227766  0.6237216  0.00000000
##   0.6     1.000000000  0.6237216  0.00000000
##   0.6     3.162277660  0.6237216  0.00000000
##   0.6    10.000000000  0.6237216  0.00000000
##   0.7     0.001000000  0.6569752  0.21270876
##   0.7     0.003162278  0.6586701  0.21496627
##   0.7     0.010000000  0.6578372  0.20643214
##   0.7     0.031622777  0.6407359  0.15319902
##   0.7     0.100000000  0.6262857  0.02562173
##   0.7     0.316227766  0.6237216  0.00000000
##   0.7     1.000000000  0.6237216  0.00000000
##   0.7     3.162277660  0.6237216  0.00000000
##   0.7    10.000000000  0.6237216  0.00000000
##   0.8     0.001000000  0.6569752  0.21270876
##   0.8     0.003162278  0.6595248  0.21657893
##   0.8     0.010000000  0.6569897  0.20311420
##   0.8     0.031622777  0.6424598  0.15610030
##   0.8     0.100000000  0.6305592  0.02536890
##   0.8     0.316227766  0.6237216  0.00000000
##   0.8     1.000000000  0.6237216  0.00000000
##   0.8     3.162277660  0.6237216  0.00000000
##   0.8    10.000000000  0.6237216  0.00000000
##   0.9     0.001000000  0.6569752  0.21270876
##   0.9     0.003162278  0.6595176  0.21739073
##   0.9     0.010000000  0.6544329  0.19510979
##   0.9     0.031622777  0.6441547  0.15584813
##   0.9     0.100000000  0.6245763  0.00494911
##   0.9     0.316227766  0.6237216  0.00000000
##   0.9     1.000000000  0.6237216  0.00000000
##   0.9     3.162277660  0.6237216  0.00000000
##   0.9    10.000000000  0.6237216  0.00000000
##   1.0     0.001000000  0.6569752  0.21270876
##   1.0     0.003162278  0.6612270  0.22121112
##   1.0     0.010000000  0.6518688  0.18887693
##   1.0     0.031622777  0.6450094  0.15504921
##   1.0     0.100000000  0.6237216  0.00000000
##   1.0     0.316227766  0.6237216  0.00000000
##   1.0     1.000000000  0.6237216  0.00000000
##   1.0     3.162277660  0.6237216  0.00000000
##   1.0    10.000000000  0.6237216  0.00000000
##
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were alpha = 1 and lambda = 0.003162278.
```

```
print(model$results)
```

```
##    alpha       lambda  Accuracy      Kappa  AccuracySD    KappaSD
## 1    0.0  0.001000000 0.6535709 0.19892175 0.027994776 0.06743390
## 2    0.0  0.003162278 0.6535709 0.19892175 0.027994776 0.06743390
## 3    0.0  0.010000000 0.6535709 0.19892175 0.027994776 0.06743390
## 4    0.0  0.031622777 0.6492974 0.17776624 0.032152132 0.07496905
## 5    0.0  0.100000000 0.6450167 0.14144740 0.026221294 0.06743679
## 6    0.0  0.316227766 0.6331233 0.03718374 0.013349759 0.04164830
## 7    0.0  1.000000000 0.6237216 0.00000000 0.002045975 0.00000000
## 8    0.0  3.162277660 0.6237216 0.00000000 0.002045975 0.00000000
## 9    0.0 10.000000000 0.6237216 0.00000000 0.002045975 0.00000000
## 10   0.1  0.001000000 0.6569752 0.21270876 0.034153263 0.08353692
## 11   0.1  0.003162278 0.6544111 0.20563810 0.030075899 0.07520845
## 12   0.1  0.010000000 0.6552731 0.20491139 0.025338648 0.06507427
## 13   0.1  0.031622777 0.6501449 0.17781050 0.031360609 0.07181368
## 14   0.1  0.100000000 0.6390410 0.11989272 0.027454459 0.06253431
## 15   0.1  0.316227766 0.6237216 0.00000000 0.002045975 0.00000000
## 16   0.1  1.000000000 0.6237216 0.00000000 0.002045975 0.00000000
## 17   0.1  3.162277660 0.6237216 0.00000000 0.002045975 0.00000000
## 18   0.1 10.000000000 0.6237216 0.00000000 0.002045975 0.00000000
## 19   0.2  0.001000000 0.6569752 0.21270876 0.034153263 0.08353692
## 20   0.2  0.003162278 0.6544111 0.20563810 0.030075899 0.07520845
## 21   0.2  0.010000000 0.6586846 0.21282150 0.028009182 0.07027192
## 22   0.2  0.031622777 0.6467261 0.16982515 0.034200173 0.07778496
## 23   0.2  0.100000000 0.6381863 0.11121605 0.029469519 0.06401603
## 24   0.2  0.316227766 0.6237216 0.00000000 0.002045975 0.00000000
## 25   0.2  1.000000000 0.6237216 0.00000000 0.002045975 0.00000000
## 26   0.2  3.162277660 0.6237216 0.00000000 0.002045975 0.00000000
## 27   0.2 10.000000000 0.6237216 0.00000000 0.002045975 0.00000000
## 28   0.3  0.001000000 0.6569752 0.21270876 0.034153263 0.08353692
## 29   0.3  0.003162278 0.6544111 0.20622551 0.031136702 0.07925183
## 30   0.3  0.010000000 0.6586846 0.21376033 0.028582889 0.06901747
## 31   0.3  0.031622777 0.6441620 0.16348590 0.034916732 0.07762783
## 32   0.3  0.100000000 0.6390410 0.10303252 0.028283960 0.06909335
## 33   0.3  0.316227766 0.6237216 0.00000000 0.002045975 0.00000000
## 34   0.3  1.000000000 0.6237216 0.00000000 0.002045975 0.00000000
## 35   0.3  3.162277660 0.6237216 0.00000000 0.002045975 0.00000000
## 36   0.3 10.000000000 0.6237216 0.00000000 0.002045975 0.00000000
## 37   0.4  0.001000000 0.6569752 0.21270876 0.034153263 0.08353692
## 38   0.4  0.003162278 0.6552586 0.20862876 0.030154903 0.07610778
## 39   0.4  0.010000000 0.6561278 0.20610692 0.027425547 0.06389815
## 40   0.4  0.031622777 0.6415906 0.15693616 0.036587865 0.08248634
## 41   0.4  0.100000000 0.6424453 0.10513678 0.029930616 0.07088241
## 42   0.4  0.316227766 0.6237216 0.00000000 0.002045975 0.00000000
## 43   0.4  1.000000000 0.6237216 0.00000000 0.002045975 0.00000000
## 44   0.4  3.162277660 0.6237216 0.00000000 0.002045975 0.00000000
## 45   0.4 10.000000000 0.6237216 0.00000000 0.002045975 0.00000000
## 46   0.5  0.001000000 0.6569752 0.21270876 0.034153263 0.08353692
## 47   0.5  0.003162278 0.6569607 0.21178068 0.028455244 0.07239212
## 48   0.5  0.010000000 0.6561278 0.20471387 0.026522815 0.06043774
## 49   0.5  0.031622777 0.6407359 0.15423739 0.038430706 0.08923284
## 50   0.5  0.100000000 0.6390627 0.08722695 0.022253217 0.04740872
```

```
## 51   0.5  0.316227766 0.6237216 0.00000000 0.002045975 0.00000000
## 52   0.5  1.000000000 0.6237216 0.00000000 0.002045975 0.00000000
## 53   0.5  3.162277660 0.6237216 0.00000000 0.002045975 0.00000000
## 54   0.5 10.000000000 0.6237216 0.00000000 0.002045975 0.00000000
## 55   0.6  0.001000000 0.6569752 0.21270876 0.034153263 0.08353692
## 56   0.6  0.003162278 0.6578154 0.21336565 0.028336788 0.07206419
## 57   0.6  0.010000000 0.6552731 0.20245295 0.029207607 0.06535220
## 58   0.6  0.031622777 0.6398812 0.15238886 0.038518489 0.09180422
## 59   0.6  0.100000000 0.6356584 0.06626350 0.020580412 0.05142497
## 60   0.6  0.316227766 0.6237216 0.00000000 0.002045975 0.00000000
## 61   0.6  1.000000000 0.6237216 0.00000000 0.002045975 0.00000000
## 62   0.6  3.162277660 0.6237216 0.00000000 0.002045975 0.00000000
## 63   0.6 10.000000000 0.6237216 0.00000000 0.002045975 0.00000000
## 64   0.7  0.001000000 0.6569752 0.21270876 0.034153263 0.08353692
## 65   0.7  0.003162278 0.6586701 0.21496627 0.028475542 0.07208604
## 66   0.7  0.010000000 0.6578372 0.20643214 0.033126828 0.07491134
## 67   0.7  0.031622777 0.6407359 0.15319902 0.038218915 0.09107688
## 68   0.7  0.100000000 0.6262857 0.02562173 0.014177383 0.04081373
## 69   0.7  0.316227766 0.6237216 0.00000000 0.002045975 0.00000000
## 70   0.7  1.000000000 0.6237216 0.00000000 0.002045975 0.00000000
## 71   0.7  3.162277660 0.6237216 0.00000000 0.002045975 0.00000000
## 72   0.7 10.000000000 0.6237216 0.00000000 0.002045975 0.00000000
## 73   0.8  0.001000000 0.6569752 0.21270876 0.034153263 0.08353692
## 74   0.8  0.003162278 0.6595248 0.21657893 0.028585243 0.07187368
## 75   0.8  0.010000000 0.6569897 0.20311420 0.034331510 0.08043205
## 76   0.8  0.031622777 0.6424598 0.15610030 0.039886665 0.09848908
## 77   0.8  0.100000000 0.6305592 0.02536890 0.011575839 0.04177211
## 78   0.8  0.316227766 0.6237216 0.00000000 0.002045975 0.00000000
## 79   0.8  1.000000000 0.6237216 0.00000000 0.002045975 0.00000000
## 80   0.8  3.162277660 0.6237216 0.00000000 0.002045975 0.00000000
## 81   0.8 10.000000000 0.6237216 0.00000000 0.002045975 0.00000000
## 82   0.9  0.001000000 0.6569752 0.21270876 0.034153263 0.08353692
## 83   0.9  0.003162278 0.6595176 0.21739073 0.028974360 0.07180314
## 84   0.9  0.010000000 0.6544329 0.19510979 0.040216614 0.09514636
## 85   0.9  0.031622777 0.6441547 0.15584813 0.040909765 0.10102580
## 86   0.9  0.100000000 0.6245763 0.00494911 0.006660134 0.02227315
## 87   0.9  0.316227766 0.6237216 0.00000000 0.002045975 0.00000000
## 88   0.9  1.000000000 0.6237216 0.00000000 0.002045975 0.00000000
## 89   0.9  3.162277660 0.6237216 0.00000000 0.002045975 0.00000000
## 90   0.9 10.000000000 0.6237216 0.00000000 0.002045975 0.00000000
## 91   1.0  0.001000000 0.6569752 0.21270876 0.034153263 0.08353692
## 92   1.0  0.003162278 0.6612270 0.22121112 0.028826316 0.07279315
## 93   1.0  0.010000000 0.6518688 0.18887693 0.037275070 0.08665291
## 94   1.0  0.031622777 0.6450094 0.15504921 0.039922459 0.09812888
## 95   1.0  0.100000000 0.6237216 0.00000000 0.002045975 0.00000000
## 96   1.0  0.316227766 0.6237216 0.00000000 0.002045975 0.00000000
## 97   1.0  1.000000000 0.6237216 0.00000000 0.002045975 0.00000000
## 98   1.0  3.162277660 0.6237216 0.00000000 0.002045975 0.00000000
## 99   1.0 10.000000000 0.6237216 0.00000000 0.002045975 0.00000000
```

```
summary(model)
```

```
##             Length Class      Mode
## a0           59    -none-     numeric
```

```
## beta        295    dgCMatrix  S4
## df           59    -none-     numeric
## dim           2    -none-     numeric
## lambda       59    -none-     numeric
## dev.ratio    59    -none-     numeric
## nulldev       1    -none-     numeric
## npasses       1    -none-     numeric
## jerr          1    -none-     numeric
## offset        1    -none-     logical
## classnames    2    -none-     character
## call          5    -none-     call
## nobs          1    -none-     numeric
## lambdaOpt     1    -none-     numeric
## xNames        5    -none-     character
## problemType   1    -none-     character
## tuneValue     2    data.frame list
## obsLevels     2    -none-     character
## param         0    -none-     list
```

```
predictions <- predict(model, newdata = testData)
confusionMatrix(predictions, testData$Dept.Type)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction non-stem stem
##   non-stem       52   44
##   stem           95  199
##
##                Accuracy : 0.6436
##                  95% CI : (0.5938, 0.6912)
##     No Information Rate : 0.6231
##     P-Value [Acc > NIR] : 0.2171
##
##                   Kappa : 0.1854
##
##  Mcnemar's Test P-Value : 2.226e-05
##
##             Sensitivity : 0.3537
##             Specificity : 0.8189
##          Pos Pred Value : 0.5417
##          Neg Pred Value : 0.6769
##              Prevalence : 0.3769
##          Detection Rate : 0.1333
##    Detection Prevalence : 0.2462
##       Balanced Accuracy : 0.5863
##
##        'Positive' Class : non-stem
##
```