IFN702: Educational Data Mining project

version 1.0 — Zhiying Zhou: 9835580

Introduction

This study will process, exploration, analyze and predict Student's academic performance based on a XAPI Educational Mining Datasets.

The XAPI Educational Mining Datasets is download from the Kaggle website. It was collected from an e-Learning system called Kalboard 360 using Experience API Web service (XAPI).

The datasets consist of 480 student records and 16 features. The features are classified into three major categories: 1. Demographic features such as gender, grade levels, topic and nationality. 2. Academic background features such as educational stage, grade Level and section. 3. Behavioral features such as raised hand on class, opening resources, answering survey by parents, and school satisfaction.

Attributes 1 Gender - student's gender (nominal: 'Male' or 'Female') 2 Nationality- student's nationality (nominal:'Kuwait', 'Lebanon', 'Egypt', 'SaudiArabia', 'USA', 'Jordan', 'Venezuela', 'Iran', 'Tunis', 'Morocco', 'Syria', 'Palestine', 'Iraq', 'Lybia') 3 Place of birth- student's Place of birth (nominal:'Kuwait', 'Lebanon', 'Egypt', 'SaudiArabia', 'USA', 'Jordan', 'Venezuela', 'Iran', 'Tunis', 'Morocco', 'Syria', 'Palestine', 'Iraq', 'Lybia') 4 Educational Stages- educational level student belongs (nominal: 'lowerlevel','MiddleSchool','HighSchool') 5 Grade Levels- grade student belongs (nominal: 'G-01', 'G-02', 'G-03', 'G-04', 'G-05', 'G-06', 'G-07', 'G-08', 'G-09', 'G-10', 'G-11', 'G-12') 6 Section ID- classroom student belongs (nominal:'A','B','C') 7 Topic- course topic (nominal:'English','Spanish', 'French','Arabic','IT','Math','Chemistry', 'Biology', 'Science','History','Quran','Geology') 8 Semester- school year semester (nominal:'First','Second') 9 Parent responsible for student (nominal:'mom','father') 10 Raised hand- how many times the student raises his/her hand on classroom (numeric:0-100) 11- Visited resources- how many times the student visits a course content(numeric:0-100) 12 Viewing announcements-how many times the student checks the new announcements(numeric:0-100) 13 Discussion groups- how many times the student participate on discussion groups (numeric:0-100) 14 Parent Answering Survey- parent answered the surveys which are provided from school or not (nominal:'Yes','No') 15 Parent School Satisfaction- the Degree of parent satisfaction from school(nominal:'Yes','No') 16 Student Absence Days-the number of absence days for each student (nominal: above-7, under-7)

The students are classified into three numerical intervals based on their total grade/mark:

Low-Level: interval includes values from 0 to 69, Middle-Level: interval includes values from 70 to 89, High-Level: interval includes values from 90-100.

We will use R Studio to analyze this datasets in R programming language, and aim to achieve the following targets:

1. Pre-process of the datasets. Clean the dataset, remove the useless columns or rows; and
2. Explore the datasets. Explore the distribution of the datasets in different features: gender, nationality, grade, topic, parental satisfaction etc. (like girls raises more hand, more discussions in high school etc.); and
3. Find underlying relationships. Like parent who are not satisfied and not answer survey, connection with study activity and performance (raising hand, discussion, absence, parental satisfaction, answering survey etc.); and
4. Build prediction model, like decision tree or regression model to predict the student's academic performance.
5. Evaluate the predictive results of models and improve the models by comparing the accuracy.

Task 1. Exploration of the students' academic performance datasets

```
# install libraries
library(ggplot2)
library(reshape2)
library(partykit)
```

```
## Warning: package 'partykit' was built under R version 3.4.2
```

```
## Loading required package: grid
```

```
library(caret)
```

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

```
library(e1071)
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.4.2
```

```
## randomForest 4.6-12
```

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

```
##
## Attaching package: 'randomForest'
```

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

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.2
```

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

```
## The following object is masked from 'package:randomForest':
##
##     combine
```

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

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

Task 1.1. Preprocess the data set:

```
# Import XAPI Educational data sets
E <- read.csv("data/xAPI-Edu-Data.csv")
```

```
str(E)
```

```
## 'data.frame':    480 obs. of  17 variables:
##  $ gender                  : Factor w/ 2 levels "F","M": 2 2 2 2 2 1 2 2 1 1 ...
##  $ NationalITy             : Factor w/ 14 levels "Egypt","Iran",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ PlaceofBirth            : Factor w/ 14 levels "Egypt","Iran",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ StageID                 : Factor w/ 3 levels "HighSchool","lowerlevel",..: 2 2 2 2 2 2 3 3 3 3 ..
##  $ GradeID                 : Factor w/ 10 levels "G-02","G-04",..: 2 2 2 2 2 2 5 5 5 5 ...
##  $ SectionID               : Factor w/ 3 levels "A","B","C": 1 1 1 1 1 1 1 1 1 2 ...
##  $ Topic                   : Factor w/ 12 levels "Arabic","Biology",..: 8 8 8 8 8 8 9 9 9 8 ...
##  $ Semester                : Factor w/ 2 levels "F","S": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Relation                : Factor w/ 2 levels "Father","Mum": 1 1 1 1 1 1 1 1 1 1 ...
##  $ raisedhands             : int  15 20 10 30 40 42 35 50 12 70 ...
##  $ VisITedResources        : int  16 20 7 25 50 30 12 10 21 80 ...
##  $ AnnouncementsView       : int  2 3 0 5 12 13 0 15 16 25 ...
##  $ Discussion              : int  20 25 30 35 50 70 17 22 50 70 ...
##  $ ParentAnsweringSurvey   : Factor w/ 2 levels "No","Yes": 2 2 1 1 1 2 1 2 2 2 ...
##  $ ParentschoolSatisfaction: Factor w/ 2 levels "Bad","Good": 2 2 1 1 1 1 1 2 2 2 ...
##  $ StudentAbsenceDays      : Factor w/ 2 levels "Above-7","Under-7": 2 2 1 1 1 1 1 2 2 2 ...
##  $ Class                   : Factor w/ 3 levels "H","L","M": 3 3 2 2 3 3 2 3 3 3 ...
```

```
summary(E)
```

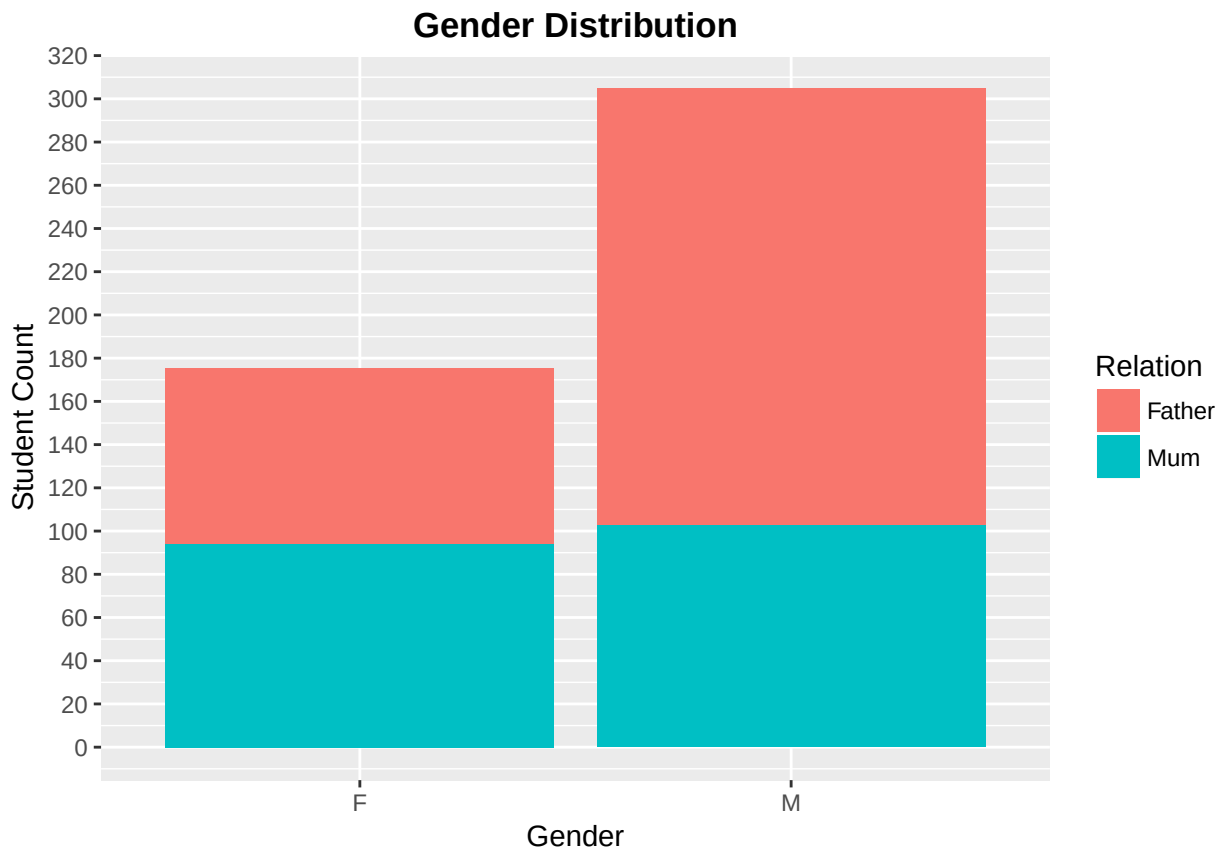
```
##  gender     NationalITy       PlaceofBirth         StageID
##  F:175   KW       :179   KuwaIT     :180   HighSchool  : 33
##  M:305   Jordan   :172   Jordan     :176   lowerlevel  :199
##          Palestine: 28   Iraq       : 22   MiddleSchool:248
##          Iraq     : 22   lebanon    : 19
##          lebanon  : 17   SaudiArabia: 16
##          Tunis    : 12   USA        : 16
##          (Other)  : 50   (Other)    : 51
##     GradeID    SectionID     Topic     Semester   Relation
##  G-02   :147   A:283     IT     : 95   F:245    Father:283
##  G-08   :116   B:167     French : 65   S:235    Mum   :197
##  G-07   :101   C: 30     Arabic : 59
##  G-04   : 48              Science: 51
##  G-06   : 32              English: 45
##  G-11   : 13              Biology: 30
##  (Other): 23              (Other):135
##   raisedhands     VisITedResources AnnouncementsView   Discussion
##  Min.   :  0.00   Min.   : 0.0     Min.   : 0.00     Min.   : 1.00
##  1st Qu.: 15.75   1st Qu.:20.0     1st Qu.:14.00     1st Qu.:20.00
##  Median : 50.00   Median :65.0     Median :33.00     Median :39.00
##  Mean   : 46.77   Mean   :54.8     Mean   :37.92     Mean   :43.28
##  3rd Qu.: 75.00   3rd Qu.:84.0     3rd Qu.:58.00     3rd Qu.:70.00
##  Max.   :100.00   Max.   :99.0     Max.   :98.00     Max.   :99.00
##
##  ParentAnsweringSurvey ParentschoolSatisfaction StudentAbsenceDays Class
##  No :210               Bad :188                 Above-7:191        H:142
##  Yes:270               Good:292                 Under-7:289        L:127
##                                                                    M:211
##
##
##
##
```

```r
# Reorder the 'Class'' column of the data set, make the
# sequence of 'Class' to be 'H','M','L' instead of
# 'H','L','M'.

E$Class = factor(E$Class, levels = c("H", "M", "L"), labels = c("H",
    "M", "L"))

# Reorder the 'StageID' column of the data set, make the
# sequence of 'StageID' to be
# 'HighSchool','MiddleSchool','Lowerlevel' instead of
# 'HighSchool','lowerlevel','MiddleSchool'.

E$StageID = factor(E$StageID, levels = c("HighSchool", "MiddleSchool",
    "lowerlevel"), labels = c("HighSchool", "MiddleSchool", "Lowerlevel"))
```

Task 1.2. Bar plots of classification values distribution:

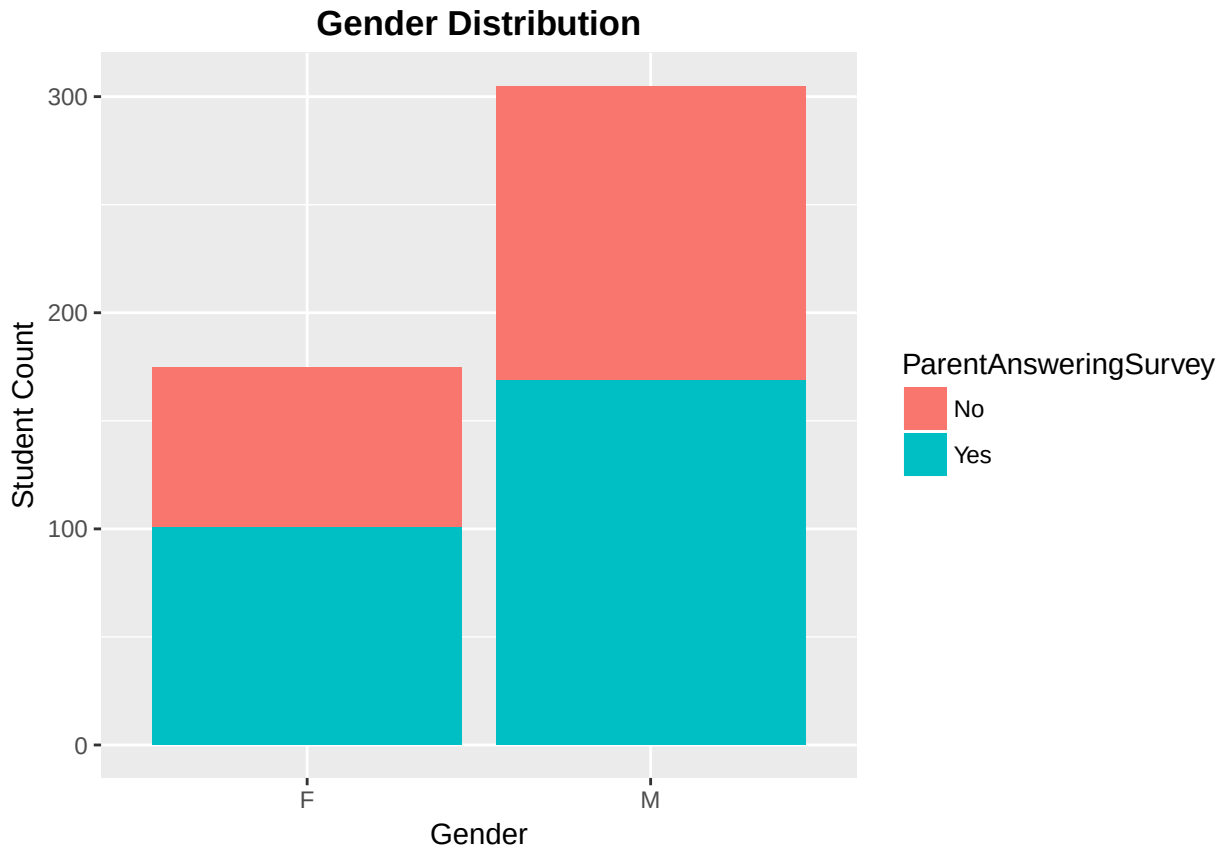
Gender distribution:

```r
ggplot(data = E, aes(x = gender, fill = Relation)) + geom_bar() +
    labs(x = "Gender", y = "Student Count") + ggtitle(label = "Gender Distribution") +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold")) + scale_y_continuous(breaks = seq(0,
    320, 20))
```



There are nearly 175 girls and 305 boys in this data set. The number of girls who have mom as guardian nearly the same as having father as guardian, whereas most boys have father as guardians.

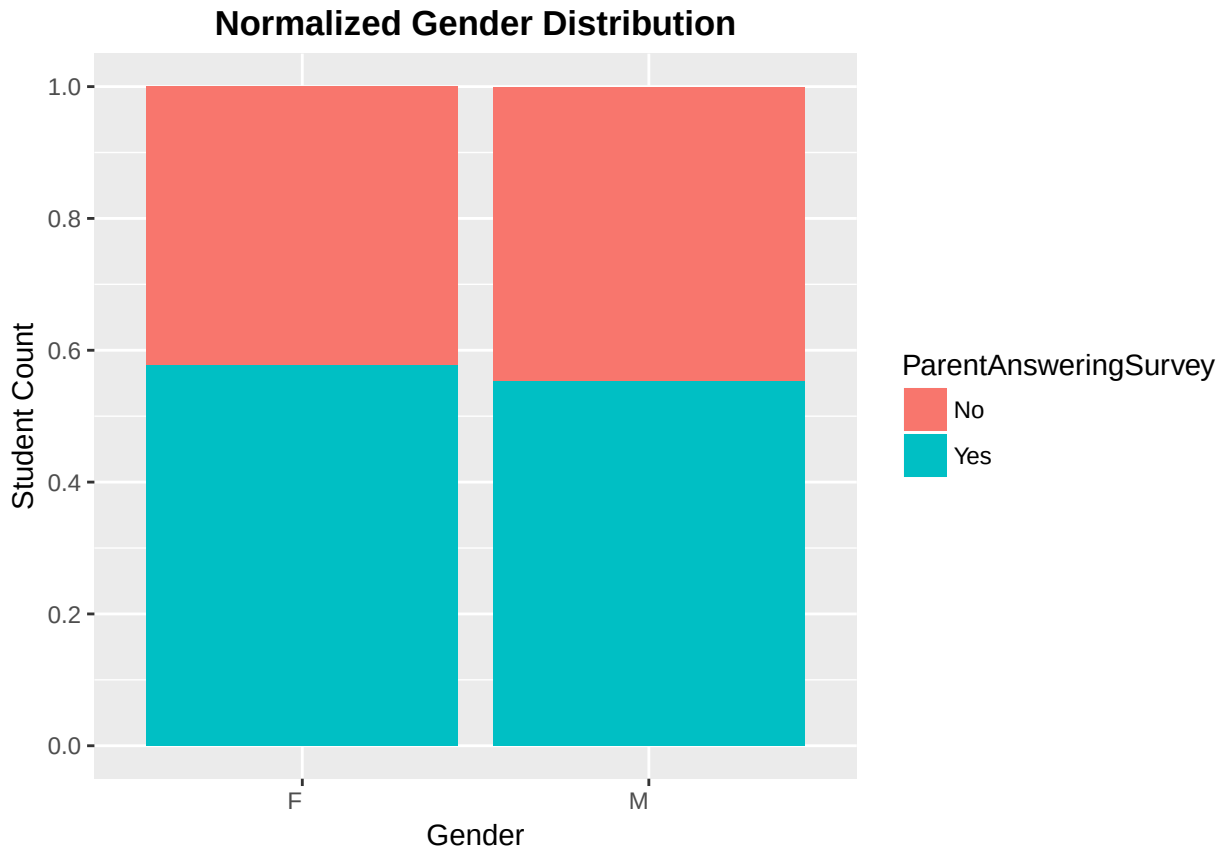
```
ggplot(data = E, aes(x = gender, fill = ParentAnsweringSurvey)) +
    geom_bar() + labs(x = "Gender", y = "Student Count") + ggtitle(label = "Gender Distribution") +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold"))
```




```
ESummary = E %>% # Get the counts
group_by(gender, ParentAnsweringSurvey) %>% summarise(count = n()) %>%
    # Get labels and position of labels
group_by(gender) %>% mutate(Percent = paste0(sprintf("%.1f",
    count/sum(count) * 100), "%"))
```

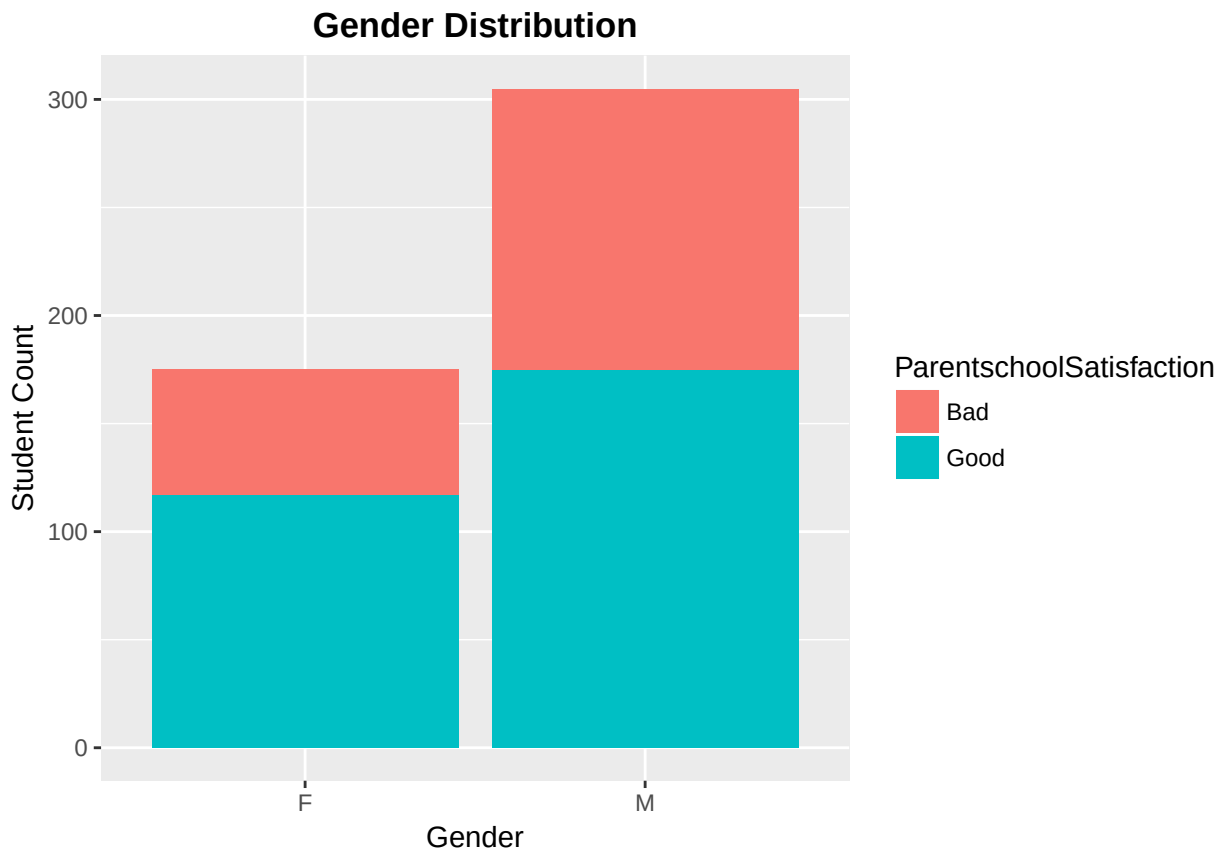
```
## Warning: package 'bindrcpp' was built under R version 3.4.2
```

```
ggplot(ESummary, aes(x = gender, y = count)) + geom_bar(aes(fill = ParentAnsweringSurvey),
    stat = "identity", position = "fill") + labs(x = "Gender",
    y = "Student Count") + ggtitle(label = "Normalized Gender Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold")) + theme(axis.text.x = element_text(angle = 0,
    hjust = 1)) + scale_y_continuous(breaks = seq(0, 1, 0.2))
```

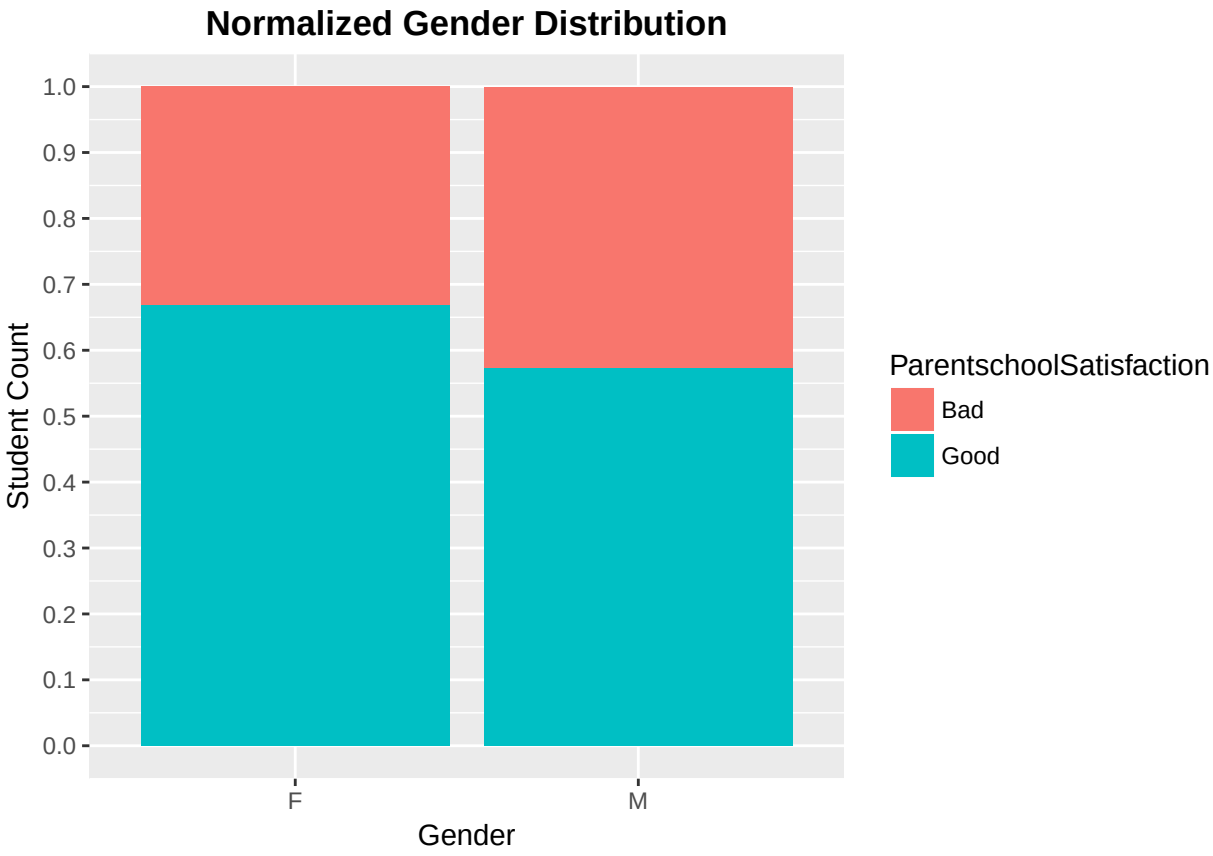


The rates of parent answering survey are nearly the same between girls and boys, which are both under 60%.

```
ggplot(data = E, aes(x = gender, fill = ParentschoolSatisfaction)) +
    geom_bar() + labs(x = "Gender", y = "Student Count") + ggtitle(label = "Gender Distribution") +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold"))
```

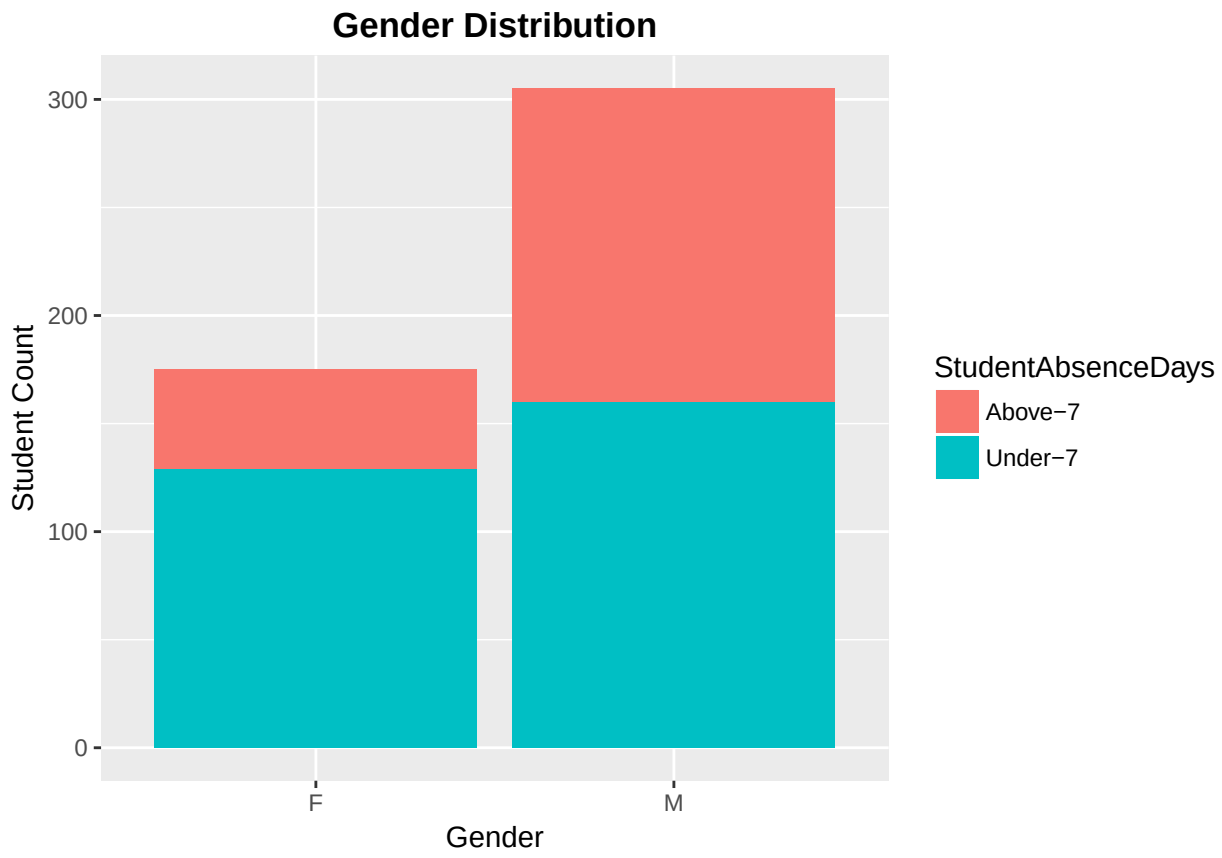



```
ESummary = E %>% # Get the counts
group_by(gender, ParentschoolSatisfaction) %>% summarise(count = n()) %>%
    # Get labels and position of labels
group_by(gender) %>% mutate(Percent = paste0(sprintf("%.1f",
    count/sum(count) * 100), "%"))

ggplot(ESummary, aes(x = gender, y = count)) + geom_bar(aes(fill = ParentschoolSatisfaction),
    stat = "identity", position = "fill") + labs(x = "Gender",
    y = "Student Count") + ggtitle(label = "Normalized Gender Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold")) + scale_y_continuous(breaks = seq(0,
    1, 0.1))
```

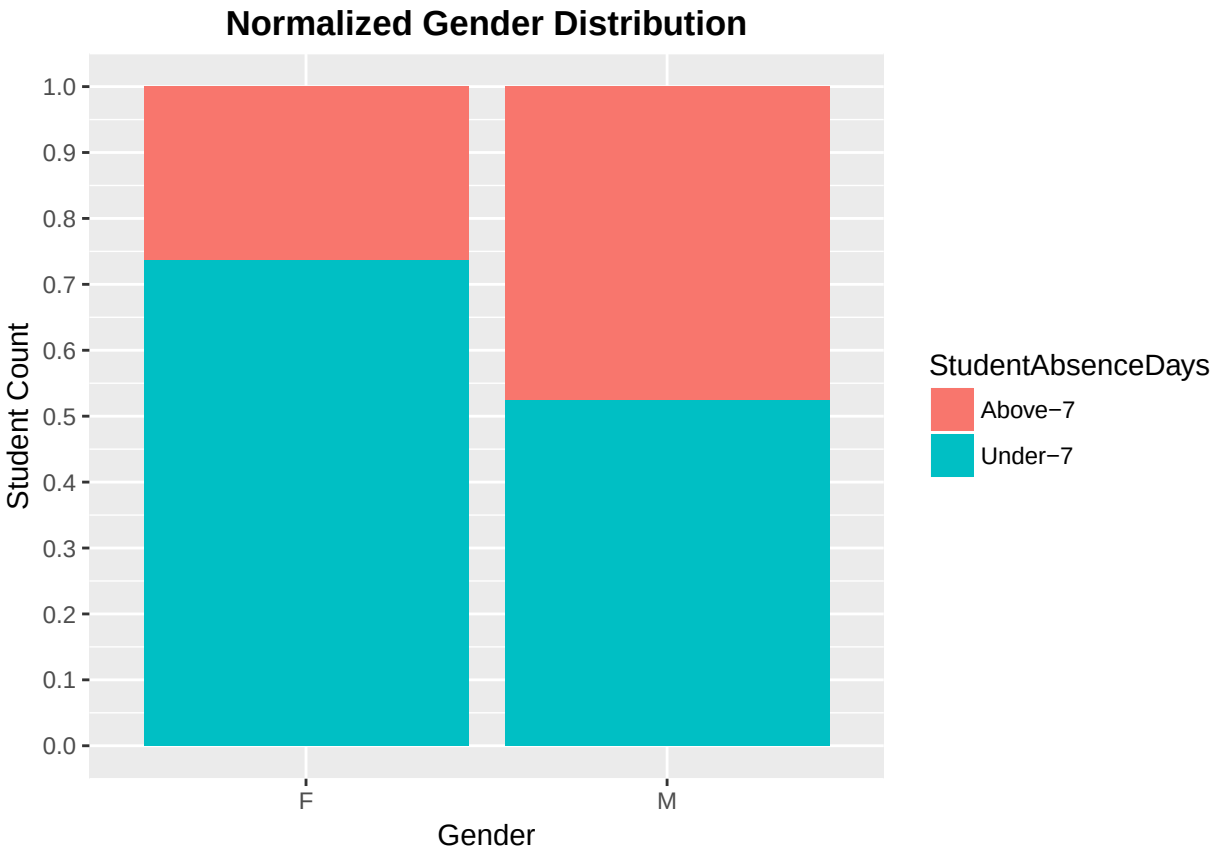


Girls' parents has higher school satisfaction than boys. Girls have higher rate(with about 67%) of parent school satisfaction in good than boys(with about 57%).

```
ggplot(data = E, aes(x = gender, fill = StudentAbsenceDays)) +
    geom_bar() + labs(x = "Gender", y = "Student Count") + ggtitle(label = "Gender Distribution") +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold"))
```

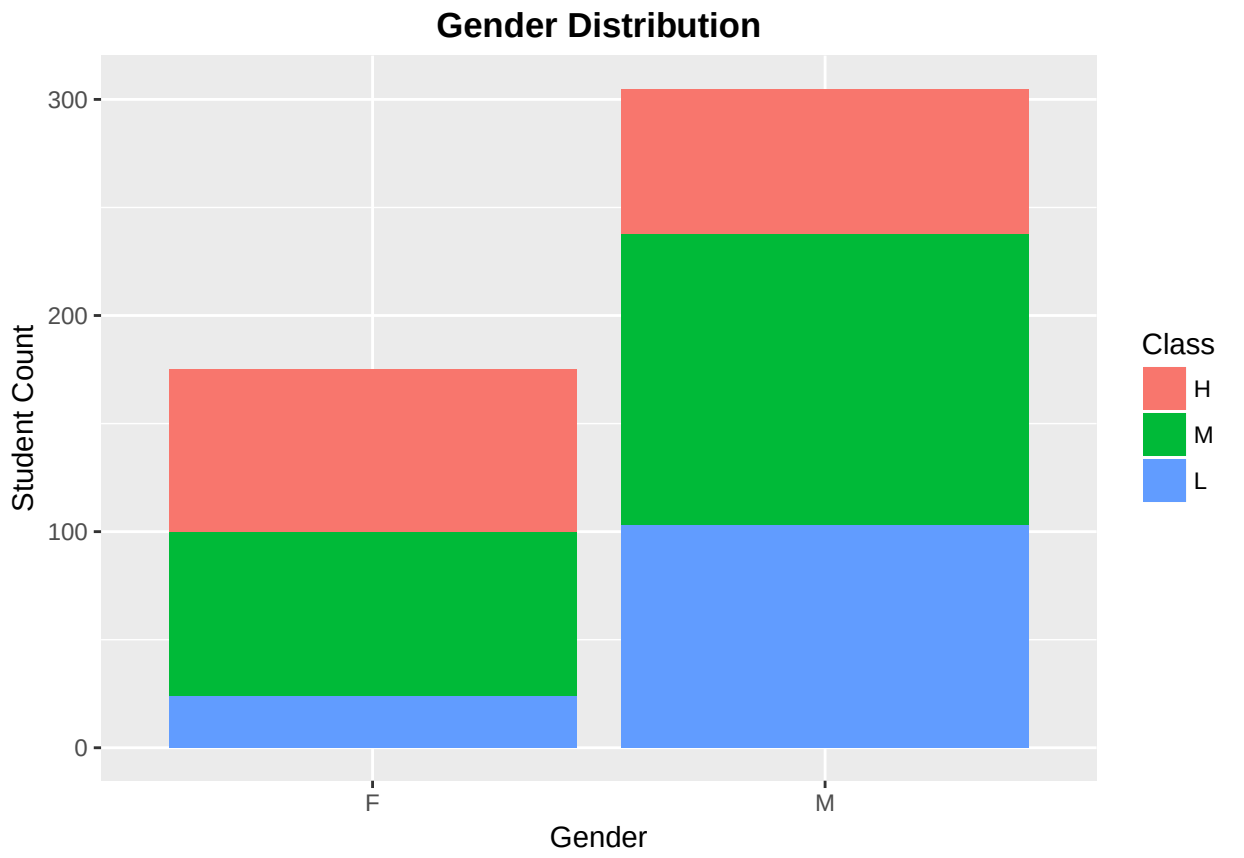



```
ESummary = E %>% # Get the counts
group_by(gender, StudentAbsenceDays) %>% summarise(count = n()) %>%
    # Get labels and position of labels
group_by(gender) %>% mutate(Percent = paste0(sprintf("%.1f",
    count/sum(count) * 100), "%"))

ggplot(ESummary, aes(x = gender, y = count)) + geom_bar(aes(fill = StudentAbsenceDays),
    stat = "identity", position = "fill") + labs(x = "Gender",
    y = "Student Count") + ggtitle(label = "Normalized Gender Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold")) + scale_y_continuous(breaks = seq(0,
    1, 0.1))
```

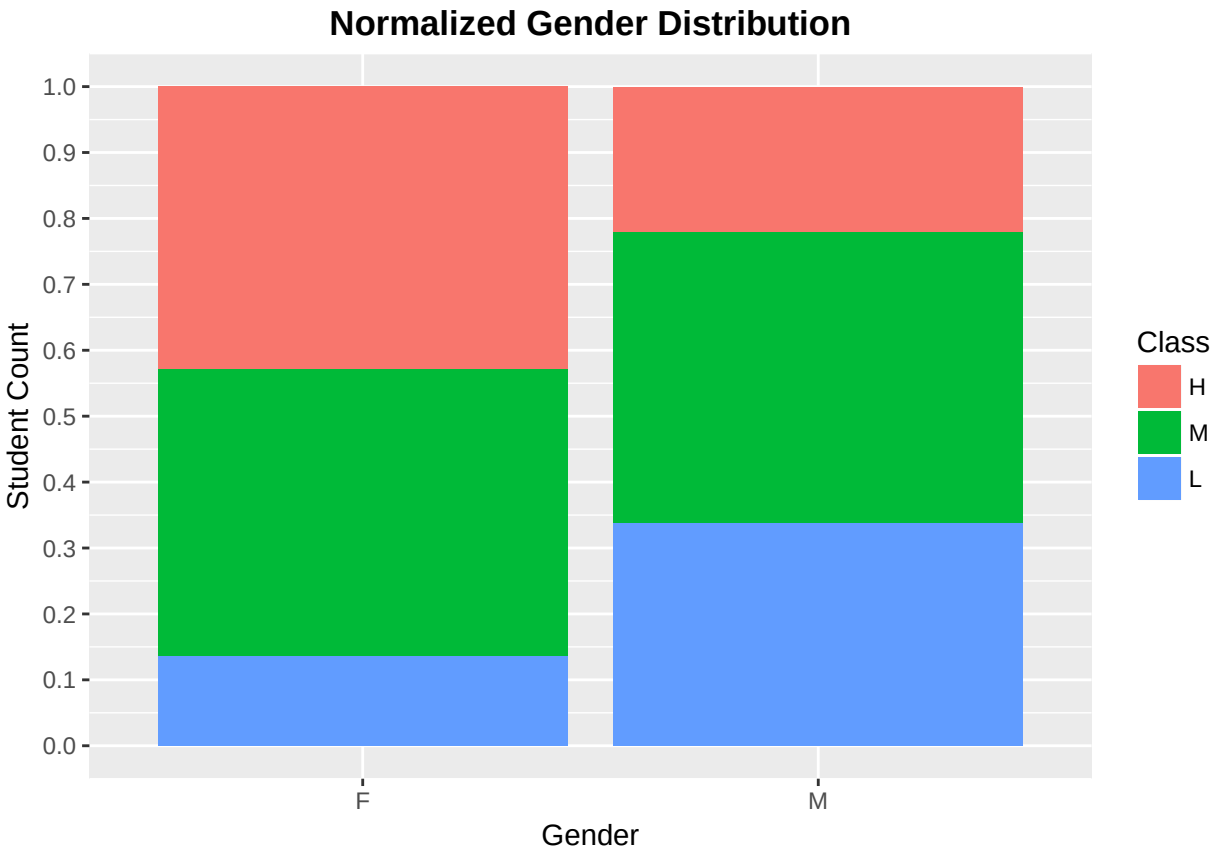


Boys has higher rate of above 7 absence days.

```
ggplot(data = E, aes(x = gender, fill = Class)) + geom_bar() +
    labs(x = "Gender", y = "Student Count") + ggtitle(label = "Gender Distribution") +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold"))
```



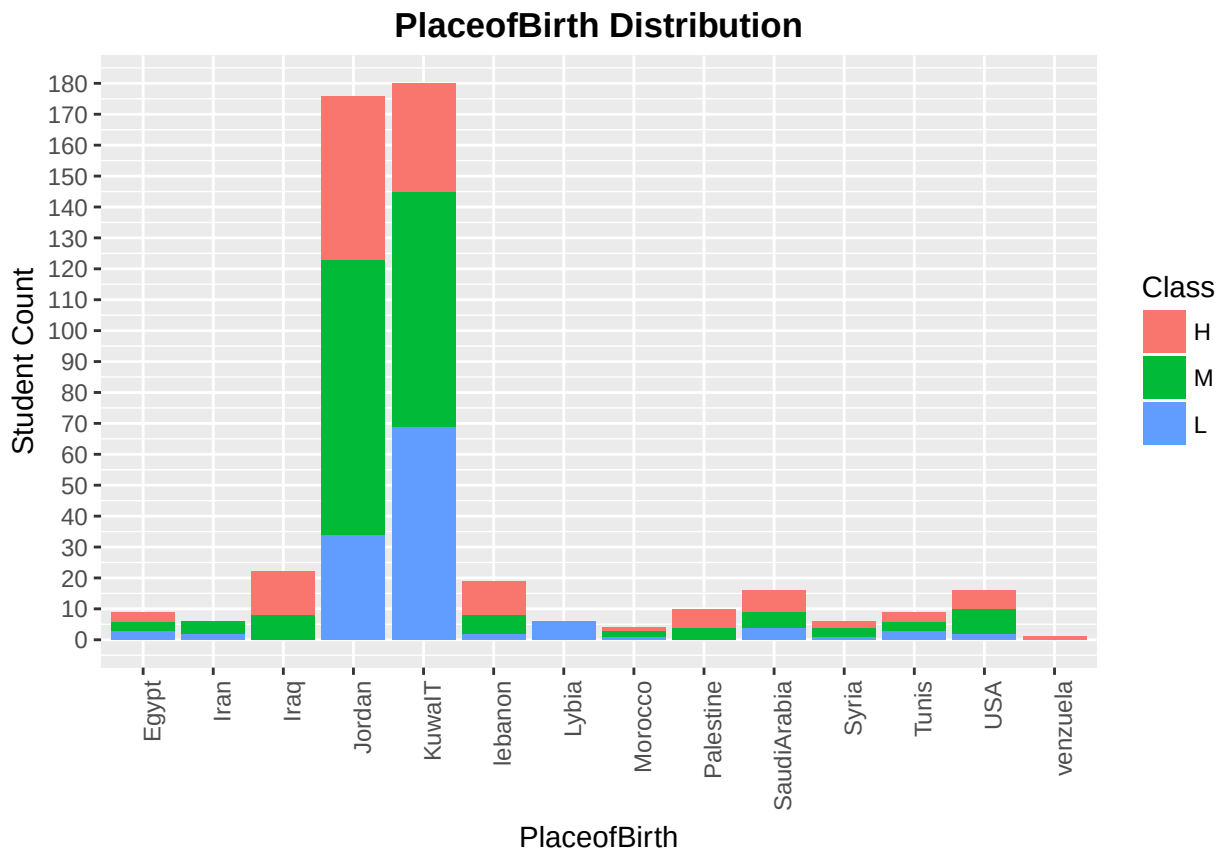

```
ESummary = E %>% # Get the counts
group_by(gender, Class) %>% summarise(count = n()) %>% # Get labels and position of labels
group_by(gender) %>% mutate(Percent = paste0(sprintf("%.1f",
    count/sum(count) * 100), "%"))

ggplot(ESummary, aes(x = gender, y = count)) + geom_bar(aes(fill = Class),
    stat = "identity", position = "fill") + labs(x = "Gender",
    y = "Student Count") + ggtitle(label = "Normalized Gender Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold")) + scale_y_continuous(breaks = seq(0,
    1, 0.1))
```



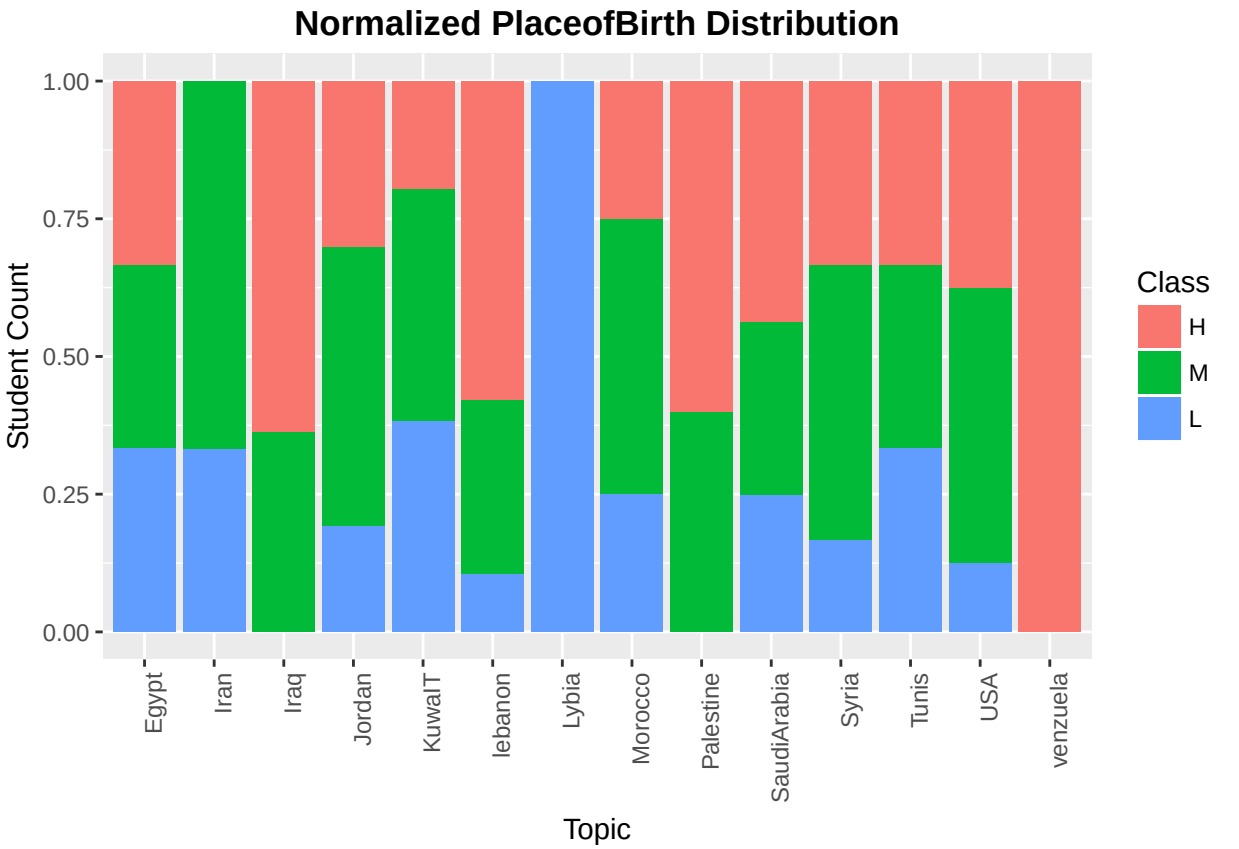
Girls have higher rate (over 40%) of people with high academic performance, whereas a small rate of boys (about 20%) have high performance. The rate of median class is almost the same both for girls and boys, which are both about 40%. About 35% rate of boys have low performance, whereas the rate of low class is very low among girls.

Place of Birth distribution:

```
ggplot(data = E, aes(x = PlaceofBirth, fill = Class)) + geom_bar() +
    labs(x = "PlaceofBirth", y = "Student Count") + ggtitle(label = "PlaceofBirth Distribution") +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold")) + theme(axis.text.x = element_text(angle = 90,
    hjust = 1)) + scale_y_continuous(breaks = seq(0, 200, 10))
```

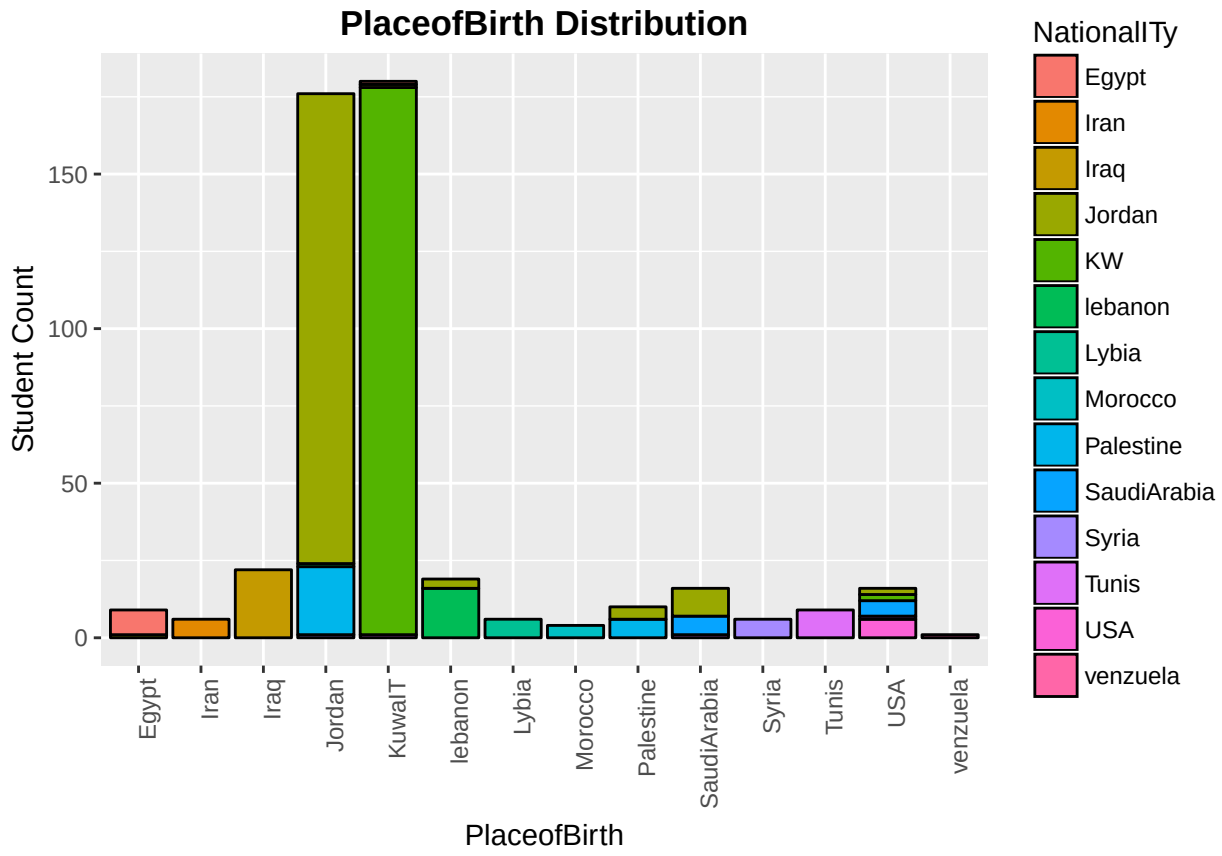



```
ESummary = E %>% # Get the counts
group_by(PlaceofBirth, Class) %>% summarise(count = n()) %>%
    # Get labels and position of labels
group_by(PlaceofBirth) %>% mutate(Percent = paste0(sprintf("%.1f",
    count/sum(count) * 100), "%"))

ggplot(ESummary, aes(x = PlaceofBirth, y = count)) + geom_bar(aes(fill = Class),
    stat = "identity", position = "fill") + labs(x = "Topic",
    y = "Student Count") + ggtitle(label = "Normalized PlaceofBirth Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold")) + theme(axis.text.x = element_text(angle = 90,
    hjust = 1))
```



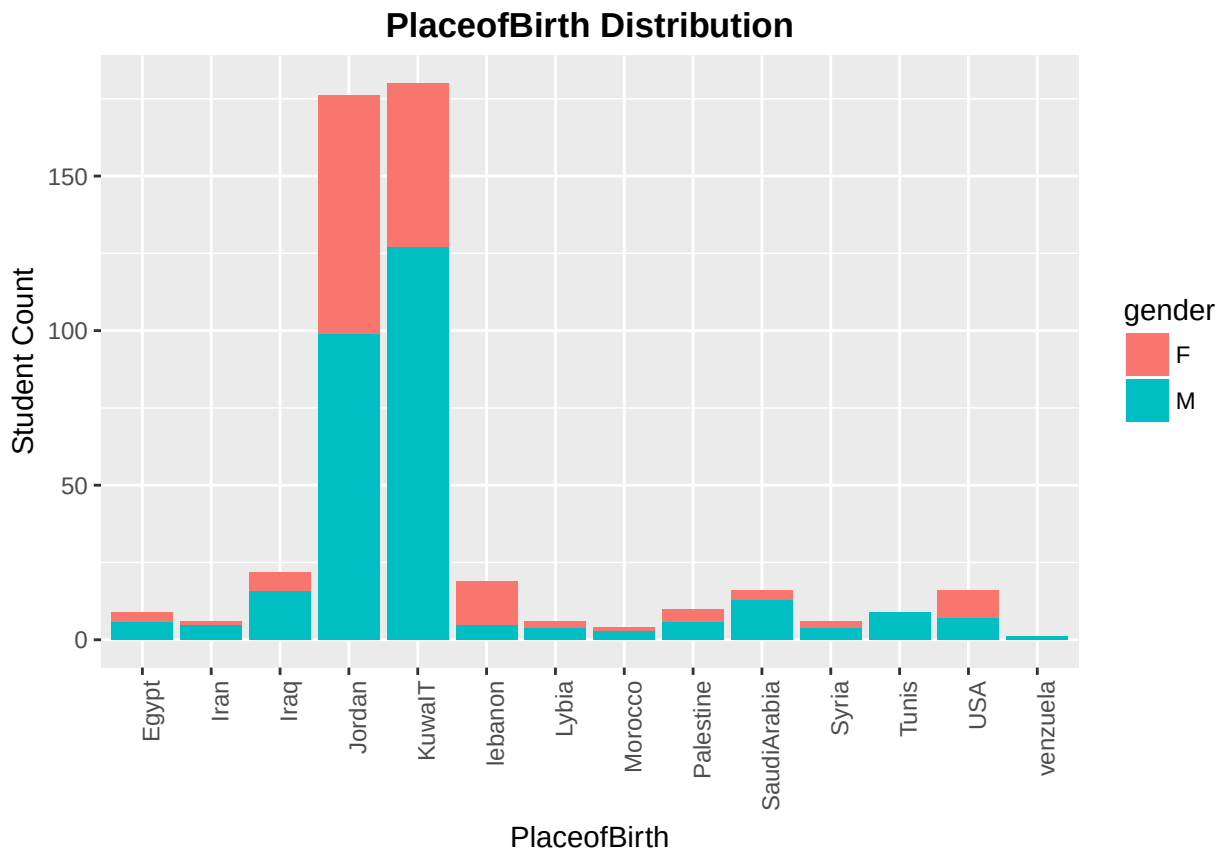
Most of the students were born in Jordan and Kuwait. Venzuela has only one student. Iraq, Lebanon, Saudiarabia, Palestine, USA and Venzuela have higher rate of student with high class, whereas Kuwait and Lybia have higher rate of students with low class.

```
ggplot(data = E, aes(x = PlaceofBirth, fill = NationalITy)) +
    geom_bar(colour = "Black") + labs(x = "PlaceofBirth", y = "Student Count") +
    ggtitle(label = "PlaceofBirth Distribution") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold")) + theme(axis.text.x = element_text(angle = 90,
    hjust = 1))
```



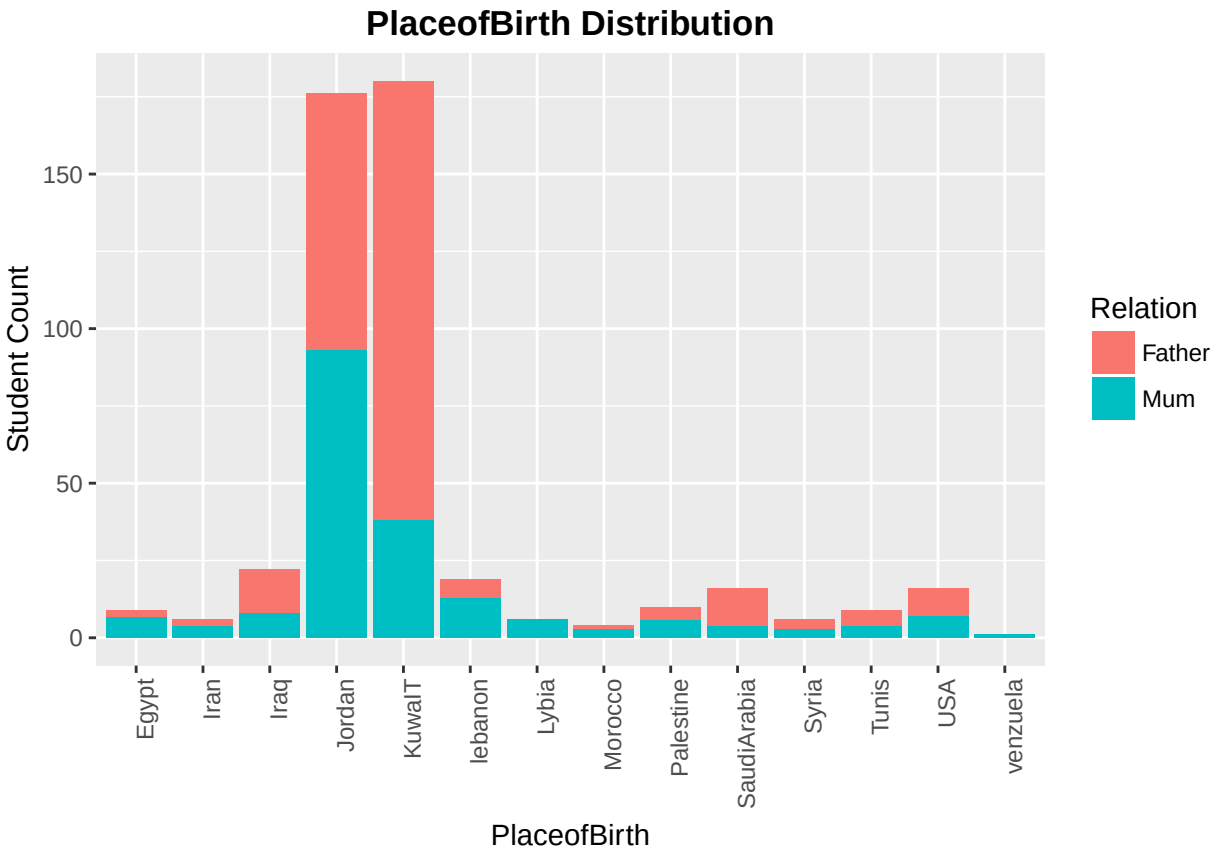
Most of the students who were born in these countries of Egypt, Iran, Iraq, Kuwait, Lebanon, Lybia, Morocco, Syria, Tunis and Venzuela have the same nationality as the place of birth. some of the students who were born in Jordan have the nationality of Palestine. The students who were born in USA have diversity of nationality. Half of the students who were born in Saudiarabia have the nationality of Jordan.

```
ggplot(data = E, aes(x = PlaceofBirth, fill = gender)) + geom_bar() +
    labs(x = "PlaceofBirth", y = "Student Count") + ggtitle(label = "PlaceofBirth Distribution") +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold")) + theme(axis.text.x = element_text(angle = 90,
    hjust = 1))
```



Lebanon has most rate of girls, half of the USA students are girls. Girls are less than boys in the other areas.

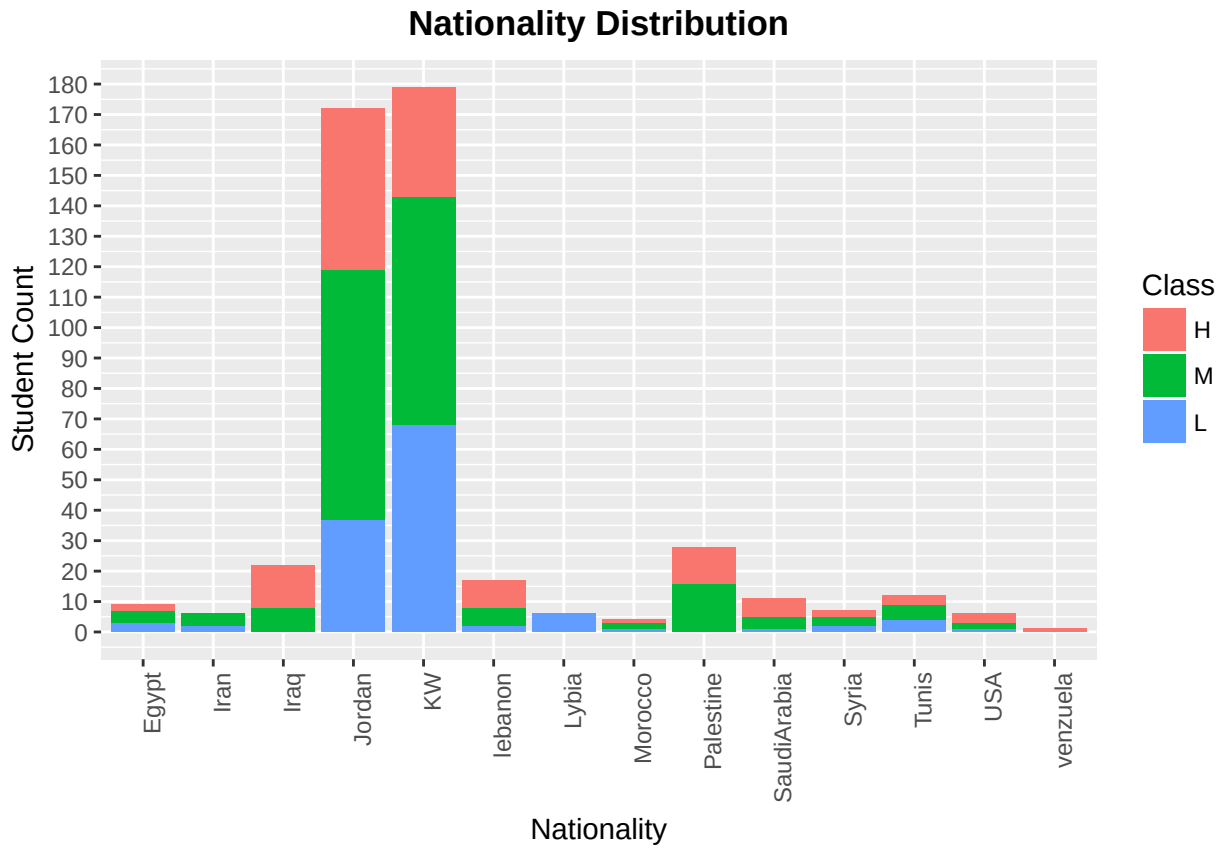
```
ggplot(data = E, aes(x = PlaceofBirth, fill = Relation)) + geom_bar() +
    labs(x = "PlaceofBirth", y = "Student Count") + ggtitle(label = "PlaceofBirth Distribution") +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold")) + theme(axis.text.x = element_text(angle = 90,
    hjust = 1))
```



Based on the place of birth, most fathers are responsible for students in some countries like: Kuwait, Iraq, Saudiarabia. Most mothers are responsible for students' study in these areas like: Egypt, Iran, Jordan, Lebanon, LYbia, Morocco, Palestine and Venzuela.

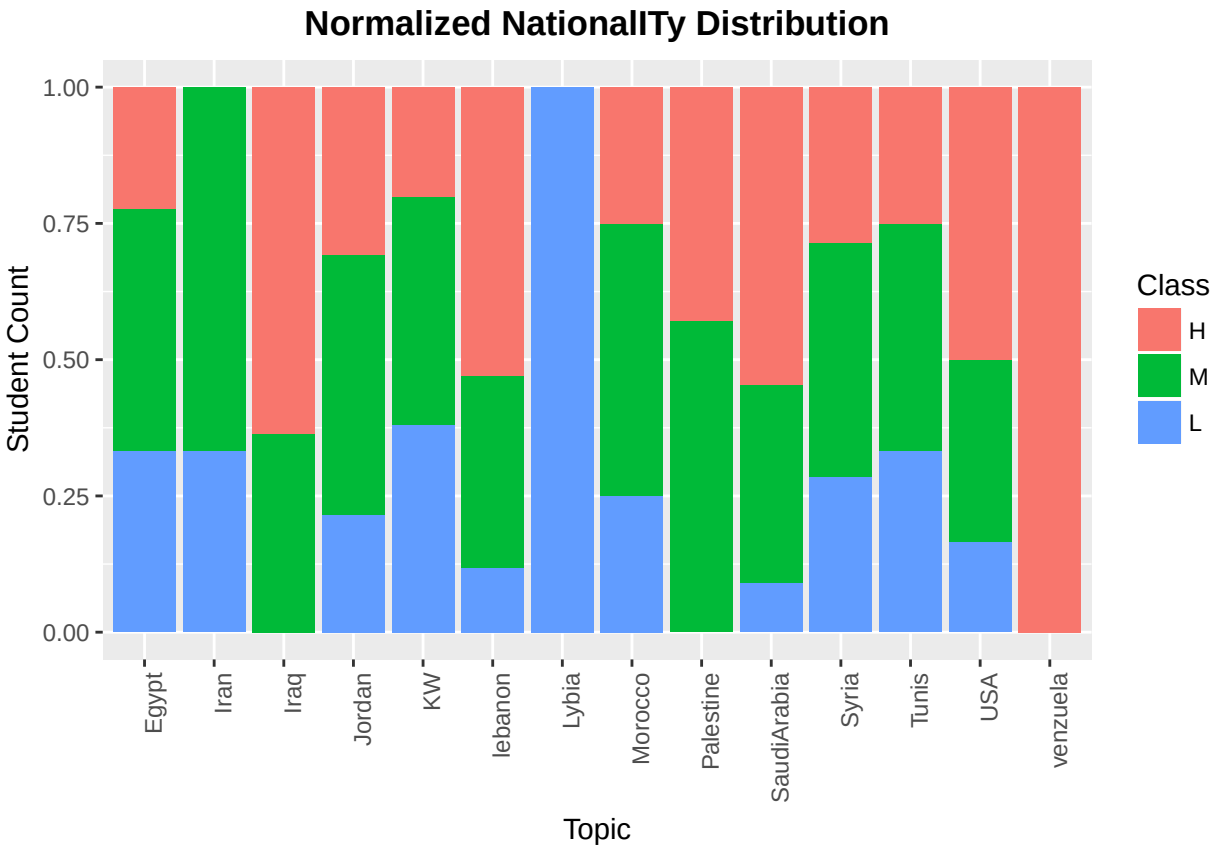
Nationality distribution:

```
ggplot(data = E, aes(x = NationalITy, fill = Class)) + geom_bar() +
    labs(x = "Nationality", y = "Student Count") + ggtitle(label = "Nationality Distribution") +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold")) + theme(axis.text.x = element_text(angle = 90,
    hjust = 1)) + scale_y_continuous(breaks = seq(0, 200, 10))
```



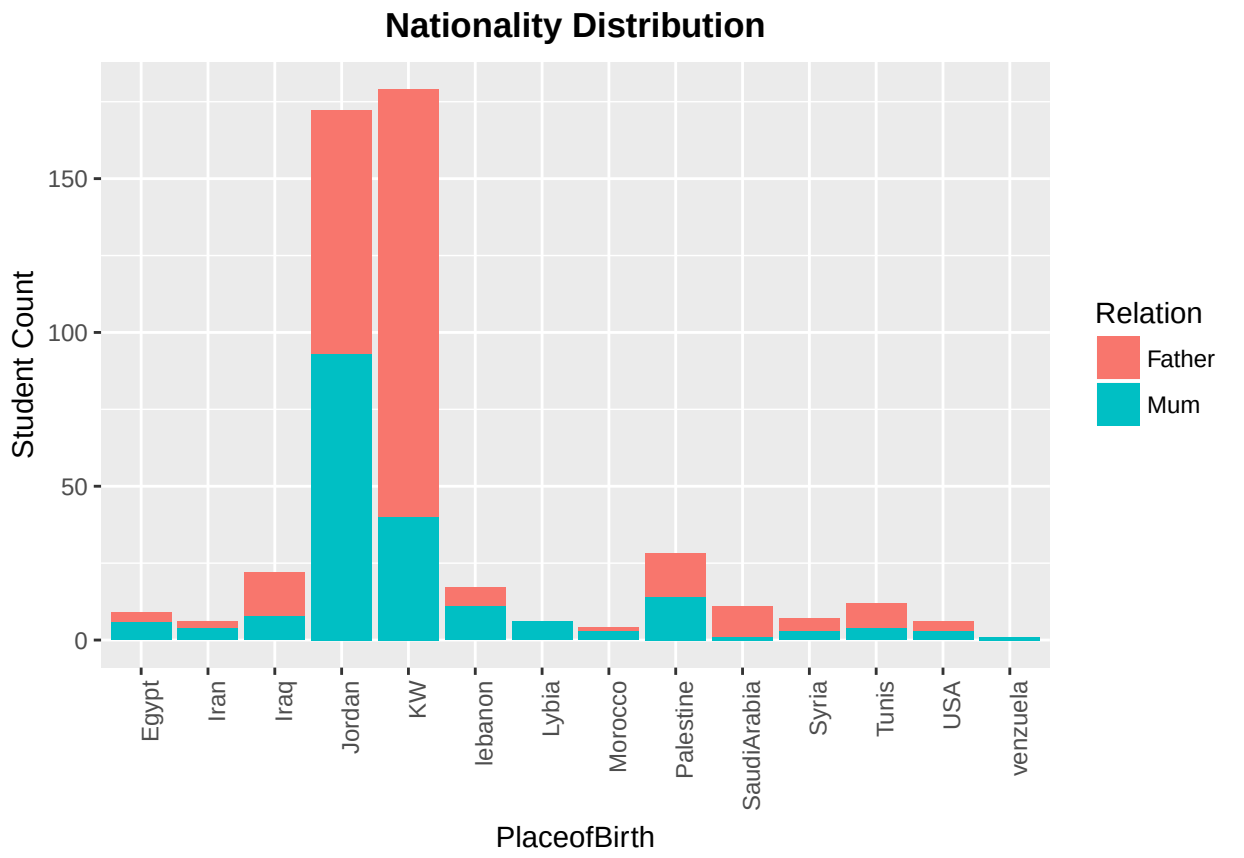
Jordan and KW(KuwaIT) has highest number of students. Venzuela has only one student.

```
ESummary = E %>% # Get the counts
group_by(NationalITy, Class) %>% summarise(count = n()) %>% # Get labels and position of labels
group_by(NationalITy) %>% mutate(Percent = paste0(sprintf("%.1f",
    count/sum(count) * 100), "%"))

ggplot(ESummary, aes(x = NationalITy, y = count)) + geom_bar(aes(fill = Class),
    stat = "identity", position = "fill") + labs(x = "Topic",
    y = "Student Count") + ggtitle(label = "Normalized NationalITy Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold")) + theme(axis.text.x = element_text(angle = 90,
    hjust = 1))
```



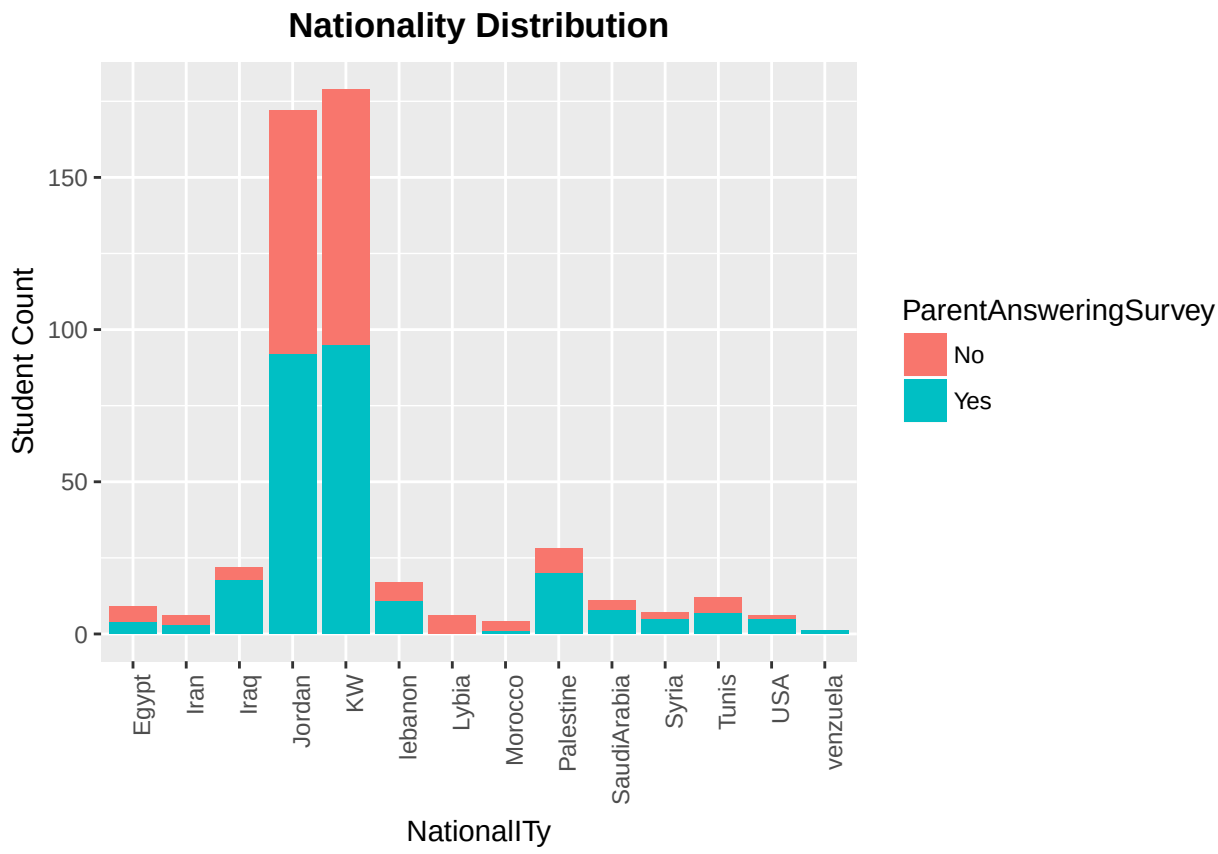
Iraq, Lebanon, Saudiarabia, Palestine, USA and Venzuela have higher rate of student with high class, whereas Kuwait and Lybia have higher rate of students with low class.

```
ggplot(data = E, aes(x = NationalITy, fill = Relation)) + geom_bar() +
    labs(x = "PlaceofBirth", y = "Student Count") + ggtitle(label = "Nationality Distribution") +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold")) + theme(axis.text.x = element_text(angle = 90,
    hjust = 1))
```



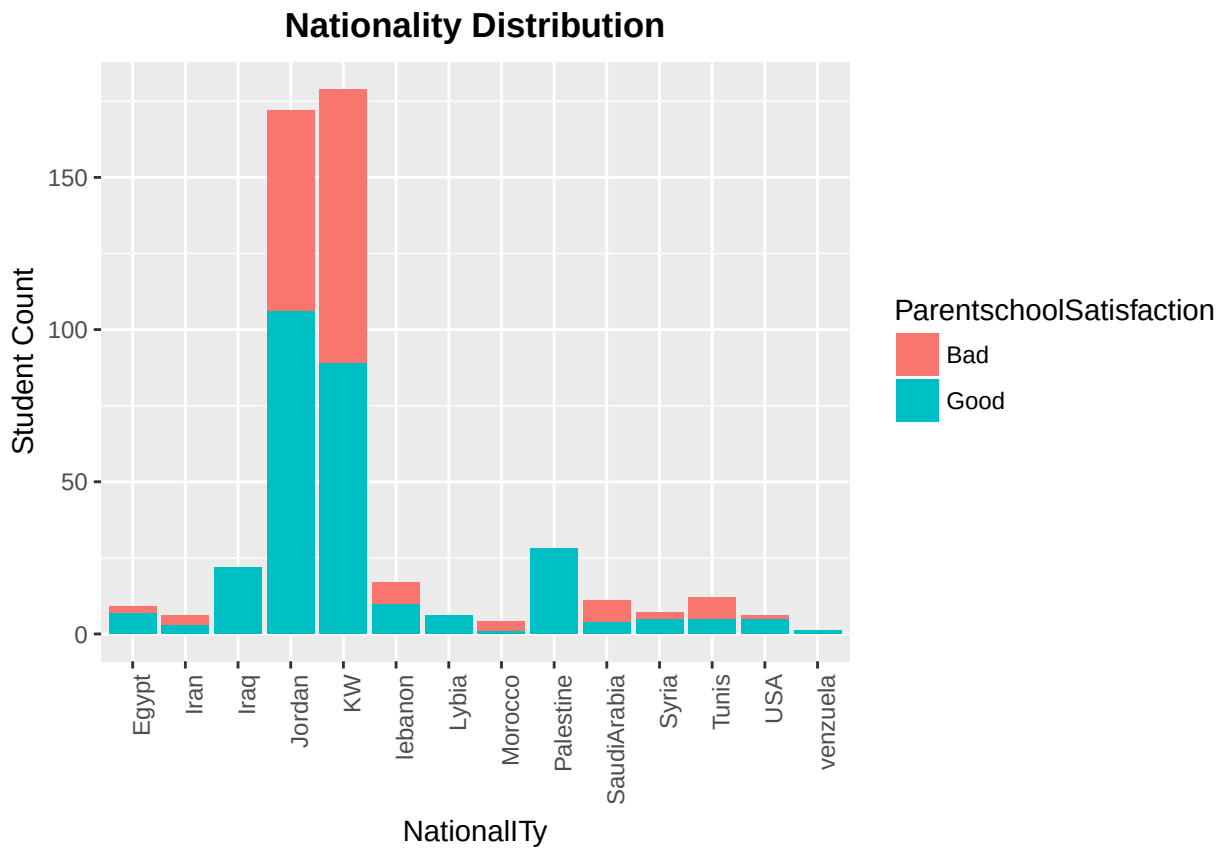
Based on the nationality,most fathers are responsible for students in some countries like: Kuwait, Iraq, Saudiarabia. Most mothers are responsible for students' study in these areas like: Egypt, Iran, Jordan, Lebanon, LYbia, Morocco, Palestine and Venzuela.

```
ggplot(data = E, aes(x = NationalITy, fill = ParentAnsweringSurvey)) +
    geom_bar() + labs(x = "NationalITy", y = "Student Count") +
    ggtitle(label = "Nationality Distribution") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold")) + theme(axis.text.x = element_text(angle = 90,
    hjust = 1))
```



Most of the areas have higher rate of parent answering survey, except Lybia.

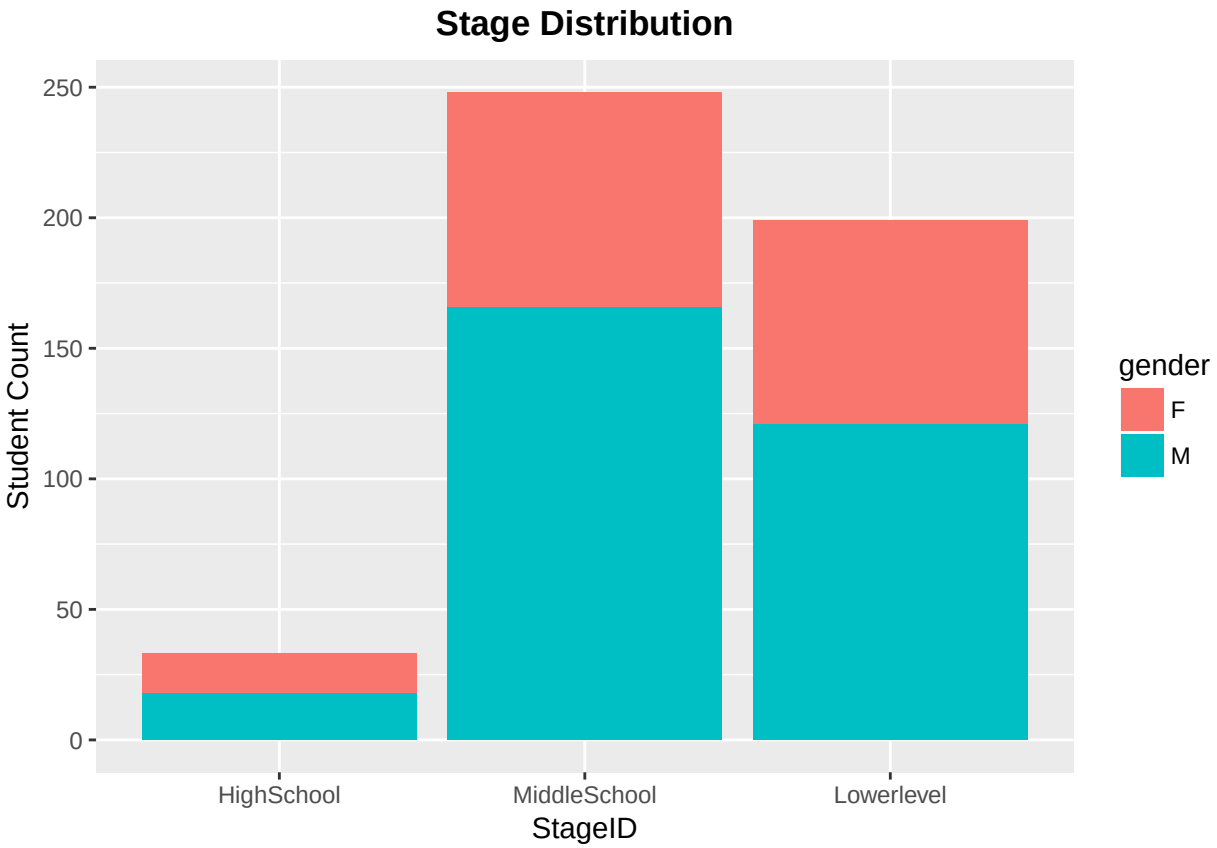
```
ggplot(data = E, aes(x = NationalITy, fill = ParentschoolSatisfaction)) +
    geom_bar() + labs(x = "NationalITy", y = "Student Count") +
    ggtitle(label = "Nationality Distribution") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold")) + theme(axis.text.x = element_text(angle = 90,
    hjust = 1))
```



Most of the areas have higher rate of good parent school satisfaction, except Suadiarabia.

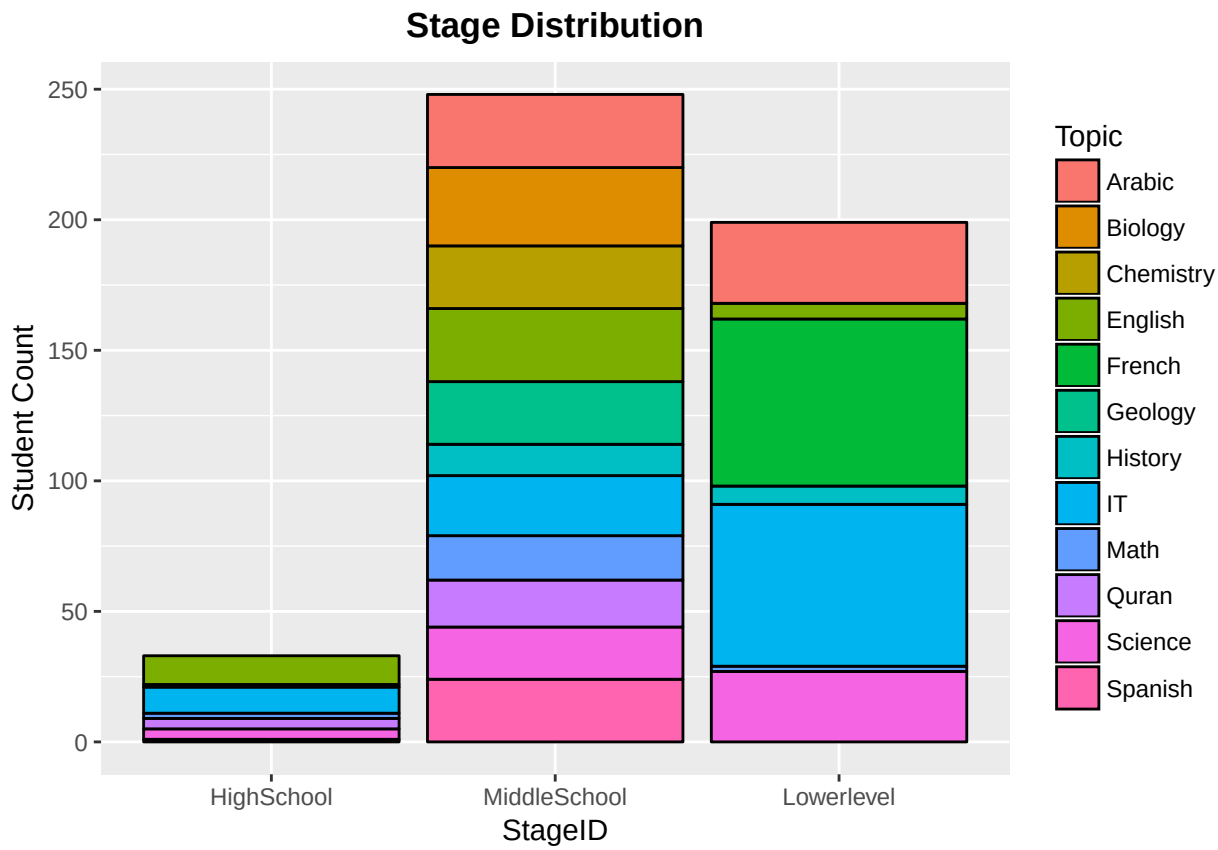
Stage ID distribution:

```
ggplot(data = E, aes(x = StageID, fill = gender)) + geom_bar() +
    labs(x = "StageID", y = "Student Count") + ggtitle(label = "Stage Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



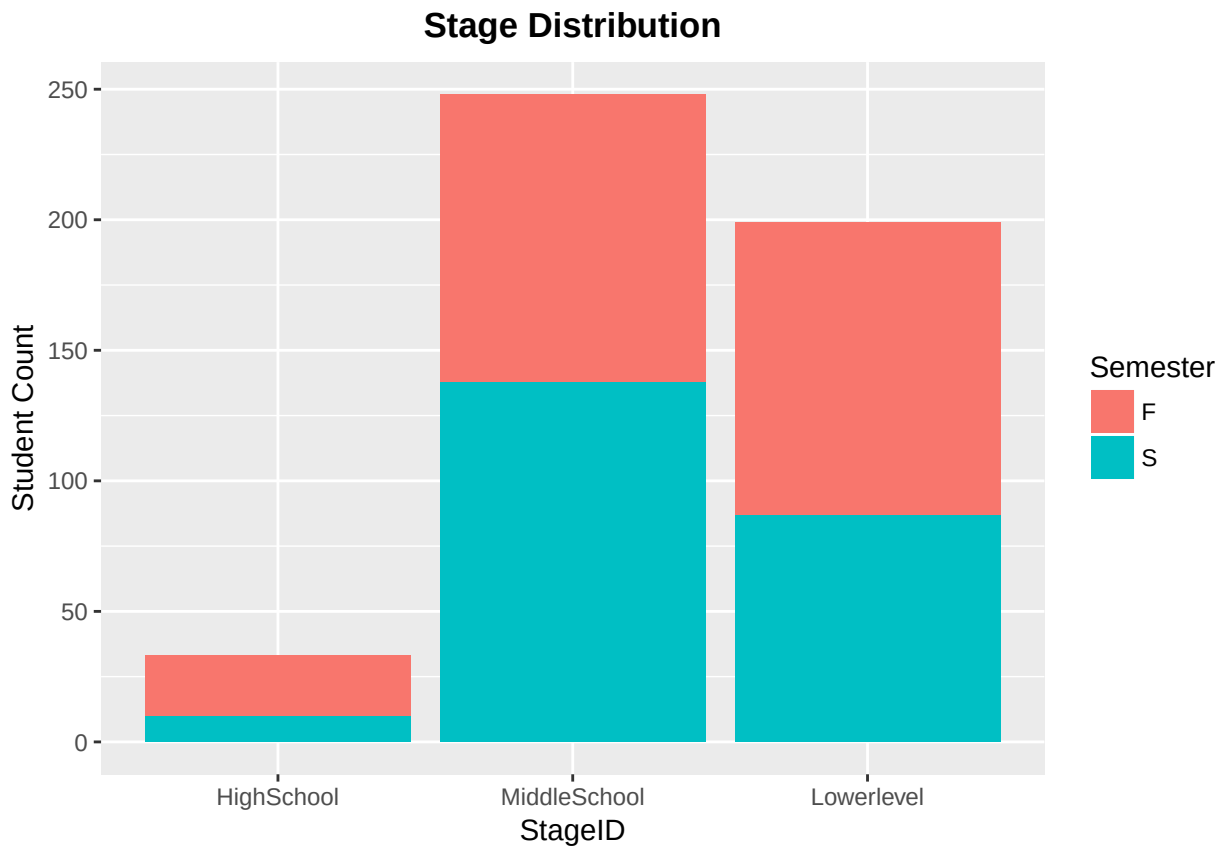
Middle school has the highest number of students. High school has very few students. Girls are less than boys in every level of school.

```
ggplot(data = E, aes(x = StageID, fill = Topic)) + geom_bar(colour = "black") +
    labs(x = "StageID", y = "Student Count") + ggtitle(label = "Stage Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



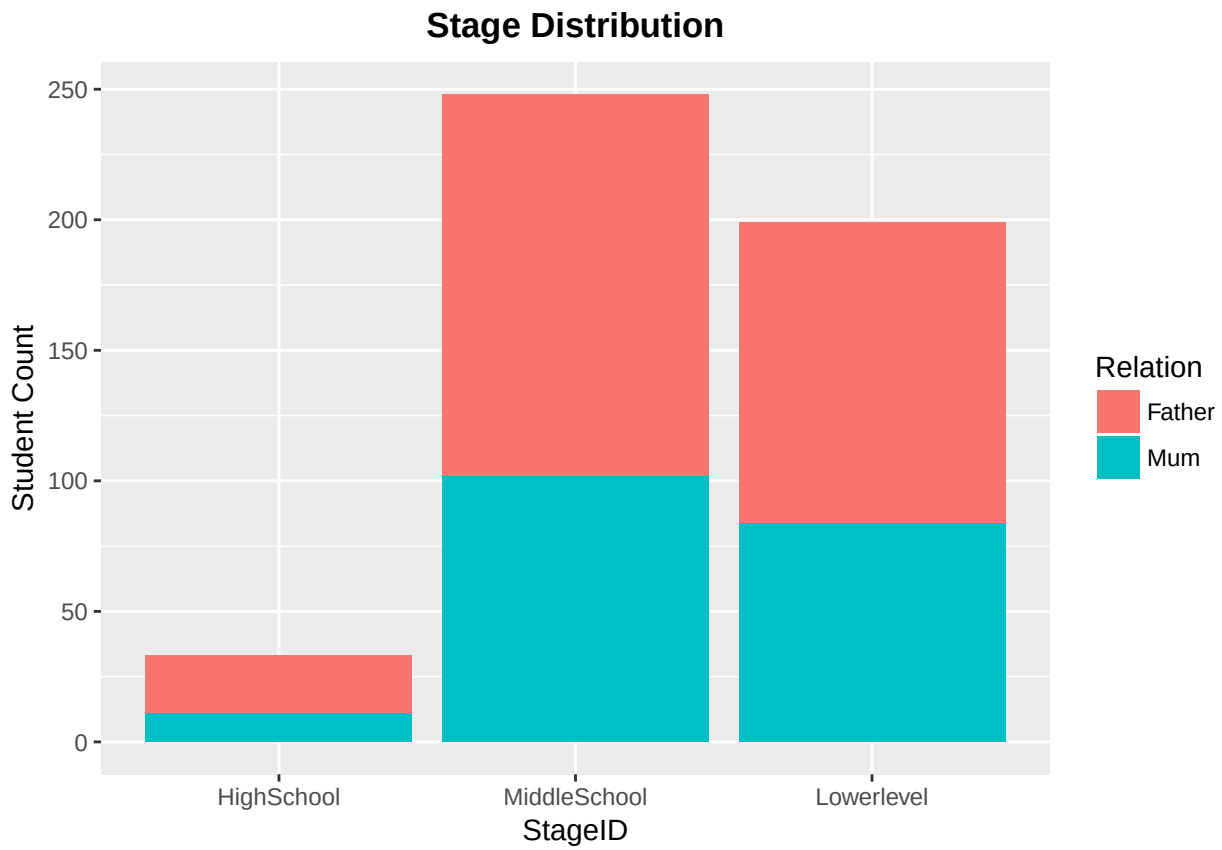
Lowerlevel has less kinds of topics. Middle school almost covers all kinds of topics.

```
ggplot(data = E, aes(x = StageID, fill = Semester)) + geom_bar() +
    labs(x = "StageID", y = "Student Count") + ggtitle(label = "Stage Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



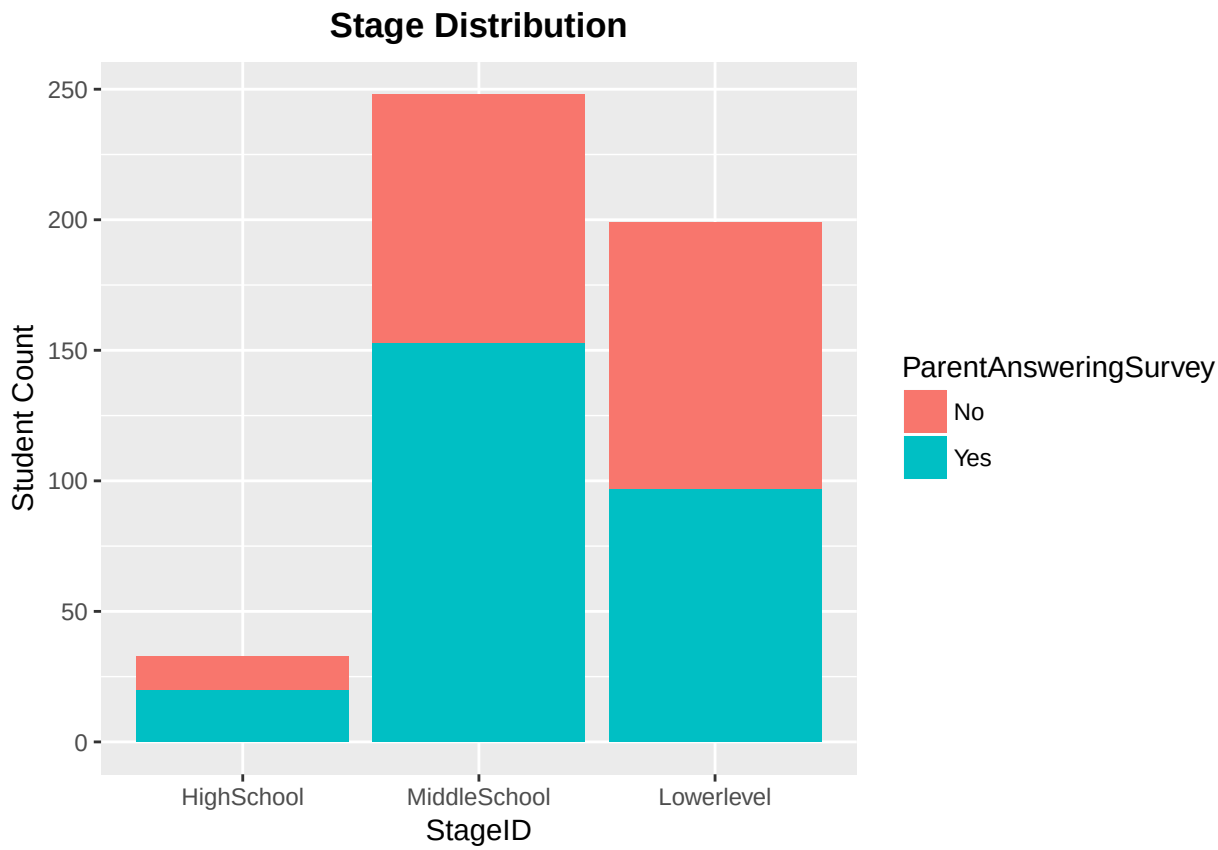
The two semesters almost cover the same rate in different level of schools.

```
ggplot(data = E, aes(x = StageID, fill = Relation)) + geom_bar() +
    labs(x = "StageID", y = "Student Count") + ggtitle(label = "Stage Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```

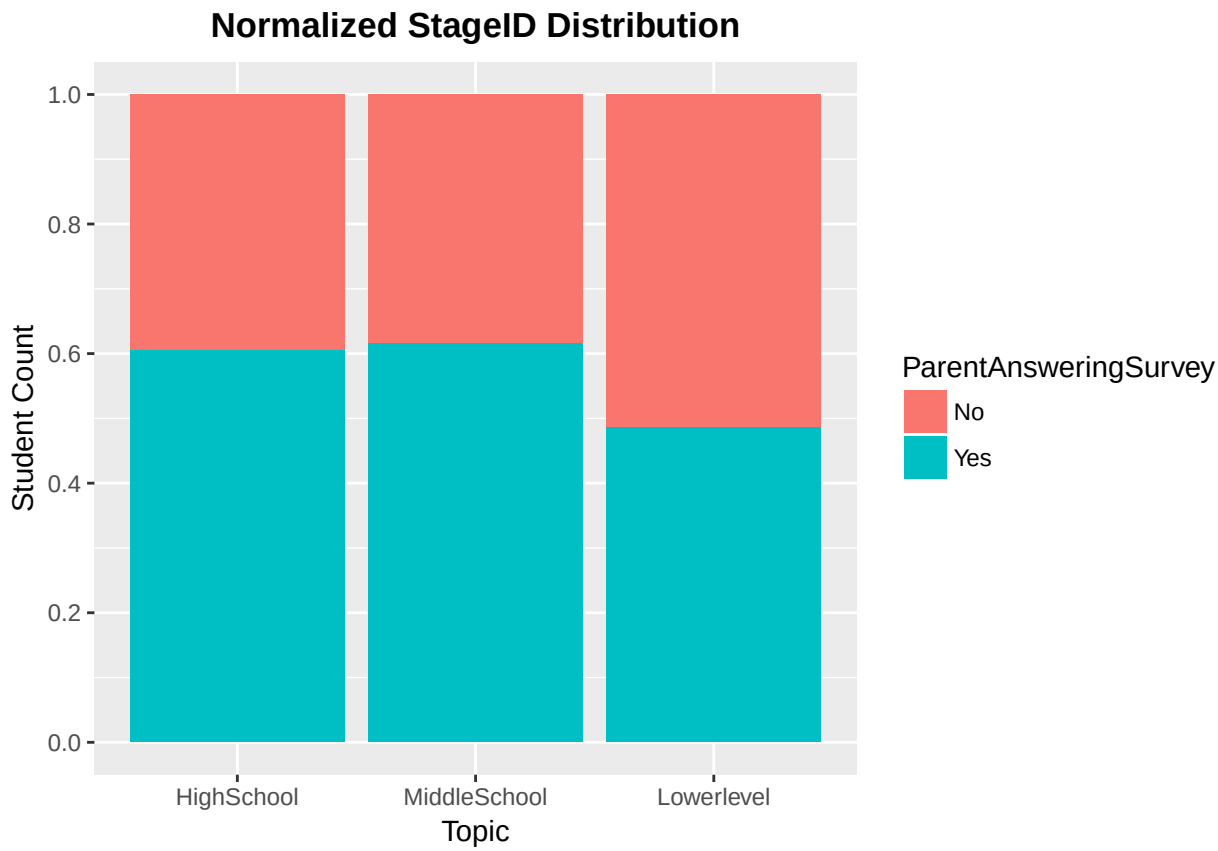


More fatheres are responsible for students' study in different level of schools.

```
ggplot(data = E, aes(x = StageID, fill = ParentAnsweringSurvey)) +
    geom_bar() + labs(x = "StageID", y = "Student Count") + ggtitle(label = "Stage Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```

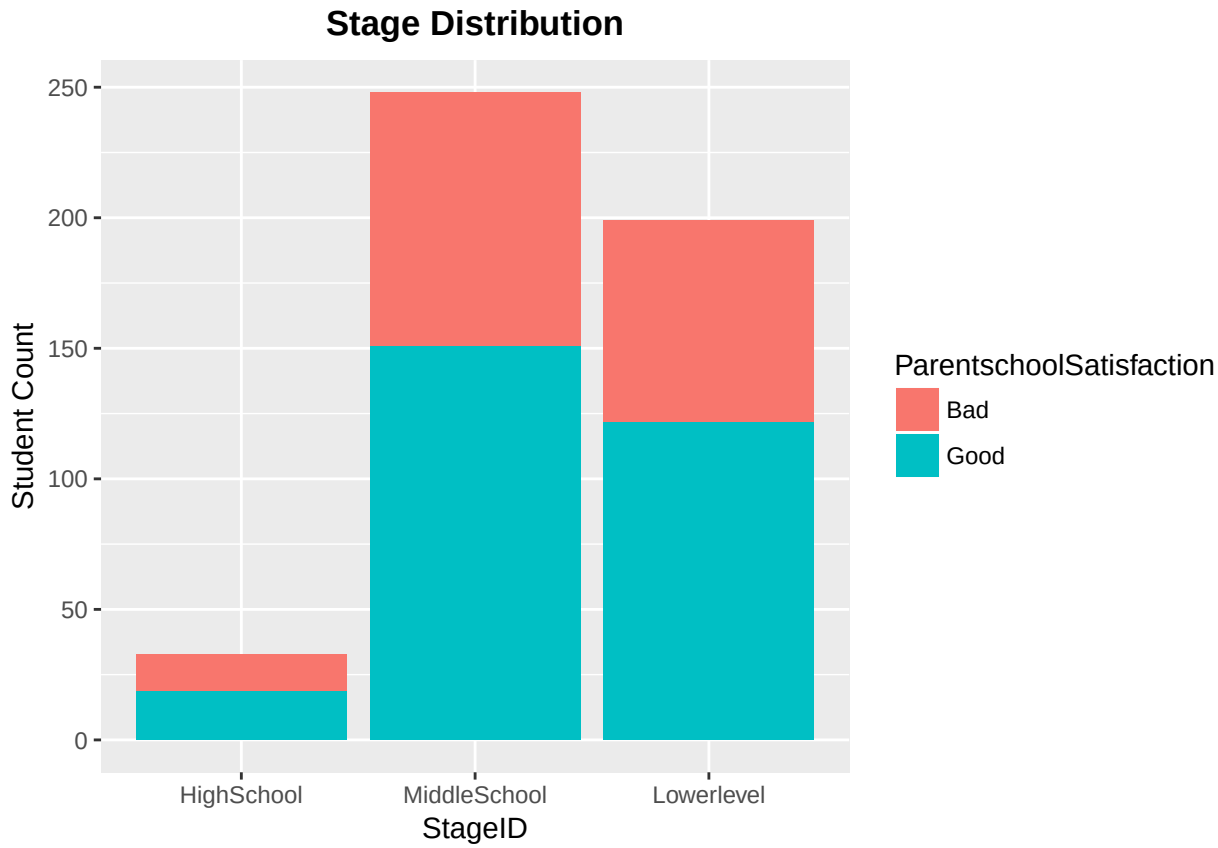



```
ESummary = E %>% # Get the counts
group_by(StageID, ParentAnsweringSurvey) %>% summarise(count = n()) %>%
    # Get labels and position of labels
group_by(StageID) %>% mutate(Percent = paste0(sprintf("%.1f",
    count/sum(count) * 100), "%"))

ggplot(ESummary, aes(x = StageID, y = count)) + geom_bar(aes(fill = ParentAnsweringSurvey),
    stat = "identity", position = "fill") + labs(x = "Topic",
    y = "Student Count") + ggtitle(label = "Normalized StageID Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold")) + scale_y_continuous(breaks = seq(0,
    1, 0.2))
```

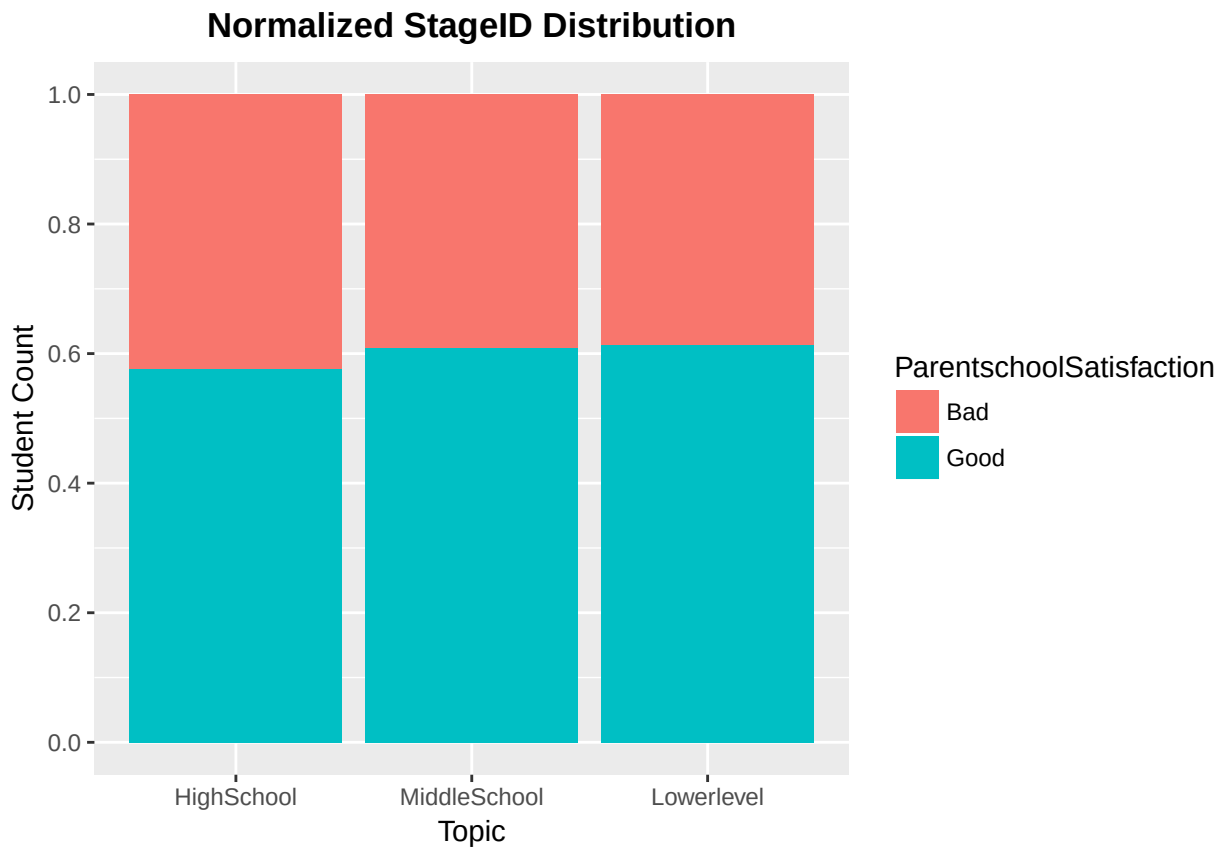


Both highschool and middleschool have more than 60% parent answering survey. Lowerlevel has less than 50% rate of parent answering survey.

```
ggplot(data = E, aes(x = StageID, fill = ParentschoolSatisfaction)) +
    geom_bar() + labs(x = "StageID", y = "Student Count") + ggtitle(label = "Stage Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```

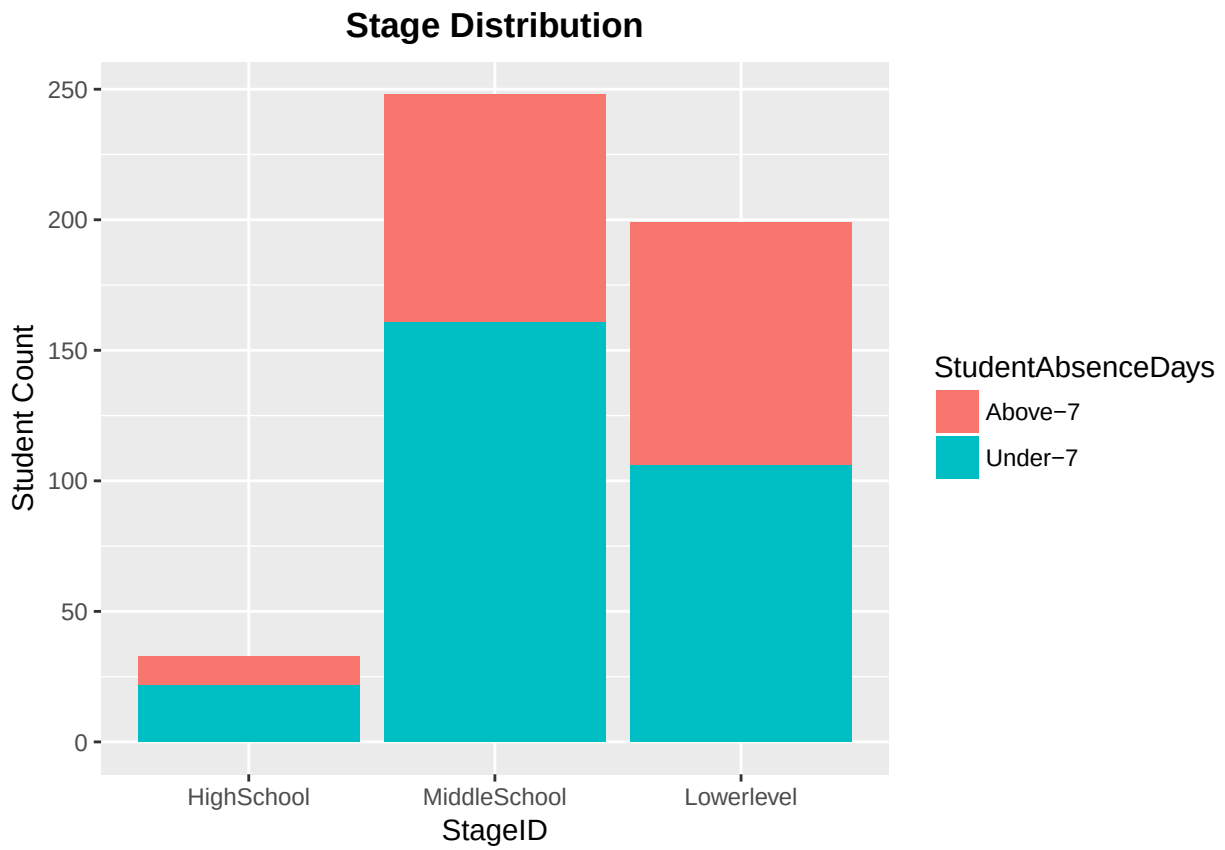



```
ESummary = E %>% # Get the counts
group_by(StageID, ParentschoolSatisfaction) %>% summarise(count = n()) %>%
    # Get labels and position of labels
group_by(StageID) %>% mutate(Percent = paste0(sprintf("%.1f",
    count/sum(count) * 100), "%"))

ggplot(ESummary, aes(x = StageID, y = count)) + geom_bar(aes(fill = ParentschoolSatisfaction),
    stat = "identity", position = "fill") + labs(x = "Topic",
    y = "Student Count") + ggtitle(label = "Normalized StageID Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold")) + scale_y_continuous(breaks = seq(0,
    1, 0.2))
```

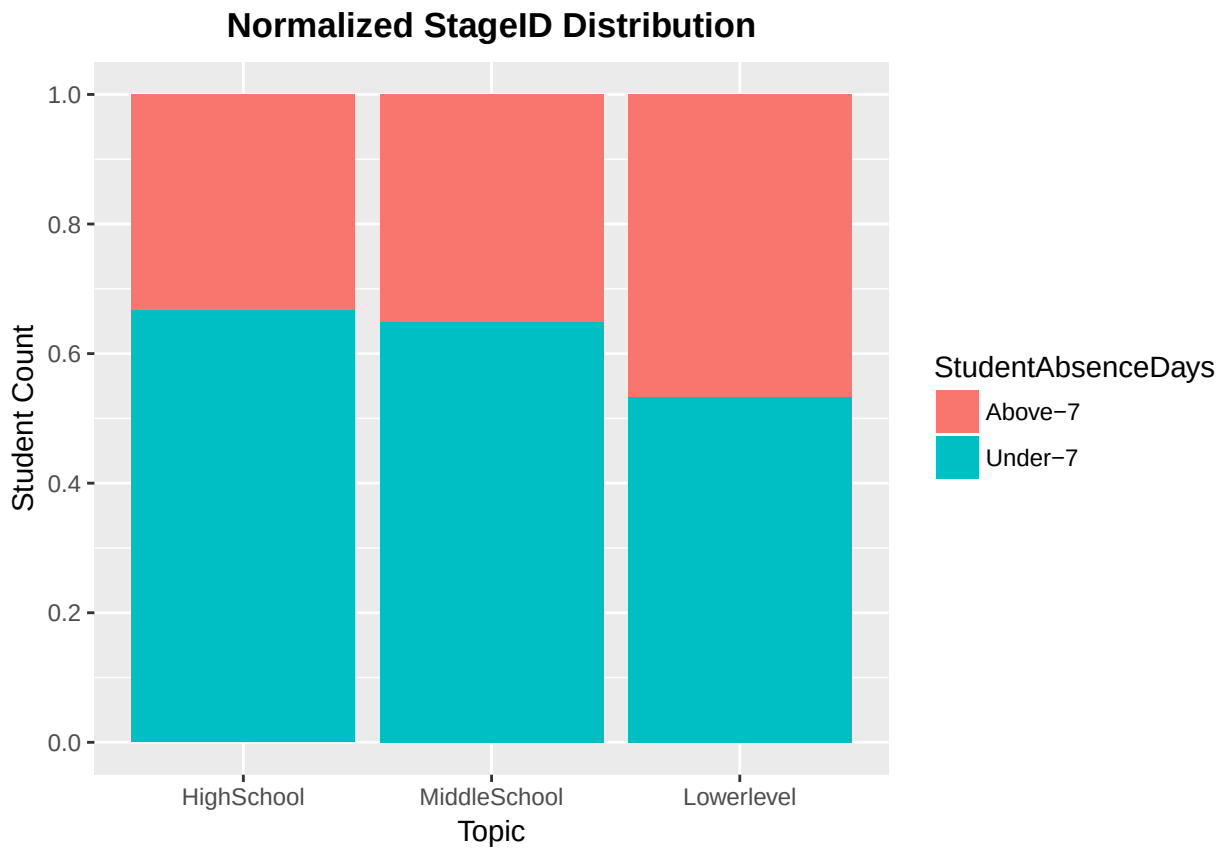


Highschool has the lowest rate(under 60%) of good parent school satisfaction. Both middleschool and lowerlevel have more than 60% rate of good parent school satisfaction.

```
ggplot(data = E, aes(x = StageID, fill = StudentAbsenceDays)) +
    geom_bar() + labs(x = "StageID", y = "Student Count") + ggtitle(label = "Stage Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```

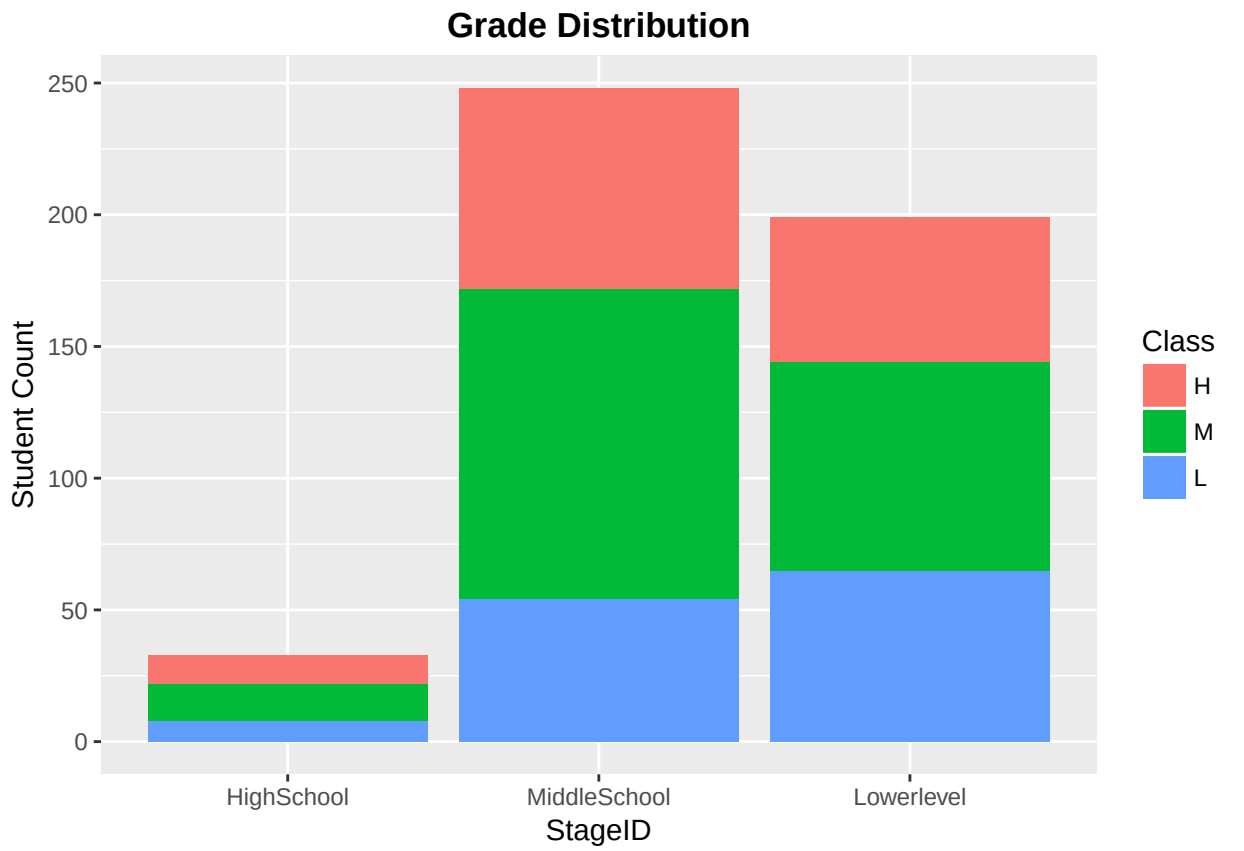



```
ESummary = E %>% # Get the counts
group_by(StageID, StudentAbsenceDays) %>% summarise(count = n()) %>%
    # Get labels and position of labels
group_by(StageID) %>% mutate(Percent = paste0(sprintf("%.1f",
    count/sum(count) * 100), "%"))

ggplot(ESummary, aes(x = StageID, y = count)) + geom_bar(aes(fill = StudentAbsenceDays),
    stat = "identity", position = "fill") + labs(x = "Topic",
    y = "Student Count") + ggtitle(label = "Normalized StageID Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold")) + scale_y_continuous(breaks = seq(0,
    1, 0.2))
```

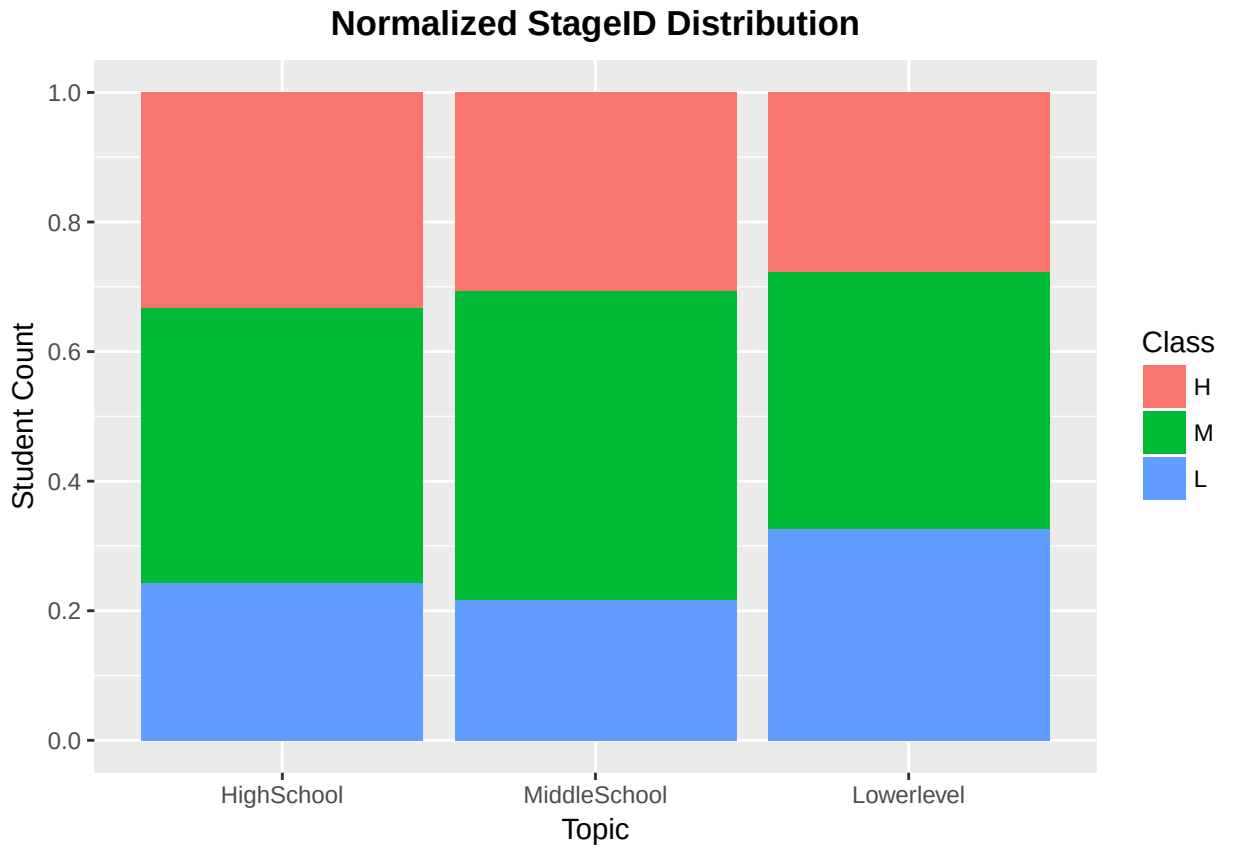


Lowerlevel has higher rate of above 7 days of student absence.

```
ggplot(data = E, aes(x = StageID, fill = Class)) + geom_bar() +
    labs(x = "StageID", y = "Student Count") + ggtitle(label = "Grade Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



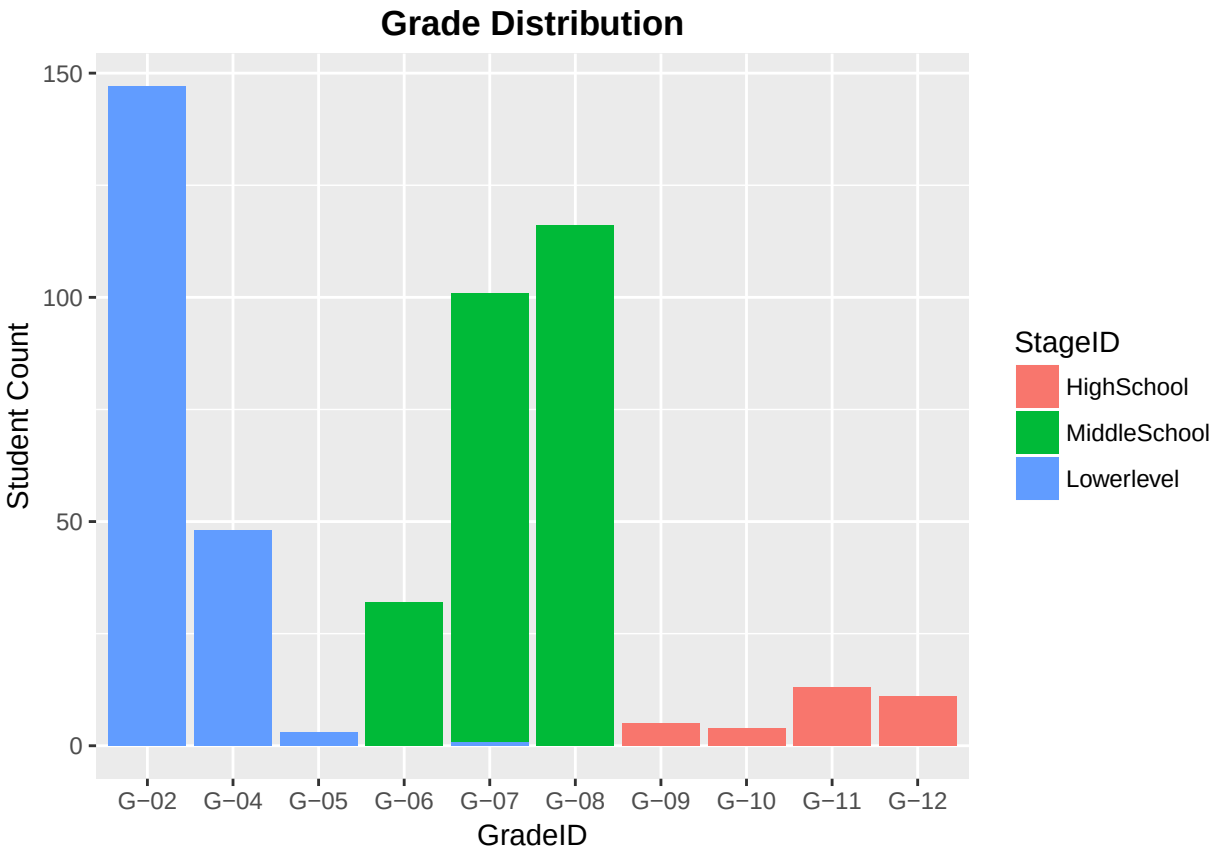

```
ESummary = E %>% # Get the counts
group_by(StageID, Class) %>% summarise(count = n()) %>% # Get labels and position of labels
group_by(StageID) %>% mutate(Percent = paste0(sprintf("%.1f",
    count/sum(count) * 100), "%"))

ggplot(ESummary, aes(x = StageID, y = count)) + geom_bar(aes(fill = Class),
    stat = "identity", position = "fill") + labs(x = "Topic",
    y = "Student Count") + ggtitle(label = "Normalized StageID Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold")) + scale_y_continuous(breaks = seq(0,
    1, 0.2))
```



Lower level has the highest rate of low class students.Highschool has the highest rate of high class students. Middle school has the highest rate of medium class students.

Grade ID distribution:

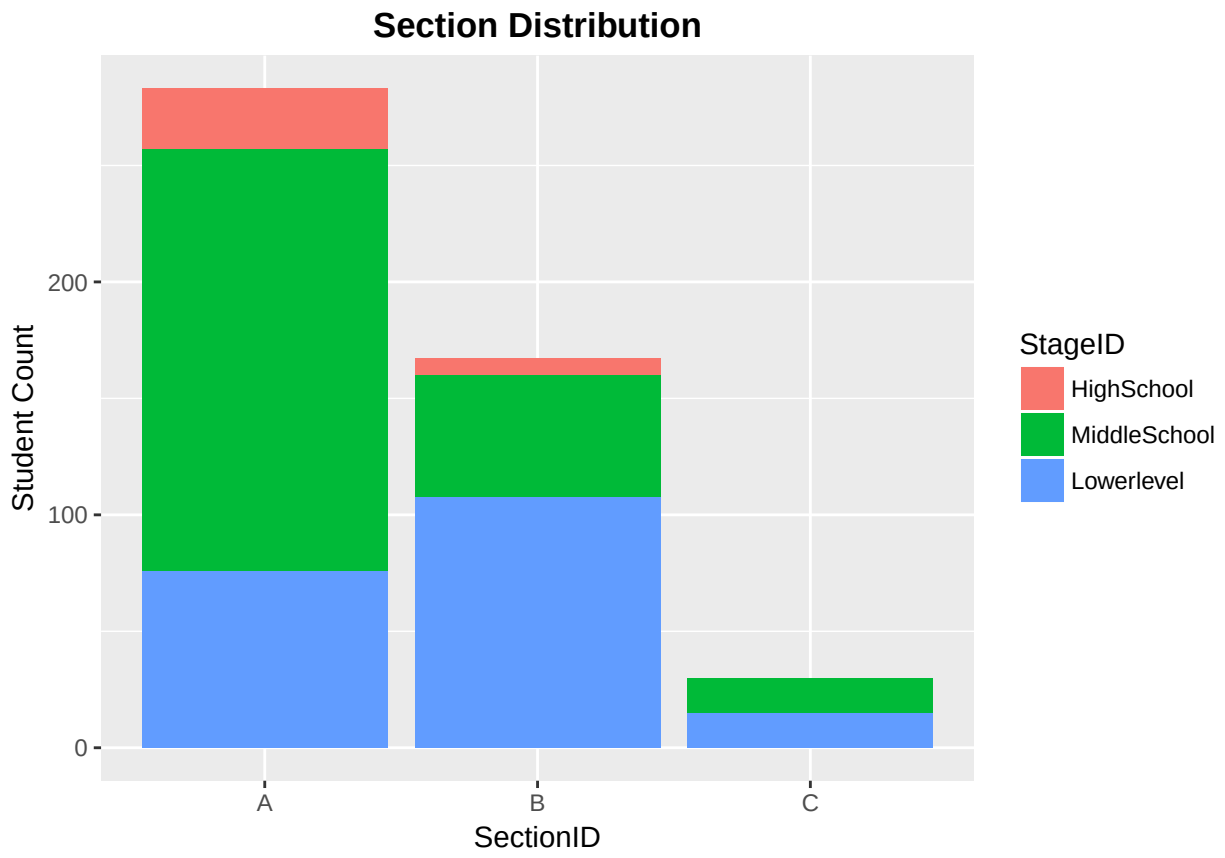
```
ggplot(data = E, aes(x = GradeID, fill = StageID)) + geom_bar() +
    labs(x = "GradeID", y = "Student Count") + theme(axis.text.x = element_text(angle = 90,
    hjust = 1)) + ggtitle(label = "Grade Distribution") + theme_grey() +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold"))
```



Grade 2, 3, 4 and 5 belong to Lowerlevel, grade 6,7 and 8 belong to Middle school, grade 9,10,11 and 12 belong to High school.

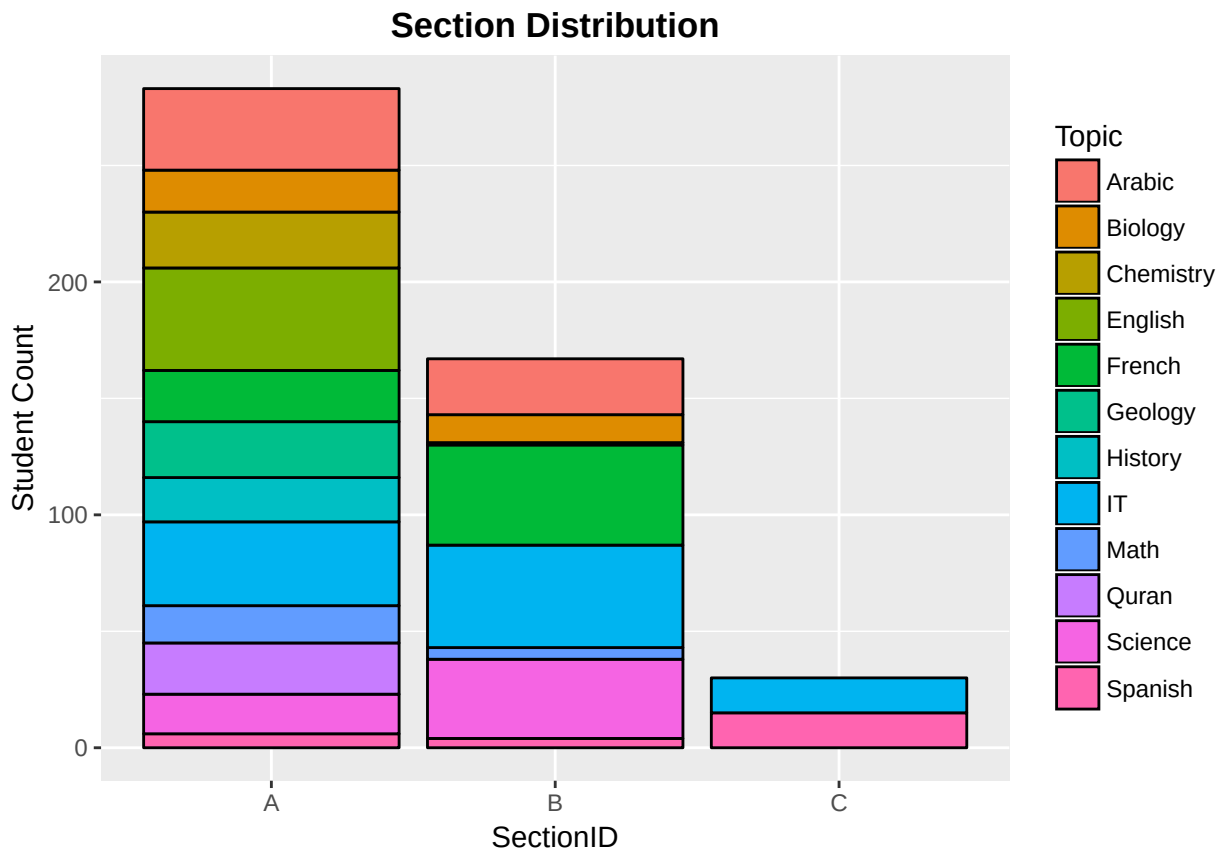
Section ID distribution:

```
ggplot(data = E, aes(x = SectionID, fill = StageID)) + geom_bar() +
    labs(x = "SectionID", y = "Student Count") + ggtitle(label = "Section Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



Section C has only Middle school and Lowerlevel students

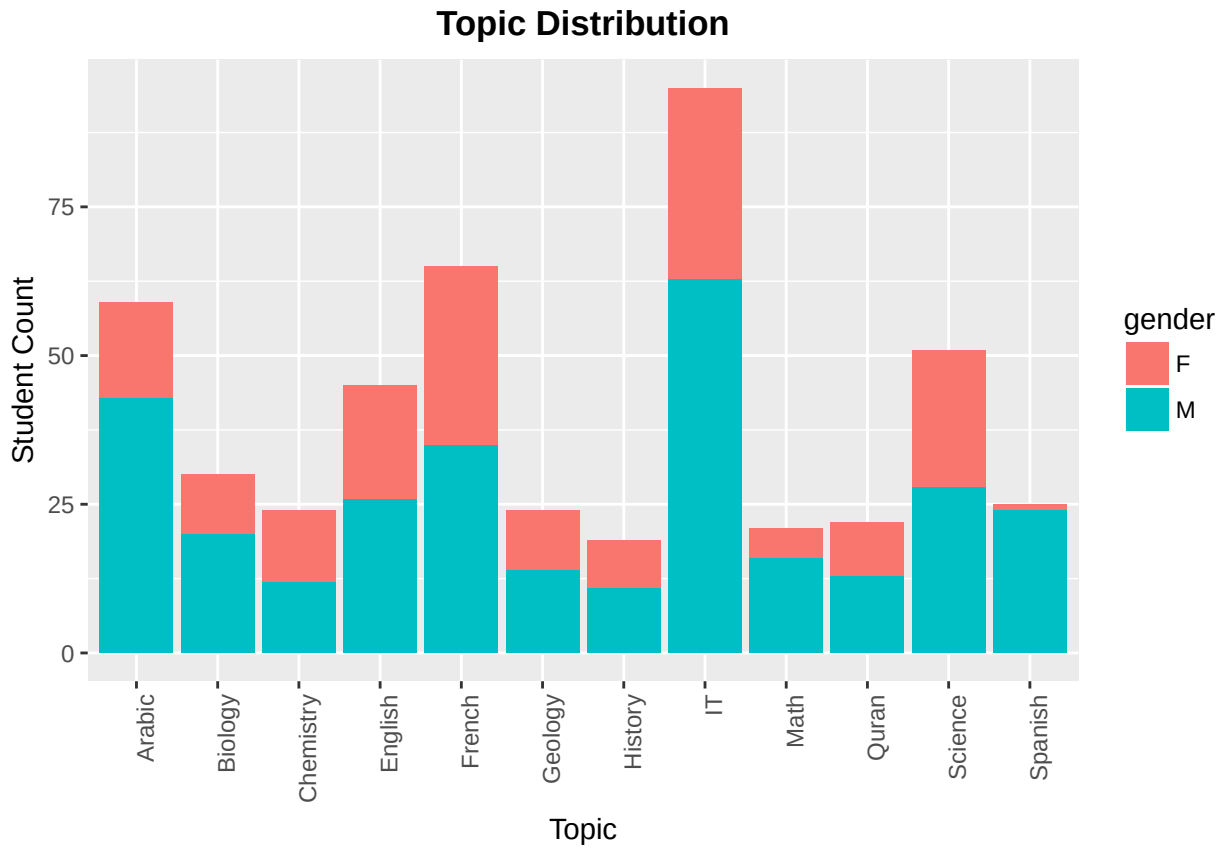
```
ggplot(data = E, aes(x = SectionID, fill = Topic)) + geom_bar(colour = "black") +
    labs(x = "SectionID", y = "Student Count") + ggtitle(label = "Section Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



Section C has only IT and science students.

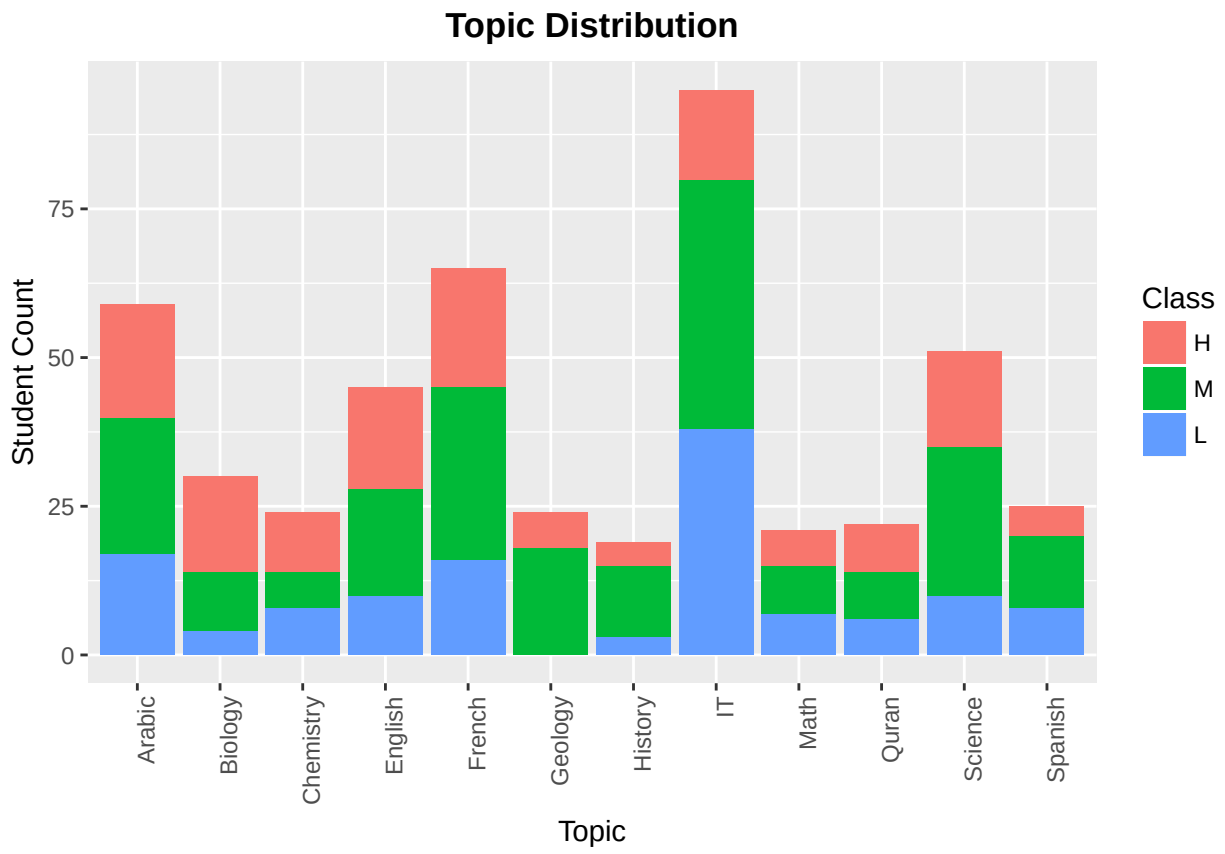
Topic distribution:

```
ggplot(data = E, aes(x = Topic, fill = gender)) + geom_bar() +
    labs(x = "Topic", y = "Student Count") + ggtitle(label = "Topic Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold")) + theme(axis.text.x = element_text(angle = 90,
    hjust = 1))
```



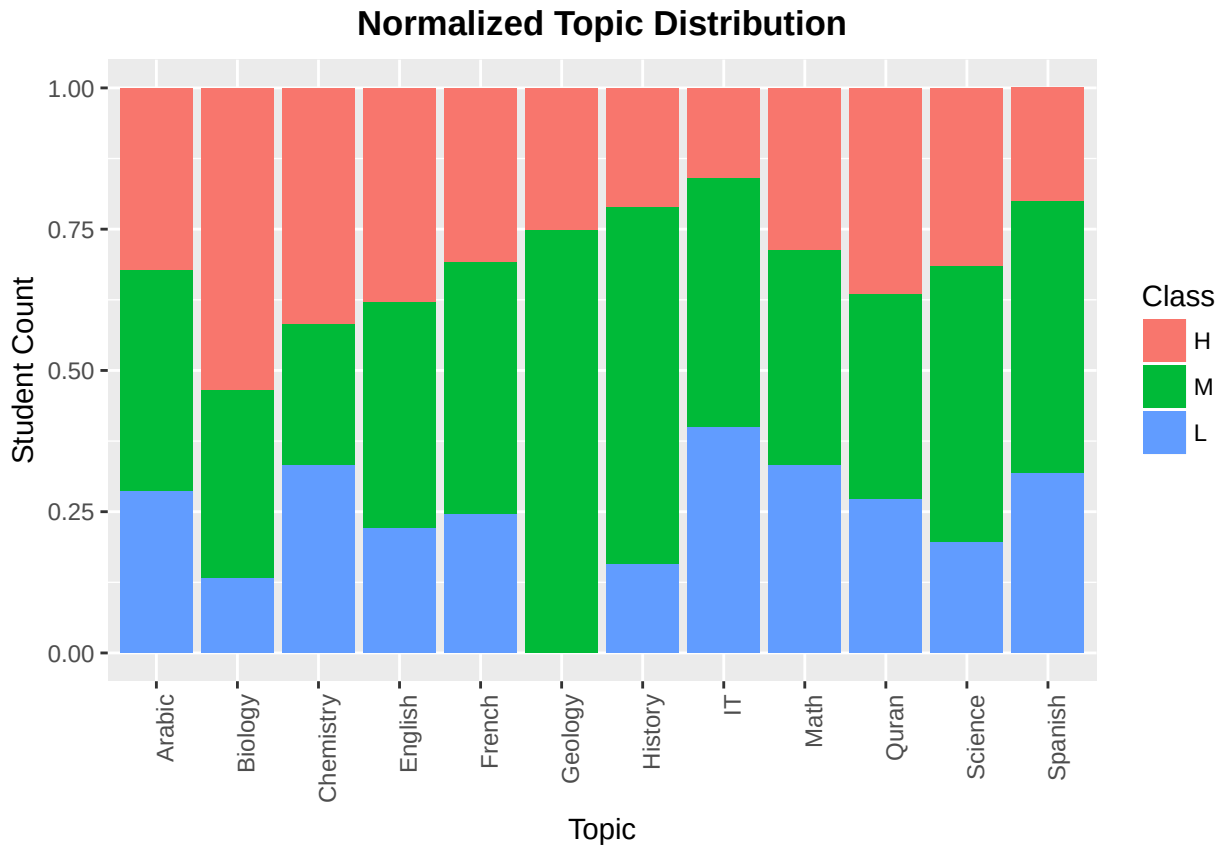
Spanish has fewest girls, whereas Science, Chemistry, English and French have higher ratio of girls.

```
ggplot(data = E, aes(x = Topic, fill = Class)) + geom_bar() +
    labs(x = "Topic", y = "Student Count") + ggtitle(label = "Topic Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold")) + theme(axis.text.x = element_text(angle = 90,
    hjust = 1))
```



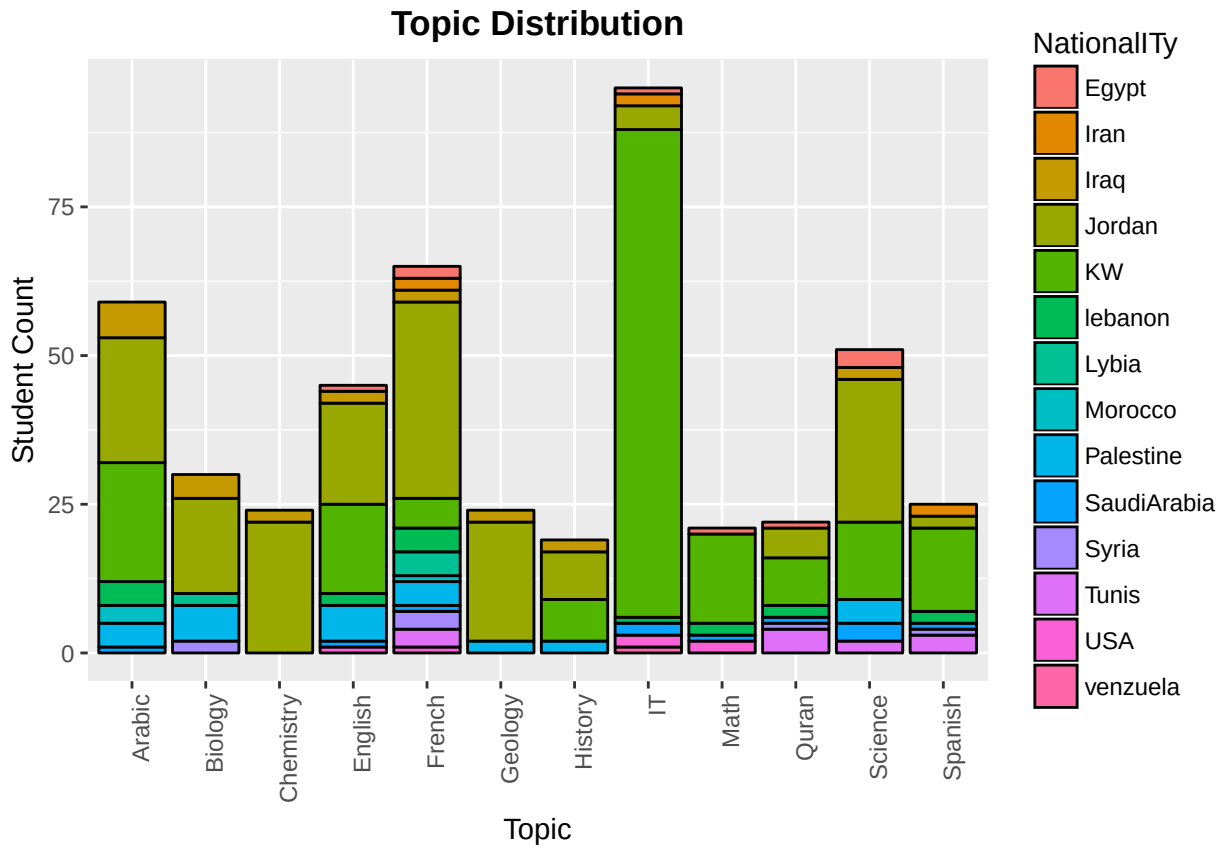
Geology has no low class students

```
ESummary = E %>% # Get the counts
group_by(Topic, Class) %>% summarise(count = n()) %>% # Get labels and position of labels
group_by(Topic) %>% mutate(Percent = paste0(sprintf("%.1f", count/sum(count) *
    100), "%"))

ggplot(ESummary, aes(x = Topic, y = count)) + geom_bar(aes(fill = Class),
    stat = "identity", position = "fill") + labs(x = "Topic",
    y = "Student Count") + ggtitle(label = "Normalized Topic Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold")) + theme(axis.text.x = element_text(angle = 90,
    hjust = 1))
```



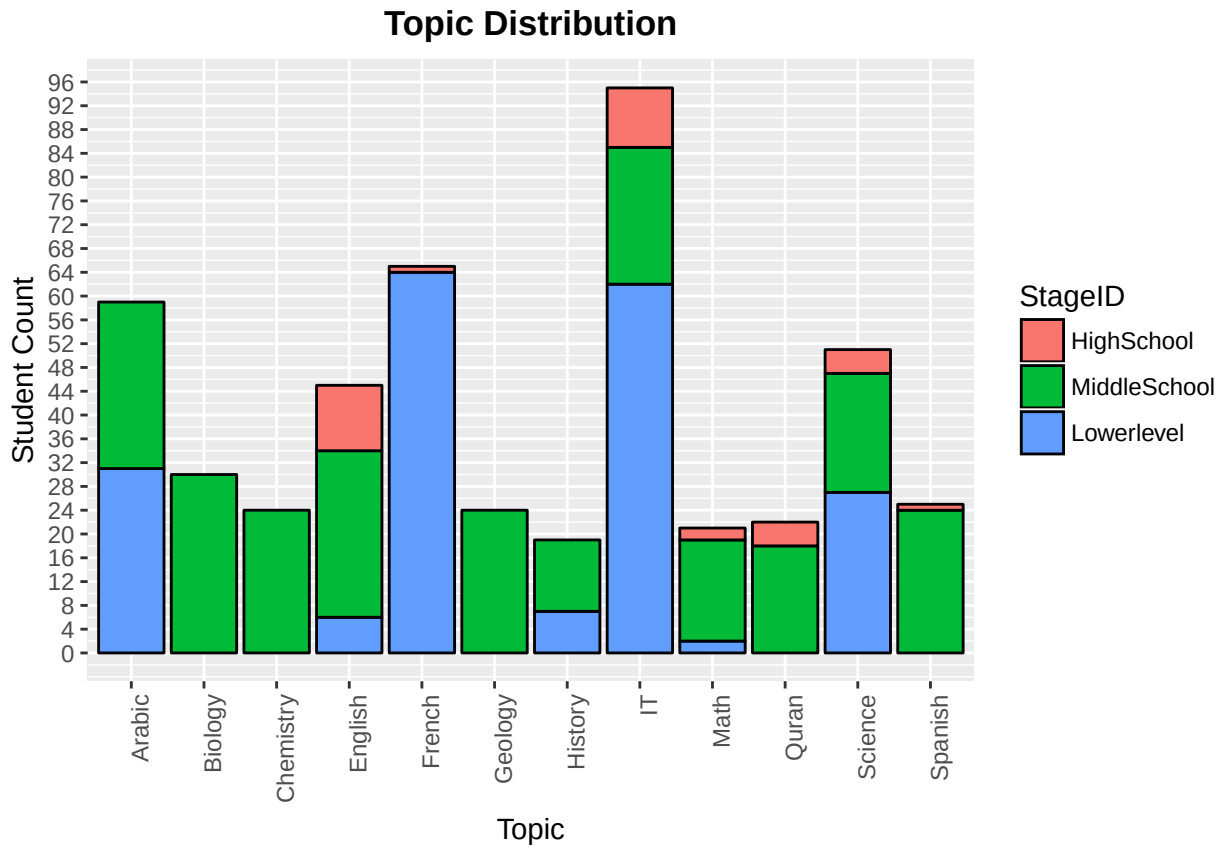
Biology has the highest ratio of high class students. It has the highest ratio of low class students. Geology has the highest ratio of medium class students.

```
ggplot(data = E, aes(x = Topic, fill = NationalITy)) + geom_bar(colour = "black") +
    labs(x = "Topic", y = "Student Count") + ggtitle(label = "Topic Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold")) + theme(axis.text.x = element_text(angle = 90,
    hjust = 1))
```



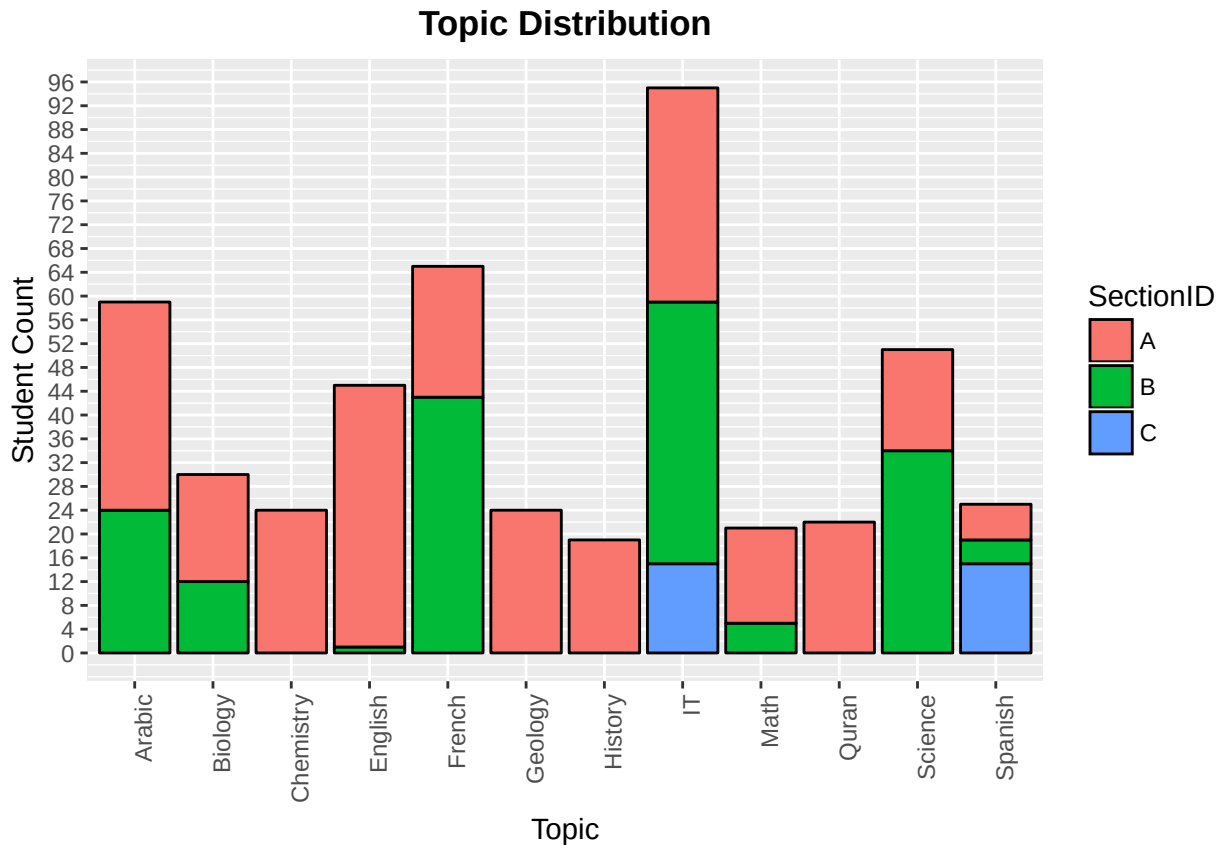
IT has most people from Kuwait. Chemistry has least diversity. French has most diversity.

```
ggplot(data = E, aes(x = Topic, fill = StageID)) + geom_bar(colour = "black") +
    labs(x = "Topic", y = "Student Count") + ggtitle(label = "Topic Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold")) + theme(axis.text.x = element_text(angle = 90,
    hjust = 1)) + scale_y_continuous(breaks = seq(0, 100, 4))
```



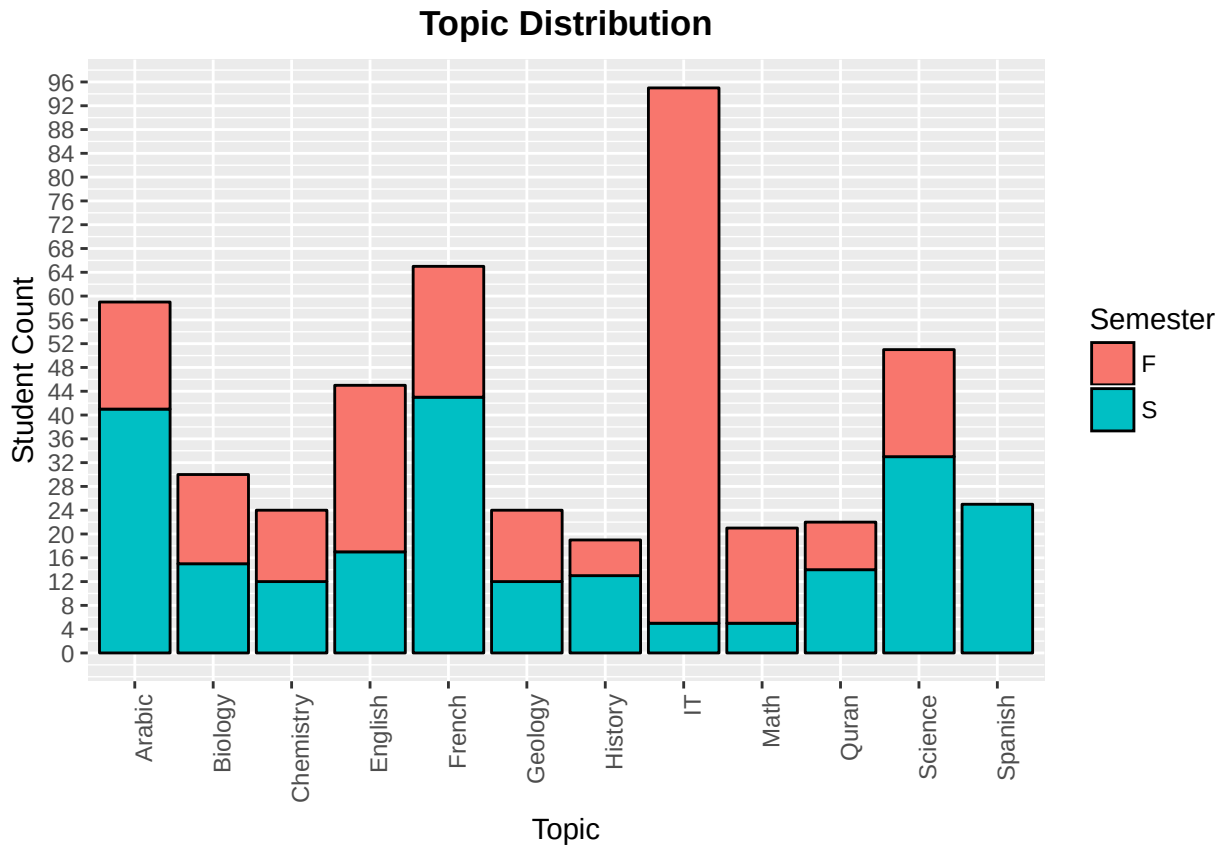
Geology, biology and chemistry are only in middle school. French has the lowest rate of highschool student.

```
ggplot(data = E, aes(x = Topic, fill = SectionID)) + geom_bar(colour = "black") +
    labs(x = "Topic", y = "Student Count") + ggtitle(label = "Topic Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold")) + theme(axis.text.x = element_text(angle = 90,
    hjust = 1)) + scale_y_continuous(breaks = seq(0, 100, 4))
```



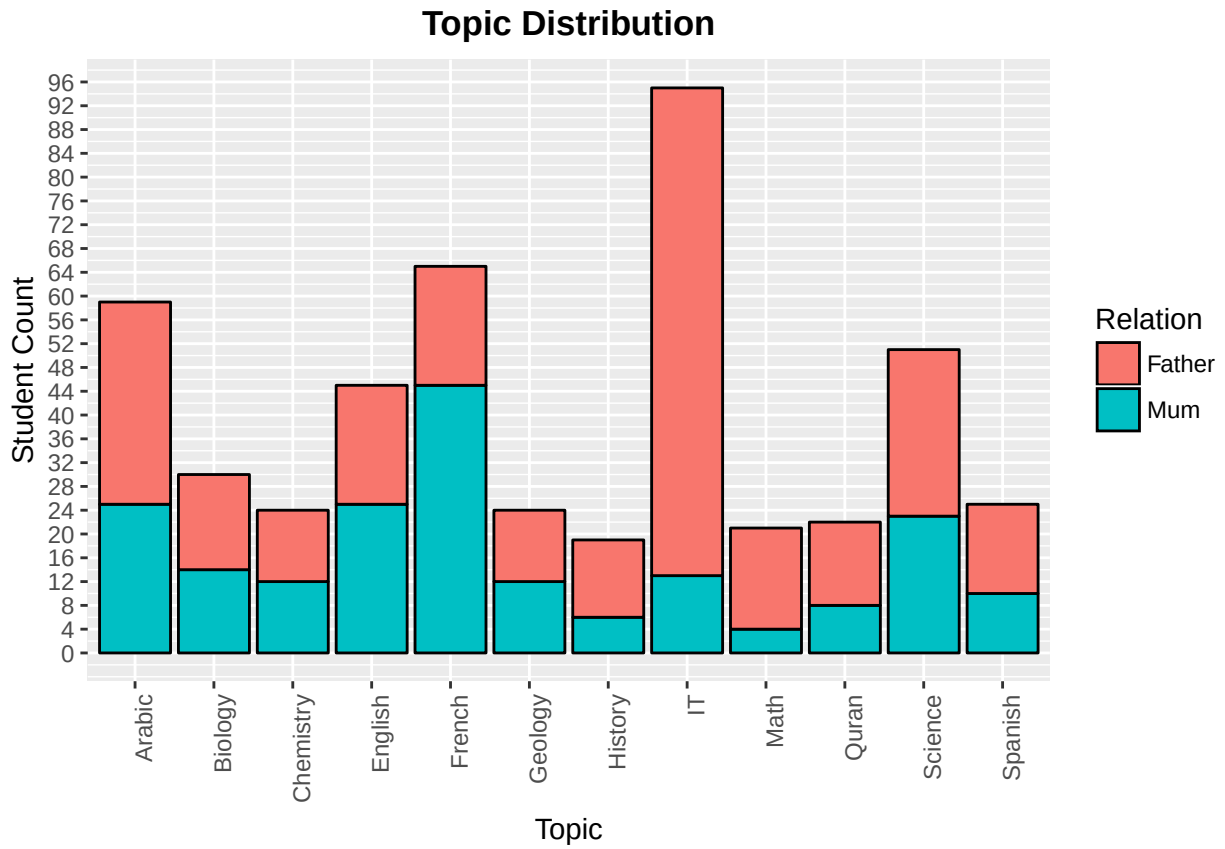
Section C only has Spanish and IT students.

```
ggplot(data = E, aes(x = Topic, fill = Semester)) + geom_bar(colour = "black") +
    labs(x = "Topic", y = "Student Count") + ggtitle(label = "Topic Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold")) + theme(axis.text.x = element_text(angle = 90,
    hjust = 1)) + scale_y_continuous(breaks = seq(0, 100, 4))
```



IT mostly has students who are in first semester.

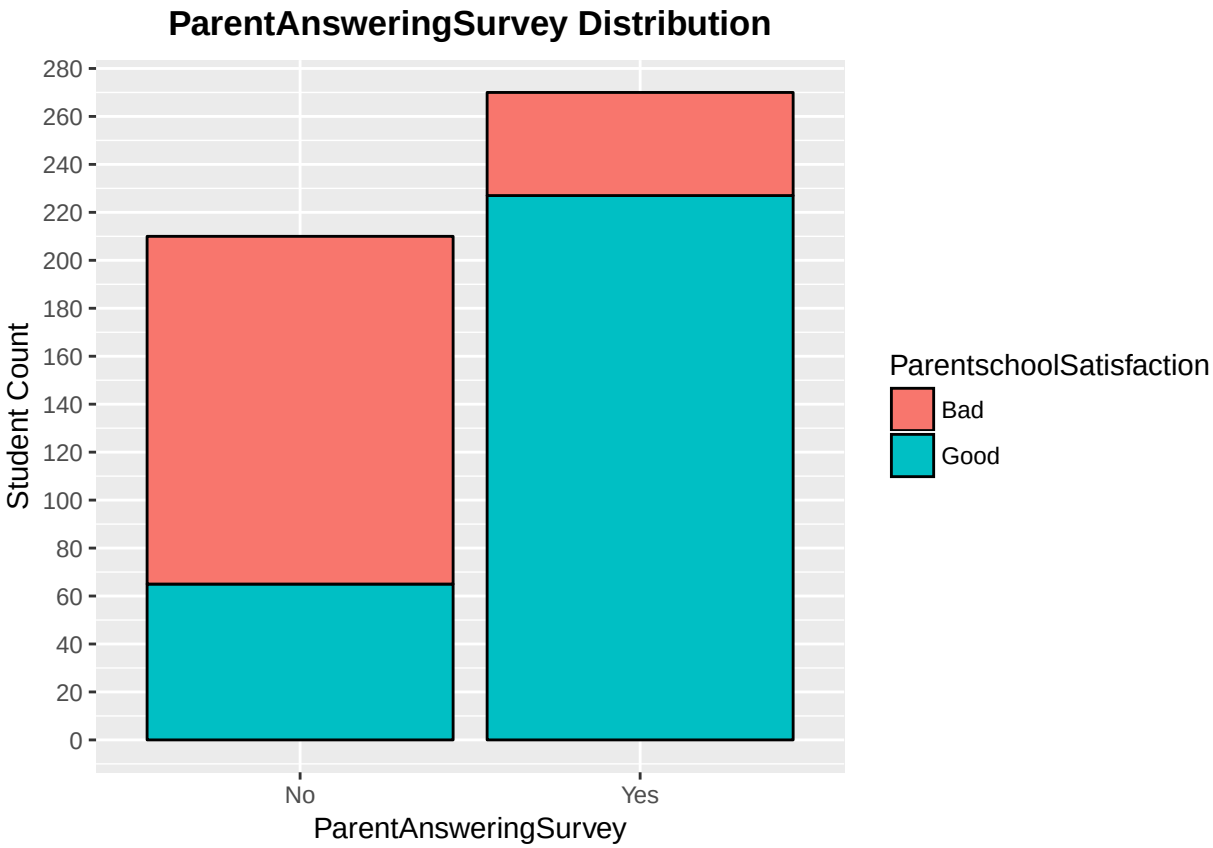
```
ggplot(data = E, aes(x = Topic, fill = Relation)) + geom_bar(colour = "black") +
    labs(x = "Topic", y = "Student Count") + ggtitle(label = "Topic Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold")) + theme(axis.text.x = element_text(angle = 90,
    hjust = 1)) + scale_y_continuous(breaks = seq(0, 100, 4))
```



Most French students have mom as guardian whereas most IT students have fathers as guardian.

Parent Answering Survey distribution:

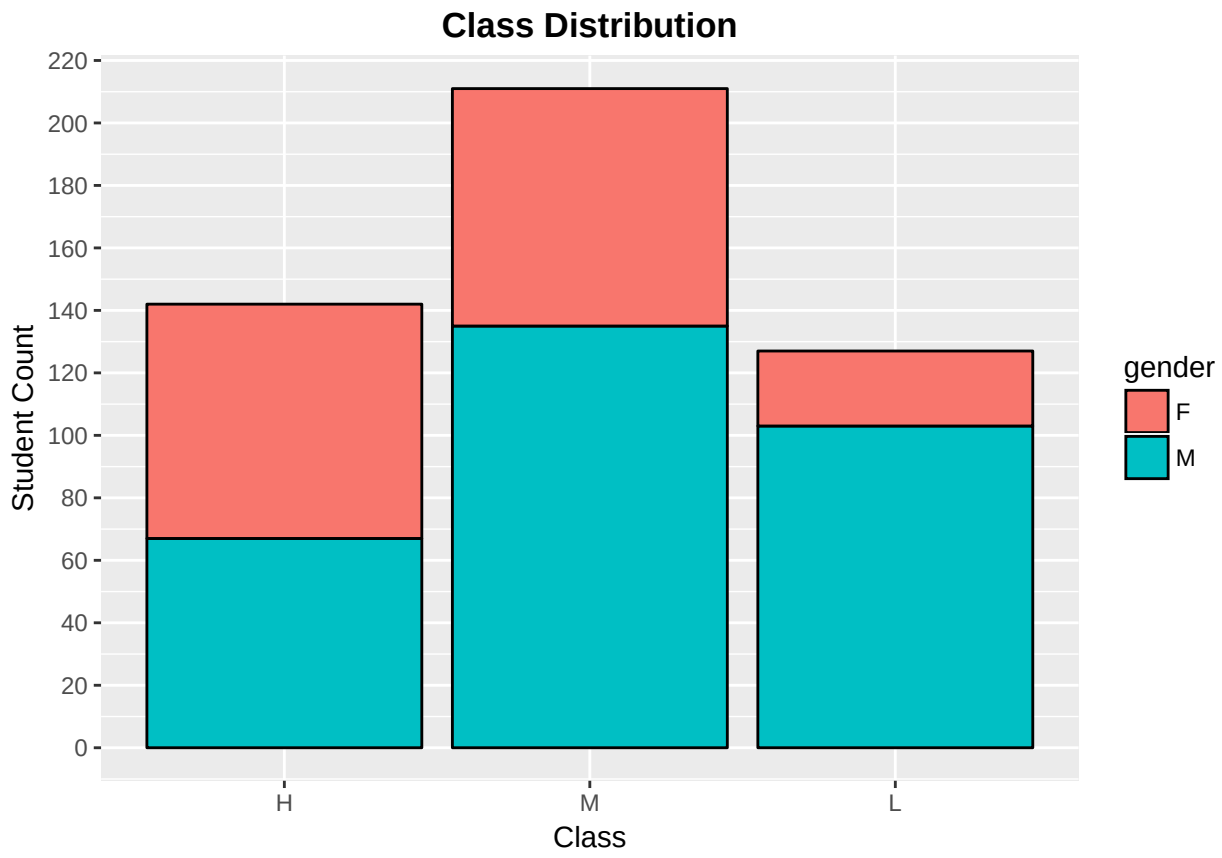
```
ggplot(data = E, aes(x = ParentAnsweringSurvey, fill = ParentschoolSatisfaction)) +
    geom_bar(colour = "black") + labs(x = "ParentAnsweringSurvey",
    y = "Student Count") + ggtitle(label = "ParentAnsweringSurvey Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold")) + scale_y_continuous(breaks = seq(0,
    500, 20))
```



Most of the parents who aren't satisfied with the school do not answer the survey.

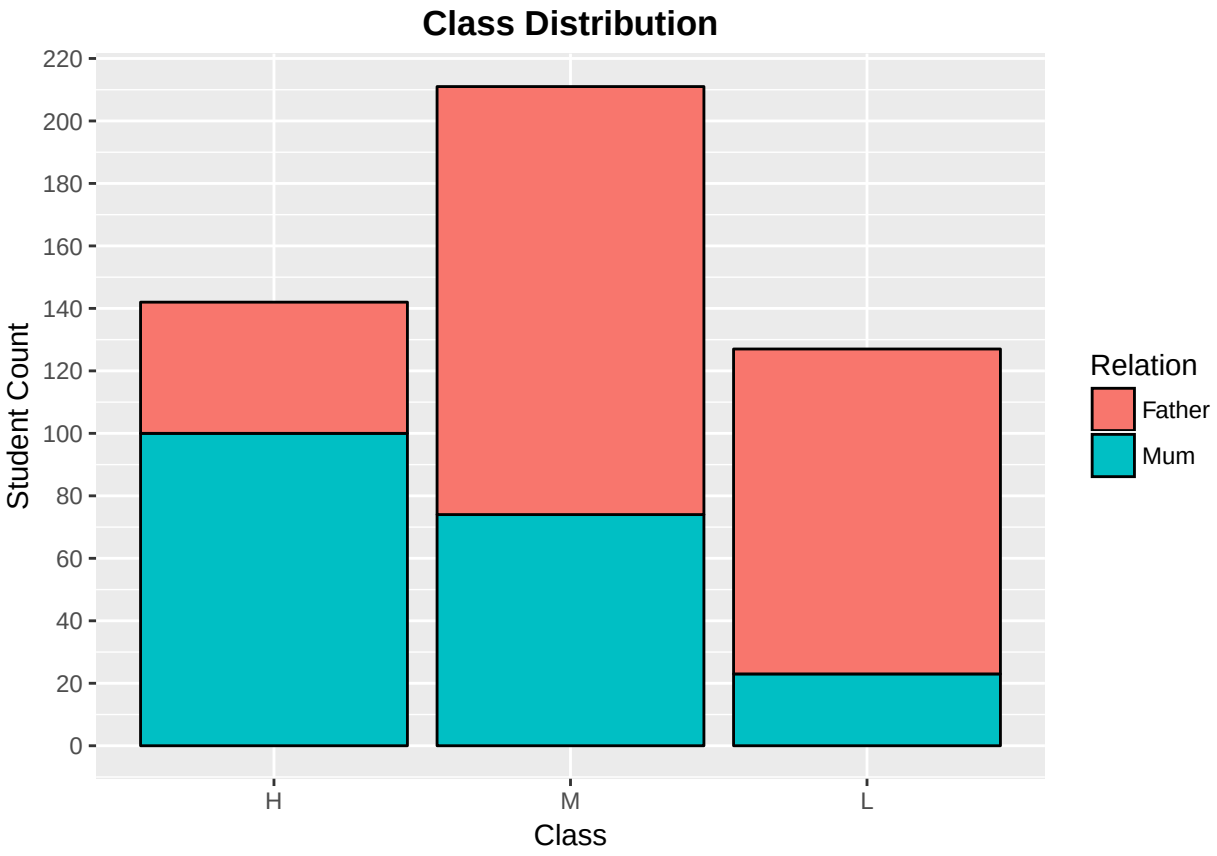
Class distribution:

```
ggplot(data = E, aes(x = Class, fill = gender)) + geom_bar(colour = "black") +
    labs(x = "Class", y = "Student Count") + ggtitle(label = "Class Distribution") +
    theme_grey() + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold")) + scale_y_continuous(breaks = seq(0,
    500, 20))
```



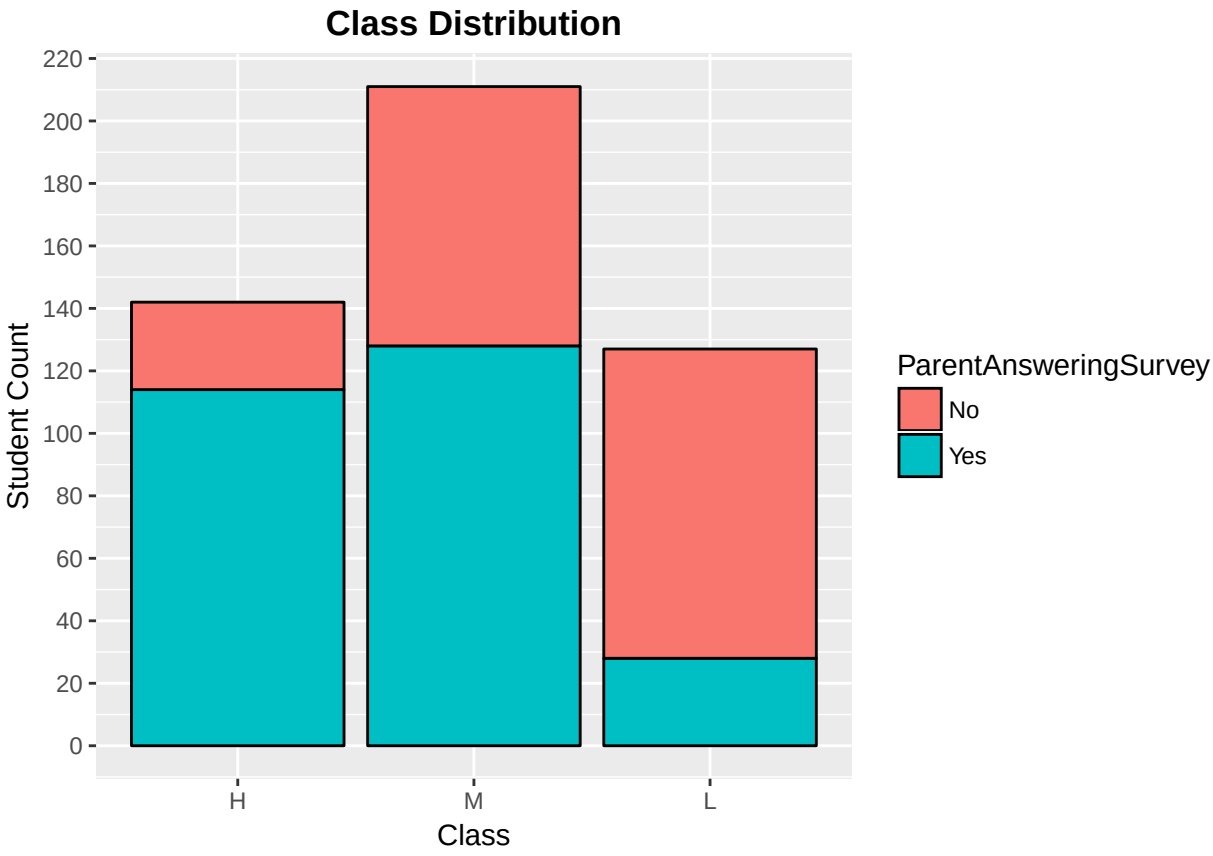
Very few girls in low class

```
ggplot(data = E, aes(x = Class, fill = Relation)) + geom_bar(colour = "black") +
    labs(x = "Class", y = "Student Count") + ggtitle(label = "Class Distribution") +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold")) + scale_y_continuous(breaks = seq(0,
    500, 20))
```



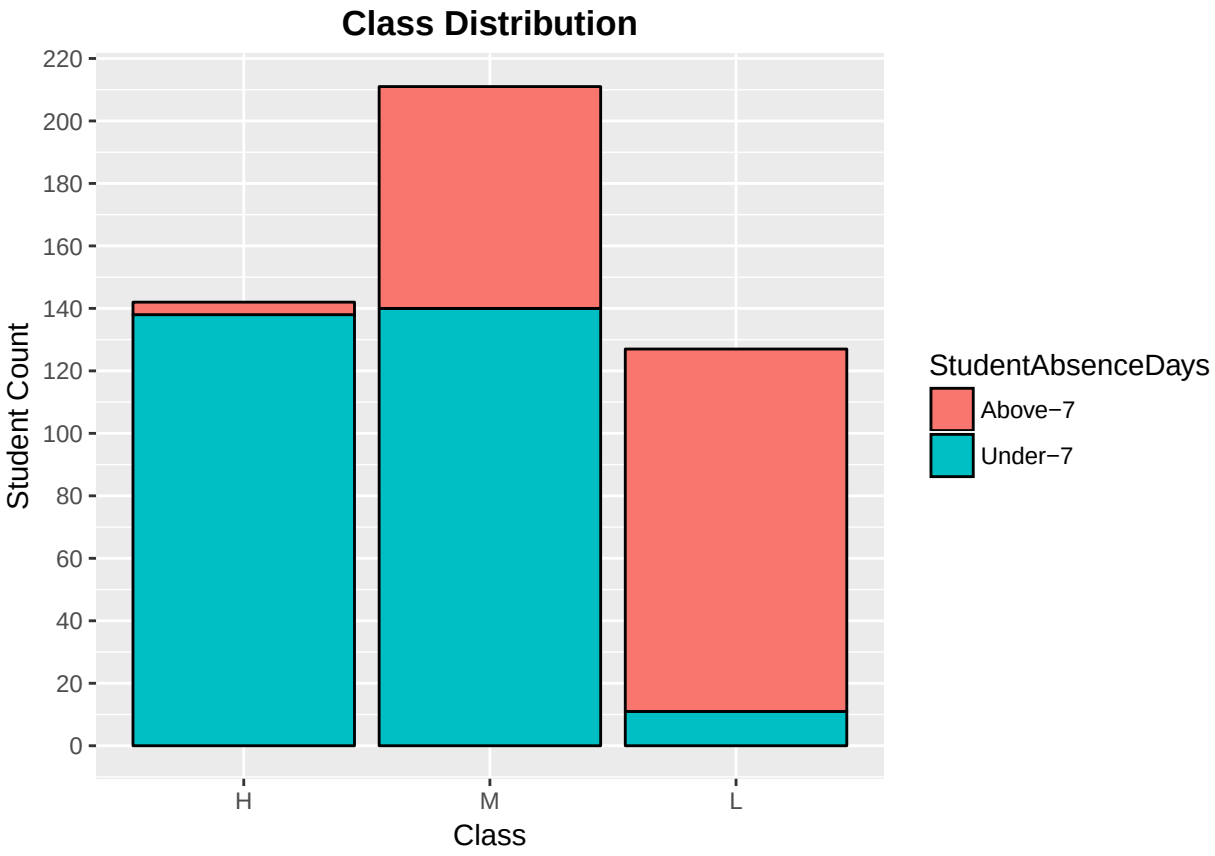
The students who have moms as guardians have higher chances to get high class marks.

```
ggplot(data = E, aes(x = Class, fill = ParentAnsweringSurvey)) +
    geom_bar(colour = "black") + labs(x = "Class", y = "Student Count") +
    ggtitle(label = "Class Distribution") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold")) + scale_y_continuous(breaks = seq(0,
    500, 20))
```



Students whose parents answer the survey are the ones getting good marks.

```
ggplot(data = E, aes(x = Class, fill = StudentAbsenceDays)) +
    geom_bar(colour = "black") + labs(x = "Class", y = "Student Count") +
    ggtitle(label = "Class Distribution") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold")) + scale_y_continuous(breaks = seq(0,
    500, 20))
```



Student getting absent are the ones getting low marks.

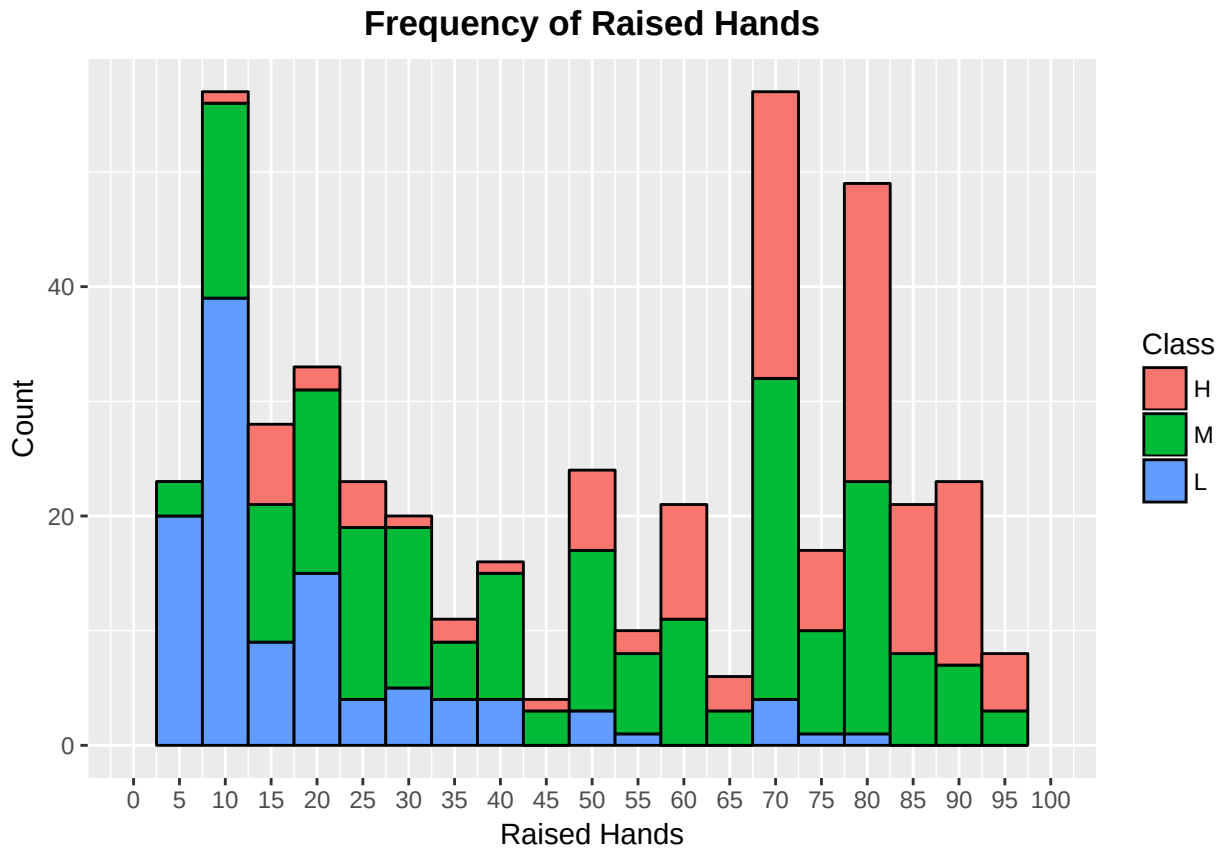
Summary of Task 1.2:

Based on the exploration work above, We conclude the following main findings which will be much useful for analysis and prediction of the academic performance in the next steps:

1. There are more girls in the high achiever's class (high class) than in the low class; boys are the opposite. Girls' parents have higher school satisfaction than boys. Boys are more frequently absent for more than 7 days.
2. There are very small differences in class performance with respect to place of birth and nationality. The findings between place of birth and nationality are very similar. This justify using only nationality in place of both for the next steps.
3. Middle school has the most students. Lower-level has higher rate of above 7 days of student absence. Lower-level has the highest rate of low class students (low achievers). Highschool has the highest rate of high class students (high achievers). Middle school has the highest rate of medium class students.
4. Biology has the highest ratio of high class students, but also the highest ratio of low class students. Geology has the highest ratio of medium class students.
5. The students who have moms as guardians have higher chances to get high class marks. Students whose parents answer the survey are the ones getting good marks. Student that have more absences are those getting low marks.

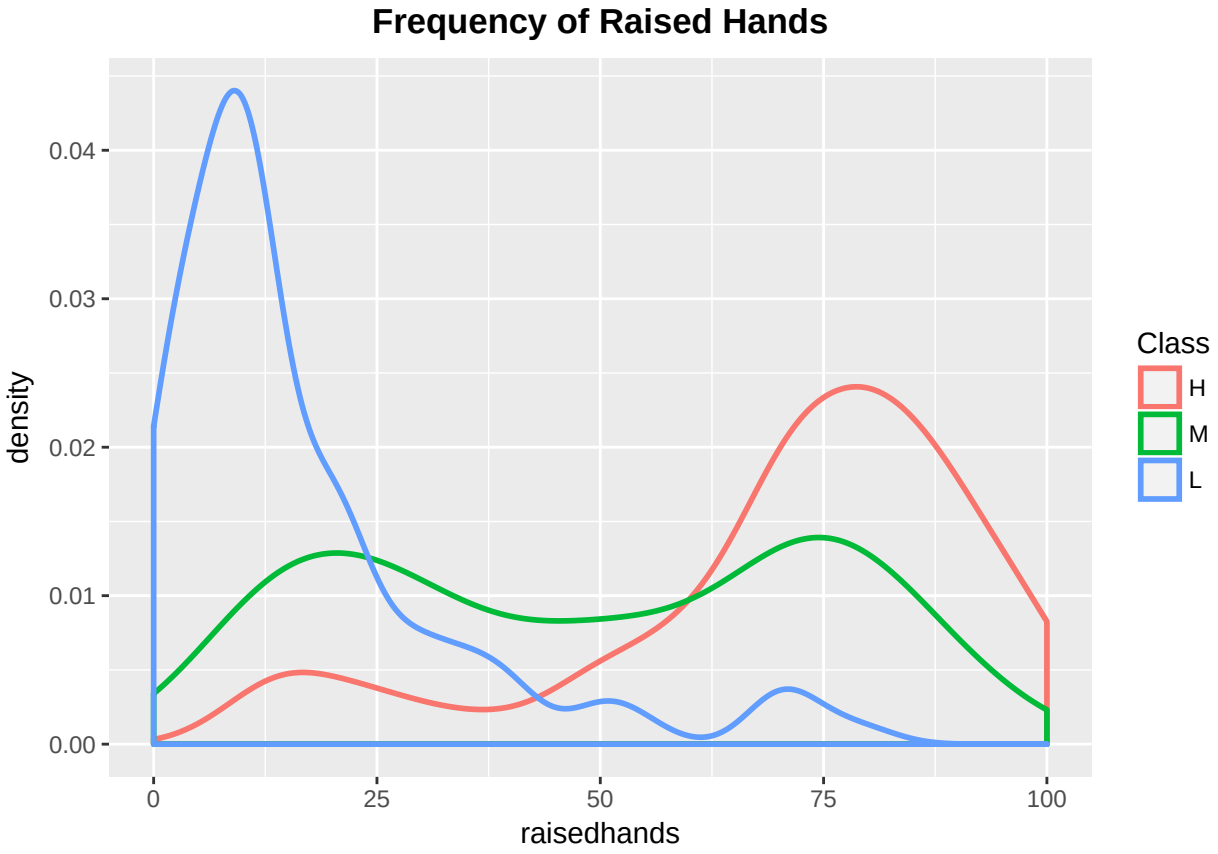
Task 1.3. Histogram of numerical values distribution:

```
ggplot(E, aes(x = raisedhands, fill = Class)) + geom_histogram(binwidth = 5,
    colour = "black") + scale_x_continuous(name = "Raised Hands",
    breaks = seq(0, 100, 5), limits = c(0, 100)) + scale_y_continuous(name = "Count") +
    ggtitle("Frequency of Raised Hands") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



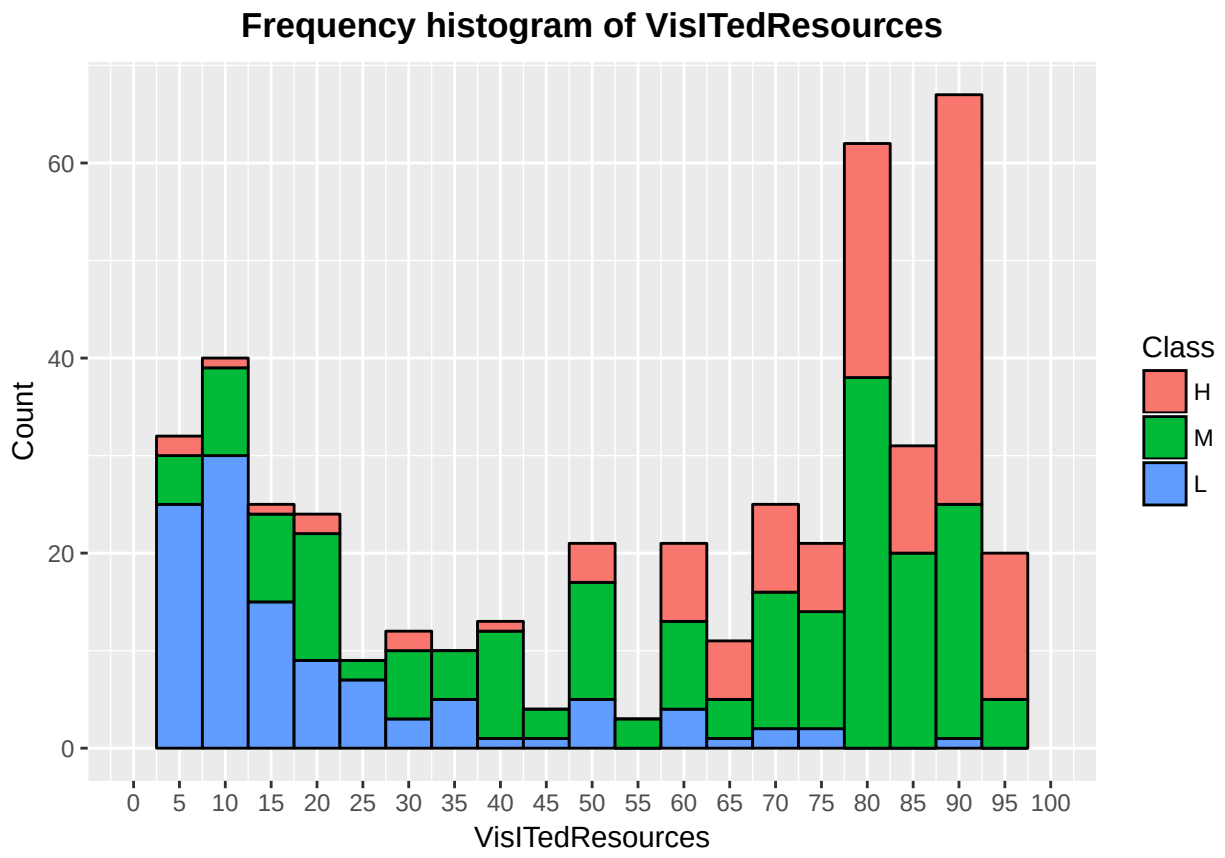
Students have the higher class the more raised hands.

```
ggplot(data = E, aes(x = raisedhands, color = Class)) + geom_density(size = 1) +
    ggtitle("Frequency of Raised Hands") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



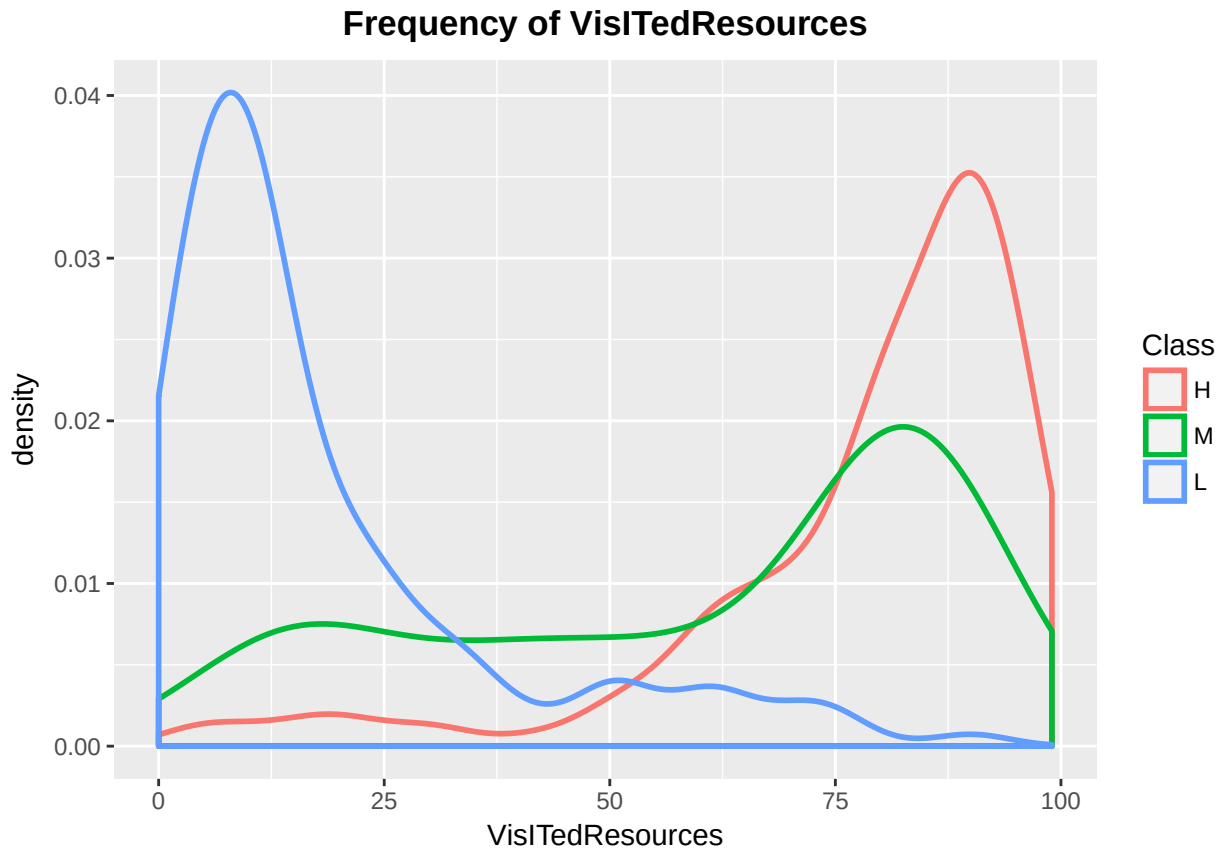
Most of the high class students raised hands many times, most of the low class students raised hands very few times.

```
ggplot(E, aes(x = VisITedResources, fill = Class)) + geom_histogram(binwidth = 5,
    colour = "black") + scale_x_continuous(name = "VisITedResources",
    breaks = seq(0, 100, 5), limits = c(0, 100)) + scale_y_continuous(name = "Count") +
    ggtitle("Frequency histogram of VisITedResources") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



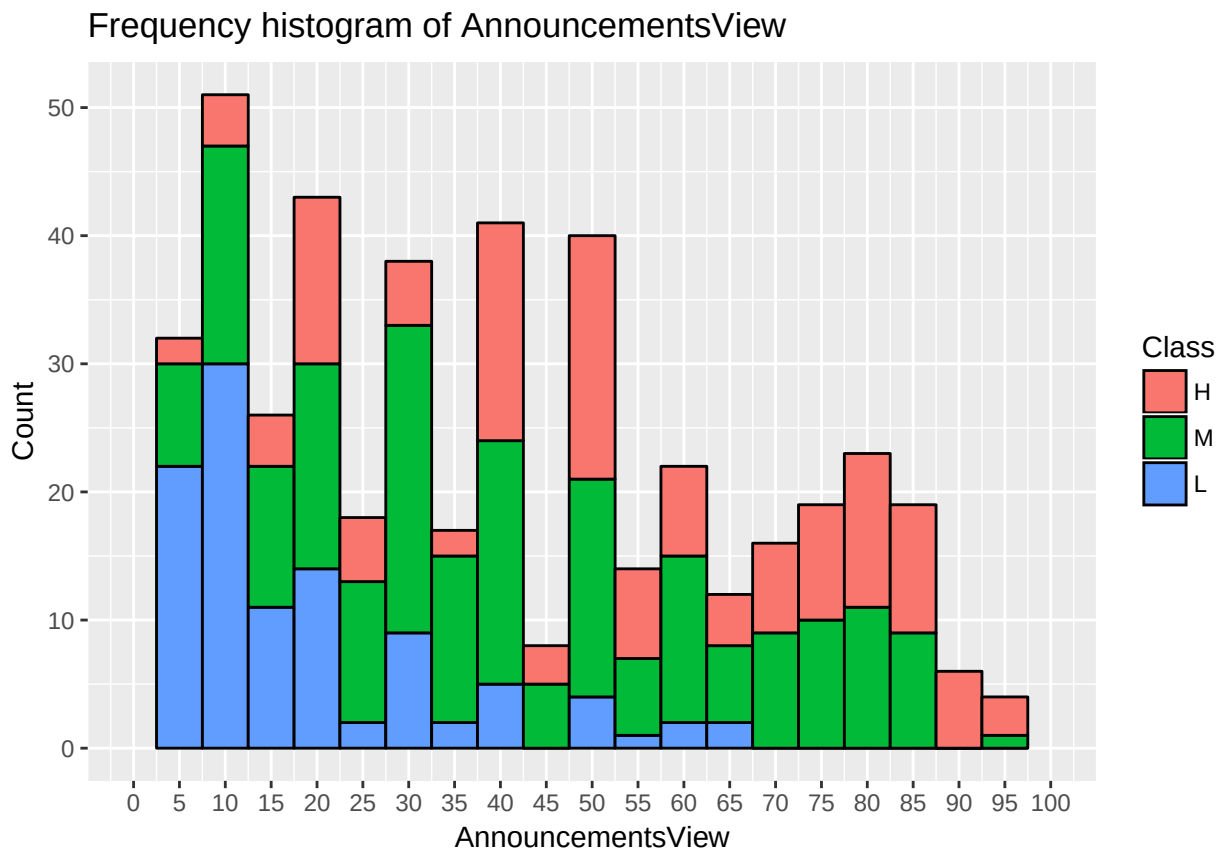
Students who have the higher class visited the resources more times.

```
ggplot(data = E, aes(x = VisITedResources, color = Class)) +
    geom_density(size = 1) + ggtitle("Frequency of VisITedResources") +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold"))
```



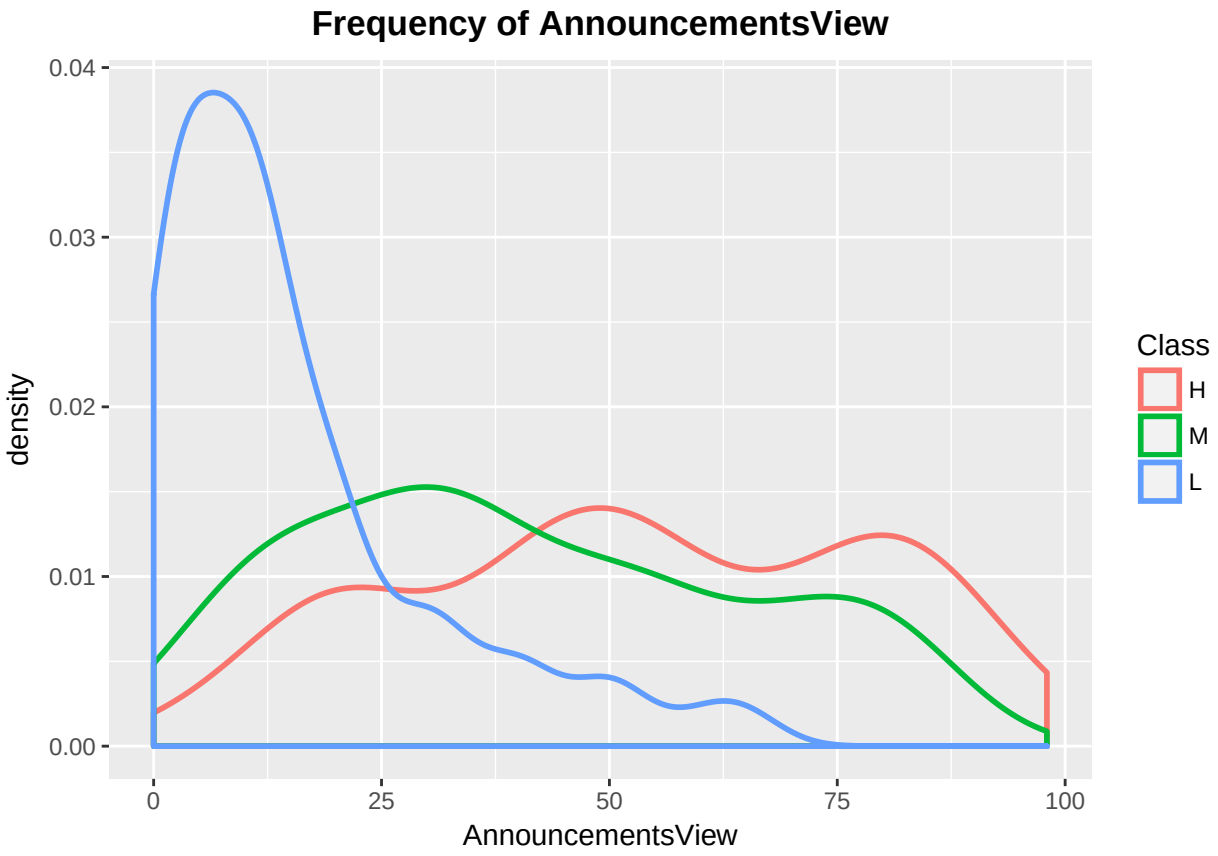
Most of the students with high class visited the resources many times, most of the students with low class visited the resources very few times.

```
ggplot(E, aes(x = AnnouncementsView, fill = Class)) + geom_histogram(binwidth = 5,
    colour = "black") + scale_x_continuous(name = "AnnouncementsView",
    breaks = seq(0, 100, 5), limits = c(0, 100)) + scale_y_continuous(name = "Count") +
    ggtitle("Frequency histogram of AnnouncementsView")
```



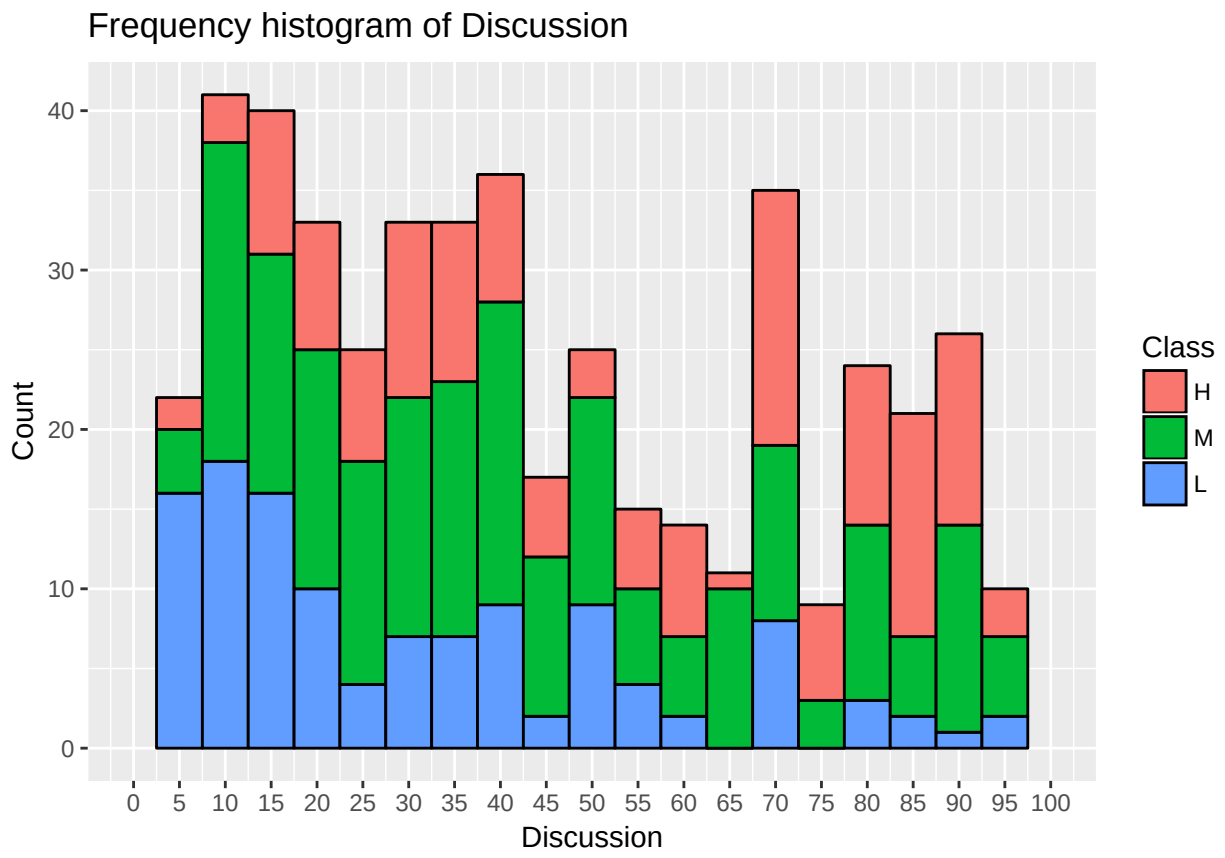
Students who have the higher class viewed the announcements more times.

```
ggplot(data = E, aes(x = AnnouncementsView, color = Class)) +
    geom_density(size = 1) + ggtitle("Frequency of AnnouncementsView") +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold"))
```



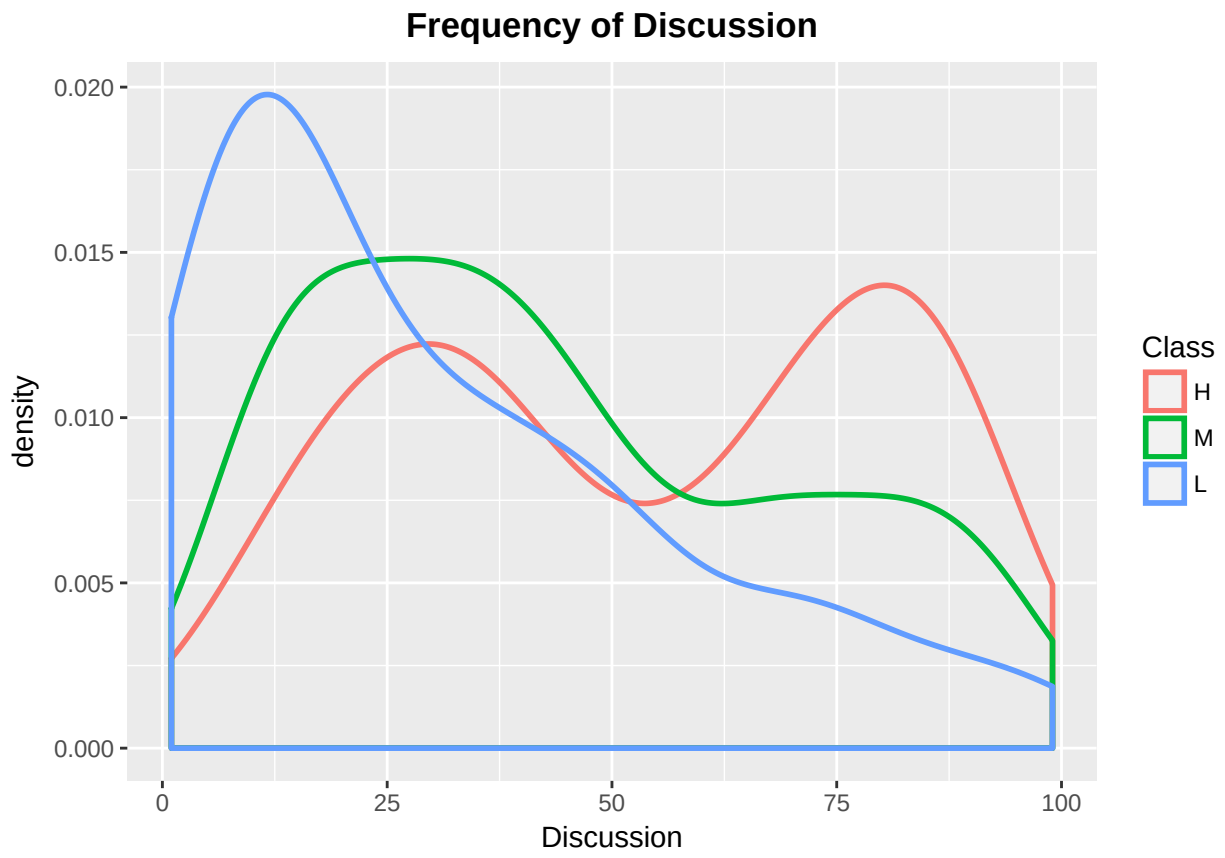
Most of the students with high class viewed the announcements many times, most of the students with low class viewed the announcements very few times. But the differences between students with high class and medium class is not very obvious

```
ggplot(E, aes(x = Discussion, fill = Class)) + geom_histogram(binwidth = 5,
    colour = "black") + scale_x_continuous(name = "Discussion",
    breaks = seq(0, 100, 5), limits = c(0, 100)) + scale_y_continuous(name = "Count") +
    ggtitle("Frequency histogram of Discussion")
```



Students who have the higher class discussed more times.

```
ggplot(data = E, aes(x = Discussion, color = Class)) + geom_density(size = 1) +
    ggtitle("Frequency of Discussion") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



Most of the students with high class discussed many times, most of the students with low class discussed very few times.

Summary of Task 1.3:

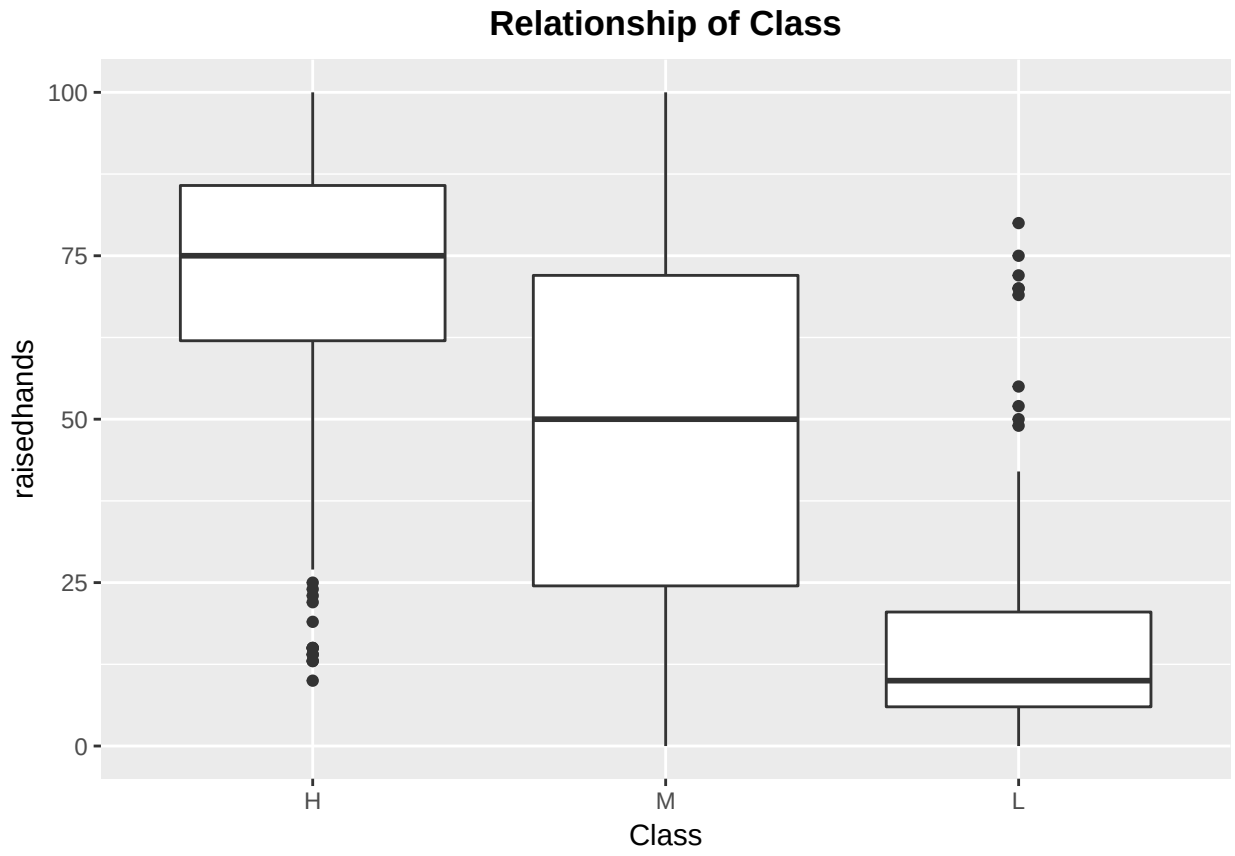
From the exploration of the numerical feature values, we have the following findings:

1. High achiever students are more likely to raise hands, visit resources, view announcements and participate in discussions. Low achiever students raised hands, visited resource, viewed announcements and participated in discussions few times.
2. There are some little differences in the density curve, especially in the announcement view and discussion.

Task 2. Exploration of the relationships between different features of the students' academic performance datasets

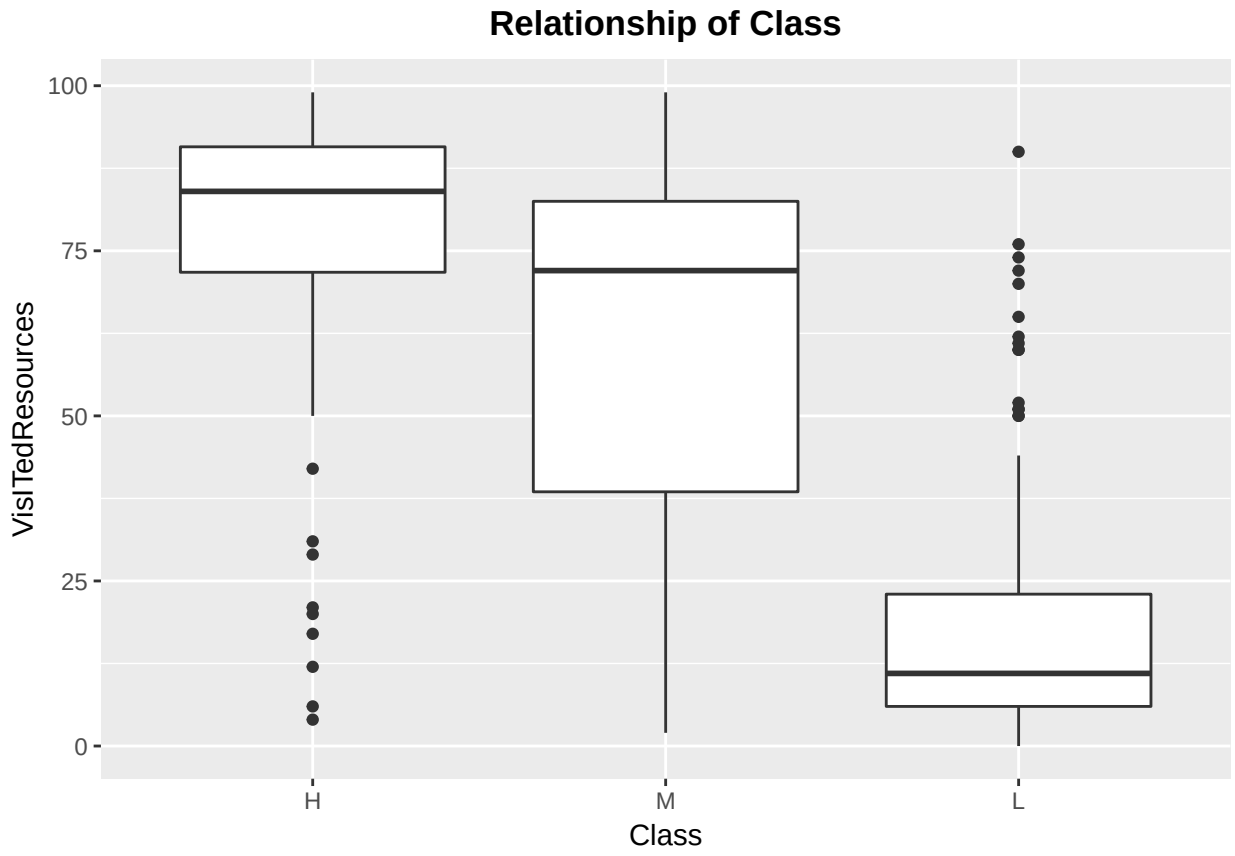
Class and the numerial features

```
ggplot(data = E, aes(x = Class, y = raisedhands)) + geom_boxplot() +
    ggtitle("Relationship of Class") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



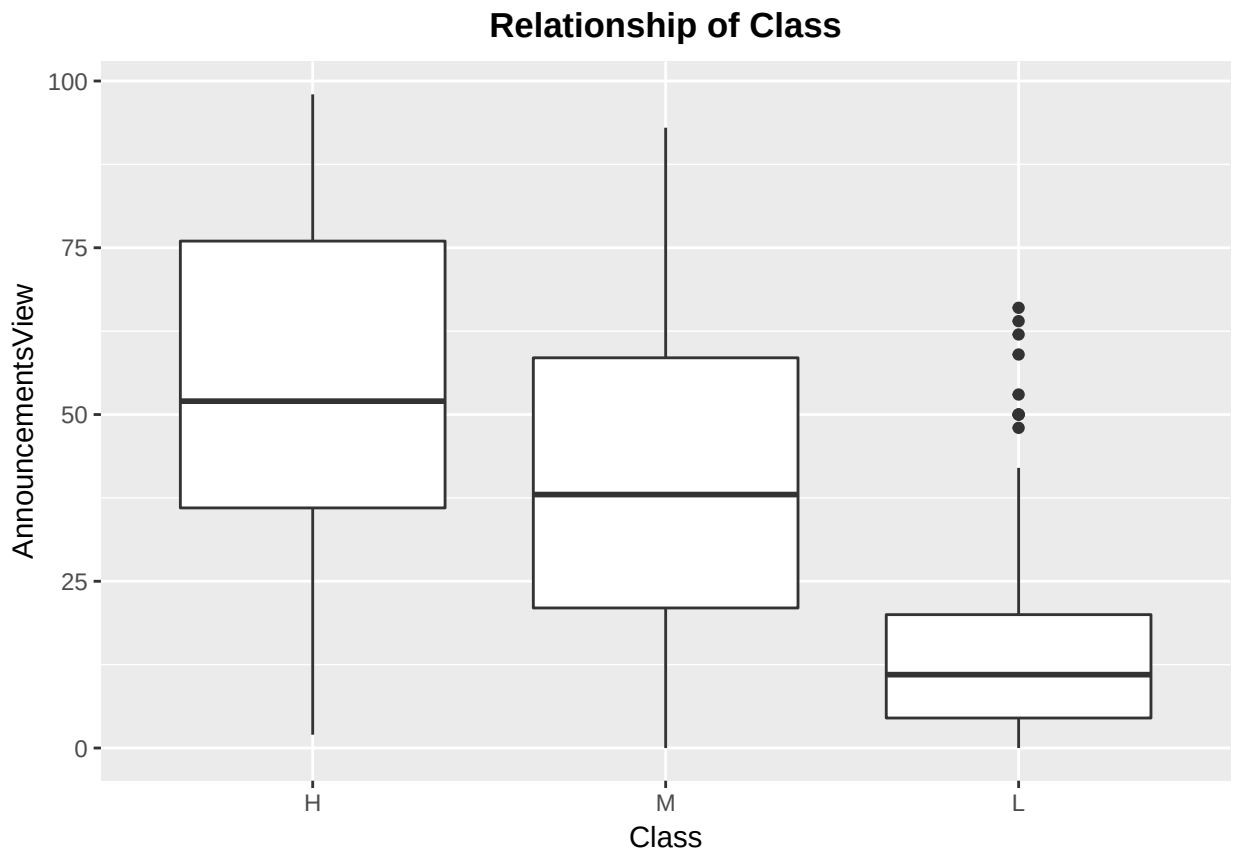
The higher class the higher times of raising hands

```
ggplot(data = E, aes(x = Class, y = VisITedResources)) + geom_boxplot() +
    ggtitle("Relationship of Class") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



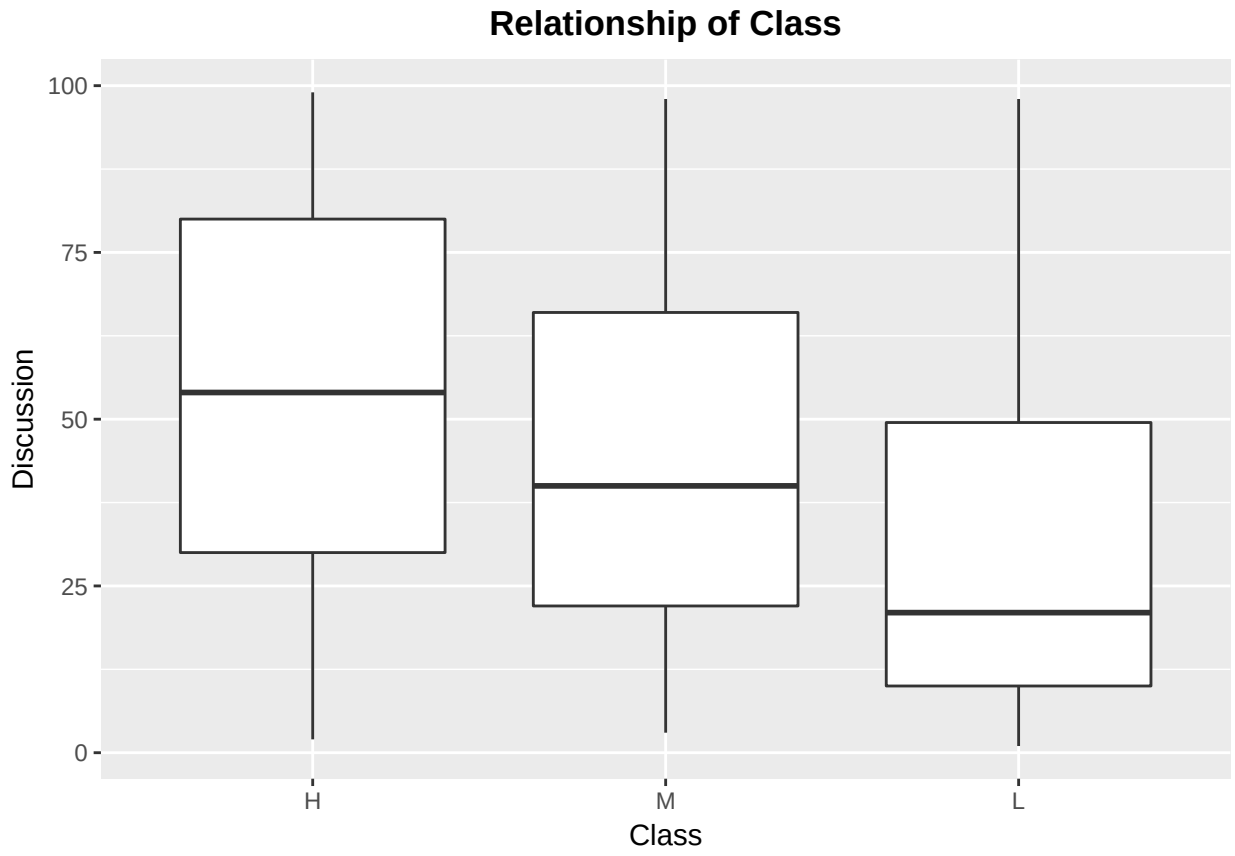
The higher class the higher times of visiting resources

```
ggplot(data = E, aes(x = Class, y = AnnouncementsView)) + geom_boxplot() +
    ggtitle("Relationship of Class") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



The higher class the higher times of viewing announcements

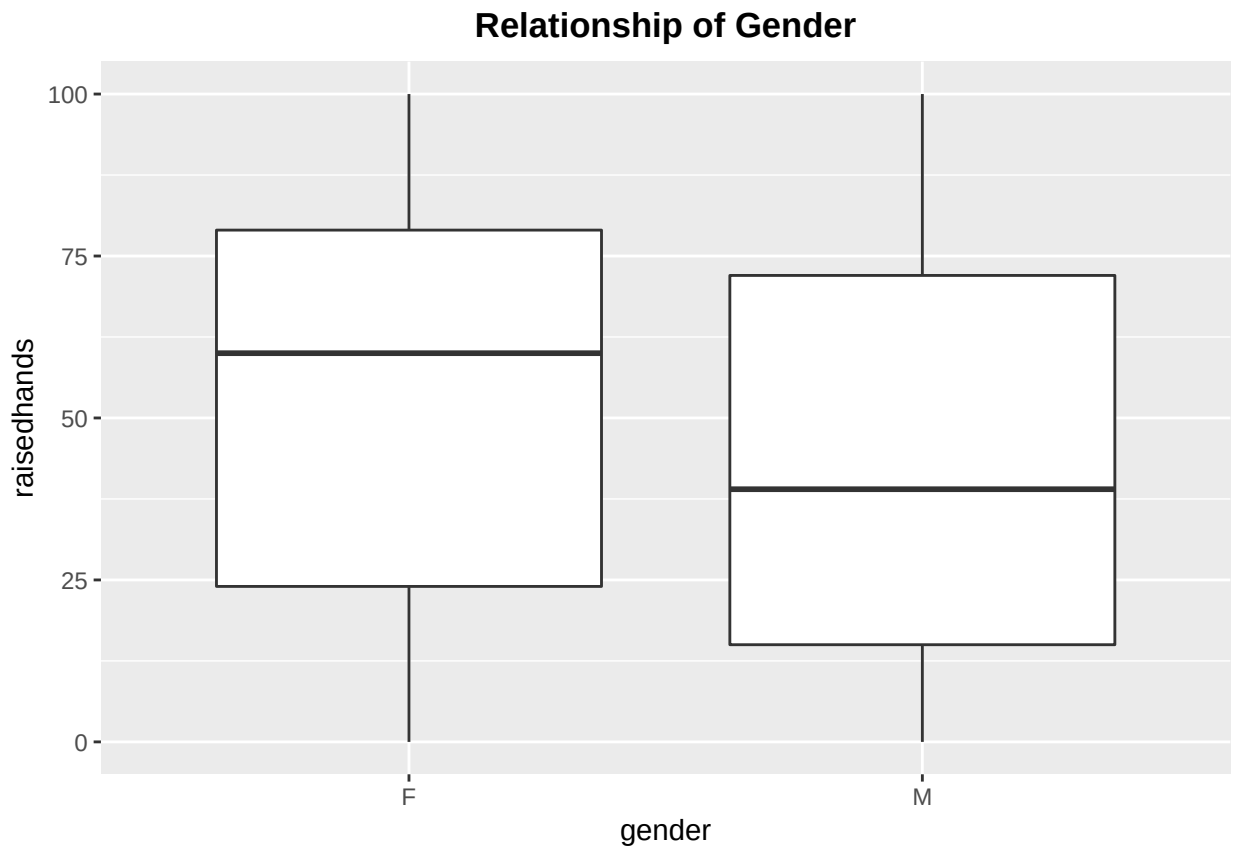
```
ggplot(data = E, aes(x = Class, y = Discussion)) + geom_boxplot() +
    ggtitle("Relationship of Class") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



The higher class the higher times of discussion

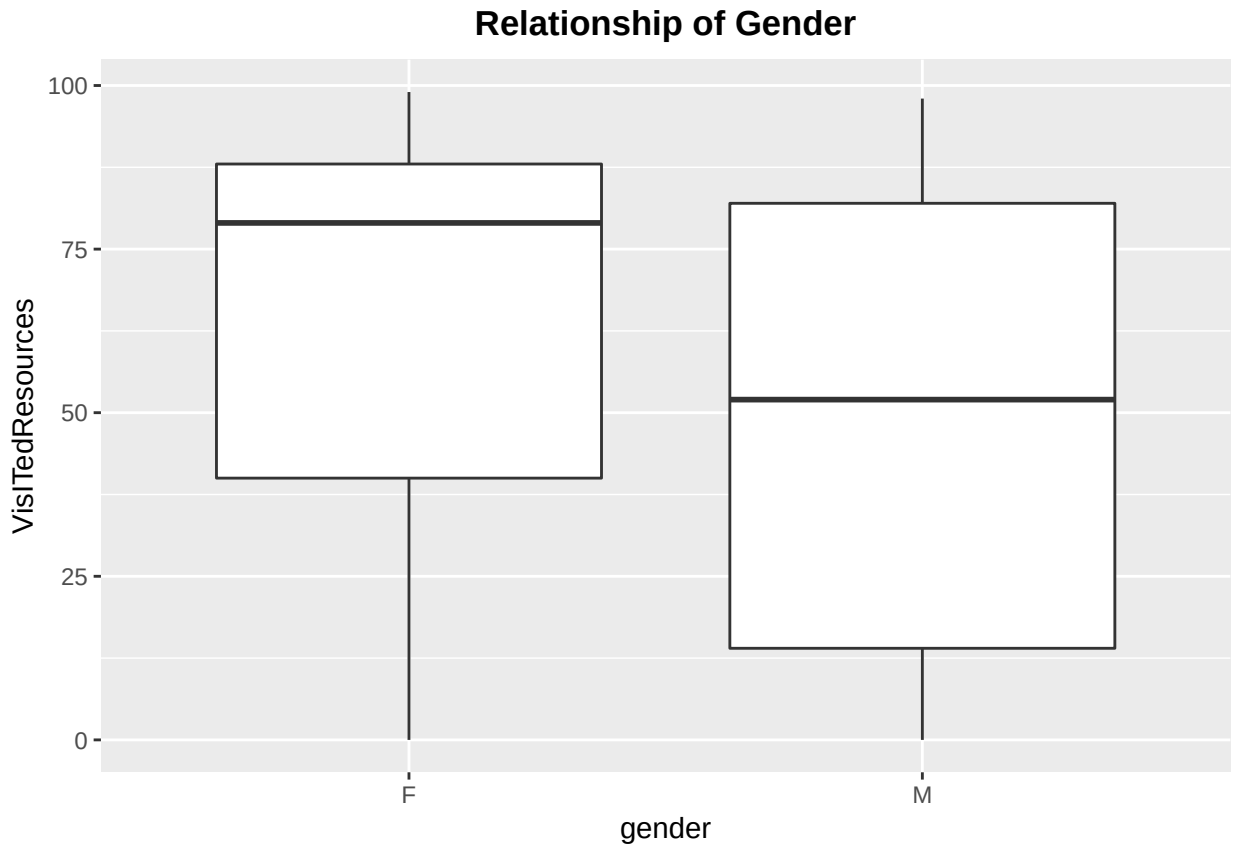
Gender and the numerial features

```
ggplot(data = E, aes(x = gender, y = raisedhands)) + geom_boxplot() +
    ggtitle("Relationship of Gender") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



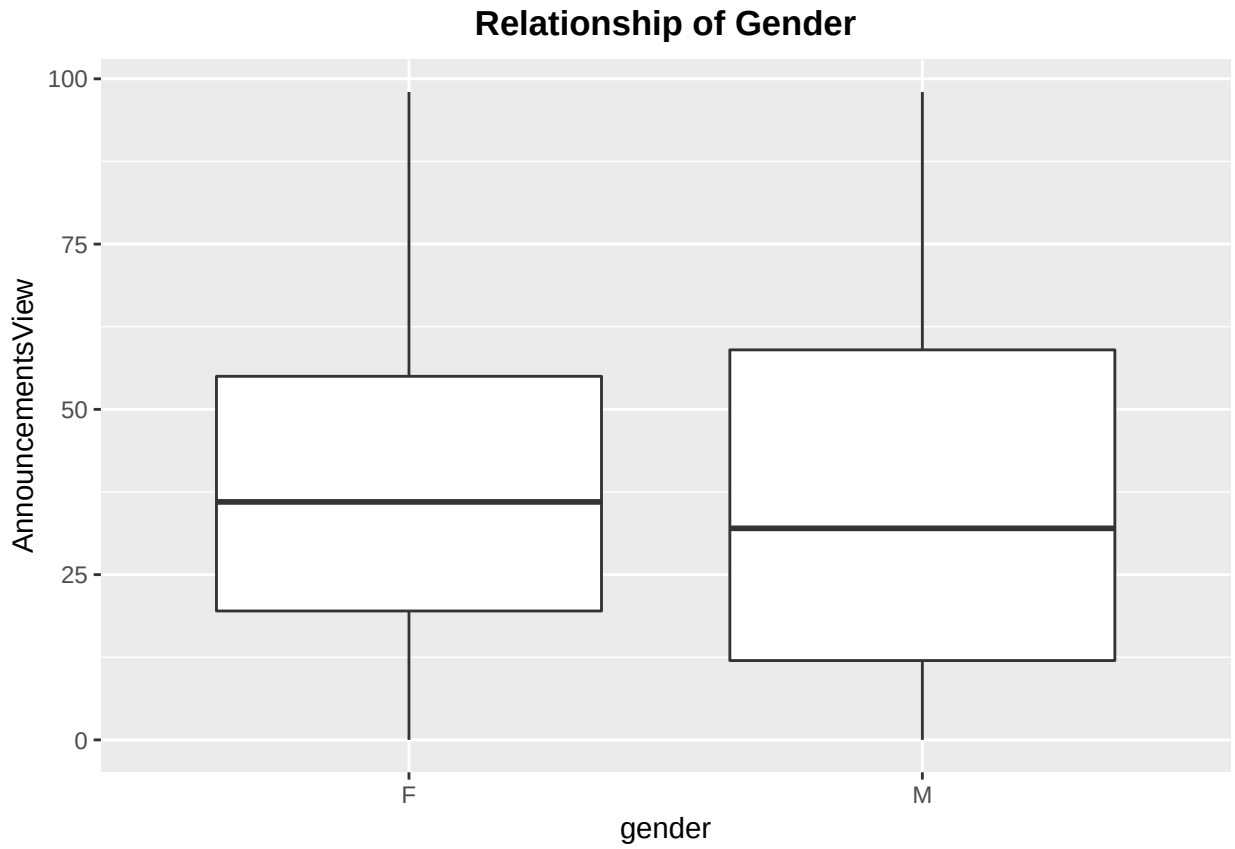
Girls have more hand raises

```
ggplot(data = E, aes(x = gender, y = VisITedResources)) + geom_boxplot() +
    ggtitle("Relationship of Gender") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



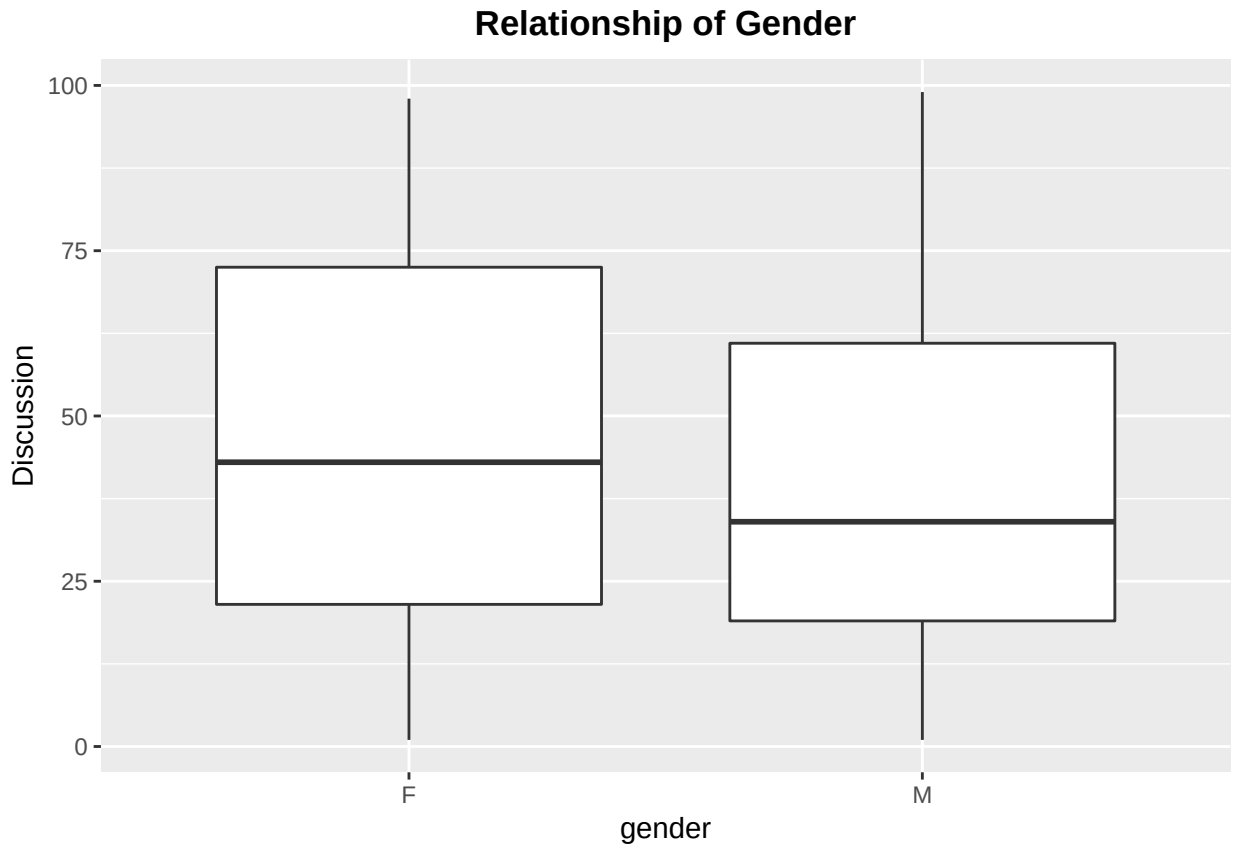
Girls visit more resources

```
ggplot(data = E, aes(x = gender, y = AnnouncementsView)) + geom_boxplot() +
    ggtitle("Relationship of Gender") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



Girls view more announcements, but the differences are not very obvious.

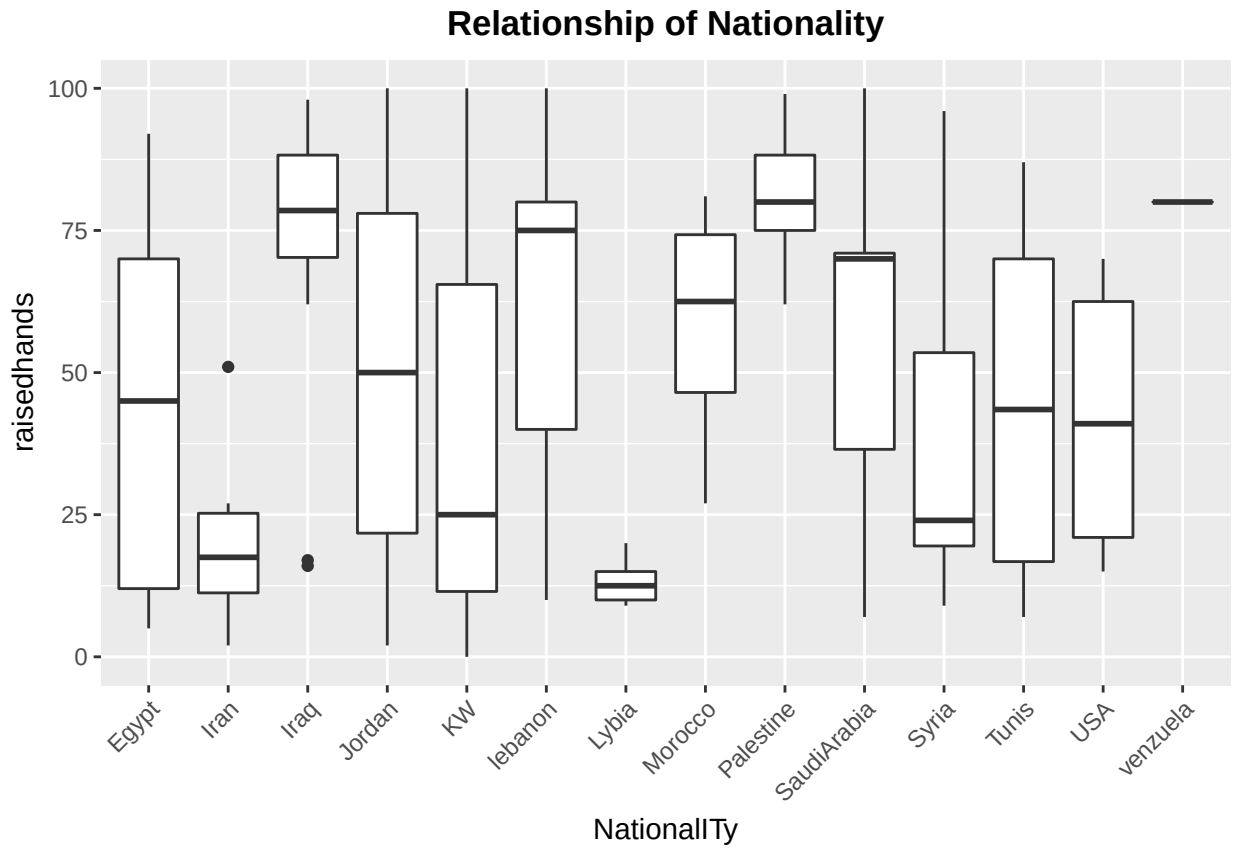
```
ggplot(data = E, aes(x = gender, y = Discussion)) + geom_boxplot() +
    ggtitle("Relationship of Gender") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



Girls have more discussion, but the differences are not very obvious.

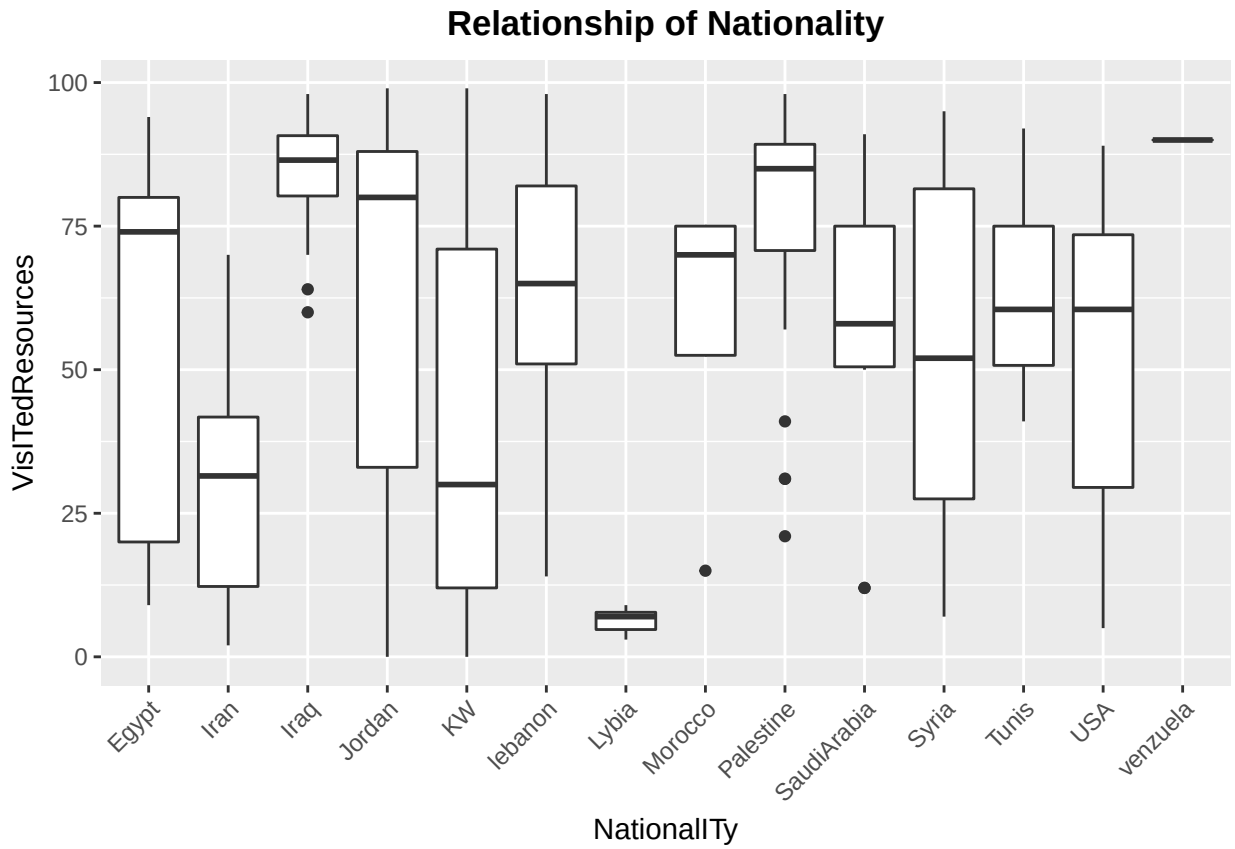
Nationality and the numerial features

```
ggplot(data = E, aes(x = NationalITy, y = raisedhands)) + geom_boxplot() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
    ggtitle("Relationship of Nationality") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



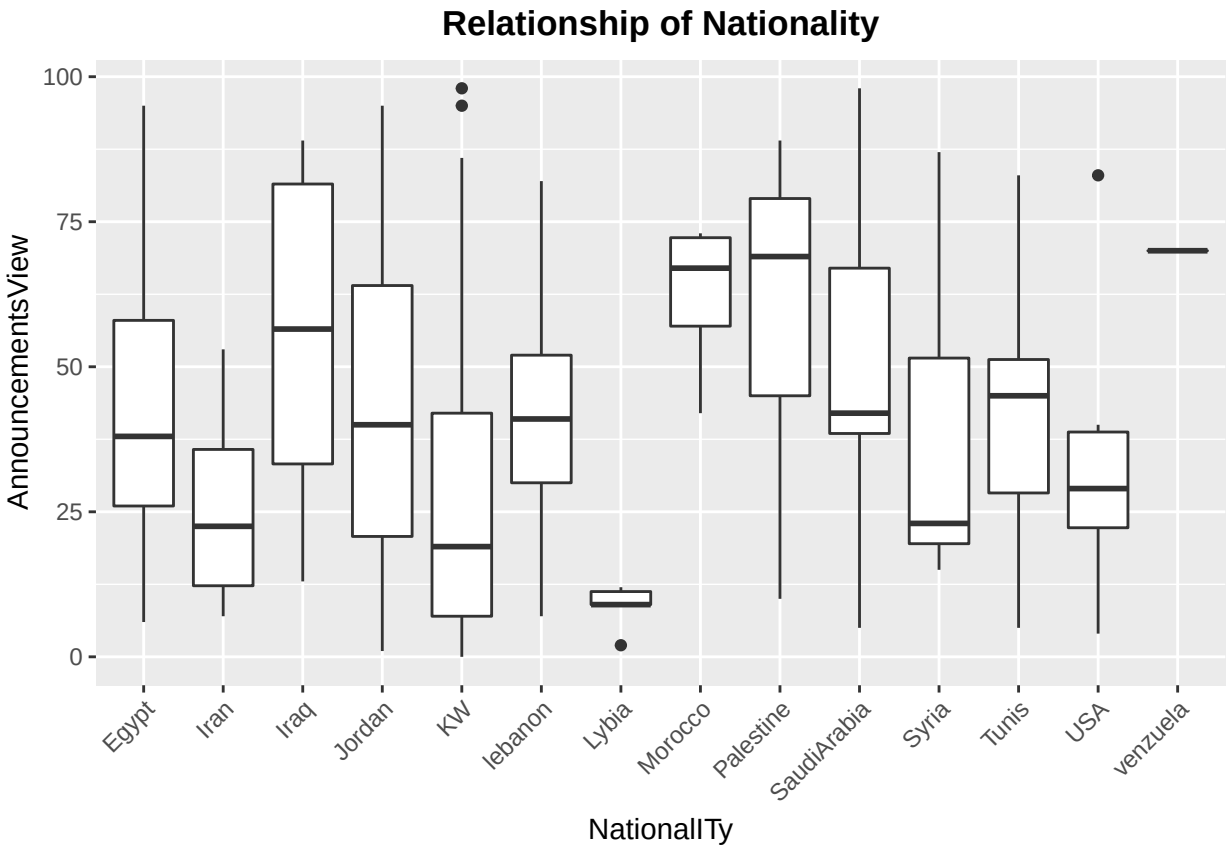
Jordan has more hand raises than Kuwait, Lybia has lowest. Iraq and Palestine have highest hand raises.

```
ggplot(data = E, aes(x = NationalITy, y = VisITedResources)) +
    geom_boxplot() + theme(axis.text.x = element_text(angle = 45,
    hjust = 1)) + ggtitle("Relationship of Nationality") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



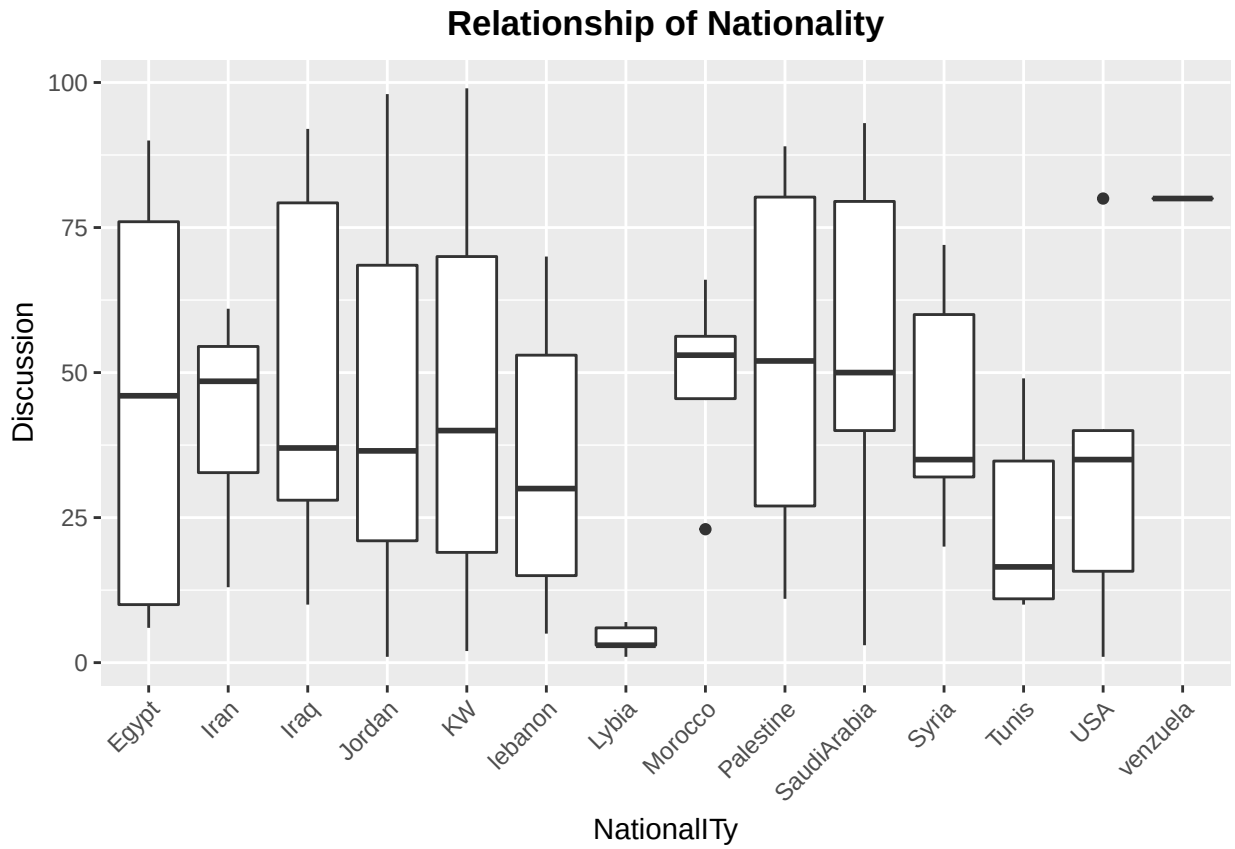
Jordan has more visiting resouces than Kuwait, Lybia has lowest. Iraq and Palestine have highest visiting resources.

```
ggplot(data = E, aes(x = NationalITy, y = AnnouncementsView)) +
    geom_boxplot() + theme(axis.text.x = element_text(angle = 45,
    hjust = 1)) + ggtitle("Relationship of Nationality") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



Jordan has more announcement view than Kuwait, Lybia has lowest. Iraq and Palestine have highest announcement view.

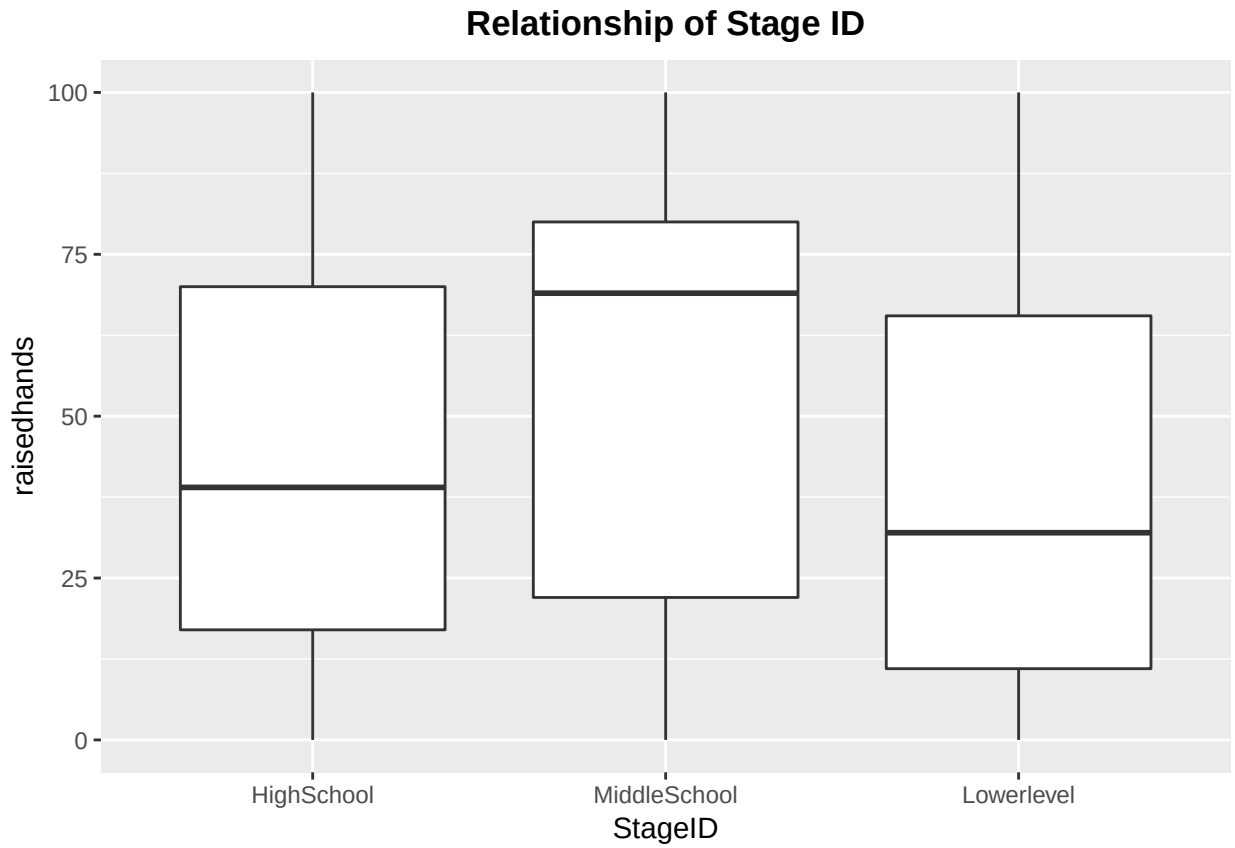
```
ggplot(data = E, aes(x = NationalITy, y = Discussion)) + geom_boxplot() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
    ggtitle("Relationship of Nationality") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



The differences between Jordan and Kuwait are not obvious, Lybia has lowest. Iraq and Palestine have highest announcement view.

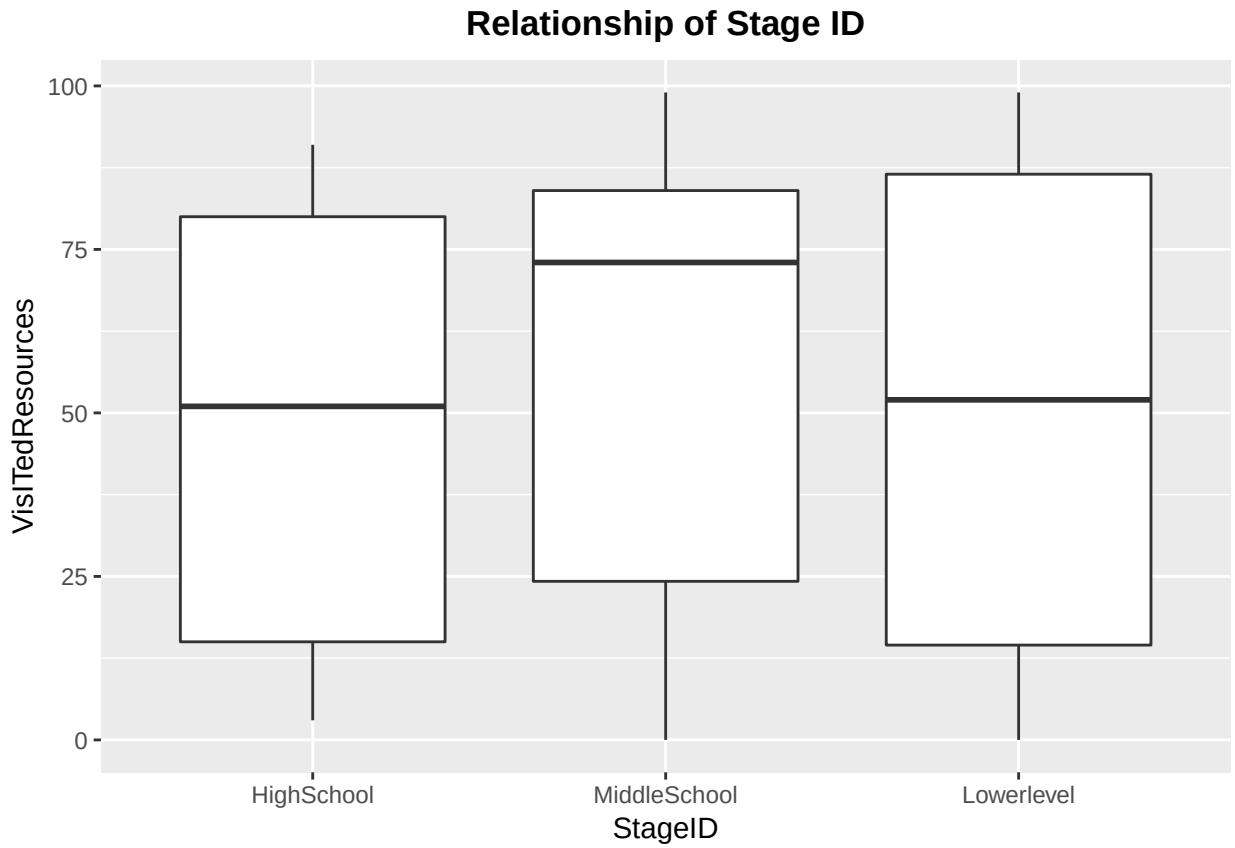
Stage ID and the numerial features

```
ggplot(data = E, aes(x = StageID, y = raisedhands)) + geom_boxplot() +
    ggtitle("Relationship of Stage ID") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



More hand raises in middle schools.

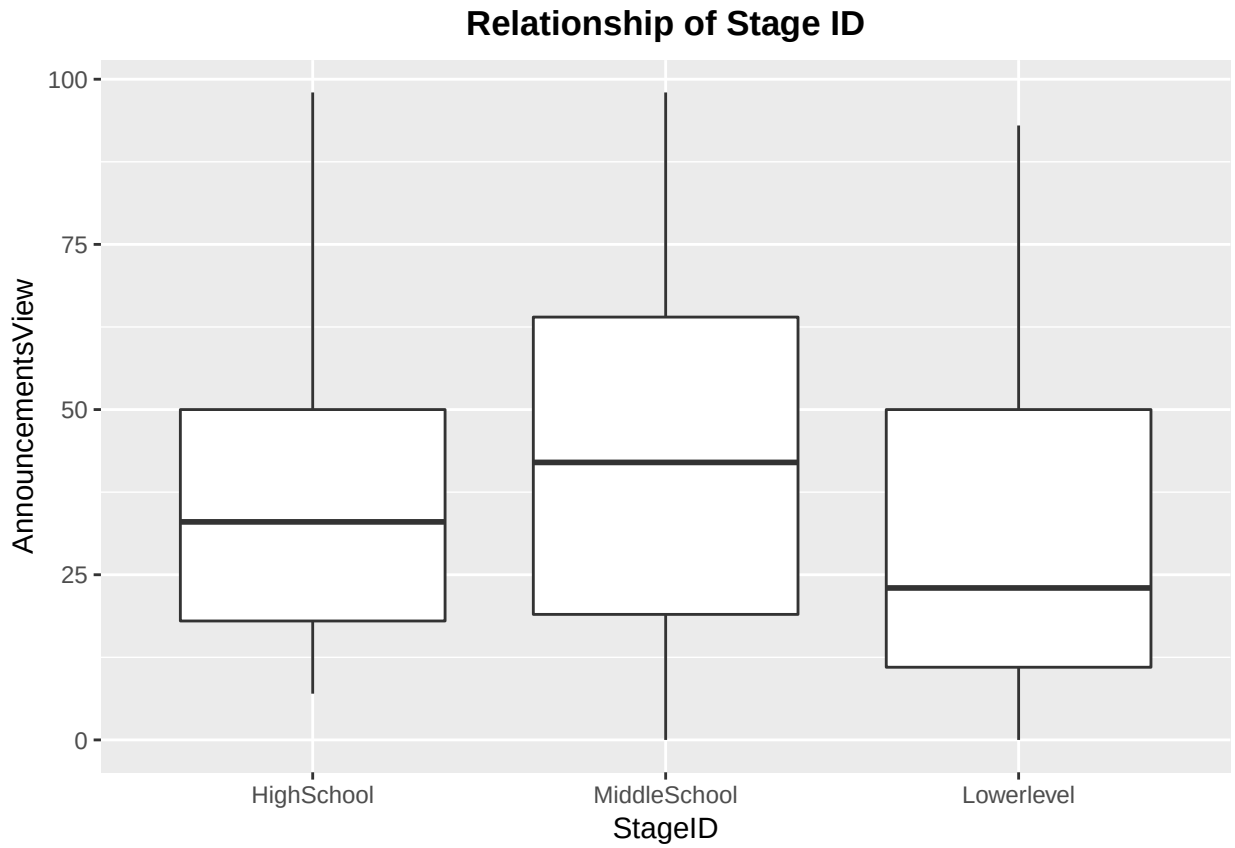
```
ggplot(data = E, aes(x = StageID, y = VisITedResources)) + geom_boxplot() +
    ggtitle("Relationship of Stage ID") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```




```
# More hand raises in middle schools.
```

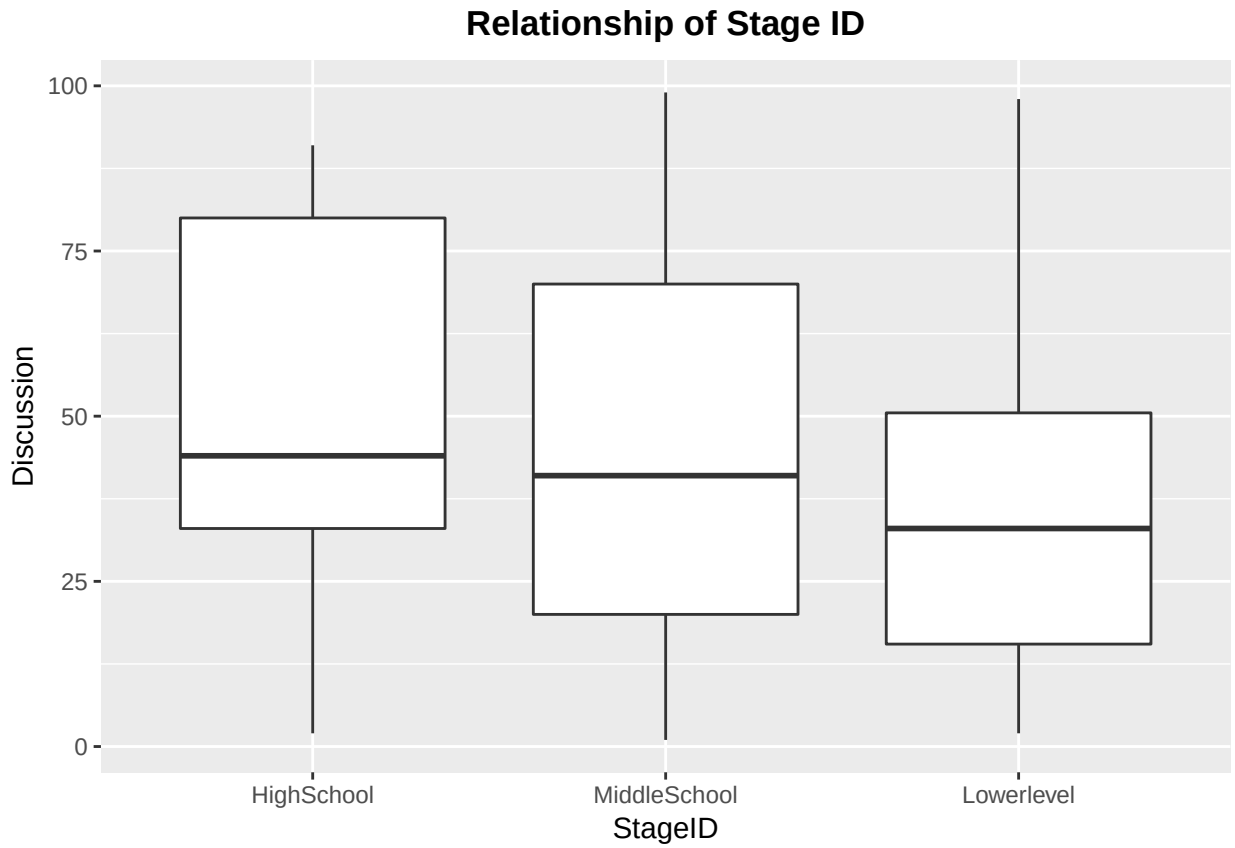
More visiting resouces in middle schools.

```
ggplot(data = E, aes(x = StageID, y = AnnouncementsView)) + geom_boxplot() +
    ggtitle("Relationship of Stage ID") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



More announcement views in middle schools.

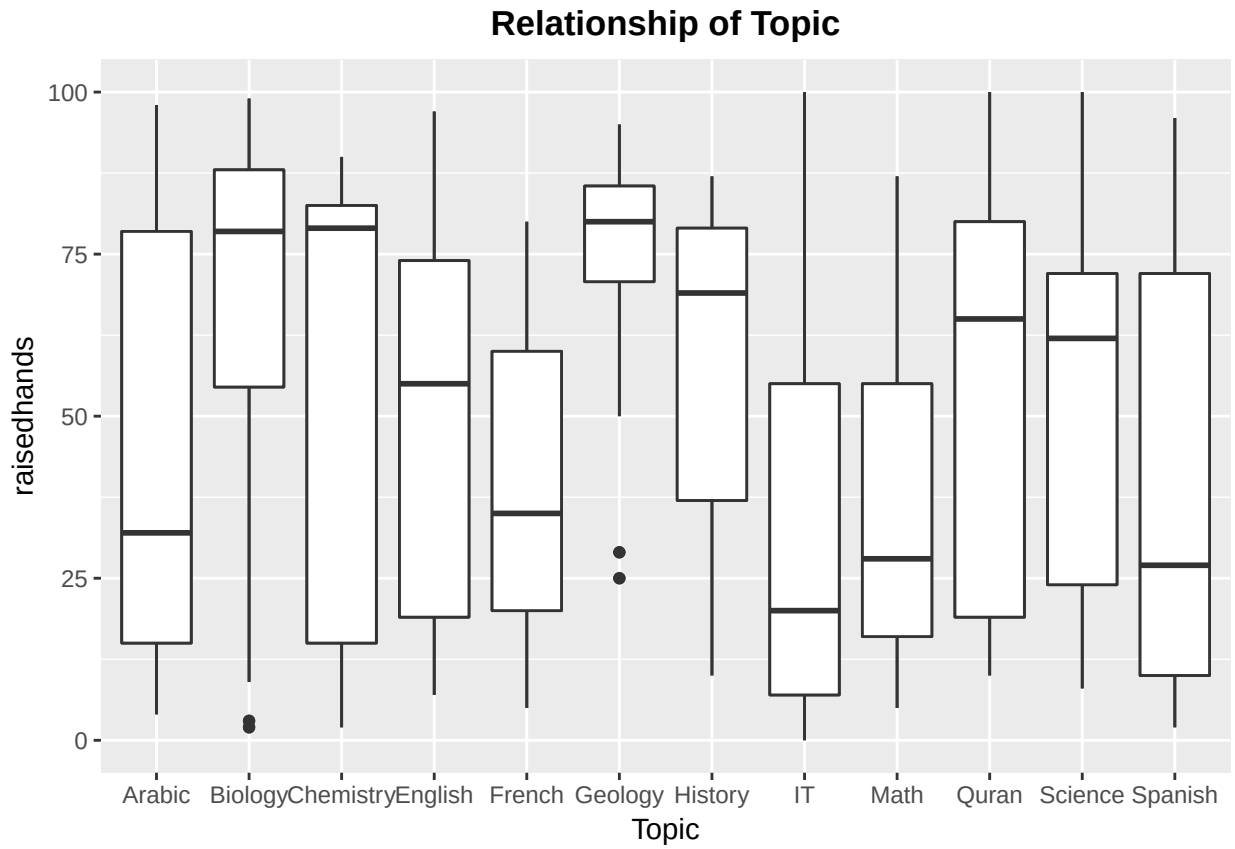
```
ggplot(data = E, aes(x = StageID, y = Discussion)) + geom_boxplot() +
    ggtitle("Relationship of Stage ID") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



More discussions in high schools.

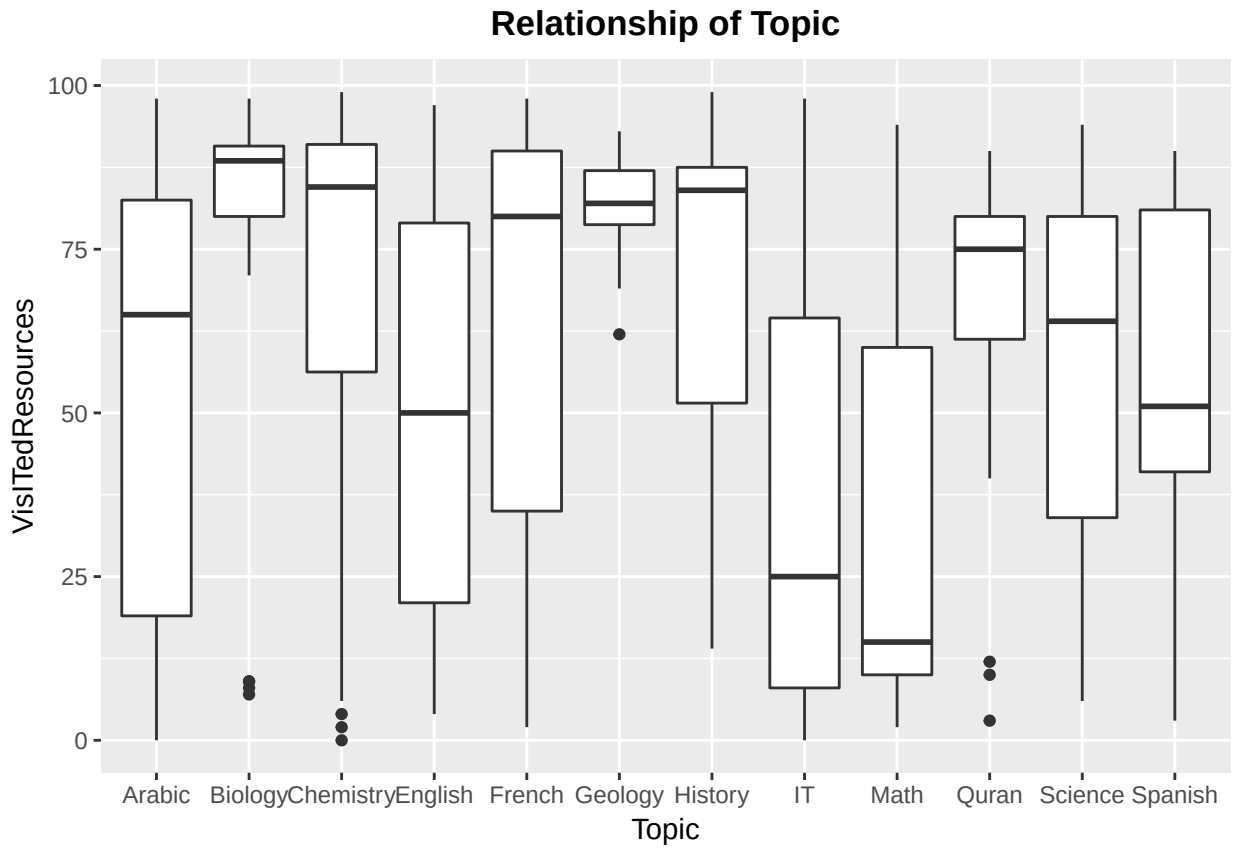
Topic and the numerial features

```
ggplot(data = E, aes(x = Topic, y = raisedhands)) + geom_boxplot() +
    geom_boxplot() + ggtitle("Relationship of Topic") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```

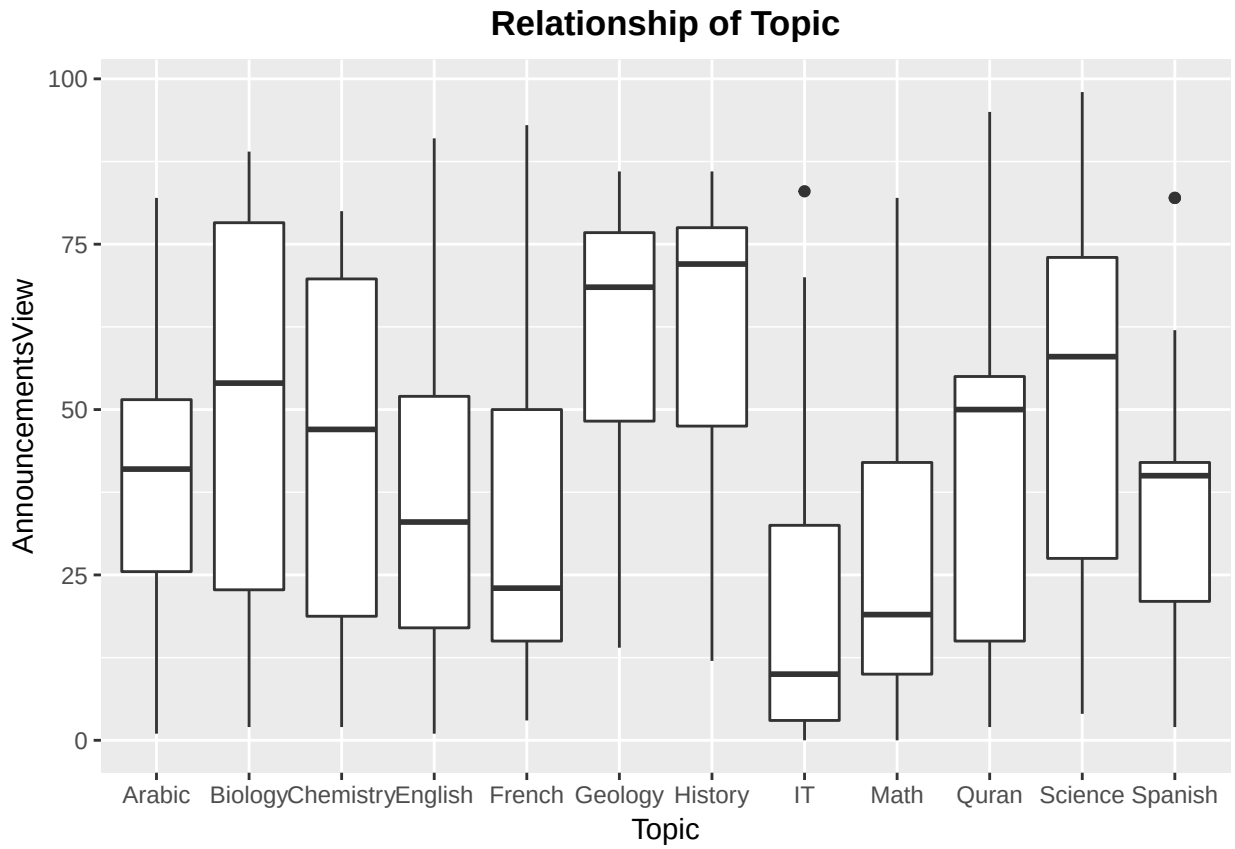


IT has very few hand raises intersetingly although most students study there.

```
ggplot(data = E, aes(x = Topic, y = VisITedResources)) + geom_boxplot() +
    ggtitle("Relationship of Topic") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```

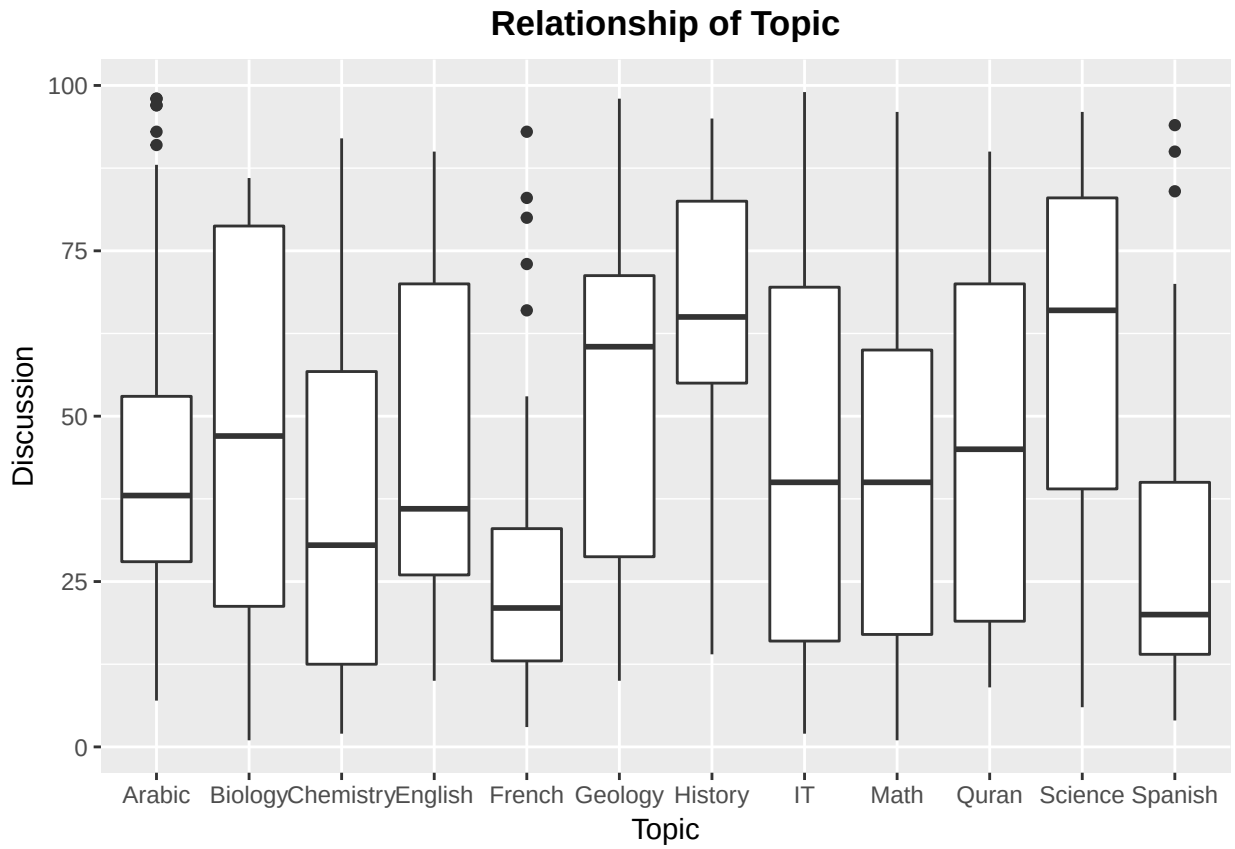



```
ggplot(data = E, aes(x = Topic, y = AnnouncementsView)) + geom_boxplot() +
    ggtitle("Relationship of Topic") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



IT has very few announcement view although most students study there.

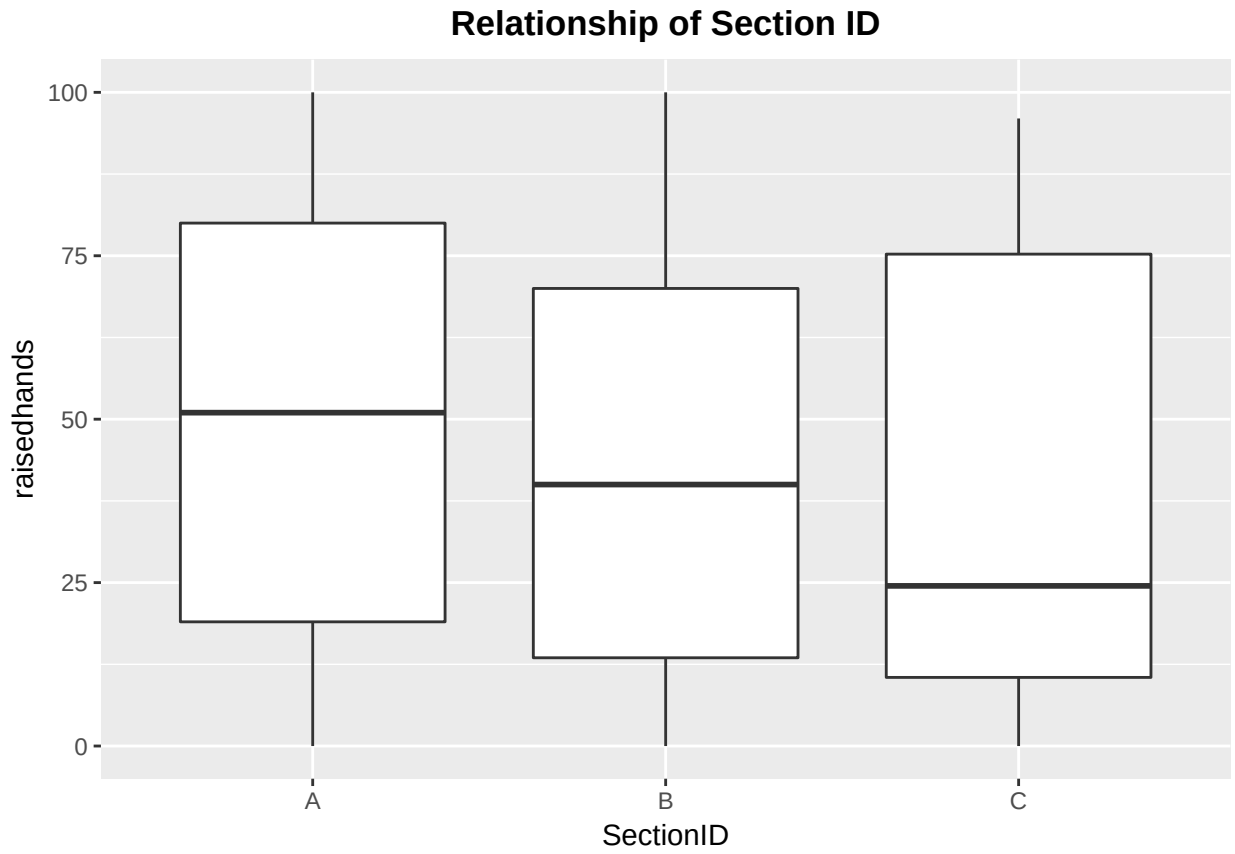
```
ggplot(data = E, aes(x = Topic, y = Discussion)) + geom_boxplot() +
    ggtitle("Relationship of Topic") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



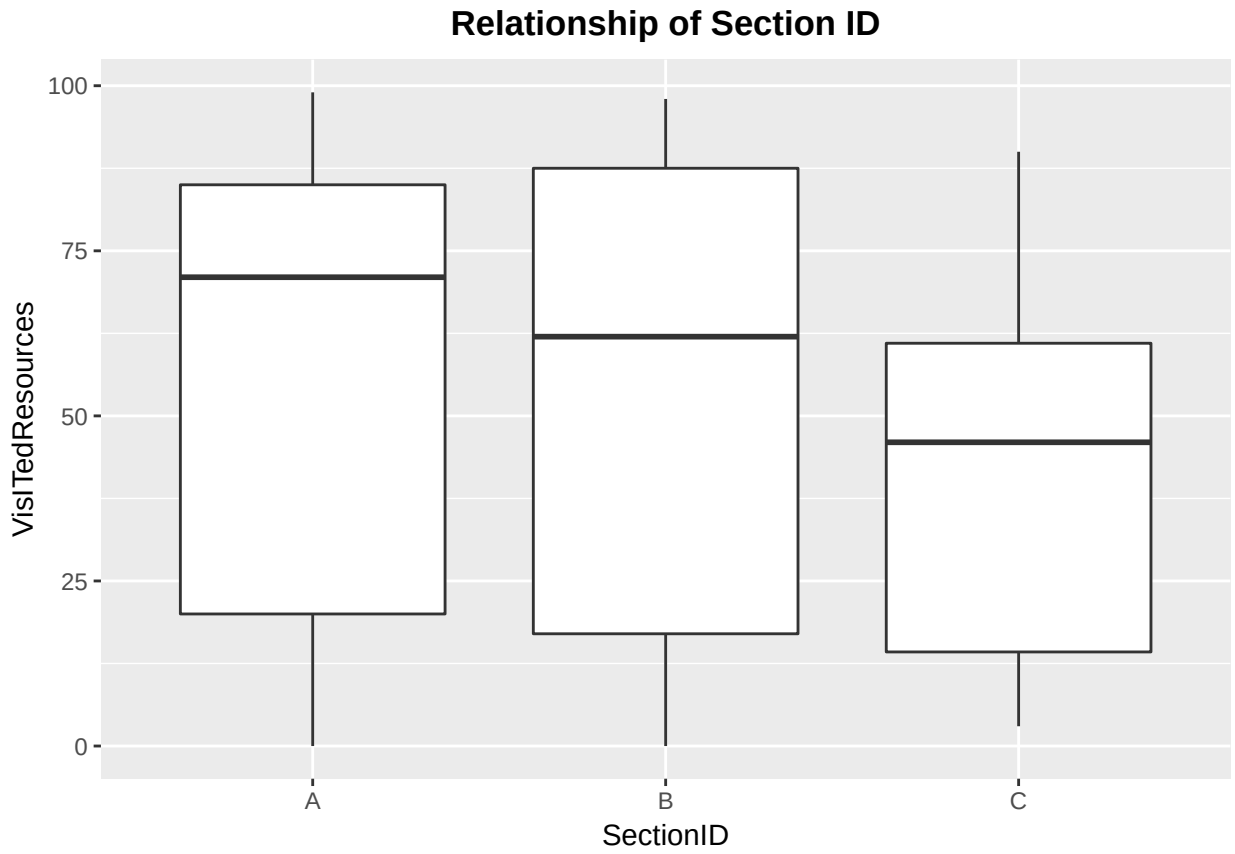
IT has very few discussion although most students study there.

Section ID and the numerial features

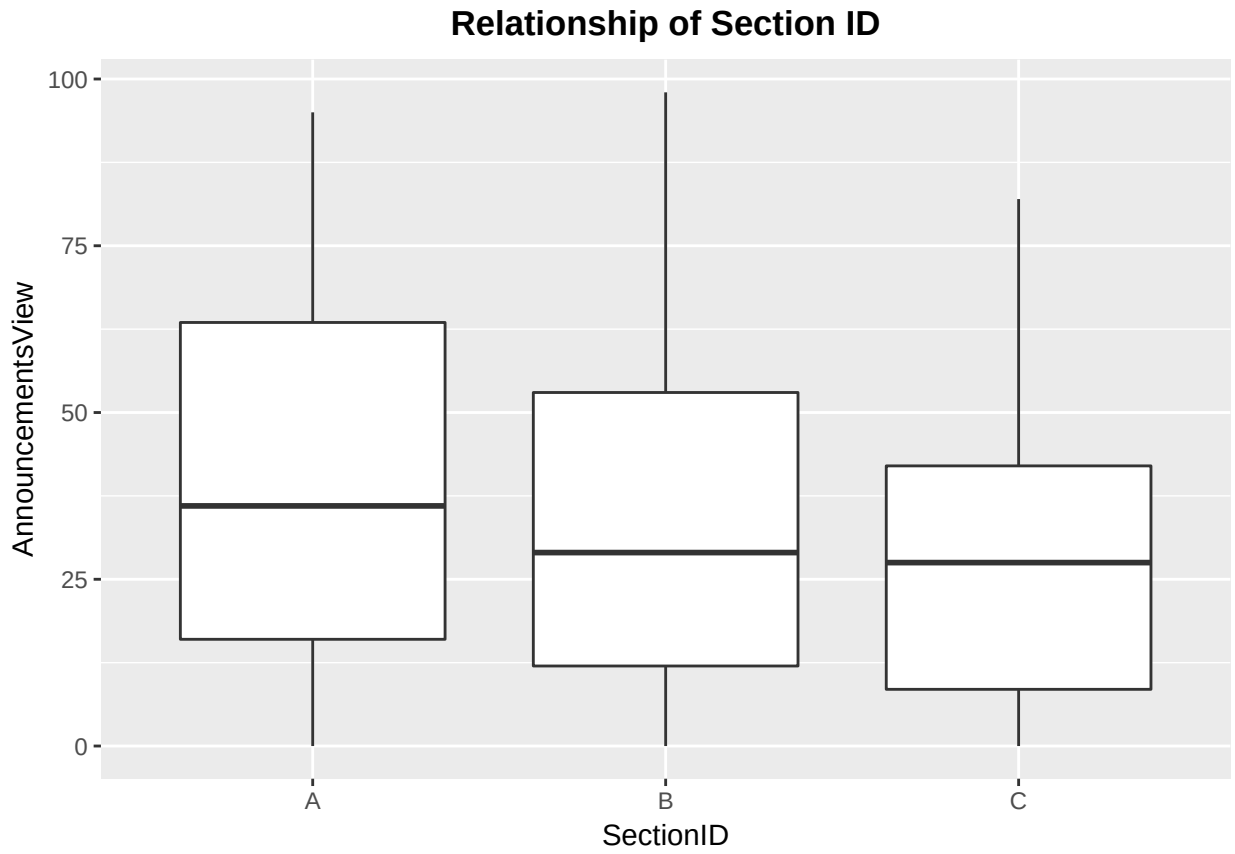
```
ggplot(data = E, aes(x = SectionID, y = raisedhands)) + geom_boxplot() +
    ggtitle("Relationship of Section ID") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



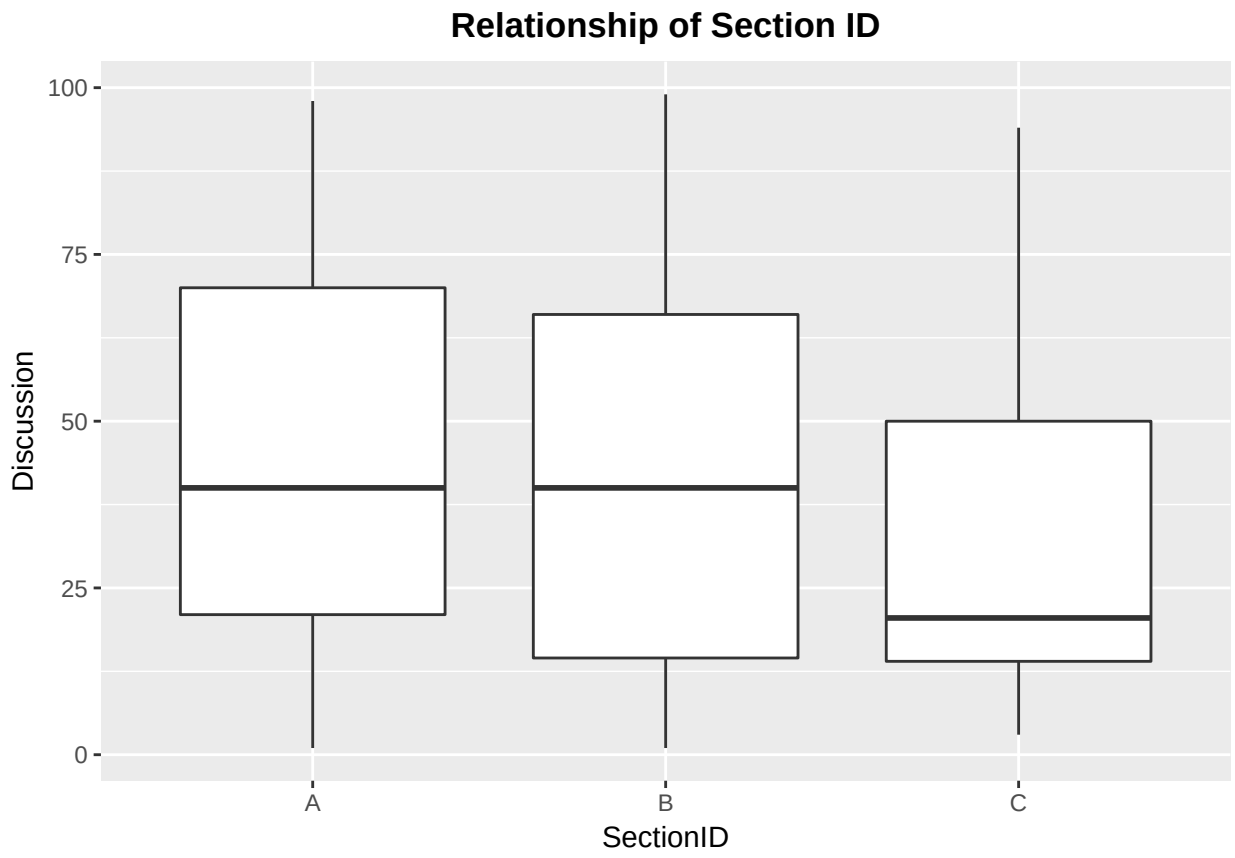

```
ggplot(data = E, aes(x = SectionID, y = VisITedResources)) +
    geom_boxplot() + ggtitle("Relationship of Section ID") +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold"))
```




```
ggplot(data = E, aes(x = SectionID, y = AnnouncementsView)) +
    geom_boxplot() + ggtitle("Relationship of Section ID") +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold"))
```



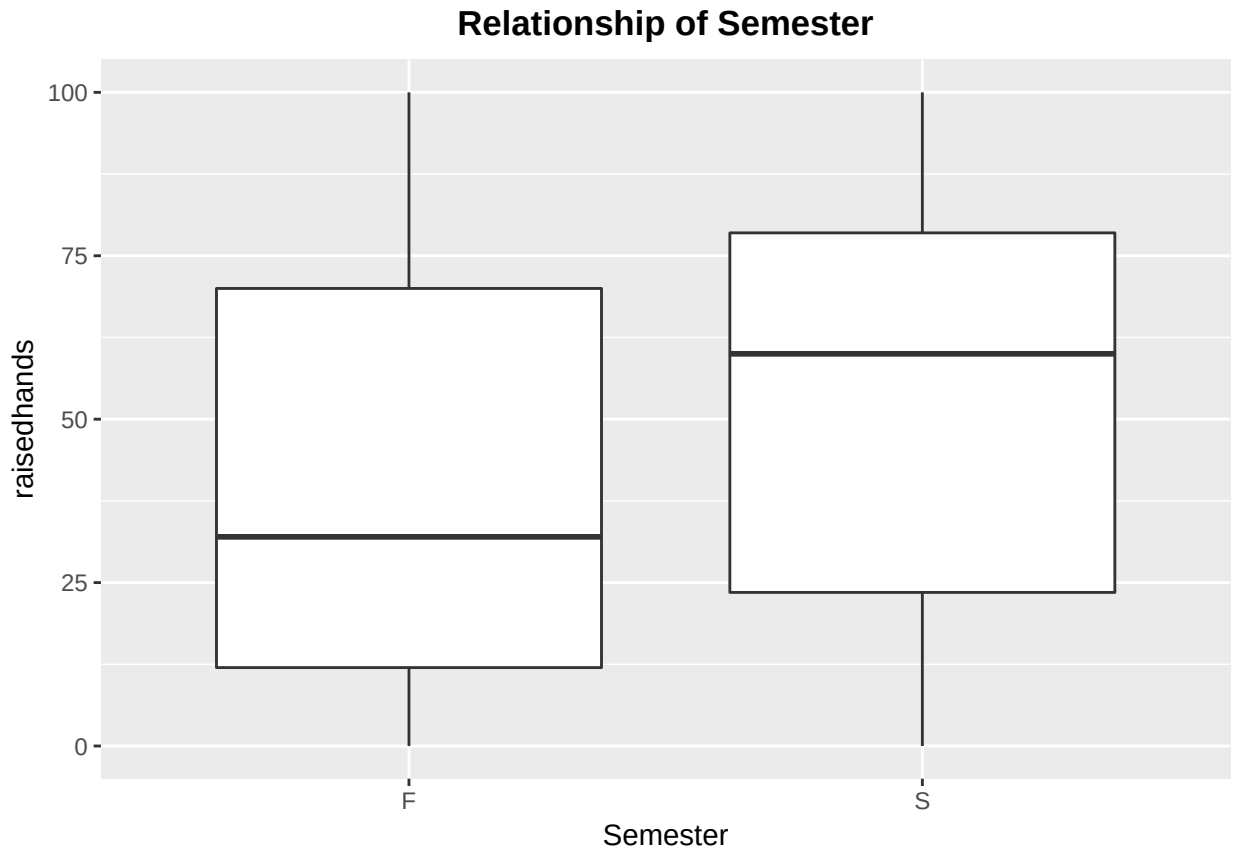

```
ggplot(data = E, aes(x = SectionID, y = Discussion)) + geom_boxplot() +
    ggtitle("Relationship of Section ID") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



Section A has the highest raised hands, visited resources, announcements view and discussion, whereas secion C has the lowest raised hands, visited resources, announcements view and discussion.

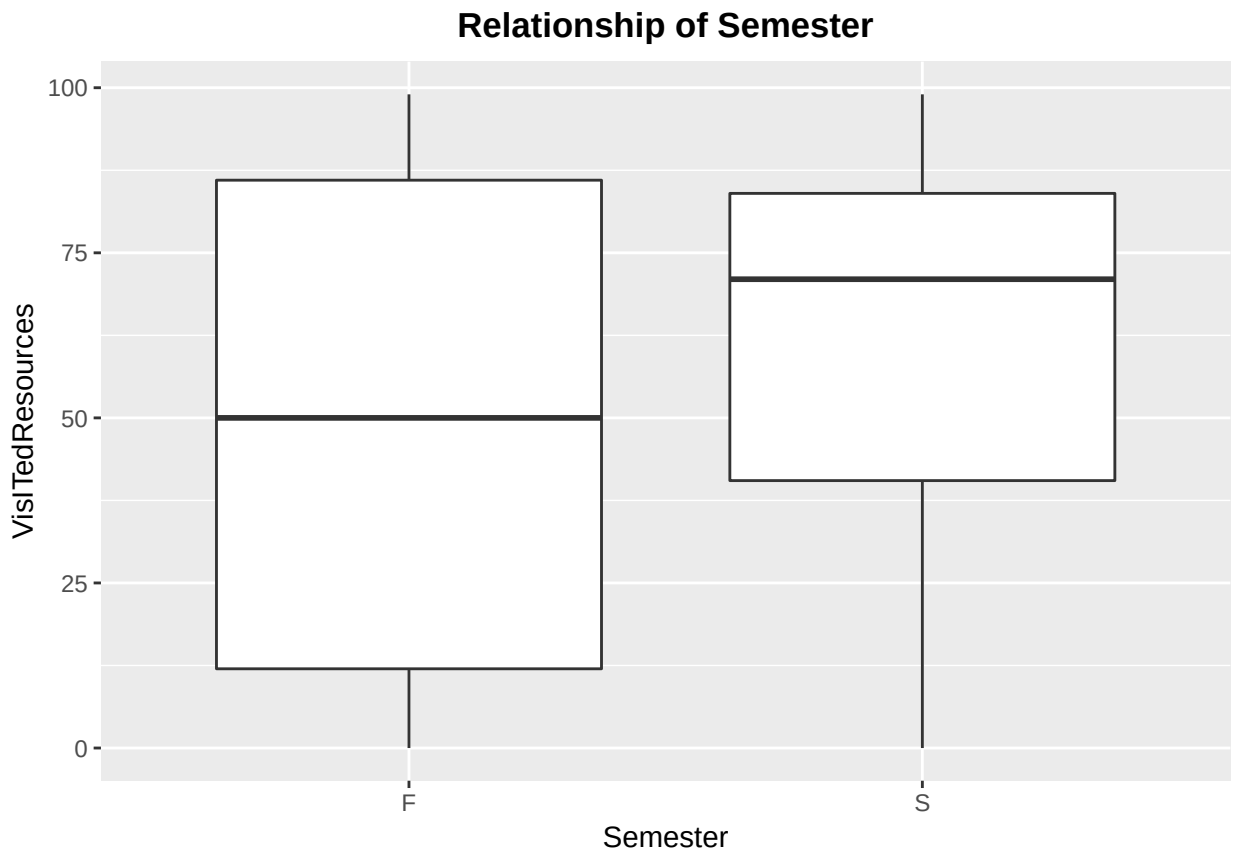
Semester and the numerial features

```
ggplot(data = E, aes(x = Semester, y = raisedhands)) + geom_boxplot() +
    ggtitle("Relationship of Semester") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



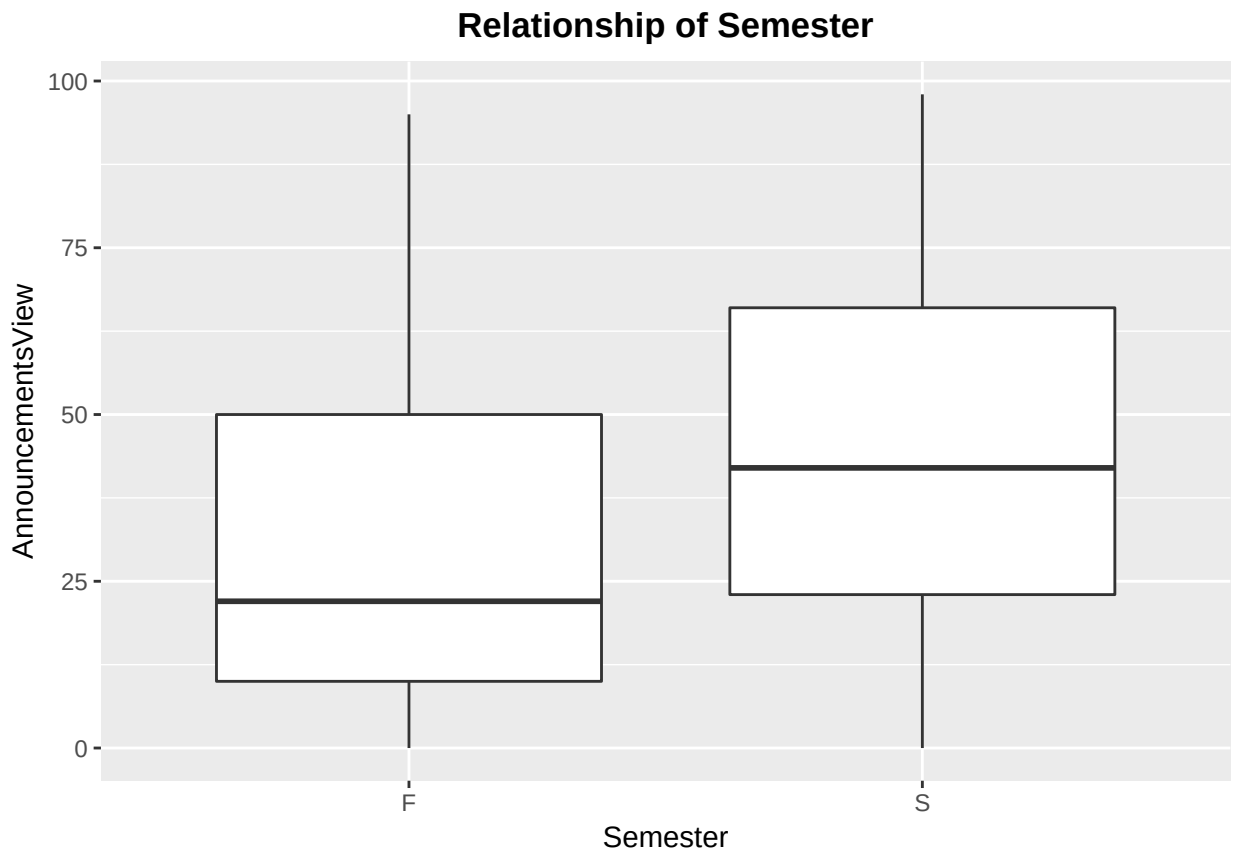
Second semester has more hand raises

```
ggplot(data = E, aes(x = Semester, y = VisITedResources)) + geom_boxplot() +
    ggtitle("Relationship of Semester") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



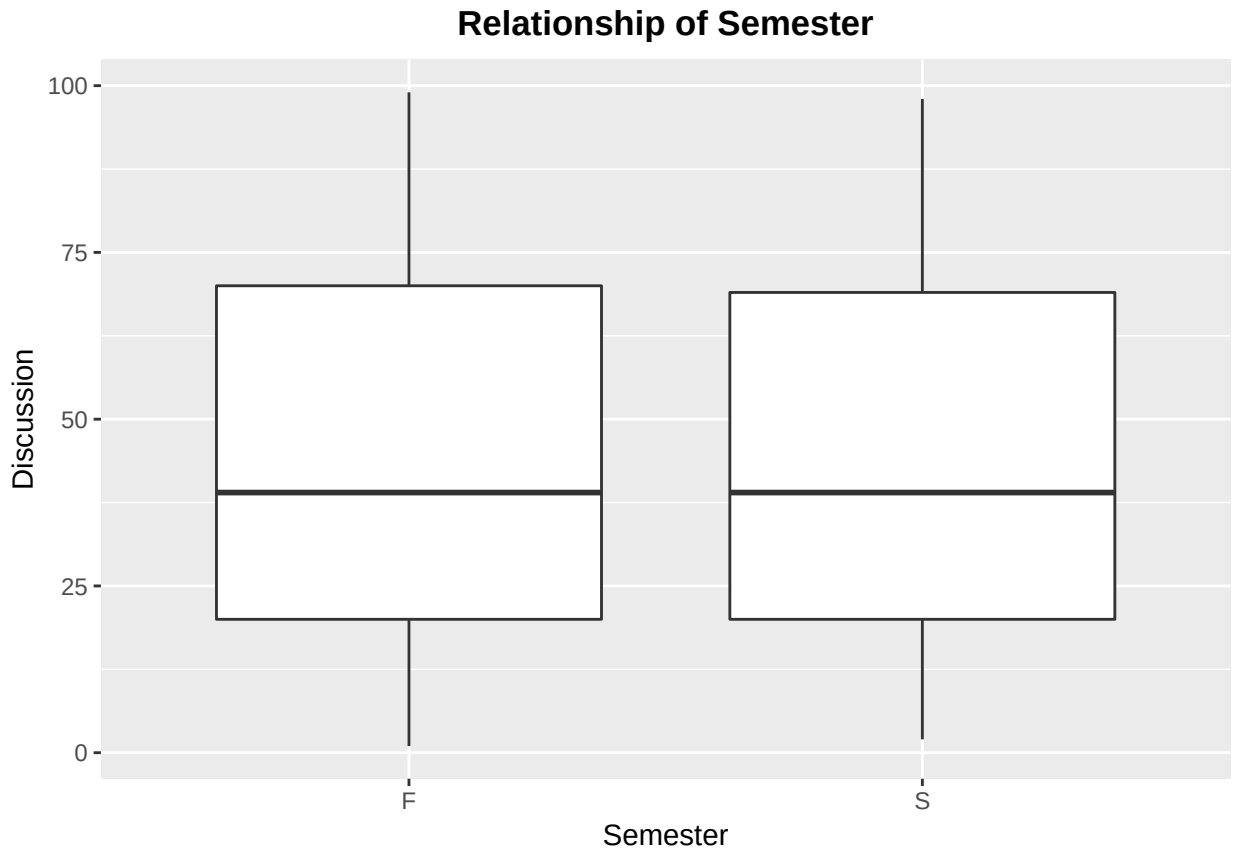
Second semester has more visited resources

```
ggplot(data = E, aes(x = Semester, y = AnnouncementsView)) +
    geom_boxplot() + ggtitle("Relationship of Semester") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



Second semester has more announcements view

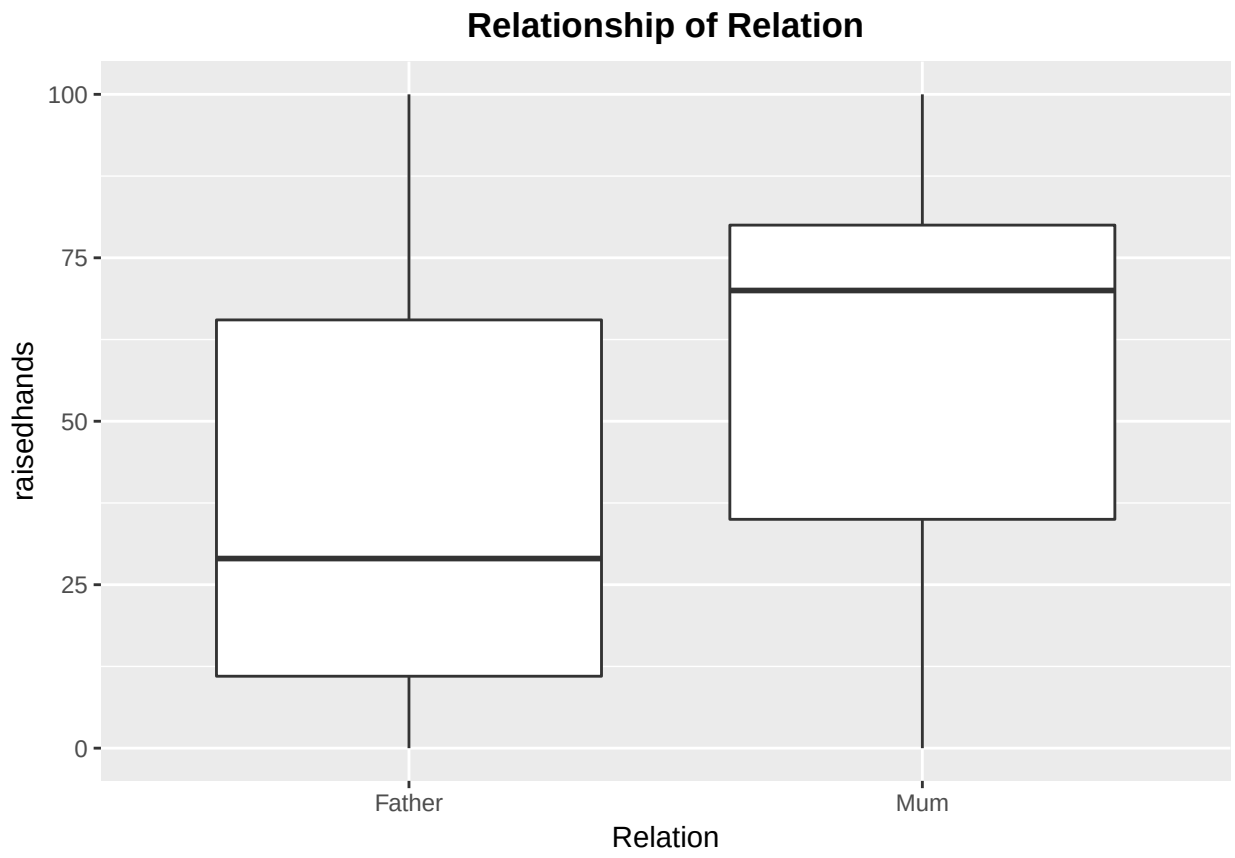
```
ggplot(data = E, aes(x = Semester, y = Discussion)) + geom_boxplot() +
    ggtitle("Relationship of Semester") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



Almost no differences between these two semesters in discussion.

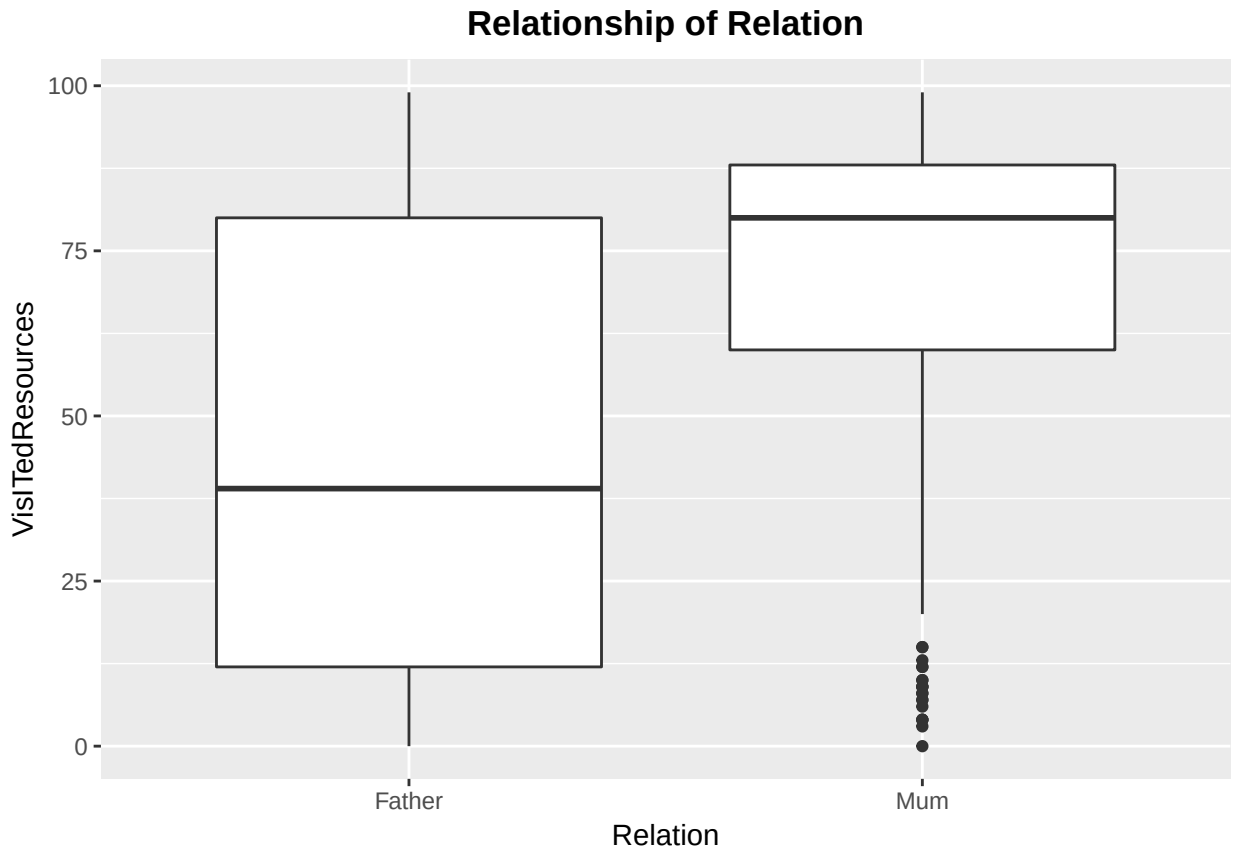
Relation and the numerial features

```
ggplot(data = E, aes(x = Relation, y = raisedhands)) + geom_boxplot() +
    ggtitle("Relationship of Relation") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



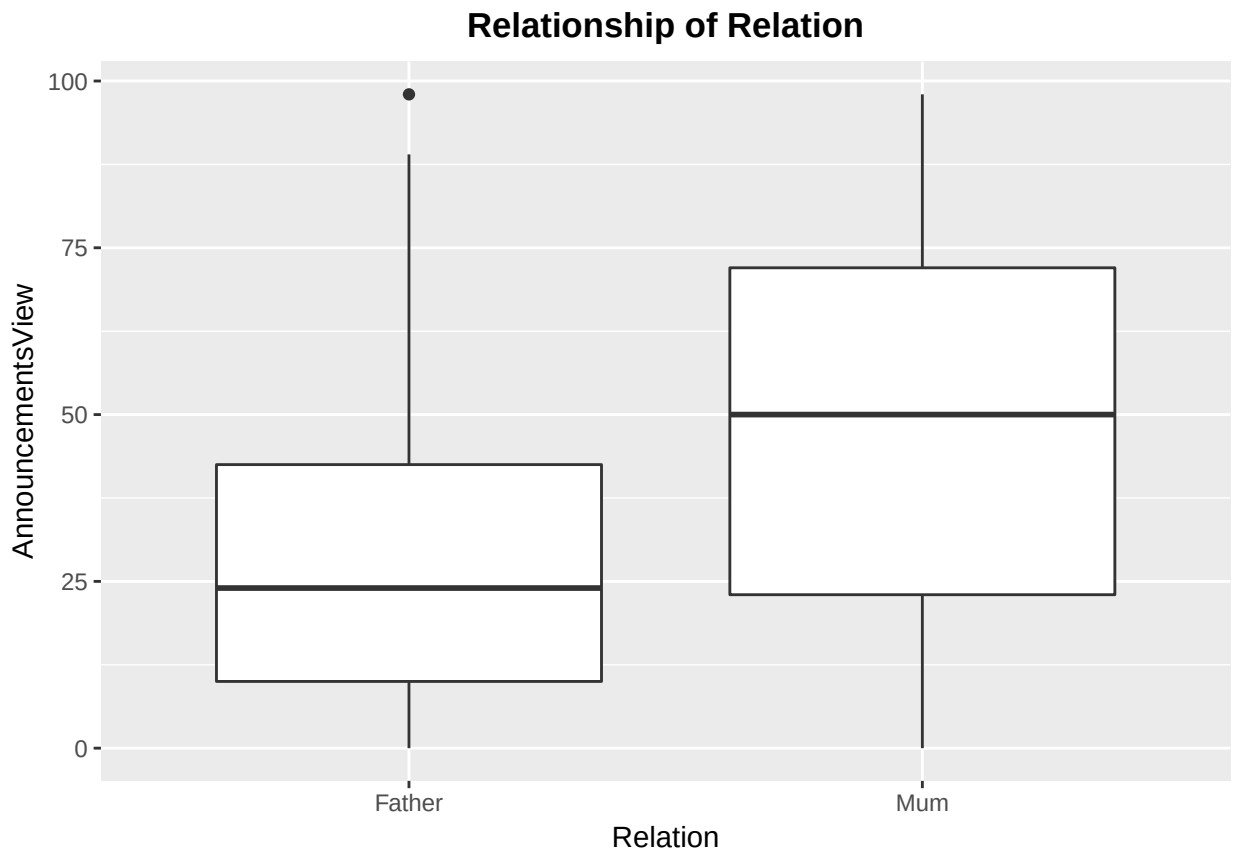
Students with Guardians mother have more hand raises.

```
ggplot(data = E, aes(x = Relation, y = VisITedResources)) + geom_boxplot() +
    ggtitle("Relationship of Relation") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



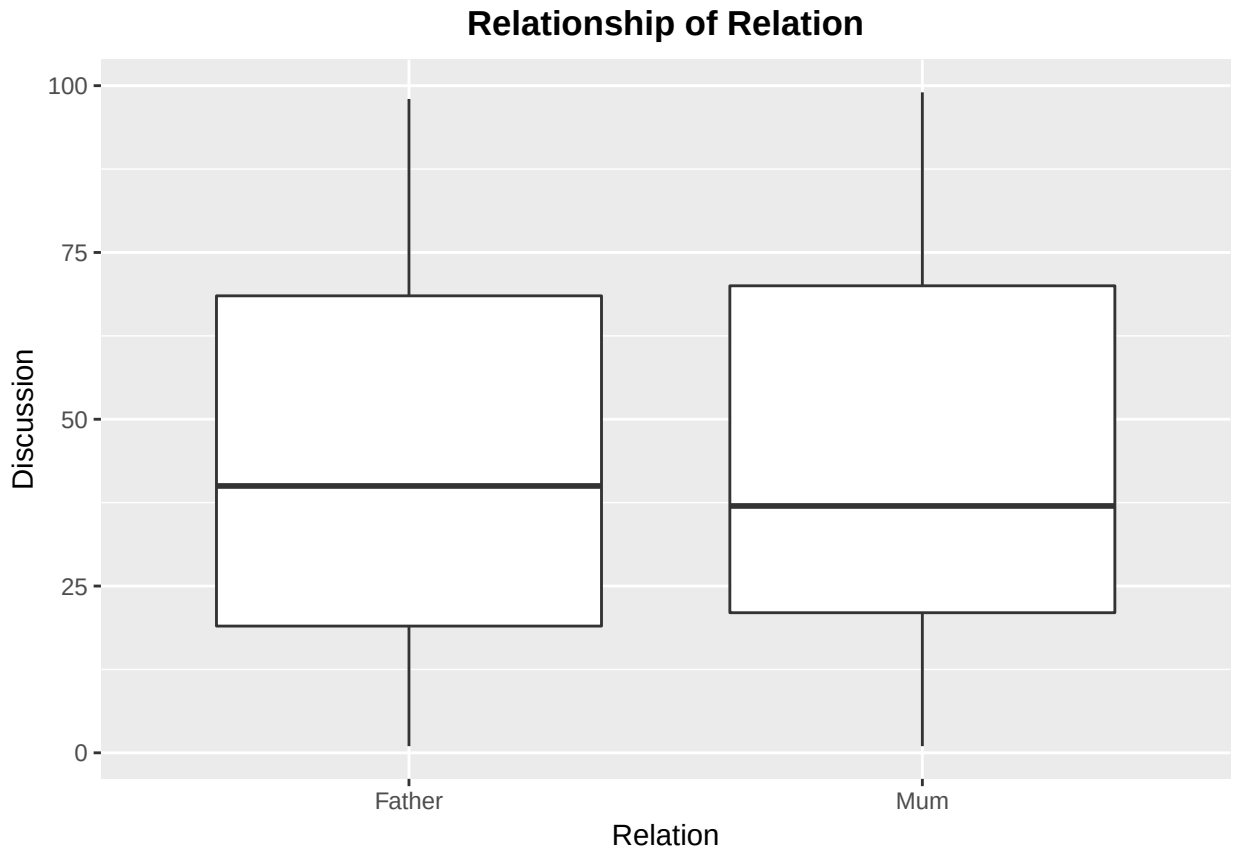
Students with Guardians mother have more visited resources.

```
ggplot(data = E, aes(x = Relation, y = AnnouncementsView)) +
    geom_boxplot() + ggtitle("Relationship of Relation") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



Students with Guardians mother have more announcements view.

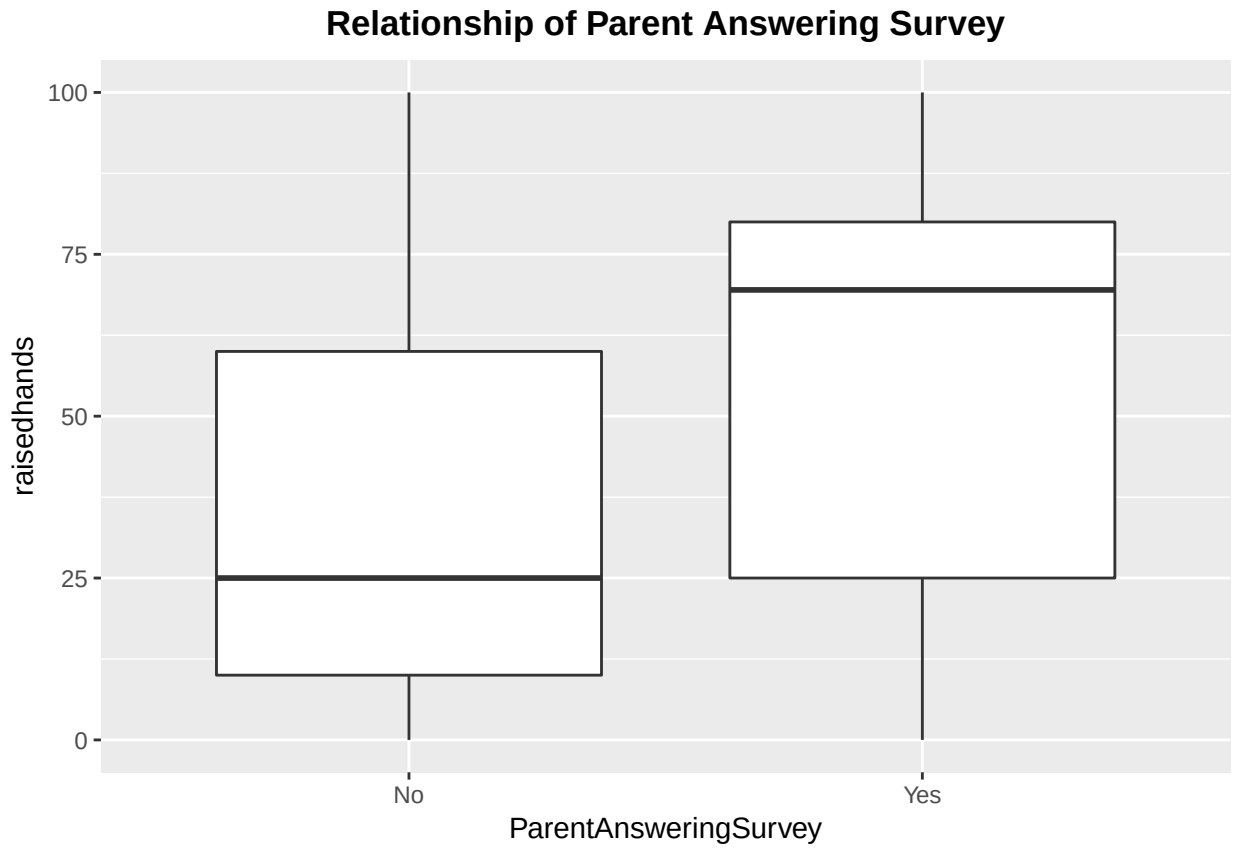
```
ggplot(data = E, aes(x = Relation, y = Discussion)) + geom_boxplot() +
    ggtitle("Relationship of Relation") + theme(plot.title = element_text(hjust = 0.5,
    lineheight = 0.8, face = "bold"))
```



No obvious differences between students with differen guardians in discussion.

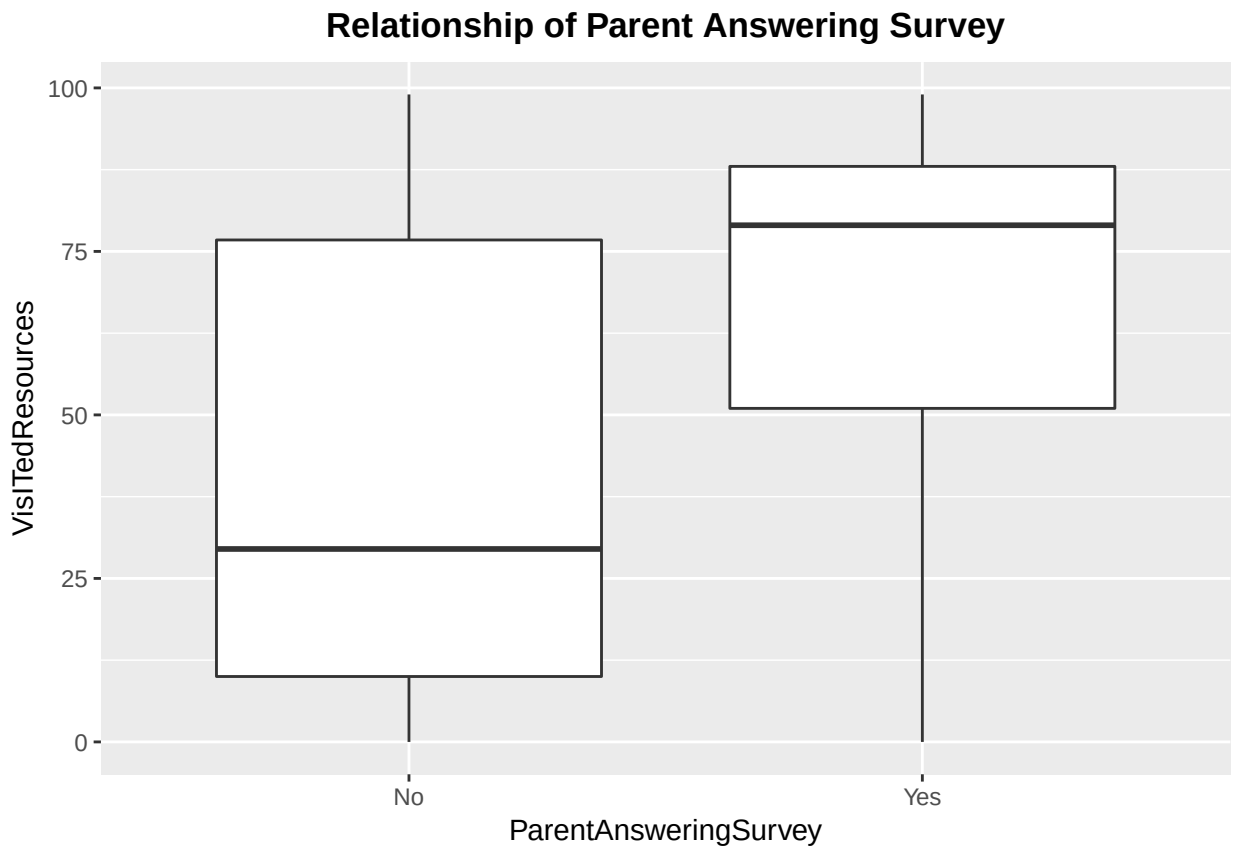
Parent Answering Survey and the numerial features

```
ggplot(data = E, aes(x = ParentAnsweringSurvey, y = raisedhands)) +
    geom_boxplot() + ggtitle("Relationship of Parent Answering Survey") +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold"))
```



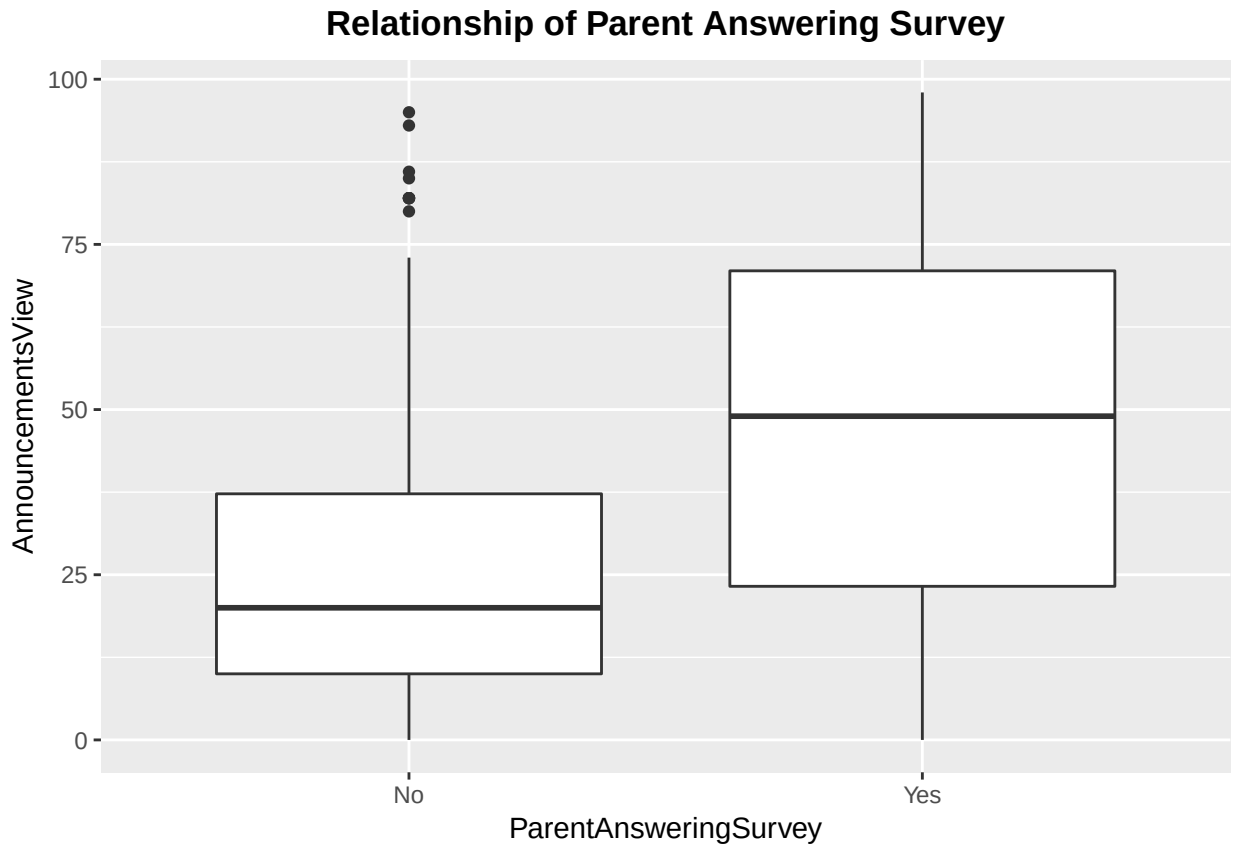
Students whose parent answering survey have more hand raises

```
ggplot(data = E, aes(x = ParentAnsweringSurvey, y = VisITedResources)) +
    geom_boxplot() + ggtitle("Relationship of Parent Answering Survey") +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold"))
```



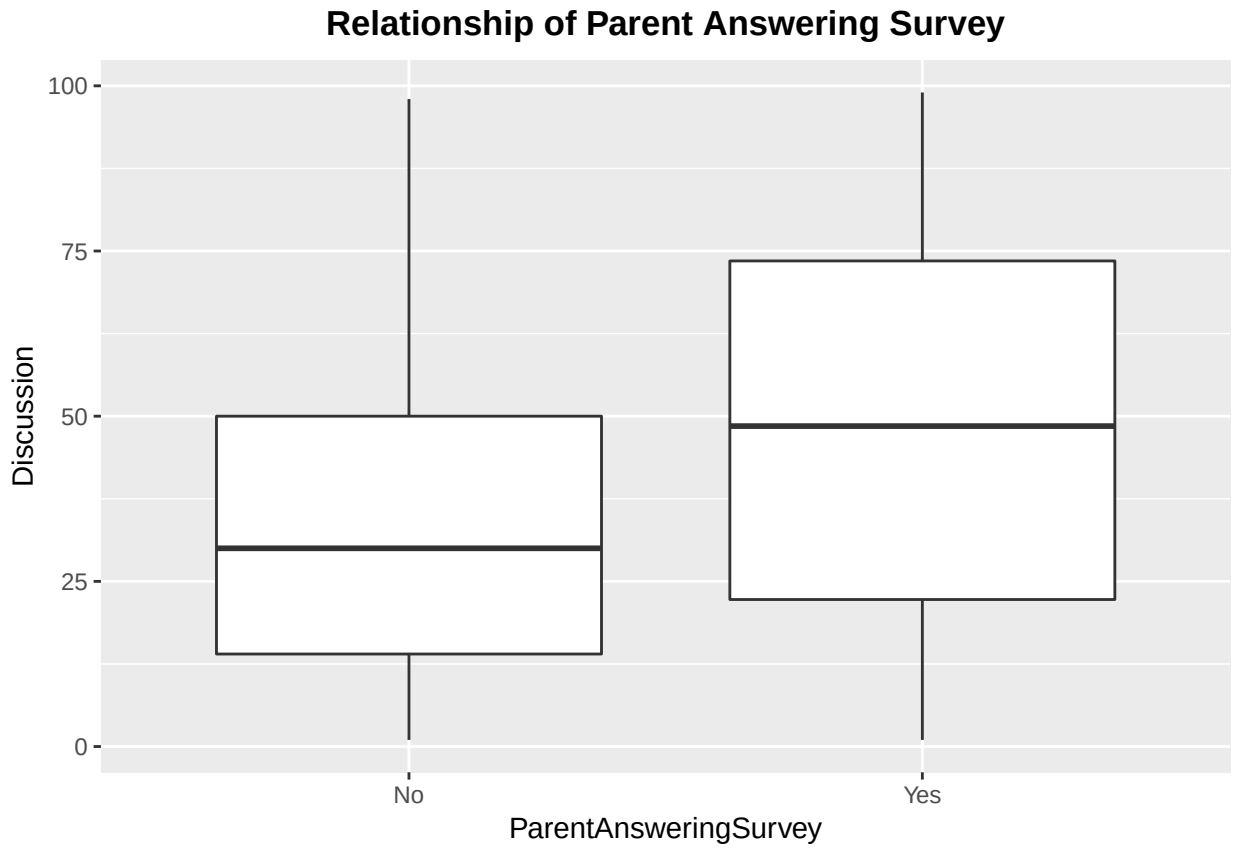
Students whose parent answering survey have more visited resources.

```
ggplot(data = E, aes(x = ParentAnsweringSurvey, y = AnnouncementsView)) +
    geom_boxplot() + ggtitle("Relationship of Parent Answering Survey") +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold"))
```



Students whose parent answering survey have more announcements view.

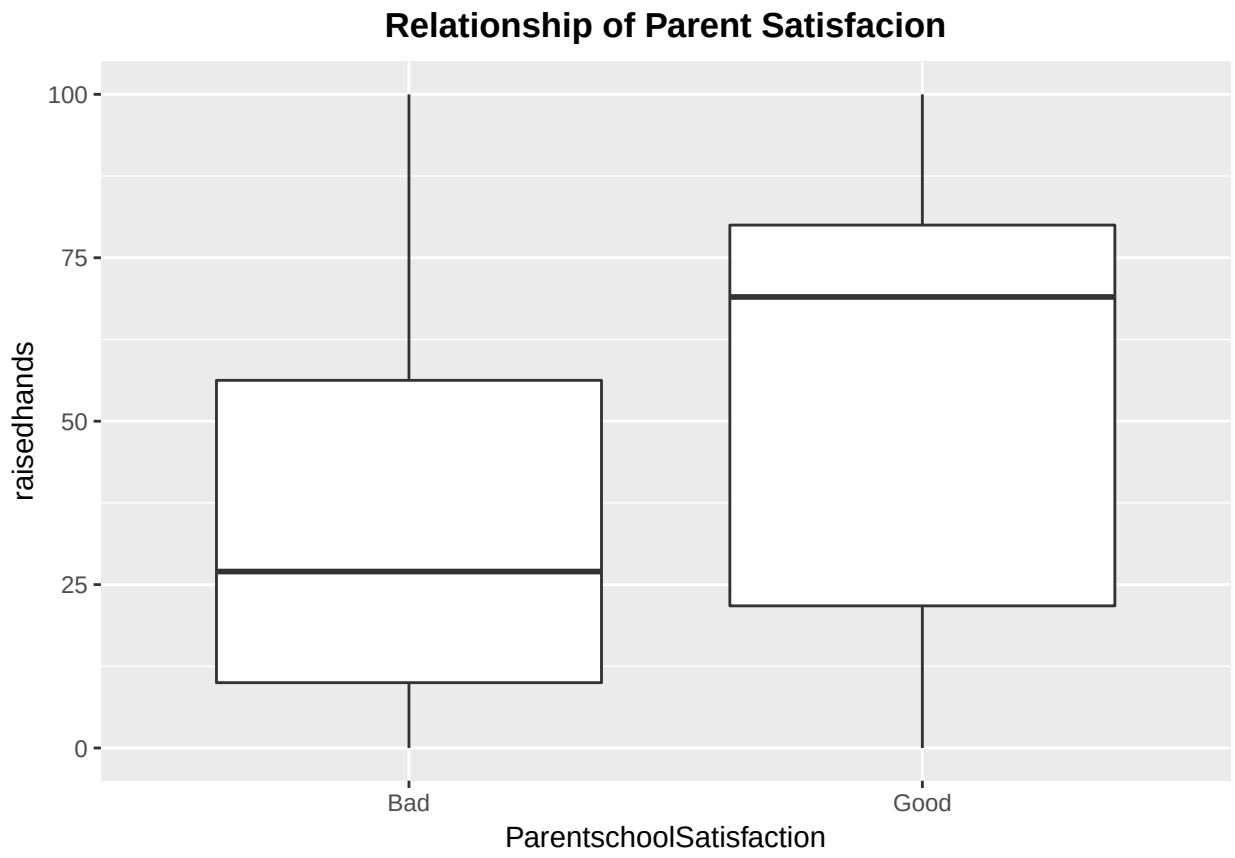
```
ggplot(data = E, aes(x = ParentAnsweringSurvey, y = Discussion)) +
    geom_boxplot() + ggtitle("Relationship of Parent Answering Survey") +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold"))
```



Students whose parent answering survey have more discussion

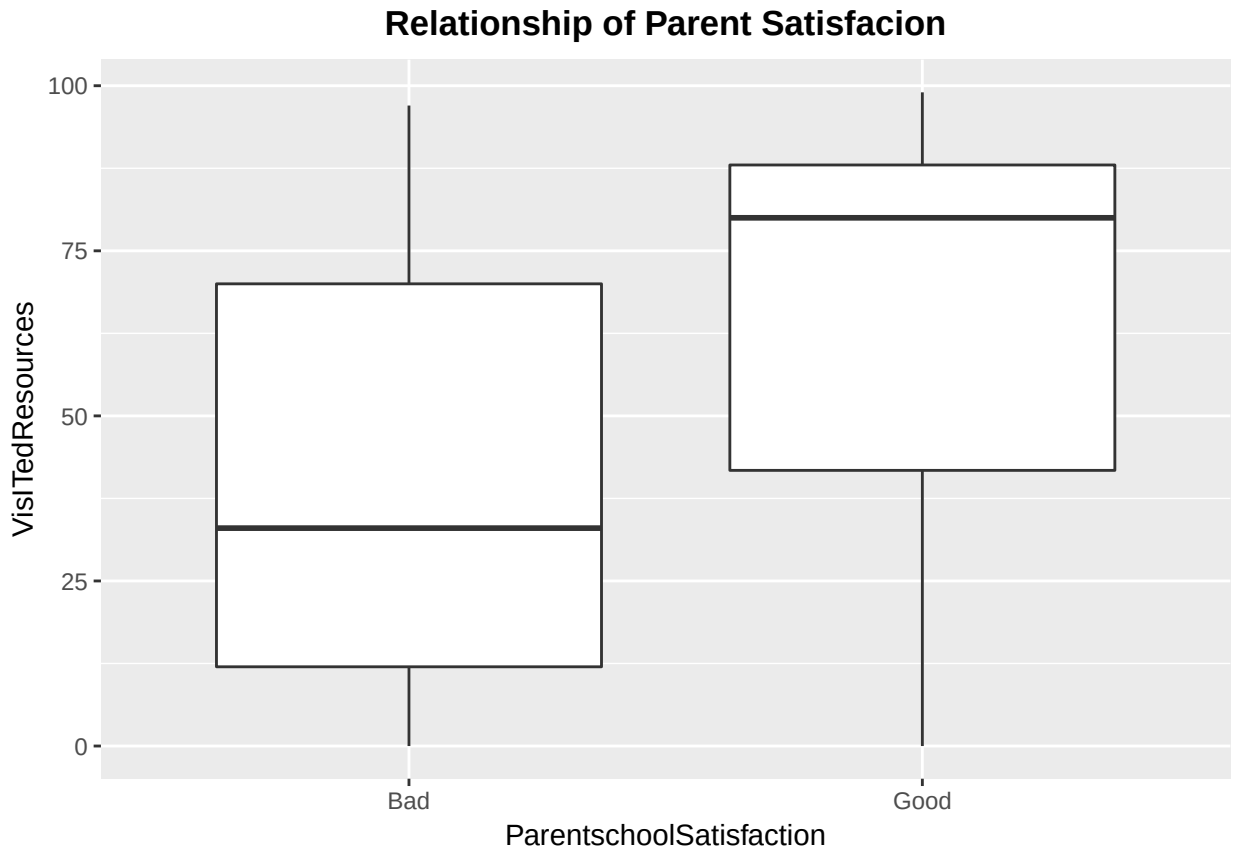
Parent school Satisfaction and the numerial features

```
ggplot(data = E, aes(x = ParentschoolSatisfaction, y = raisedhands)) +
    geom_boxplot() + ggtitle("Relationship of Parent Satisfacion") +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold"))
```



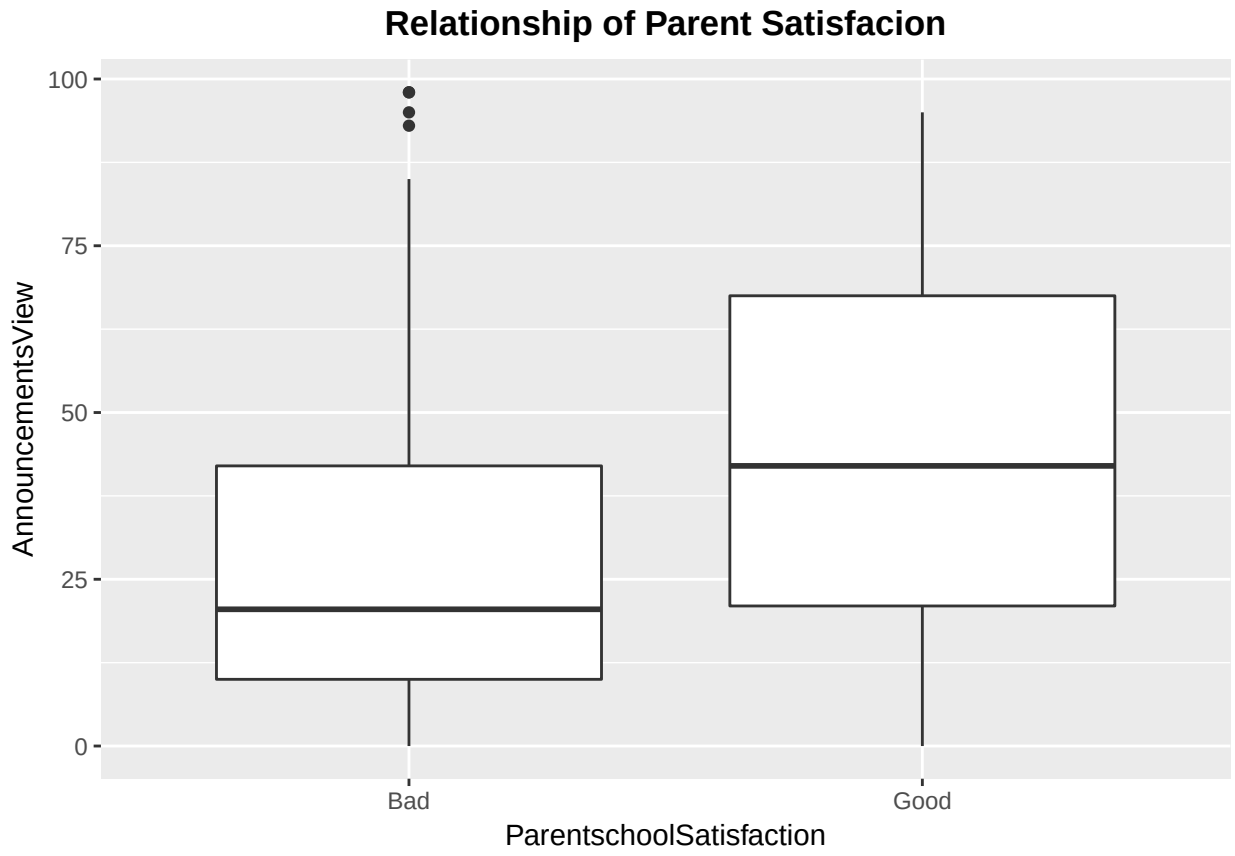
Students whose parents with good satisfaction have more hand raises.

```
ggplot(data = E, aes(x = ParentschoolSatisfaction, y = VisITedResources)) +
    geom_boxplot() + ggtitle("Relationship of Parent Satisfacion") +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold"))
```



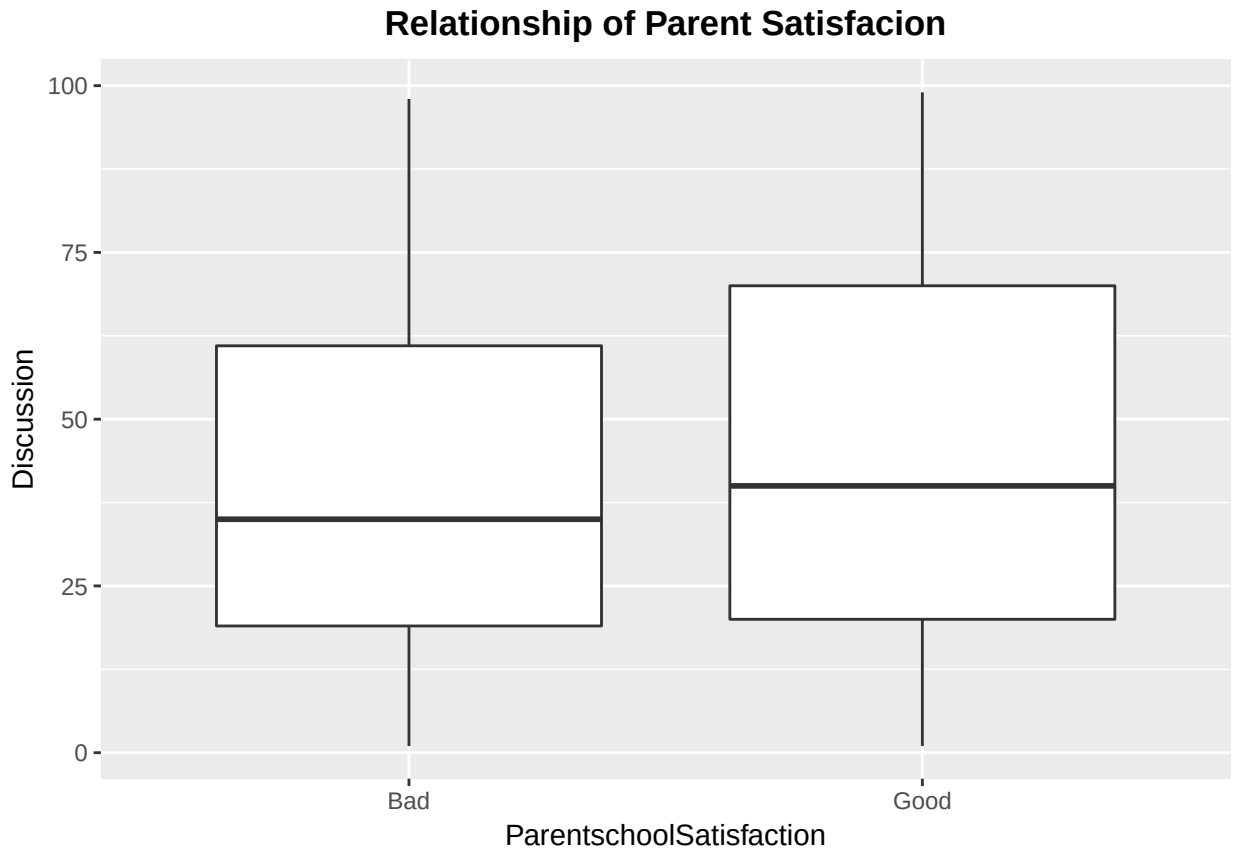
Students whose parents with good satisfaction have more visited resources.

```
ggplot(data = E, aes(x = ParentschoolSatisfaction, y = AnnouncementsView)) +
    geom_boxplot() + ggtitle("Relationship of Parent Satisfacion") +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold"))
```



Students whose parents with good satisfaction have more announcement view.

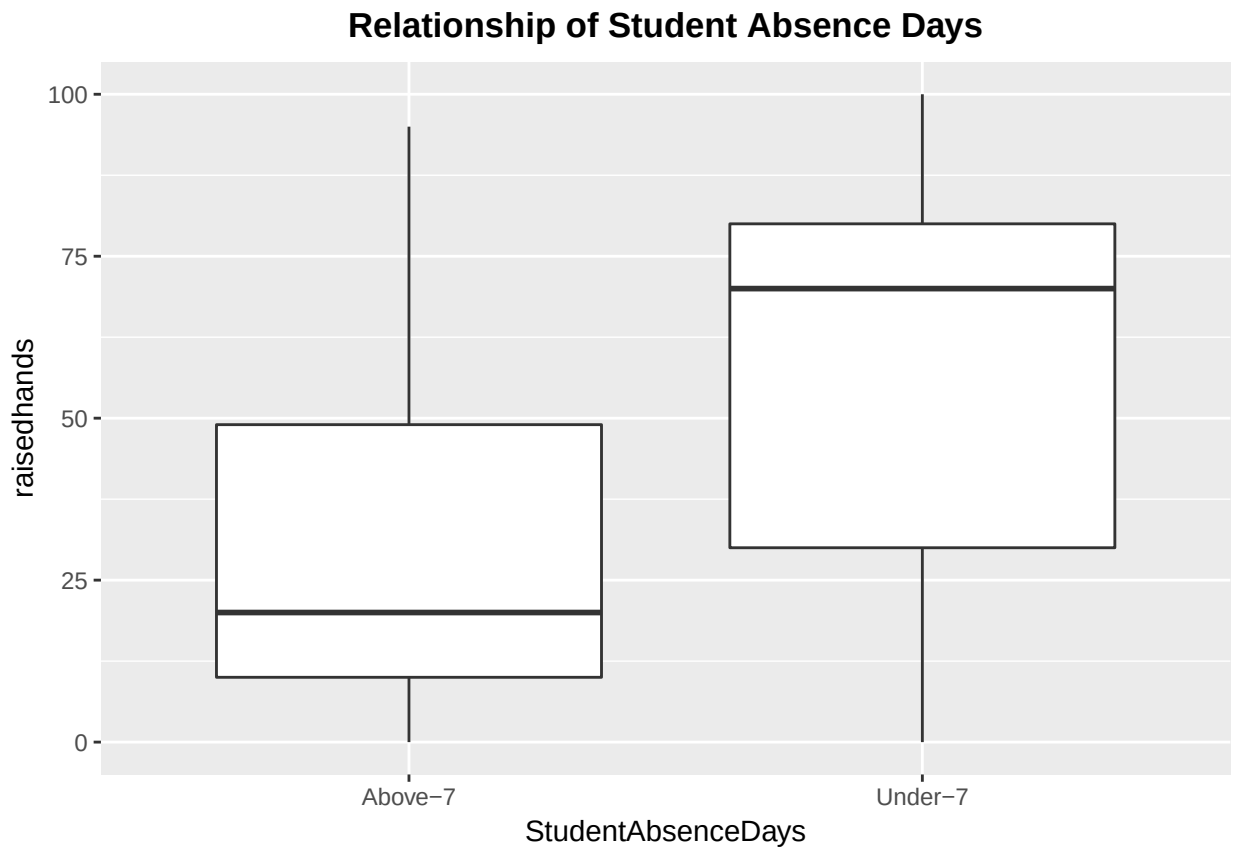
```
ggplot(data = E, aes(x = ParentschoolSatisfaction, y = Discussion)) +
    geom_boxplot() + ggtitle("Relationship of Parent Satisfacion") +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold"))
```



No obvious differences in discution between students whose parents with different satisfaction .

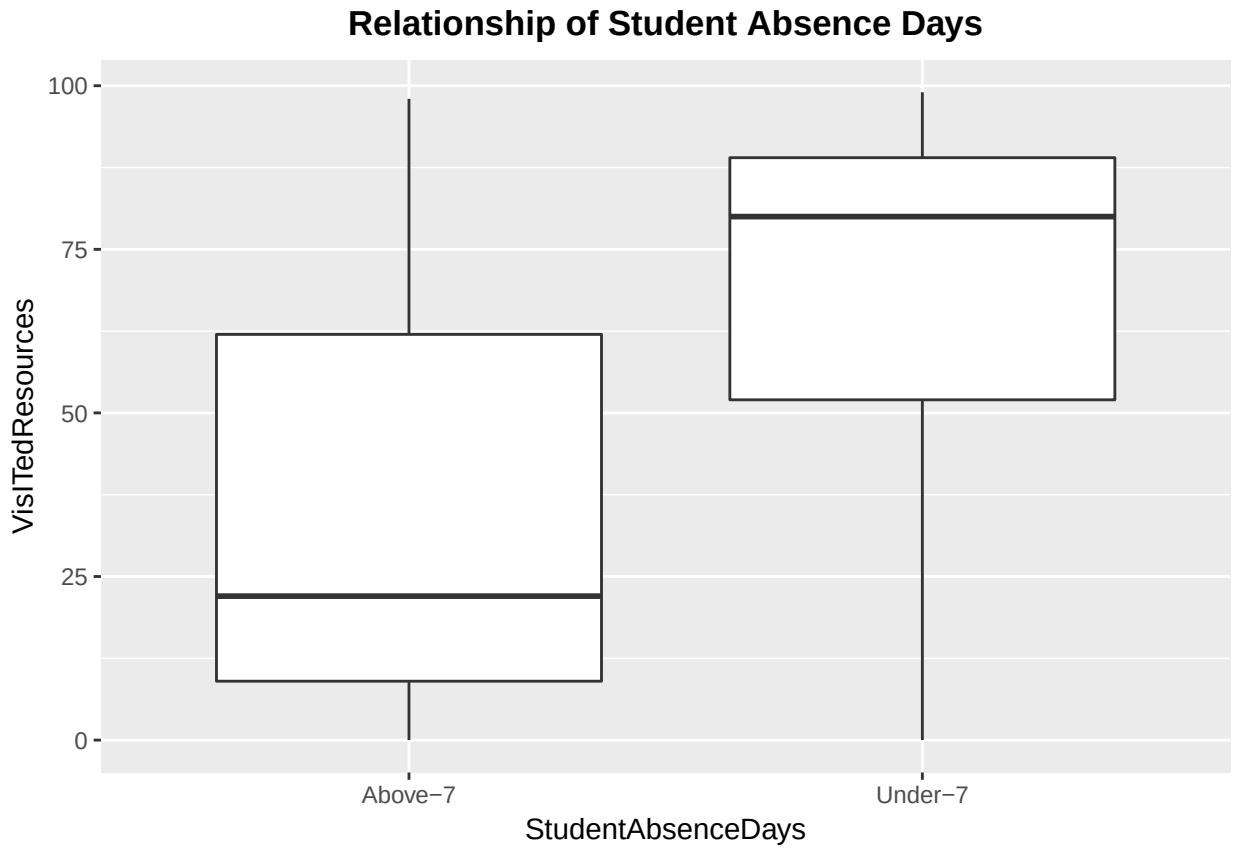
Student Absence Days and the numerial features

```
ggplot(data = E, aes(x = StudentAbsenceDays, y = raisedhands)) +
    geom_boxplot() + ggtitle("Relationship of Student Absence Days") +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold"))
```



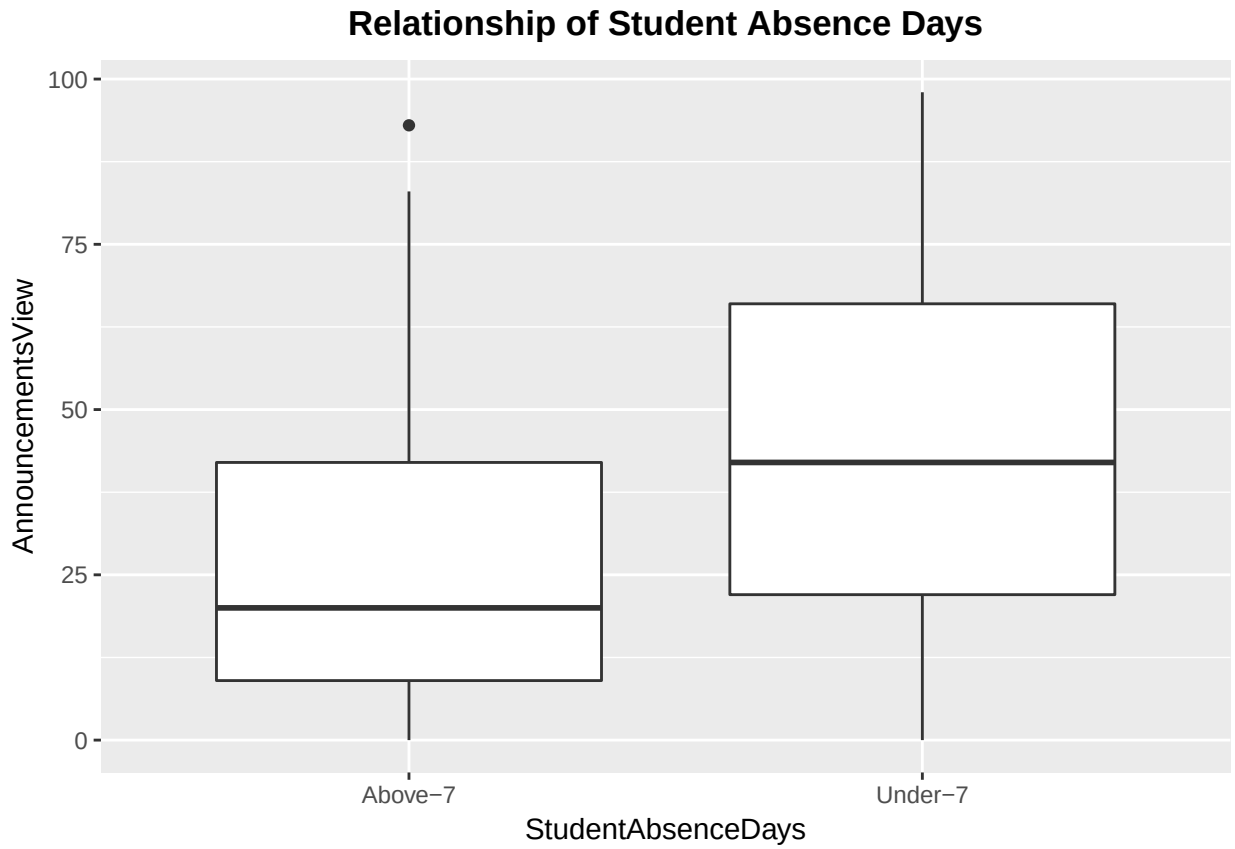
Students have the more leaves the less hand raises.

```
ggplot(data = E, aes(x = StudentAbsenceDays, y = VisITedResources)) +
    geom_boxplot() + ggtitle("Relationship of Student Absence Days") +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold"))
```



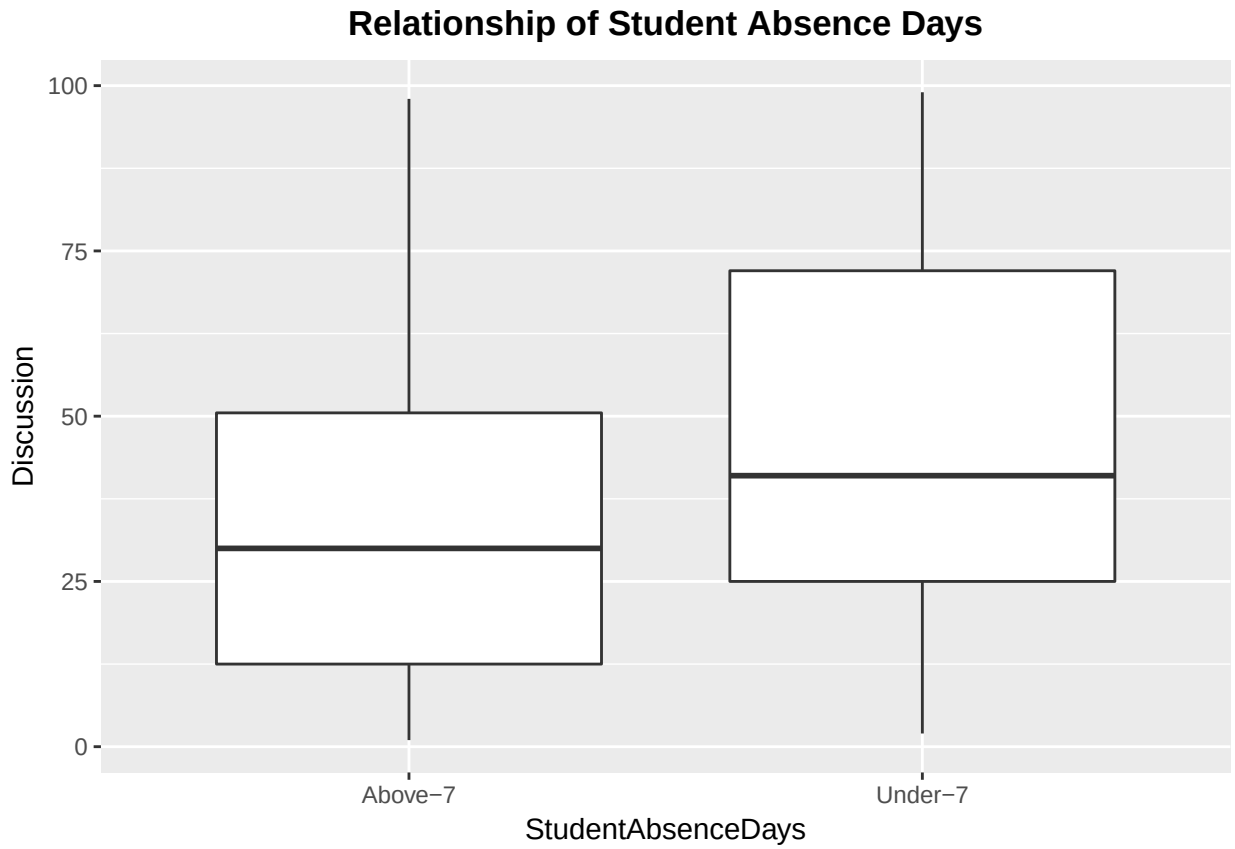
Students have the more leaves the less visited resources.

```
ggplot(data = E, aes(x = StudentAbsenceDays, y = AnnouncementsView)) +
    geom_boxplot() + ggtitle("Relationship of Student Absence Days") +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold"))
```



Students have the more leaves the less announcements view.

```
ggplot(data = E, aes(x = StudentAbsenceDays, y = Discussion)) +
    geom_boxplot() + ggtitle("Relationship of Student Absence Days") +
    theme(plot.title = element_text(hjust = 0.5, lineheight = 0.8,
        face = "bold"))
```



Students have the more leaves the less discussion.

Summary of Task 2:

From the exploration relationship between features, we have the following findings:

1. The analysis further confirmed our findings in the previous step: High achiever students raised hands, visited resource, viewed announcements and discussed more times.
2. Girls have more hand raises and visit more resources.
3. The results on raised hands, visited resources and announcements view are very similar. The difference of discussions between Jordan and Kuwait is not obvious.
4. There are more raised hands, visited resources and announcements view in middle schools. More discussions in high schools.
5. Interestingly, IT subjects have very few hand raises although most students study there.
6. Students with guardians mother have more hand raises, visited resources and announcements view.
7. Students whose parents answered survey and had good satisfaction, have more hand raised, visited resources, announcements view and discussion.
8. Students with more absences have the least raised hands, visited resources, announcements and discussion.

Conclusion of Task 1 and Task 2:

From the exploration and analysis in the previous steps, we found some features have very crucial connection with the students' academic class, and some other features are less related with the students' academic class. From task2, we found that the features like "raisedhands", "VisITedResources","AnnouncementsView", "Discussion" and "StudentsAbsenceDays"have direct connection with students' academic class. There are also some features have less connection with students' academic class, like "StageID", "Section ID", "Semester". We will do some experiments in the following Task 3 to confirm our findings.

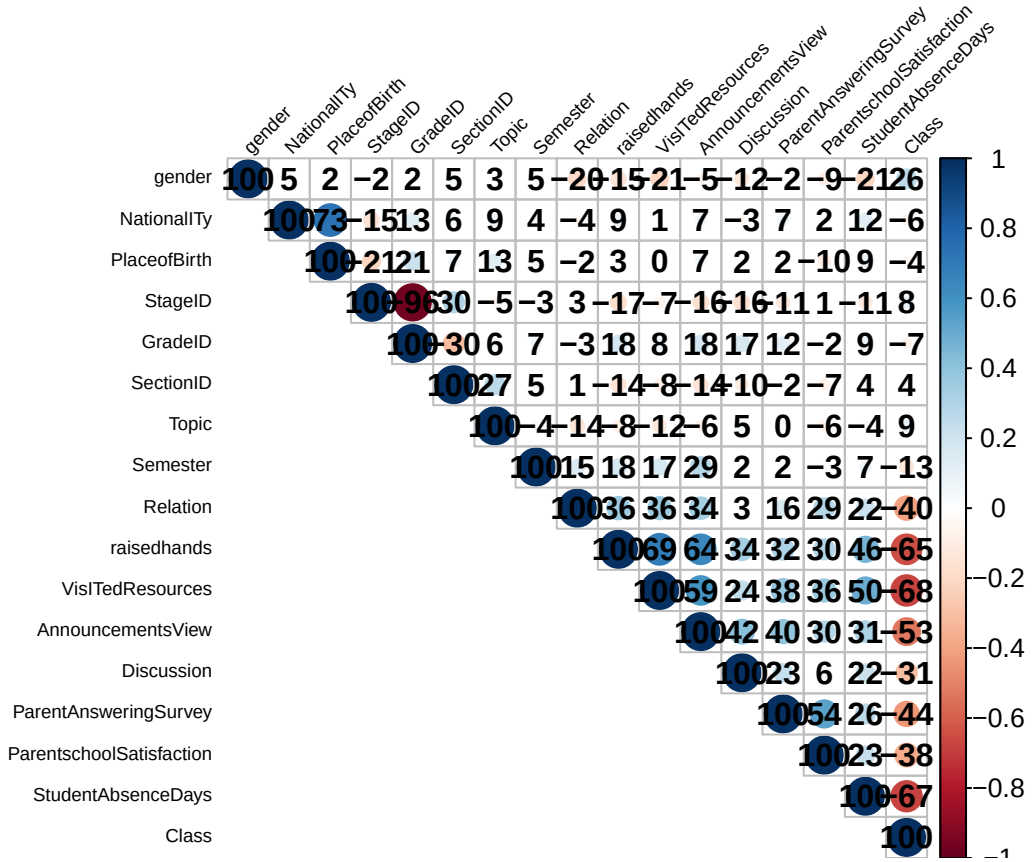
Task 3. Prediction of the students' academic performance. Build prediction model and evaluate the results of the model

Preprocess: Select the important features which are important to Class, and remove some unimportant features.

```
# create data with some correlation structure
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.4.2
```

```
Edu1 <- E
for (i in 1:ncol(Edu1)) {
    Edu1[, i] <- as.integer(Edu1[, i])
}
corrplot::corrplot(cor(as.matrix(Edu1)), type = "upper", method = "circle",
    tl.pos = "lt", tl.col = "black", tl.cex = 0.6, tl.srt = 45,
    addCoef.col = "black", addCoefasPercent = TRUE, sig.level = 0.5,
    insig = "blank")
```



From the correlation plot above, we can find that features like raisedhands, visitedresources,announcementsview,studentabsenceda and discussion have very close correlation with class. On the other hand, some features like sectionID, gradeID, stageID,placeofbirth and nationality have very less correlation with class. So we decide to remove useless features before building our prediction models.

```
# Remove some useless features , such as: GradeID, SectionID
# and Semester.
E1 <- E[, -c(5, 6, 8)]
```

We have 14 features left after removing 3 useless features.

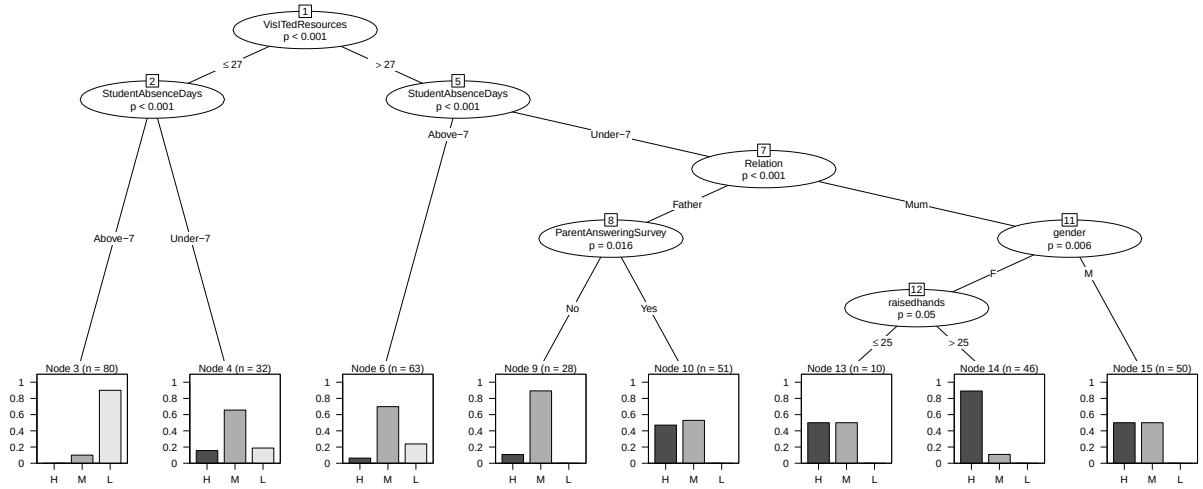
Prediction model 1 – Decision tree

```
# Decision trees use probability to choose how to split, so
# here we 'seed' a random number generator so our results are
# always the same.
set.seed(55)


# Split the data into 75% training, 25% test: this means we
# use 75% of the dataset to build our predictive model. We
# then use the remaining 25% of data as the unseen data
# (test) in order to evaluate the predictions.

ind <- sample(2, nrow(E1), replace = TRUE, prob = c(0.75, 0.25))
```

```
trainingData <- E1[ind == 1, ]
testData <- E1[ind == 2, ]
```

```
# Use the training data to train a model.
formula <- Class ~ gender + NationalITy + PlaceofBirth + StageID +
    Topic + Relation + raisedhands + VisITedResources + AnnouncementsView +
    Discussion + ParentAnsweringSurvey + ParentschoolSatisfaction +
    StudentAbsenceDays

E_ctree <- ctree(formula, data = trainingData)
```

```
plot(E_ctree)
```




```
tree.predict <- predict(E_ctree, newdata = testData)
confusionMatrix(tree.predict, testData$Class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  H  M  L
##          H 24  7  0
##          M 11 40 13
##          L  0  4 21
##
## Overall Statistics
##
##                Accuracy : 0.7083
##                  95% CI : (0.6184, 0.7877)
##     No Information Rate : 0.425
##     P-Value [Acc > NIR] : 3.275e-10
##
##                   Kappa : 0.5435
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: H Class: M Class: L
## Sensitivity            0.6857   0.7843   0.6176
```

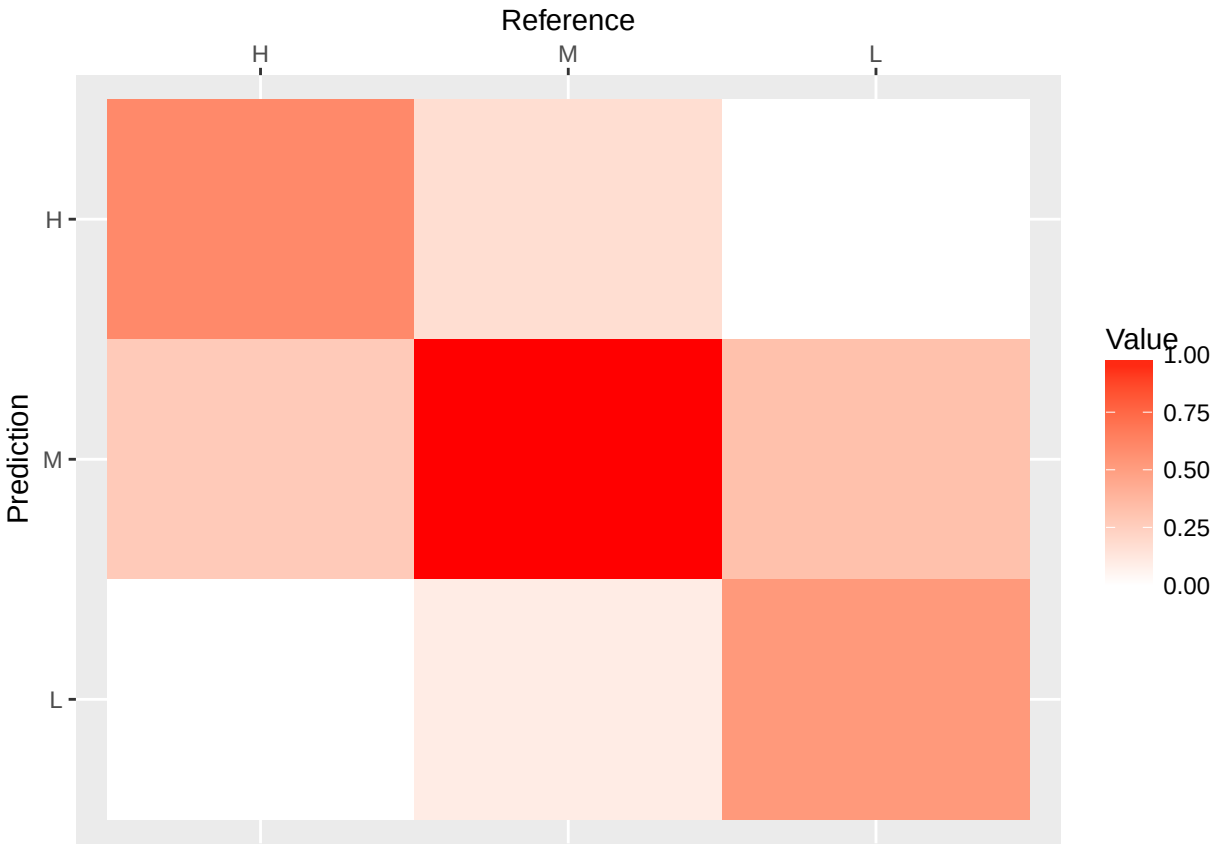
```
## Specificity            0.9176   0.6522   0.9535
## Pos Pred Value         0.7742   0.6250   0.8400
## Neg Pred Value         0.8764   0.8036   0.8632
## Prevalence             0.2917   0.4250   0.2833
## Detection Rate         0.2000   0.3333   0.1750
## Detection Prevalence   0.2583   0.5333   0.2083
## Balanced Accuracy      0.8017   0.7182   0.7856
```

Heatmap 1

```
E_table <- table(tree.predict, testData$Class)
E_melt <- melt(E_table)
names(E_melt) <- c("Prediction", "Reference", "Value")
E_melt$Value <- (E_melt$Value - min(E_melt$Value))/(max(E_melt$Value) -
    min(E_melt$Value))
E_melt
```

```
##   Prediction Reference Value
## 1          H         H 0.600
## 2          M         H 0.275
## 3          L         H 0.000
## 4          H         M 0.175
## 5          M         M 1.000
## 6          L         M 0.100
## 7          H         L 0.000
## 8          M         L 0.325
## 9          L         L 0.525
```

```
ggplot(E_melt, aes(x = Reference, y = Prediction, fill = Value)) +
    geom_tile() + scale_fill_gradient(low = "white", high = "red") +
    scale_x_discrete(position = "top") + scale_y_discrete(limits = rev(levels(E_melt$Prediction)))
```



Summary of model 1:

we conclude from the tree mode above:

1. From the plot of the tree, some variables (VisITedResources, StudentAbsenceDays, Relation, gender, ParentAnsweringSurvey and raisedhands) are selected as splitting rules, which means these features are crucial in splitting the tree. so they are very important features in this tree model.
2. From the confusion matrix and heatmap, there is no prediction error between class L and H. However, the performance is not so good between class M and H, M and L. It means that this model has some problem in predicting the adjacent classes but not the distant classes.
3. The accuracy of this model is 0.7038 from the statistics result. The accuracy is a description of systematic errors, a measure of statistical bias. It reflects the percentage of the true positive cases out of all the cases. In this case, we can see that 24 of H class have been predicted correctly which means the numbers of the true positive of H class, and 7 of M classes have been predicted as H classes but actually not H, which indicates numbers of the false positive of H class. To calculate accuracy, the true positive cases are accumulated as 84 (24+40+21) including all the H, M and L classes, and the total cases are 120 (24+7+11+40+13+4+21) including all the TP, TN, FP and FN cases. The formula of accuracy is TP/TP+TN+FP+FT, so the accuracy is 84/120 = 0.7038 in this model.The accuracy reflects how the model performs in predicting the correct classes out of all the cases.
4. Precision is a better method for us to judge the performance of a model. Precision is a description of random errors, a measure of statistical variability. The formula of precision is TP/TP+FP, which can reflect the rate of true positive cases out of all the positive cases including false positive cases. In this case, there are 13 L classes that predicted as M classes, these 13 cases are the FP cases in predicting M

classes, and there are also 7 M classes predicted as H classes, these 7 cases is the FP cases in predicting H classes. We can get a better insight of the model by referring to the precision of each class.

5. Since the precision is related to the true positive and false positive, we can evaluate the performance of the predictive model by calculating the precision of each class. In this confusion matrix, the specificity is equal to precision, we can use the specificity of the results of confusion matrix to observe the performance of the model.

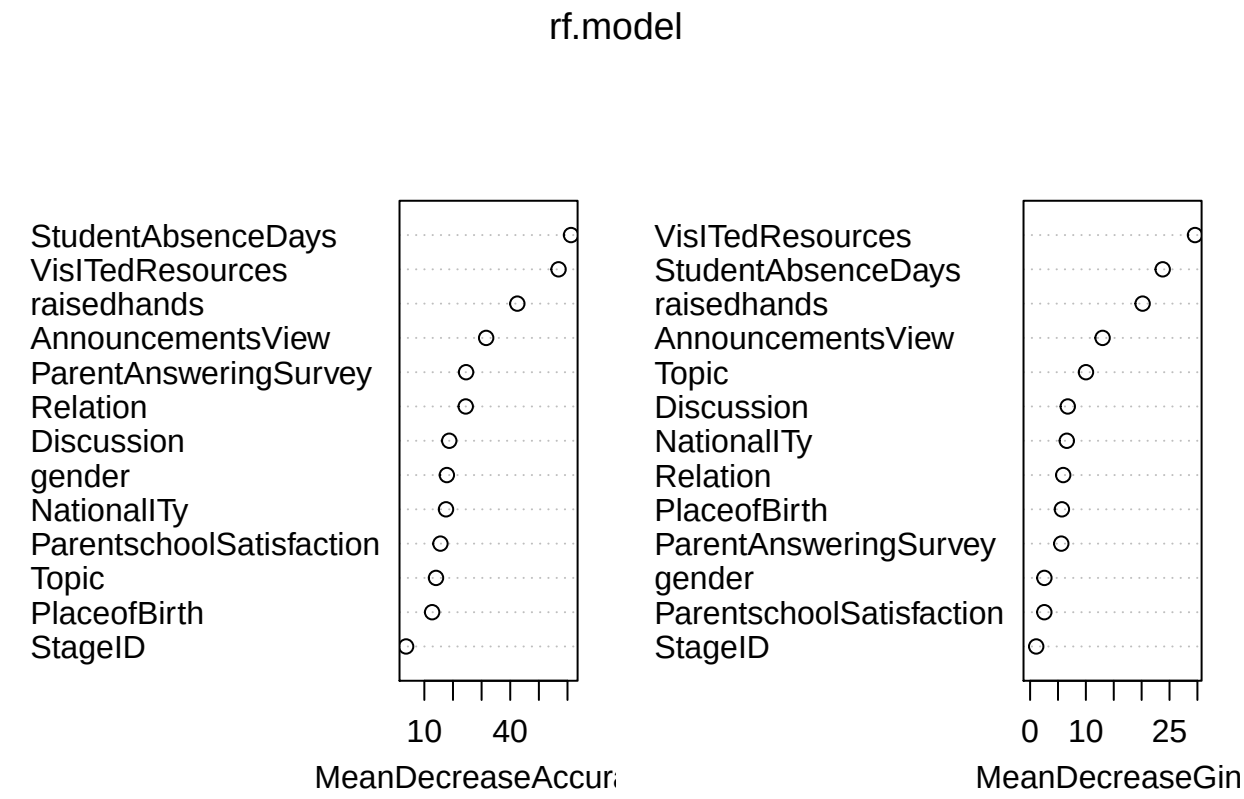
In this tree model, precision is: H(0.9176), M(0.6522), L(0.9535), and the mean precision is 0.8411.

Prediction model 2 – Random Forest

```
rf.model <- randomForest(formula, data = trainingData, importance = TRUE,
    ntree = 2000, nodesize = 20)

rf.predict <- predict(rf.model, testData)
confusionMatrix(testData$Class, rf.predict)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  H  M  L
##          H 22 13  0
##          M  5 42  4
##          L  0  7 27
##
## Overall Statistics
##
##                Accuracy : 0.7583
##                  95% CI : (0.6717, 0.8318)
##     No Information Rate : 0.5167
##     P-Value [Acc > NIR] : 4.645e-08
##
##                   Kappa : 0.6233
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: H Class: M Class: L
## Sensitivity            0.8148   0.6774   0.8710
## Specificity            0.8602   0.8448   0.9213
## Pos Pred Value         0.6286   0.8235   0.7941
## Neg Pred Value         0.9412   0.7101   0.9535
## Prevalence             0.2250   0.5167   0.2583
## Detection Rate         0.1833   0.3500   0.2250
## Detection Prevalence   0.2917   0.4250   0.2833
## Balanced Accuracy      0.8375   0.7611   0.8962
```

```
varImpPlot(rf.model)
```

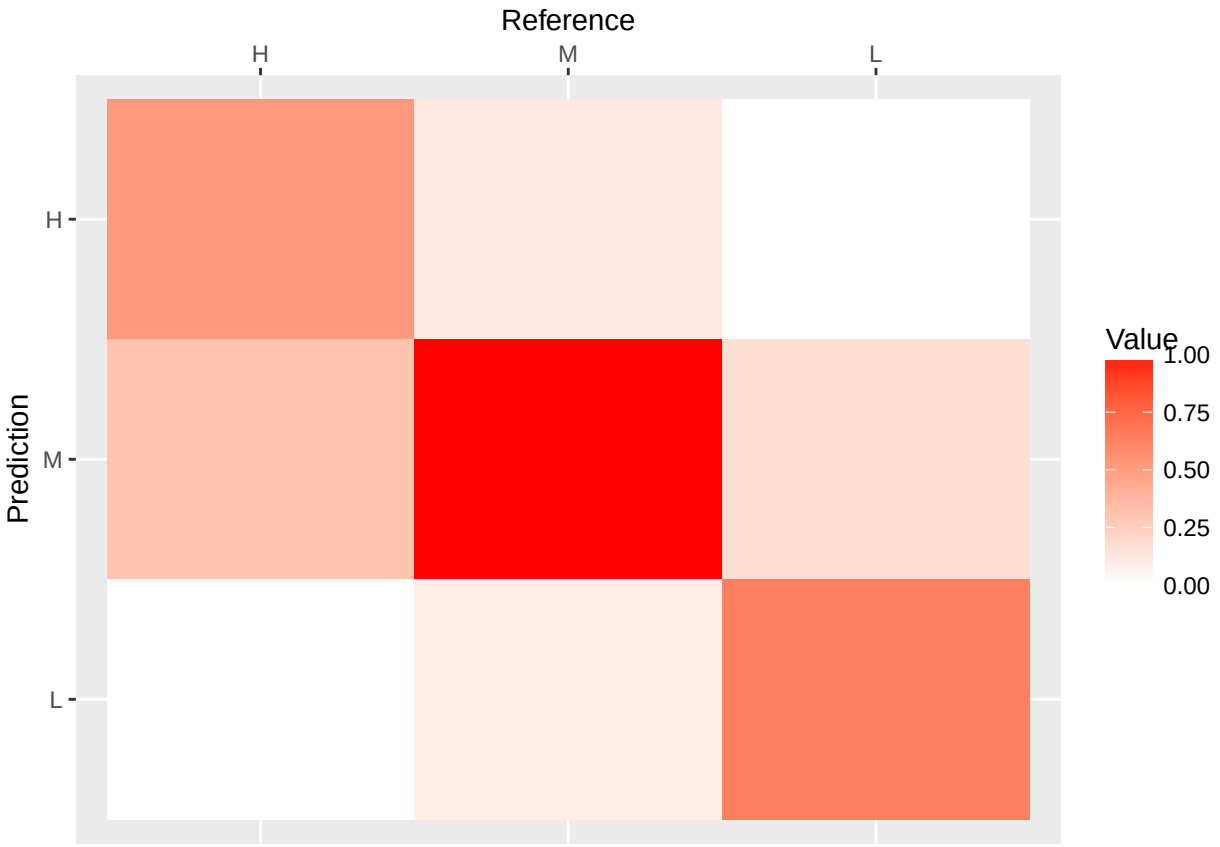


Heatmap 2

```
E_table <- table(rf.predict, testData$Class)
E_melt <- melt(E_table)
names(E_melt) <- c("Prediction", "Reference", "Value")
E_melt$Value <- (E_melt$Value - min(E_melt$Value))/(max(E_melt$Value) -
    min(E_melt$Value))
E_melt
```

```
##   Prediction Reference     Value
## 1          H         H 0.5238095
## 2          M         H 0.3095238
## 3          L         H 0.0000000
## 4          H         M 0.1190476
## 5          M         M 1.0000000
## 6          L         M 0.0952381
## 7          H         L 0.0000000
## 8          M         L 0.1666667
## 9          L         L 0.6428571
```

```
ggplot(E_melt, aes(x = Reference, y = Prediction, fill = Value)) +
    geom_tile() + scale_fill_gradient(low = "white", high = "red") +
    scale_x_discrete(position = "top") + scale_y_discrete(limits = rev(levels(E_melt$Prediction)))
```



Summary of model 2:

we conclude from the tree mode above:

1. From the plot of variable importance, some variables (StudentAbsenceDays, VisITedResources, raised-hands, AnnouncementsView, ParentAnsweringSurvey and Relation) are more important, and the other variables are less important in this model such as StageID, PlaceofBirth and Topic.
2. From the confusion matrix, there is also no prediction error between class L and H in this random forest model. There are some improvements: all the number of correct prediction has been increased in all the classes (26, 42 and 26) than the previous model. And the FP of M classes have been reduced a lot. However, the FP cases of H and L classes have been slightly increased.
3. The accuracy of this model 2 is 0.7583, which is a little higher than model 1.
4. In this random forest model, precision is: H(0.8681), M(0.8333), L(0.9213), and the mean precision is 0.8742.

Overall, this model is better than the tree model both in accuracy and precision.

Prediction model 3 – Support Vector Machines

```
svm.model <- svm(formula, data = trainingData, kernel = "radial",
    cost = 10, gamma = 0.15)
```

```r
svm.predict <- predict(svm.model, testData)
confusionMatrix(testData$Class, svm.predict)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  H  M  L
##          H 29  6  0
##          M  7 41  3
##          L  1  8 25
##
## Overall Statistics
##
##                Accuracy : 0.7917
##                  95% CI : (0.708, 0.8604)
##     No Information Rate : 0.4583
##     P-Value [Acc > NIR] : 7.632e-14
##
##                   Kappa : 0.6791
##  Mcnemar's Test P-Value : 0.3408
##
## Statistics by Class:
##
##                      Class: H Class: M Class: L
## Sensitivity            0.7838   0.7455   0.8929
## Specificity            0.9277   0.8462   0.9022
## Pos Pred Value         0.8286   0.8039   0.7353
## Neg Pred Value         0.9059   0.7971   0.9651
## Prevalence             0.3083   0.4583   0.2333
## Detection Rate         0.2417   0.3417   0.2083
## Detection Prevalence   0.2917   0.4250   0.2833
## Balanced Accuracy      0.8557   0.7958   0.8975
```

Summary of model 3:

we conclude from the tree mode above:

1. From the confusion matrix, different from the previous two models, there is a prediction error between class L and H, one H class is predicted as L class. There are some improvements: all the number of correct prediction has been increased in all the classes (29, 41 and 25), much better than the model 1, but similar to model 2. And the FP of H and M classes have been reduced, which is the best among these 3 models.
2. The accuracy of this model 3 is 0.7917, which is a little higher than model 1 and model 2.
3. In this random forest model, precision is: H(0.9277), M(0.8462), L(0.9022), and the mean precision is 0.8920.

Overall, this model is best among these 3 models both in accuracy and precision.

Task 4. Improvement of these prediction models using N-folder cross validation for the prediction models above

Preprocess:Using N-folder cross validation on the training data.

```
# using 10-folder cross validation to train this model
TrainingParameters <- trainControl(method = "cv", number = 10)
```

Using N-folder cross validation on prediction model 1 – Decision tree

```
# train model with randomforest networks
trModel <- train(formula, data = trainingData, method = "ctree",
    trControl = TrainingParameters, preProcess = c("scale", "center"),
    na.action = na.omit)
```

```
## Loading required package: party
## Loading required package: mvtnorm
## Loading required package: modeltools
## Loading required package: stats4
## Loading required package: strucchange
## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
## Loading required package: sandwich
##
## Attaching package: 'party'
## The following objects are masked from 'package:partykit':
##
##     cforest, ctree, ctree_control, edge_simple, mob, mob_control,
##     node_barplot, node_bivplot, node_boxplot, node_inner,
##     node_surv, node_terminal
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: NationalITyvenzuela,
## PlaceofBirthvenzuela

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: NationalITyvenzuela,
## PlaceofBirthvenzuela

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
```

```
## uniqueCut = 10, : These variables have zero variances: NationalITyvenzuela,
## PlaceofBirthvenzuela
```

```
trPredictions <- predict(trModel, testData)
# Create confusion matrix
confusionMatrix(testData$Class, trPredictions)
```

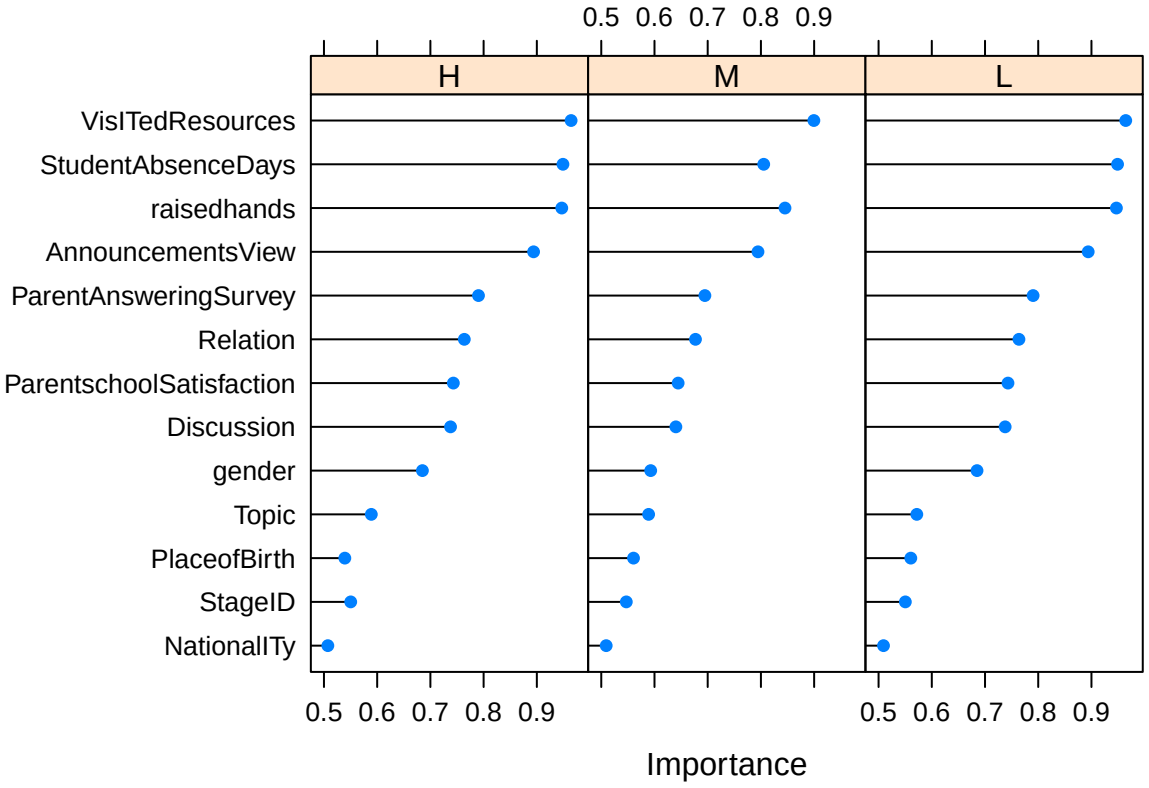
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  H  M  L
##          H 30  5  0
##          M  8 39  4
##          L  1 12 21
##
## Overall Statistics
##
##                Accuracy : 0.75
##                  95% CI : (0.6627, 0.8245)
##     No Information Rate : 0.4667
##     P-Value [Acc > NIR] : 2.484e-10
##
##                   Kappa : 0.6141
##  Mcnemar's Test P-Value : 0.1276
##
## Statistics by Class:
##
##                      Class: H Class: M Class: L
## Sensitivity            0.7692   0.6964   0.8400
## Specificity            0.9383   0.8125   0.8632
## Pos Pred Value         0.8571   0.7647   0.6176
## Neg Pred Value         0.8941   0.7536   0.9535
## Prevalence             0.3250   0.4667   0.2083
## Detection Rate         0.2500   0.3250   0.1750
## Detection Prevalence   0.2917   0.4250   0.2833
## Balanced Accuracy      0.8538   0.7545   0.8516
```

```
# estimate variable importance
importance <- varImp(trModel, scale = FALSE)
# summarize importance
print(importance)
```

```
## ROC curve variable importance
##
##   variables are sorted by maximum importance across the classes
##                               H      M      L
## VisITedResources          0.9644 0.8995 0.9644
## StudentAbsenceDays        0.9491 0.8052 0.9491
## raisedhands               0.9469 0.8452 0.9469
## AnnouncementsView         0.8939 0.7945 0.8939
## ParentAnsweringSurvey     0.7904 0.6947 0.7904
## Relation                  0.7637 0.6770 0.7637
## ParentschoolSatisfaction 0.7431 0.6445 0.7431
## Discussion                0.7379 0.6402 0.7379
## gender                    0.6850 0.5929 0.6850
```

```
## Topic                0.5890 0.5890 0.5719
## PlaceofBirth         0.5392 0.5606 0.5606
## StageID              0.5503 0.5470 0.5503
## NationalITy          0.5073 0.5092 0.5092
```

```r
# plot importance
plot(importance)
```



Summary of model 1:

we condlude from the tree mode above:

1. From the plot of variable importance, the most important variables are StudentAbsenceDays, VisITedResources, raisedhands and AnnouncementsView, the less important variables are ParentAnswering-Survey, relation, ParentschoolSatisfaction, Discussion and gender, and the unimportant variables are Nationality, StageID, PlaceofBirth and Topic for every class. This conclusion further confirmed our findings in the previous steps.
2. From the confusion matrix, there is a prediction error between class L and H, one H class is predicted as L class. There are some improvements: The number of TP of H class with 30 is much better, it's also the highest till now.
3. The accuracy of this model is 0.75, which is higher than the accuracy of model1 without cross validation(0.7083).
4. In this model, precision is: H(0.9383), M(0.8125), L(0.8632), and the mean precision is 0.8713, which is higher than that of model 1 without cross validation (0.8411).

Overall, this model is better than the model 1 without cross validation both in accuracy and precision, and similar to the performance of model 2 without cross validation

Using N-folder cross validation on prediction model 2 – Random Forest

```
# train model with randomforest networks
rfModel <- train(formula, data = trainingData, method = "rf",
    trControl = TrainingParameters, preProcess = c("scale", "center"),
    na.action = na.omit)
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: NationalITyvenzuela,
## PlaceofBirthvenzuela

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: NationalITyvenzuela,
## PlaceofBirthvenzuela

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: NationalITyvenzuela,
## PlaceofBirthvenzuela
```

```
rfPredictions <- predict(rfModel, testData)
# Create confusion matrix
confusionMatrix(rfPredictions, testData$Class)
```

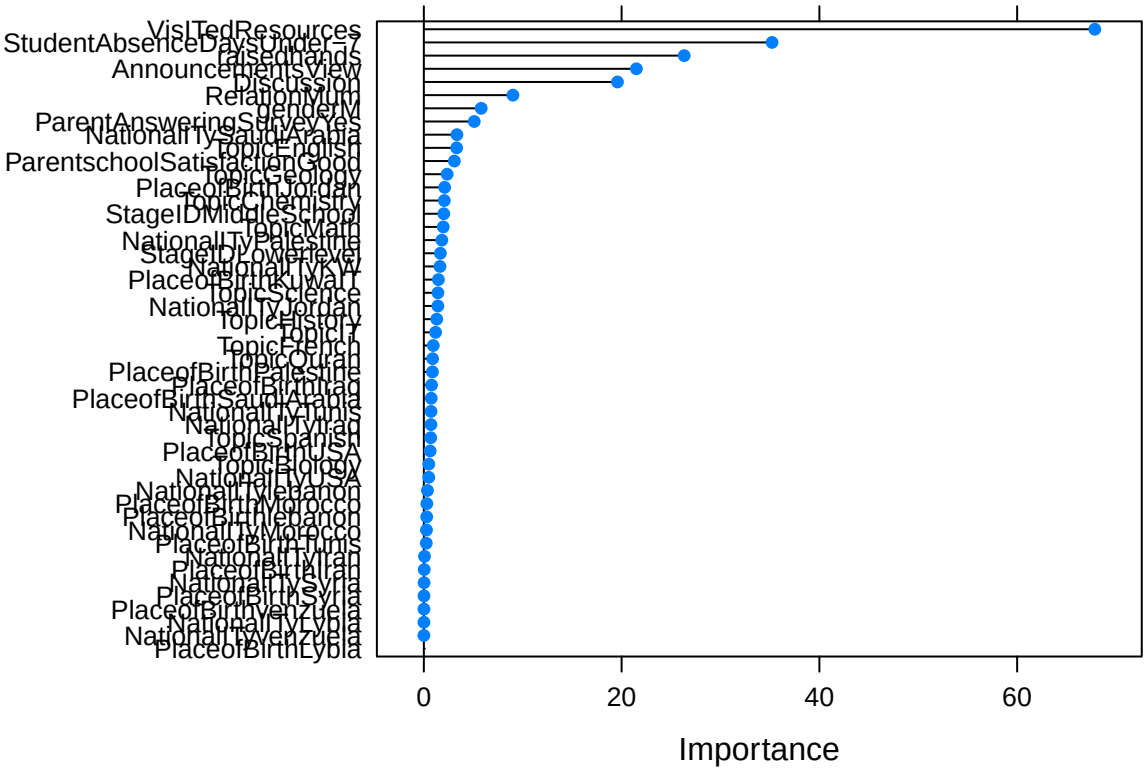
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  H  M  L
##          H 25  5  0
##          M 10 42  7
##          L  0  4 27
##
## Overall Statistics
##
##                Accuracy : 0.7833
##                  95% CI : (0.6989, 0.8533)
##     No Information Rate : 0.425
##     P-Value [Acc > NIR] : 1.259e-15
##
##                   Kappa : 0.664
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: H Class: M Class: L
## Sensitivity            0.7143   0.8235   0.7941
## Specificity            0.9412   0.7536   0.9535
## Pos Pred Value         0.8333   0.7119   0.8710
## Neg Pred Value         0.8889   0.8525   0.9213
## Prevalence             0.2917   0.4250   0.2833
```

```
## Detection Rate         0.2083   0.3500   0.2250
## Detection Prevalence   0.2500   0.4917   0.2583
## Balanced Accuracy      0.8277   0.7886   0.8738
```

```
# estimate variable importance
importance <- varImp(rfModel, scale = FALSE)
# summarize importance
print(importance)
```

```
## rf variable importance
##
##   only 20 most important variables shown (out of 48)
##
##                              Overall
## VisITedResources              67.853
## StudentAbsenceDaysUnder-7     35.216
## raisedhands                   26.329
## AnnouncementsView             21.496
## Discussion                    19.582
## RelationMum                    9.009
## genderM                        5.801
## ParentAnsweringSurveyYes       5.096
## NationalITySaudiArabia         3.347
## TopicEnglish                   3.320
## ParentschoolSatisfactionGood   3.084
## TopicGeology                   2.356
## PlaceofBirthJordan             2.102
## TopicChemistry                 2.071
## StageIDMiddleSchool            2.015
## TopicMath                      1.961
## NationalITyPalestine           1.819
## StageIDLowerlevel              1.673
## NationalITyKW                  1.631
## PlaceofBirthKuwaIT             1.473
```

```
# plot importance
plot(importance)
```



Summary of model 2:

we conclude from the tree mode above:

1. From the plot of variable importance, the most important variables are VisITedResources, raisedhands, StudentAbsenceDays, AnnouncementsView and Discussion, the less important variables are Relation-Mum, ParentAnsweringSurvey, genderM and ParentschoolSatisfaction, and the unimportant variables are Topic, StageID,Nationality and PlaceofBirth for every class. This figures that represent the level of importance is much clear, which also further confirmed our findings in the previous steps.
2. From the confusion matrix, there is no prediction error between class L and H in this random forest model. There are some improvements: The number of correct prediciton has been increased a lot in M and L class (43 and 28) than the model 1. And there are only 4 m class predicted as L class, which has been improved a lot than model 1.
3. The accuracy of this model is 0.8083, which is the highest accuracy among all the models.
4. In this model, precision is: H(0.9529), M(0.7826), L(0.9535), and the mean precision is 0.8963, which is highest precision among all the models till now.

Overall, this model is much better than the model 1 with cross validation both in accuracy and precision.

Using N-folder cross validation on prediction model 3 – Support Vector Machines

```
# train model with SVM
SVMModel <- train(formula, data = trainingData, method = "svmRadial",
    cost = 10, gamma = 0.15, trControl = TrainingParameters,
    preProcess = c("scale", "center"), na.action = na.omit)
```

```
## Loading required package: kernlab

## Warning: package 'kernlab' was built under R version 3.4.1

##
## Attaching package: 'kernlab'

## The following object is masked from 'package:modeltools':
##
##     prior

## The following object is masked from 'package:ggplot2':
##
##     alpha

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: NationalITyvenzuela,
## PlaceofBirthvenzuela

## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: NationalITyvenzuela,
## PlaceofBirthvenzuela

## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: NationalITyvenzuela,
## PlaceofBirthvenzuela

## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.
```

```
SVMPredictions <- predict(SVMModel, testData)
# Create confusion matrix
confusionMatrix(SVMPredictions, testData$Class)
```

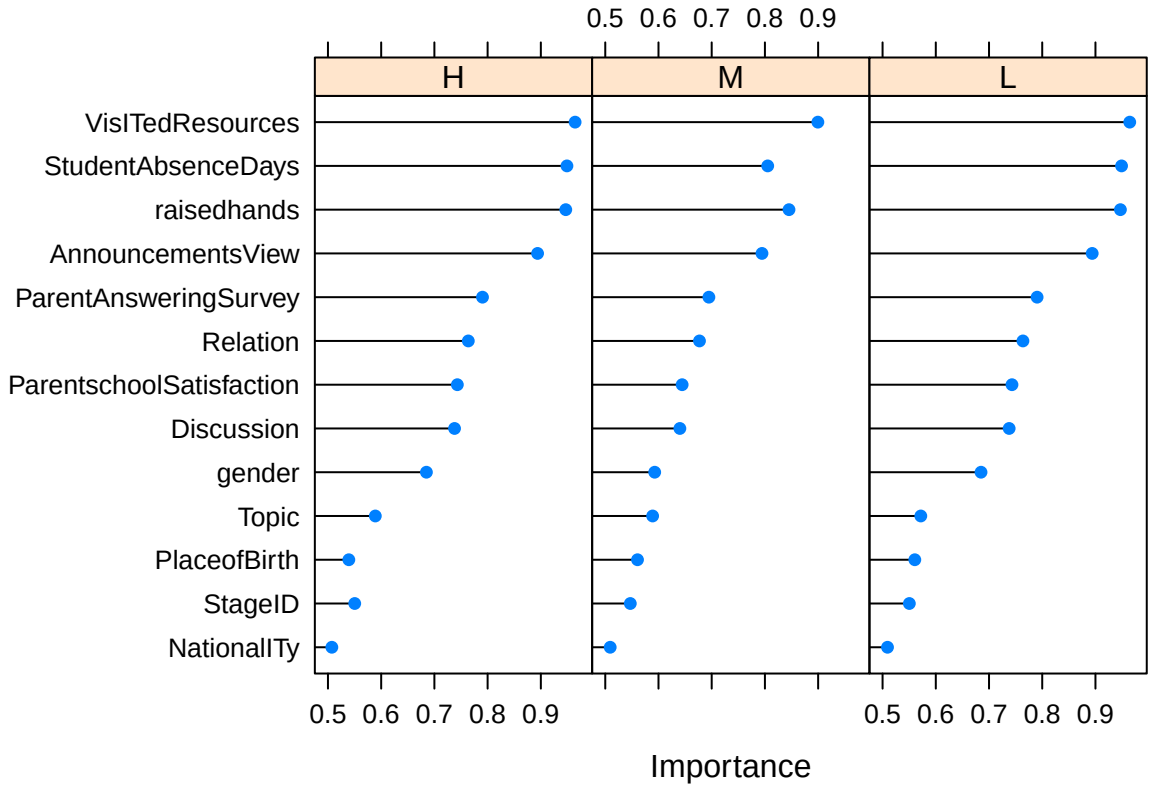
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  H  M  L
##          H 22  2  0
##          M 13 45  9
##          L  0  4 25
##
## Overall Statistics
##
##                Accuracy : 0.7667
##                  95% CI : (0.6807, 0.839)
##     No Information Rate : 0.425
##     P-Value [Acc > NIR] : 2.733e-14
```

```
##
##                   Kappa : 0.6331
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: H Class: M Class: L
## Sensitivity            0.6286   0.8824   0.7353
## Specificity            0.9765   0.6812   0.9535
## Pos Pred Value         0.9167   0.6716   0.8621
## Neg Pred Value         0.8646   0.8868   0.9011
## Prevalence             0.2917   0.4250   0.2833
## Detection Rate         0.1833   0.3750   0.2083
## Detection Prevalence   0.2000   0.5583   0.2417
## Balanced Accuracy      0.8025   0.7818   0.8444
```

```r
# estimate variable importance
importance <- varImp(SVMModel, scale = FALSE)
# summarize importance
print(importance)
```

```
## ROC curve variable importance
##
##   variables are sorted by maximum importance across the classes
##                              H      M      L
## VisITedResources         0.9644 0.8995 0.9644
## StudentAbsenceDays       0.9491 0.8052 0.9491
## raisedhands              0.9469 0.8452 0.9469
## AnnouncementsView        0.8939 0.7945 0.8939
## ParentAnsweringSurvey    0.7904 0.6947 0.7904
## Relation                 0.7637 0.6770 0.7637
## ParentschoolSatisfaction 0.7431 0.6445 0.7431
## Discussion               0.7379 0.6402 0.7379
## gender                   0.6850 0.5929 0.6850
## Topic                    0.5890 0.5890 0.5719
## PlaceofBirth             0.5392 0.5606 0.5606
## StageID                  0.5503 0.5470 0.5503
## NationalITy              0.5073 0.5092 0.5092
```

```r
# plot importance
plot(importance)
```



Summary of model 3:

we conclude from the tree mode above:

1. From the plot of variable importance, the most important variables are VisITedResources, StudentAbsenceDays, raisedhands and AnnouncementsView, the less important variables are ParentAnswering-Survey, Relation, ParentschoolSatisfaction, Discussion and gender, and the unimportant variables are Topic, StageID, Nationality and PlaceofBirth for every class. The figures that represent the level of importance is much clear, which also further confirmed our findings in the previous steps.

2. From the confusion matrix, there is no prediction error between class L and H in this random forest model. There are some improvements than model 1 but not than model 2: The number of correct prediction has been increased a lot in M and L class (43 and 25) than the model 1. And there are only 4 m class predicted as L class, which has been improved a lot than model 1. However, the number of TP of H class is the lowest among those 3 models.

3. The accuracy of this model is 0.75, which is the same as model 1 but less than the model 3 without cross validation.

4. In this model, precision is: H(0.9529), M(0.6812), L(0.9535), and the mean precision is 0.8625.

Overall, this model worse than the model 3 without cross validation both in accuracy and precision.

Final Conclusion:

1. Random forests model outperformed all the models by training N-fold cross validation data set in this specific educational datasets. It got the highest rate of accuracy and precision. As a result, the random forest prediction model is recommended for this educational data mining project.
2. The most important activities that related to students' academic performance are VisITedResources, raisedhands, StudentAbsenceDays, AnnouncementsView and Discussion. Then the Relation, ParentAnsweringSurvey, gender and ParentschoolSatisfaction also has some connection with the academic performance. Students' academic performance has little connection with topic, StageID, Nationality, PlaceofBirth, Grade ID, Section ID and semester based on this educational datasets.