Project

Charan Reddy Kandula Venkata, Sona Shree Reddy Gutha, Vaishnavi Papudesi Babu

2025-04-23

Load and Combine Datasets

```r
# Load both datasets
mat <- fread("student/student-mat.csv", sep = ";")
por <- fread("student/student-por.csv", sep = ";")

# Quick checks
cat("Math dataset dimensions:", dim(mat), "\n")
```

```
## Math dataset dimensions: 395 33
```

```r
cat("Portuguese dataset dimensions:", dim(por), "\n")
```

```
## Portuguese dataset dimensions: 649 33
```

```r
head(mat)
```

```
##    school    sex   age address famsize Pstatus  Medu  Fedu     Mjob     Fjob
##    <char> <char> <int>  <char>  <char>  <char> <int> <int>   <char>   <char>
## 1:     GP      F    18       U     GT3       A     4     4  at_home  teacher
## 2:     GP      F    17       U     GT3       T     1     1  at_home    other
## 3:     GP      F    15       U     LE3       T     1     1  at_home    other
## 4:     GP      F    15       U     GT3       T     4     2   health services
## 5:     GP      F    16       U     GT3       T     3     3    other    other
## 6:     GP      M    16       U     LE3       T     4     3 services    other
##        reason guardian traveltime studytime failures schoolsup famsup   paid
##        <char>   <char>      <int>     <int>    <int>    <char> <char> <char>
## 1:     course   mother          2         2        0       yes     no     no
## 2:     course   father          1         2        0        no    yes     no
## 3:      other   mother          1         2        3       yes     no    yes
## 4:       home   mother          1         3        0        no    yes    yes
## 5:       home   father          1         2        0        no    yes    yes
## 6: reputation   mother          1         2        0        no    yes    yes
##    activities nursery higher internet romantic famrel freetime goout  Dalc
##        <char>  <char> <char>   <char>   <char>  <int>    <int> <int> <int>
## 1:         no     yes    yes       no       no      4        3     4     1
## 2:         no      no    yes      yes       no      5        3     3     1
## 3:         no     yes    yes      yes       no      4        3     2     2
```

```
## 4:        yes     yes    yes      yes      yes      3        2     2     1
## 5:         no     yes    yes       no       no      4        3     2     1
## 6:        yes     yes    yes      yes       no      5        4     2     1
##     Walc health absences    G1    G2    G3
##    <int>  <int>    <int> <int> <int> <int>
## 1:     1      3        6     5     6     6
## 2:     1      3        4     5     5     6
## 3:     3      3       10     7     8    10
## 4:     1      5        2    15    14    15
## 5:     2      5        4     6    10    10
## 6:     2      5       10    15    15    15
```

```r
head(por)
```

```
##    school    sex   age address famsize Pstatus  Medu  Fedu     Mjob     Fjob
##    <char> <char> <int>  <char>  <char>  <char> <int> <int>   <char>   <char>
## 1:     GP      F    18       U     GT3       A     4     4  at_home  teacher
## 2:     GP      F    17       U     GT3       T     1     1  at_home    other
## 3:     GP      F    15       U     LE3       T     1     1  at_home    other
## 4:     GP      F    15       U     GT3       T     4     2   health services
## 5:     GP      F    16       U     GT3       T     3     3    other    other
## 6:     GP      M    16       U     LE3       T     4     3 services    other
##        reason guardian traveltime studytime failures schoolsup famsup   paid
##        <char>   <char>      <int>     <int>    <int>    <char> <char> <char>
## 1:     course   mother          2         2        0       yes     no     no
## 2:     course   father          1         2        0        no    yes     no
## 3:      other   mother          1         2        0       yes     no     no
## 4:       home   mother          1         3        0        no    yes     no
## 5:       home   father          1         2        0        no    yes     no
## 6: reputation   mother          1         2        0        no    yes     no
##    activities nursery higher internet romantic famrel freetime goout  Dalc
##        <char>  <char> <char>   <char>   <char>  <int>    <int> <int> <int>
## 1:         no     yes    yes       no       no      4        3     4     1
## 2:         no      no    yes      yes       no      5        3     3     1
## 3:         no     yes    yes      yes       no      4        3     2     2
## 4:        yes     yes    yes      yes      yes      3        2     2     1
## 5:         no     yes    yes       no       no      4        3     2     1
## 6:        yes     yes    yes      yes       no      5        4     2     1
##     Walc health absences    G1    G2    G3
##    <int>  <int>    <int> <int> <int> <int>
## 1:     1      3        4     0    11    11
## 2:     1      3        2     9    11    11
## 3:     3      3        6    12    13    12
## 4:     1      5        0    14    14    14
## 5:     2      5        0    11    13    13
## 6:     2      5        6    12    12    13
```

Step 2: Combine the Datasets

Our project is on Student Alcohol Consumption (and you want a full complete project), we should combine both datasets smartly.

Important:

The same student may appear in both mat and por datasets.

UCI suggests matching students based on certain key columns (like school, sex, age, address, etc.).

We'll need to avoid counting the same student twice.

```
# Key columns to match students
join_by_cols <- c("school", "sex", "age", "address", "famsize", "Pstatus",
                  "Medu", "Fedu", "Mjob", "Fjob", "reason", "guardian",
                  "traveltime", "studytime", "failures", "schoolsup", "famsup",
                  "paid", "activities", "nursery", "higher", "internet", "romantic")

# Perform an inner join
students <- inner_join(mat, por, by = join_by_cols, suffix = c(".math", ".por"))
```

Step 3: Exploratory Data Analysis (EDA)

We'll break EDA into two parts:

Understand dataset structure (dimensions, types, missing values)

Visualize important patterns (alcohol consumption, grades, etc.)

Part 1: Quick Data Checks

```
# Check dimensions
dim(students)
```

```
## [1] 162  43
```

```
# See the first few rows
head(students)
```

```
##    school    sex   age address famsize Pstatus  Medu  Fedu     Mjob    Fjob
##    <char> <char> <int>  <char>  <char>  <char> <int> <int>   <char>  <char>
## 1:     GP      F    18       U     GT3       A     4     4  at_home teacher
## 2:     GP      F    17       U     GT3       T     1     1  at_home   other
## 3:     GP      M    16       U     LE3       T     2     2    other   other
## 4:     GP      F    17       U     GT3       A     4     4    other teacher
## 5:     GP      F    15       U     GT3       T     2     1 services   other
## 6:     GP      M    15       U     GT3       A     2     2    other   other
##        reason guardian traveltime studytime failures schoolsup famsup   paid
##        <char>   <char>      <int>     <int>    <int>    <char> <char> <char>
## 1:     course   mother          2         2        0       yes     no     no
## 2:     course   father          1         2        0        no    yes     no
## 3:       home   mother          1         2        0        no     no     no
## 4:       home   mother          2         2        0       yes    yes     no
## 5: reputation   father          3         3        0        no    yes     no
## 6:       home    other          1         3        0        no    yes     no
##    activities nursery higher internet romantic famrel.math freetime.math
##        <char>  <char> <char>   <char>   <char>       <int>         <int>
```

```
## 1:         no     yes    yes      no       no           4             3
## 2:         no      no    yes     yes       no           5             3
## 3:         no     yes    yes     yes       no           4             4
## 4:         no     yes    yes      no       no           4             1
## 5:        yes     yes    yes     yes       no           5             2
## 6:         no     yes    yes     yes      yes           4             5
##    goout.math Dalc.math Walc.math health.math absences.math G1.math G2.math
##         <int>     <int>     <int>       <int>         <int>   <int>   <int>
## 1:          4         1         1           3             6       5       6
## 2:          3         1         1           3             4       5       5
## 3:          4         1         1           3             0      12      12
## 4:          4         1         1           1             6       6       5
## 5:          2         1         1           4             4      10      12
## 6:          2         1         1           3             0      14      16
##    G3.math famrel.por freetime.por goout.por Dalc.por Walc.por health.por
##      <int>      <int>        <int>     <int>    <int>    <int>      <int>
## 1:       6          4            3         4        1        1          3
## 2:       6          5            3         3        1        1          3
## 3:      11          4            4         4        1        1          3
## 4:       6          4            1         4        1        1          1
## 5:      12          5            2         2        1        1          4
## 6:      16          4            5         2        1        1          3
##    absences.por G1.por G2.por G3.por
##           <int>  <int>  <int>  <int>
## 1:            4      0     11     11
## 2:            2      9     11     11
## 3:            0     13     12     13
## 4:            2     10     13     13
## 5:            0     10     12     13
## 6:            0     14     14     15
```

```r
# Check data types
str(students)
```

```
## Classes 'data.table' and 'data.frame':   162 obs. of  43 variables:
##  $ school       : chr  "GP" "GP" "GP" "GP" ...
##  $ sex          : chr  "F" "F" "M" "F" ...
##  $ age          : int  18 17 16 17 15 15 16 16 15 15 ...
##  $ address      : chr  "U" "U" "U" "U" ...
##  $ famsize      : chr  "GT3" "GT3" "LE3" "GT3" ...
##  $ Pstatus      : chr  "A" "T" "T" "A" ...
##  $ Medu         : int  4 1 2 4 2 2 4 3 4 4 ...
##  $ Fedu         : int  4 1 2 4 1 2 4 3 3 4 ...
##  $ Mjob         : chr  "at_home" "at_home" "other" "other" ...
##  $ Fjob         : chr  "teacher" "other" "other" "teacher" ...
##  $ reason       : chr  "course" "course" "home" "home" ...
##  $ guardian     : chr  "mother" "father" "mother" "mother" ...
##  $ traveltime   : int  2 1 1 2 3 1 1 3 1 1 ...
##  $ studytime    : int  2 2 2 2 3 3 1 2 2 1 ...
##  $ failures     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ schoolsup    : chr  "yes" "no" "no" "yes" ...
##  $ famsup       : chr  "no" "yes" "no" "yes" ...
##  $ paid         : chr  "no" "no" "no" "no" ...
##  $ activities   : chr  "no" "no" "no" "no" ...
```

```
##  $ nursery      : chr  "yes" "no" "yes" "yes" ...
##  $ higher       : chr  "yes" "yes" "yes" "yes" ...
##  $ internet     : chr  "no" "yes" "yes" "no" ...
##  $ romantic     : chr  "no" "no" "no" "no" ...
##  $ famrel.math  : int  4 5 4 4 5 4 4 5 4 5 ...
##  $ freetime.math: int  3 3 4 1 2 5 4 3 4 4 ...
##  $ goout.math   : int  4 3 4 4 2 2 4 2 1 2 ...
##  $ Dalc.math    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Walc.math    : int  1 1 1 1 1 1 2 1 1 1 ...
##  $ health.math  : int  3 3 3 1 4 3 2 4 1 5 ...
##  $ absences.math: int  6 4 0 6 4 0 4 4 0 0 ...
##  $ G1.math      : int  5 5 12 6 10 14 14 8 13 12 ...
##  $ G2.math      : int  6 5 12 5 12 16 14 10 14 15 ...
##  $ G3.math      : int  6 6 11 6 12 16 14 10 15 15 ...
##  $ famrel.por   : int  4 5 4 4 5 4 4 5 4 5 ...
##  $ freetime.por : int  3 3 4 1 2 5 4 3 4 4 ...
##  $ goout.por    : int  4 3 4 4 2 2 4 2 1 2 ...
##  $ Dalc.por     : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Walc.por     : int  1 1 1 1 1 1 2 1 1 1 ...
##  $ health.por   : int  3 3 3 1 4 3 2 4 1 5 ...
##  $ absences.por : int  4 2 0 2 0 0 6 2 0 0 ...
##  $ G1.por       : int  0 9 13 10 10 14 17 13 12 11 ...
##  $ G2.por       : int  11 11 12 13 12 14 17 14 13 12 ...
##  $ G3.por       : int  11 11 13 13 13 15 17 14 14 12 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```r
nrow(students)
```

```
## [1] 162
```

```r
ncol(students)
```

```
## [1] 43
```

```r
# Summary statistics
summary(students)
```

```
##     school              sex                 age           address         
##  Length:162         Length:162         Min.   :15.00   Length:162        
##  Class :character   Class :character   1st Qu.:16.00   Class :character  
##  Mode  :character   Mode  :character   Median :16.00   Mode  :character  
##                                        Mean   :16.48                     
##                                        3rd Qu.:17.00                     
##                                        Max.   :22.00                     
##    famsize            Pstatus               Medu            Fedu      
##  Length:162         Length:162         Min.   :0.000   Min.   :0.000  
##  Class :character   Class :character   1st Qu.:2.000   1st Qu.:2.000  
##  Mode  :character   Mode  :character   Median :3.000   Median :3.000  
##                                        Mean   :2.765   Mean   :2.599  
##                                        3rd Qu.:4.000   3rd Qu.:4.000  
##                                        Max.   :4.000   Max.   :4.000  
##      Mjob               Fjob              reason            guardian        
```

```
##  Length:162         Length:162         Length:162         Length:162
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##    traveltime      studytime       failures       schoolsup
##  Min.   :1.000   Min.   :1.000   Min.   :0.0000   Length:162
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:0.0000   Class :character
##  Median :1.000   Median :2.000   Median :0.0000   Mode  :character
##  Mean   :1.481   Mean   :1.988   Mean   :0.1296
##  3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:0.0000
##  Max.   :4.000   Max.   :4.000   Max.   :3.0000
##     famsup              paid            activities          nursery
##  Length:162         Length:162         Length:162         Length:162
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##     higher            internet           romantic          famrel.math
##  Length:162         Length:162         Length:162         Min.   :1.000
##  Class :character   Class :character   Class :character   1st Qu.:4.000
##  Mode  :character   Mode  :character   Mode  :character   Median :4.000
##                                                           Mean   :3.994
##                                                           3rd Qu.:5.000
##                                                           Max.   :5.000
##  freetime.math    goout.math      Dalc.math       Walc.math      health.math
##  Min.   :1.00   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.00
##  1st Qu.:3.00   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:3.00
##  Median :3.00   Median :3.000   Median :1.000   Median :2.000   Median :4.00
##  Mean   :3.29   Mean   :3.019   Mean   :1.432   Mean   :2.185   Mean   :3.58
##  3rd Qu.:4.00   3rd Qu.:4.000   3rd Qu.:1.750   3rd Qu.:3.000   3rd Qu.:5.00
##  Max.   :5.00   Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.00
##  absences.math       G1.math         G2.math         G3.math
##  Min.   : 0.000   Min.   : 5.00   Min.   : 0.00   Min.   : 0.00
##  1st Qu.: 0.000   1st Qu.: 8.25   1st Qu.: 9.00   1st Qu.: 9.00
##  Median : 4.000   Median :11.00   Median :11.00   Median :11.00
##  Mean   : 5.901   Mean   :11.31   Mean   :10.99   Mean   :10.94
##  3rd Qu.: 8.000   3rd Qu.:14.00   3rd Qu.:14.00   3rd Qu.:14.00
##  Max.   :75.000   Max.   :19.00   Max.   :19.00   Max.   :20.00
##    famrel.por     freetime.por    goout.por        Dalc.por        Walc.por
##  Min.   :1.000   Min.   :1.00   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:4.000   1st Qu.:3.00   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:1.000
##  Median :4.000   Median :3.00   Median :3.000   Median :1.000   Median :2.000
##  Mean   :3.994   Mean   :3.29   Mean   :3.019   Mean   :1.432   Mean   :2.185
##  3rd Qu.:5.000   3rd Qu.:4.00   3rd Qu.:4.000   3rd Qu.:1.750   3rd Qu.:3.000
##  Max.   :5.000   Max.   :5.00   Max.   :5.000   Max.   :5.000   Max.   :5.000
##    health.por    absences.por        G1.por          G2.por          G3.por
##  Min.   :1.00   Min.   : 0.000   Min.   : 0.00   Min.   : 7.00   Min.   : 0.0
##  1st Qu.:3.00   1st Qu.: 0.000   1st Qu.:10.00   1st Qu.:11.00   1st Qu.:11.0
##  Median :4.00   Median : 2.000   Median :12.00   Median :12.00   Median :13.0
##  Mean   :3.58   Mean   : 3.889   Mean   :12.12   Mean   :12.31   Mean   :12.6
##  3rd Qu.:5.00   3rd Qu.: 6.000   3rd Qu.:14.00   3rd Qu.:14.00   3rd Qu.:14.0
```

```
##  Max.   :5.00   Max.   :32.000   Max.   :19.00   Max.   :18.00   Max.   :18.0
```

```r
# Check missing values
colSums(is.na(students))
```

```
##        school           sex           age       address       famsize
##             0             0             0             0             0
##       Pstatus          Medu          Fedu          Mjob          Fjob
##             0             0             0             0             0
##        reason      guardian    traveltime     studytime      failures
##             0             0             0             0             0
##     schoolsup        famsup          paid    activities       nursery
##             0             0             0             0             0
##        higher      internet      romantic   famrel.math freetime.math
##             0             0             0             0             0
##    goout.math     Dalc.math     Walc.math   health.math absences.math
##             0             0             0             0             0
##       G1.math       G2.math       G3.math    famrel.por  freetime.por
##             0             0             0             0             0
##     goout.por      Dalc.por      Walc.por    health.por  absences.por
##             0             0             0             0             0
##        G1.por        G2.por        G3.por
##             0             0             0
```

```r
# check for duplicates
sum(duplicated(students))
```

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

There are no missing data in the dataset.

Part 2: Understanding Data and Visualizations

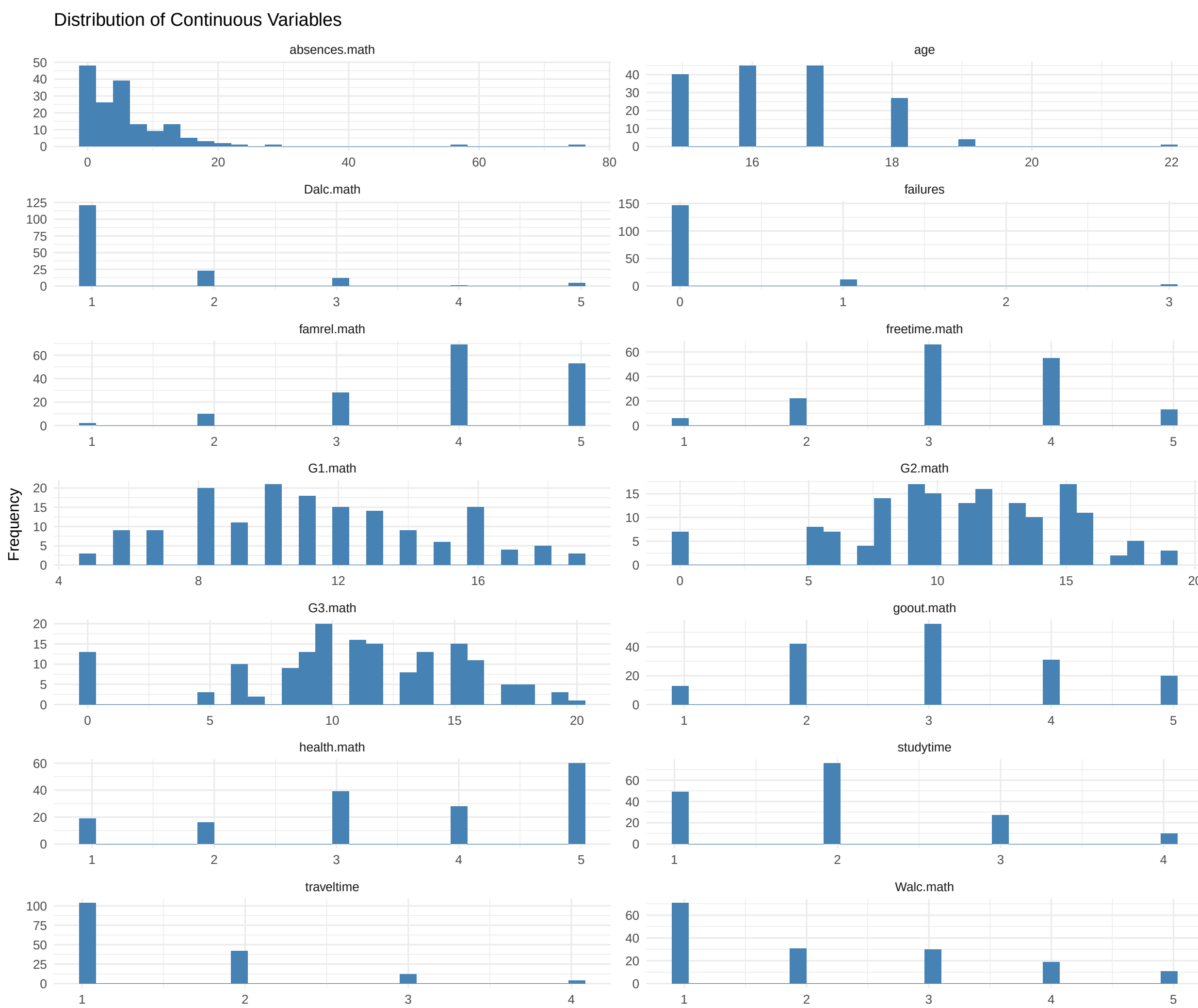
We'll group variables into:

Quantitative (Numeric) Qualitative Ordinal (ordered categories) Qualitative Nominal Qualitative Binary (Yes/No, F/M, etc.)

```r
quantitative_vars <- c("age", "absences.math", "G1.math", "G2.math", "G3.math",
                       "traveltime", "studytime", "failures",
                       "famrel.math", "freetime.math", "goout.math", "Dalc.math", "Walc.math", "health.

ordinal_vars <- c("Medu", "Fedu")
binary_vars <- c("sex", "address", "famsize", "Pstatus", "schoolsup", "famsup",
                 "paid", "activities", "nursery", "higher", "internet", "romantic")

nominal_vars <- c("school", "Mjob", "Fjob", "reason", "guardian")
```

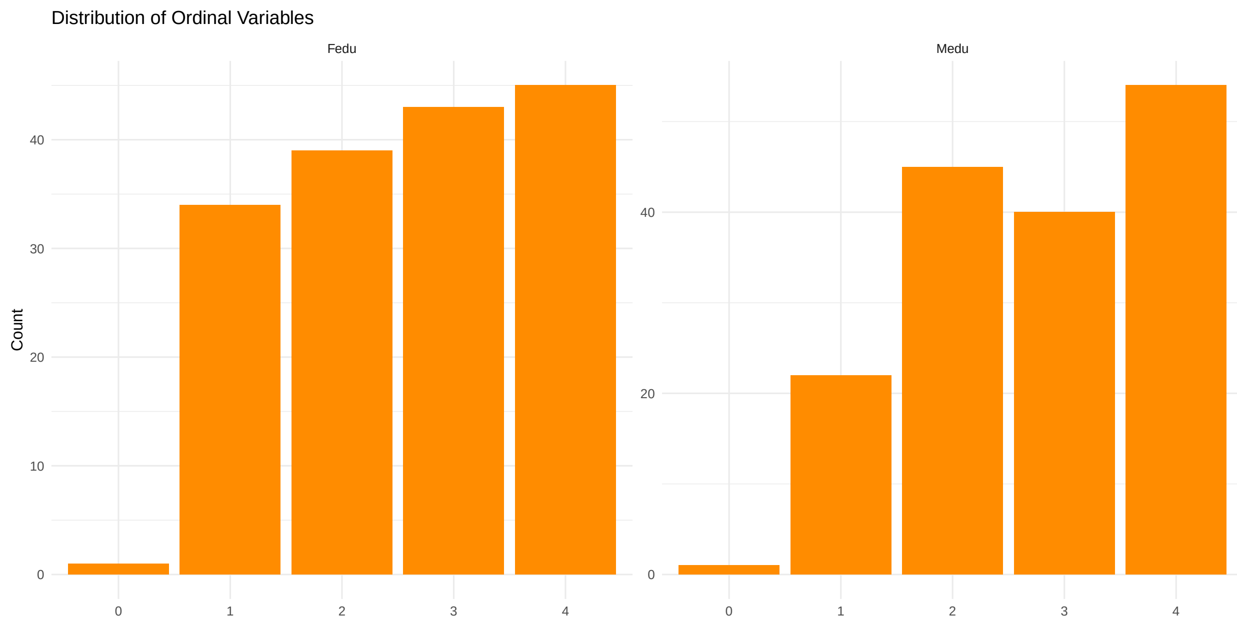
Distribution of continuous variables (Quantitative)

```r
library(ggplot2)
library(tidyr)

students %>%
  select(all_of(quantitative_vars)) %>%
  pivot_longer(cols = everything(), names_to = "variable", values_to = "value") %>%
  ggplot(aes(x = value)) +
  geom_histogram(fill = "steelblue", bins = 30) +
  facet_wrap(~variable, scales = "free", ncol = 2) +
  theme_minimal() +
  labs(title = "Distribution of Continuous Variables", x = NULL, y = "Frequency")
```



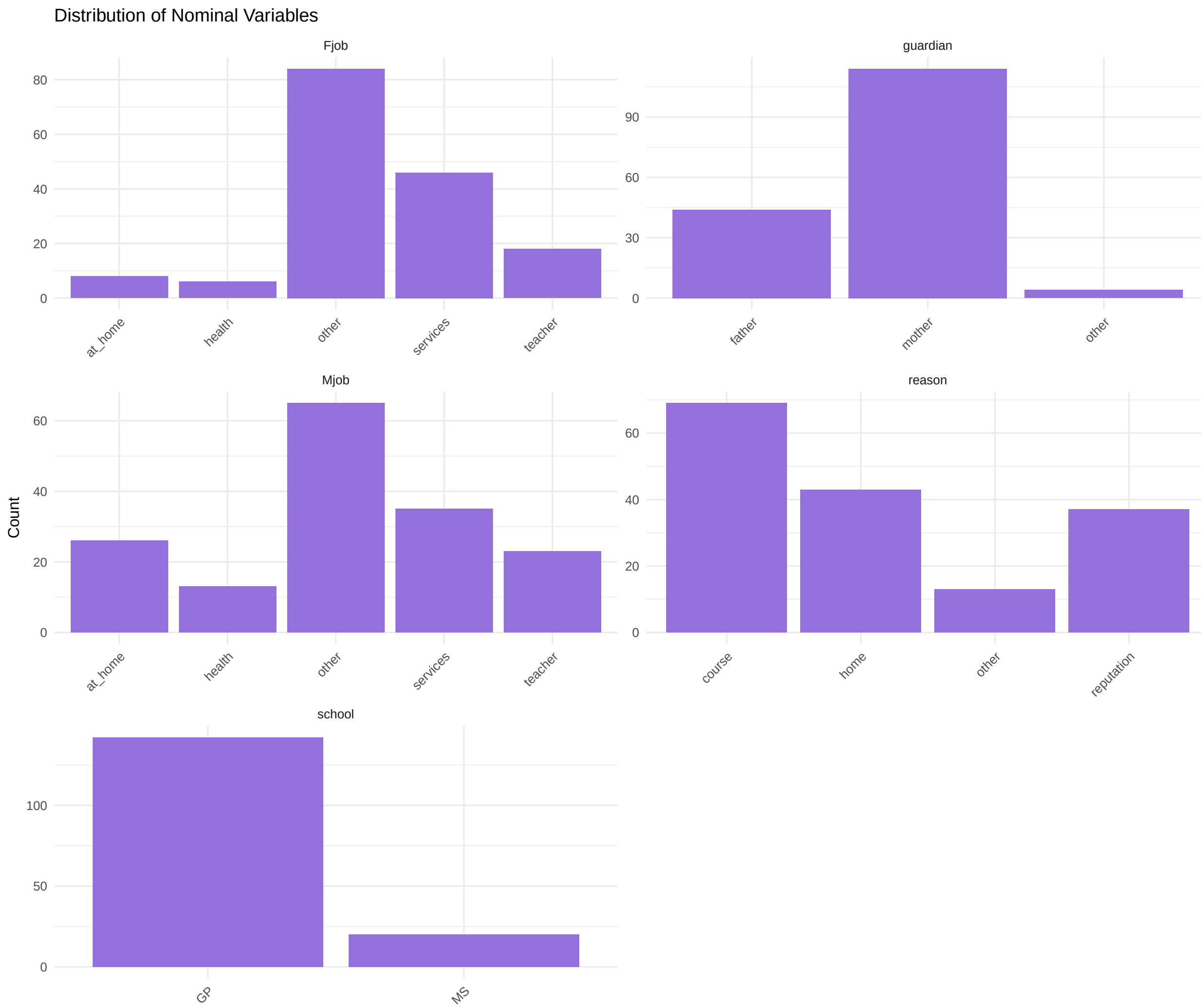
Ordinal Variables Distribution

```r
students %>%
  select(all_of(ordinal_vars)) %>%
  pivot_longer(cols = everything(), names_to = "variable", values_to = "value") %>%
  ggplot(aes(x = as.factor(value))) +
  geom_bar(fill = "darkorange") +
  facet_wrap(~variable, scales = "free", ncol = 2) +
  theme_minimal() +
  labs(title = "Distribution of Ordinal Variables", x = NULL, y = "Count")
```



Nominal Variables Distribution

```r
students %>%
  select(all_of(nominal_vars)) %>%
  pivot_longer(cols = everything(), names_to = "variable", values_to = "value") %>%
  ggplot(aes(x = value)) +
  geom_bar(fill = "mediumpurple") +
  facet_wrap(~variable, scales = "free", ncol = 2) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Distribution of Nominal Variables", x = NULL, y = "Count")
```

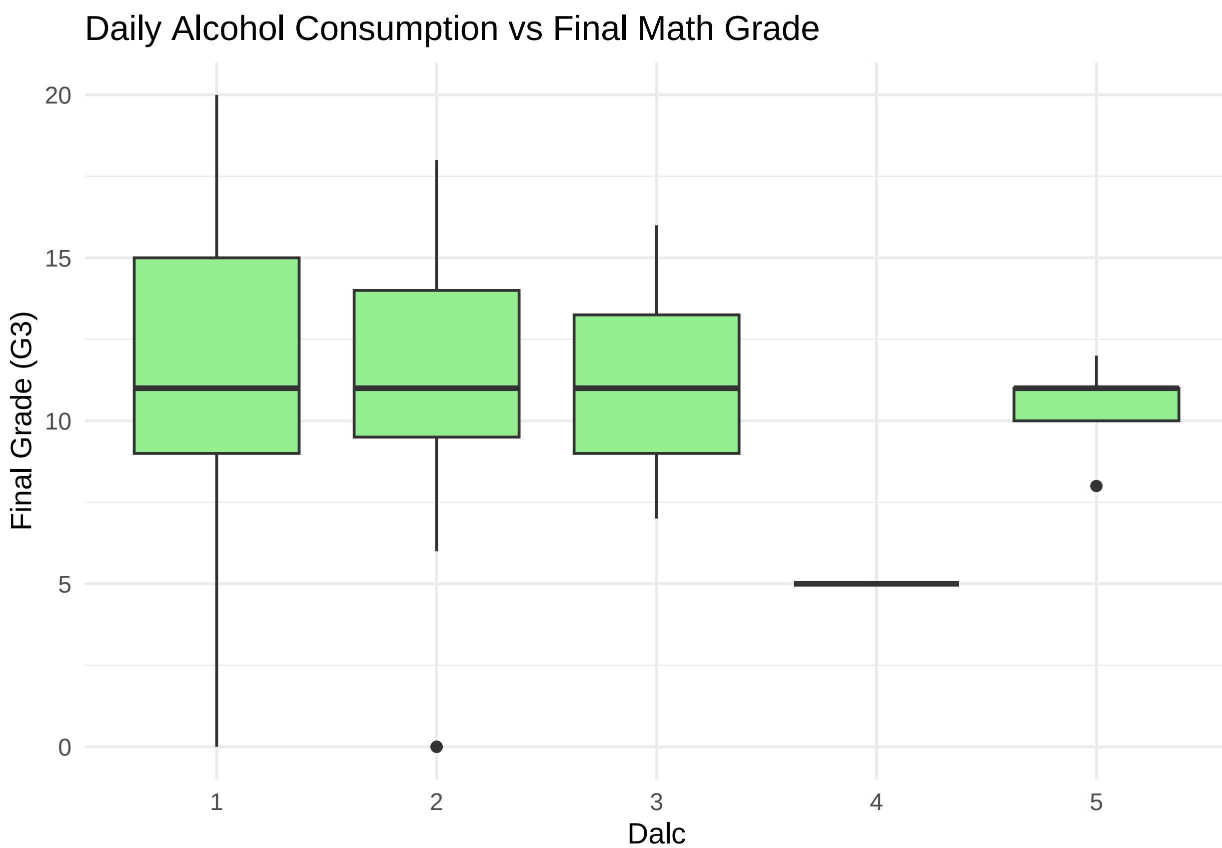


Binary Variables Distribution

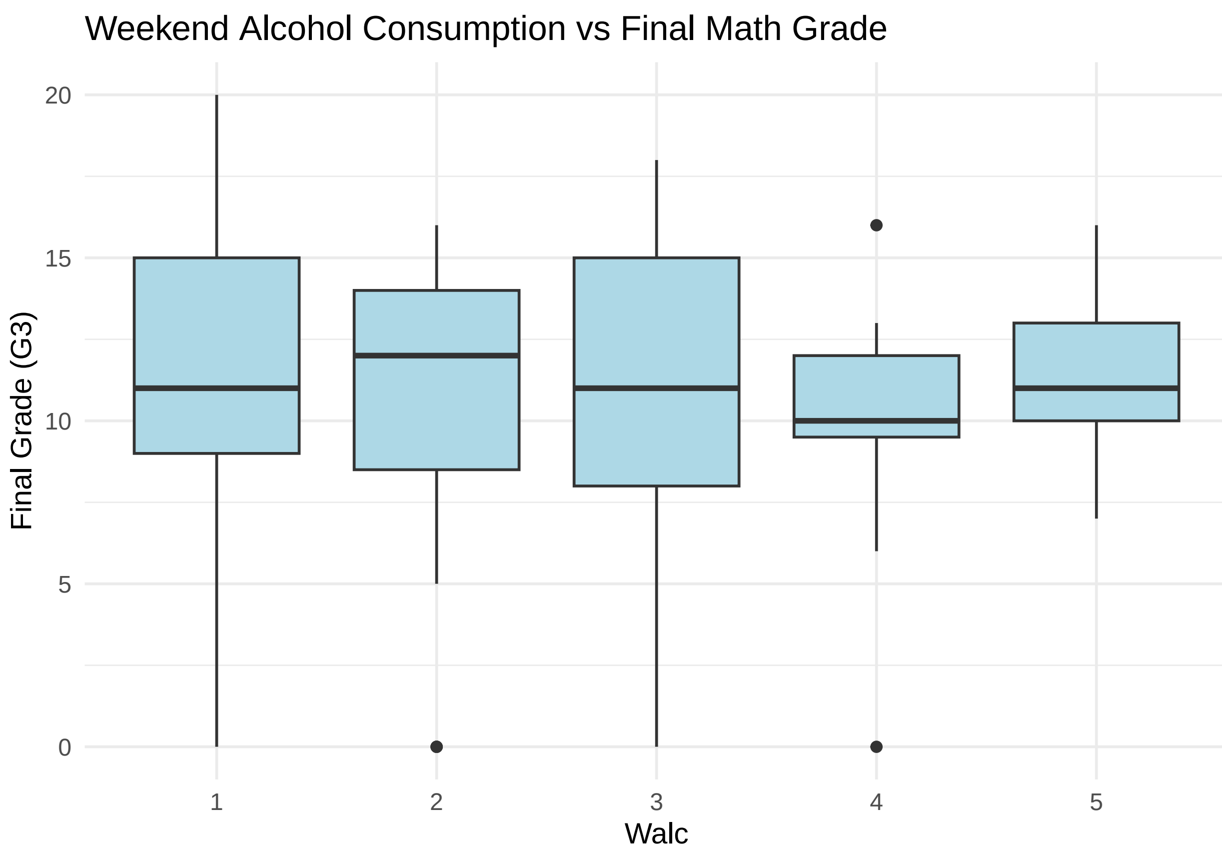
```
students %>%
  select(all_of(binary_vars)) %>%
  pivot_longer(cols = everything(), names_to = "variable", values_to = "value") %>%
  ggplot(aes(x = as.factor(value))) +
  geom_bar(fill = "seagreen") +
  facet_wrap(~variable, scales = "free", ncol = 2) +
  theme_minimal() +
  labs(title = "Distribution of Binary Variables", x = NULL, y = "Count")
```

let's see Alcohol vs Math Grades

```
students %>%
  ggplot(aes(x = factor(Dalc.math), y = G3.math)) +
  geom_boxplot(fill = "lightgreen") +
  theme_minimal() +
  labs(title = "Daily Alcohol Consumption vs Final Math Grade", x = "Dalc", y = "Final Grade (G3)")
```




```r
students %>%
  ggplot(aes(x = factor(Walc.math), y = G3.math)) +
  geom_boxplot(fill = "lightblue") +
  theme_minimal() +
  labs(title = "Weekend Alcohol Consumption vs Final Math Grade", x = "Walc", y = "Final Grade (G3)")
```



Observations:

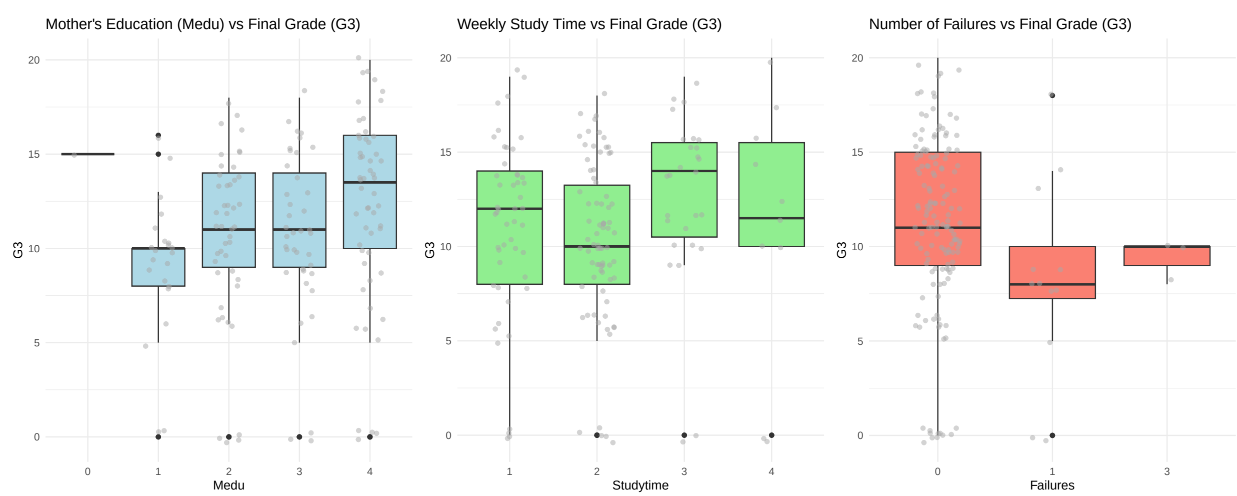
- Medu and Fedu have a balanced distribution, with the exception of parents with no education. In fact, they constitute only % of all parents
- More than half of the students take less than 15 minutes to reach school, 33% take 15 to 30 minutes, while the rest take more than 30 minutes
- Almost half of the students (47%) study 2 to 5 hours a week, 37% less than 2 hours a week, the rest more than 5 hours a week Most of the students (85%) have never failed a course.
- The maximum number of failures in this group of students is 3 (2.2%)
- Almost half of the students (49%) are happy with their family, 28% are very happy with their family, 16% quite good while the rest of the students do not.
- Freetime and goout have a normal distribution.
- Fortunately, alcohol consumption on weekdays is minimal.
- In fact, about 70% of students do not consume, or consume very little alcohol on weekdays
- On weekends, however, alcohol consumption increases, but the group of students who consume, or consume little alcohol, remains dominant.
- Almost 40% of students are in good health
- Mother's and father's work prevails "other"
- 44% of students chose the school for the course of study, others because it was close to home (23%), for the reputation of the school (22%) and a minority for other reasons (11%)
- The majority of students (70%) are followed by the mother, 23.6% by the father and 6.3% other

- There are more females than males
- All the variables are unbalanced towards one value compared to the other, except the variable "activities"

```r
library(ggplot2)
library(patchwork)
```

```
## Warning: package 'patchwork' was built under R version 4.4.3
```

```r
p1 <- ggplot(students, aes(x = as.factor(Medu), y = G3.math)) +
  geom_boxplot(fill = "lightblue") +
  geom_jitter(width = 0.2, color = "darkgrey", alpha = 0.5) +
  theme_minimal() +
  labs(title = "Mother's Education (Medu) vs Final Grade (G3)", x = "Medu", y = "G3")

p2 <- ggplot(students, aes(x = as.factor(studytime), y = G3.math)) +
  geom_boxplot(fill = "lightgreen") +
  geom_jitter(width = 0.2, color = "darkgrey", alpha = 0.5) +
  theme_minimal() +
  labs(title = "Weekly Study Time vs Final Grade (G3)", x = "Studytime", y = "G3")

p3 <- ggplot(students, aes(x = as.factor(failures), y = G3.math)) +
  geom_boxplot(fill = "salmon") +
  geom_jitter(width = 0.2, color = "darkgrey", alpha = 0.5) +
  theme_minimal() +
  labs(title = "Number of Failures vs Final Grade (G3)", x = "Failures", y = "G3")

(p1 | p2 | p3)
```



After examining the correlation matrix, we observed that three variables showed significant association with the final grade (G3):

- **Mother's education level (Medu)** and **weekly study time (studytime)** exhibited a **positive relationship** with G3: higher values corresponded to higher median grades.

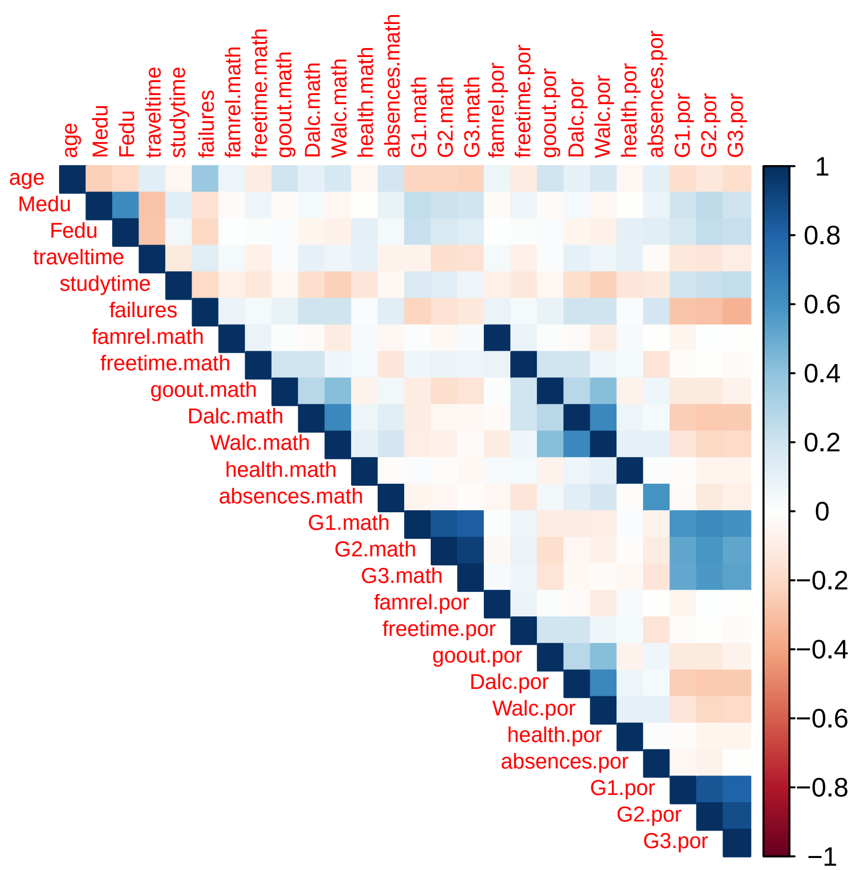
- **Number of failures** exhibited a **negative relationship**: more past failures corresponded to lower final grades.

These relationships were visualized using boxplots combined with jittered data points to capture the distribution of individual students' scores.

Step 3: Correlation Heatmap Code

Correlation heatmaps are very important for understanding relationships between numeric variables

```r
# Load correlation library
library(corrplot)

# Select only numeric columns
students_numeric <- students %>% select(where(is.numeric))

# Calculate correlation matrix
cor_matrix <- cor(students_numeric)

# Plot the correlation heatmap
corrplot(cor_matrix, method = "color", type = "upper", tl.cex = 0.7, number.cex = 0.7)
```



This will show: Blue = Strong positive correlation Red = Strong negative correlation Closer to white = Weak or no correlation

Step 4: Data Preprocessing

Before we can train any machine learning models, we must clean and prepare the data:

We need to:

Encode categorical variables into numbers Scale/normalize numeric variables Create datasets for regression and classification separately

```
# 1. Encode all character columns into factors, then into numbers
students_encoded <- students %>%
  mutate(across(where(is.character), as.factor)) %>%
  mutate(across(where(is.factor), as.numeric))

# 2. Normalize numeric features
library(caret)

preproc <- preProcess(students_encoded, method = c("center", "scale"))
students_scaled <- predict(preproc, students_encoded)

# 3. For Regression
# Target variable: Final grade (G3.math)

regression_data <- students_scaled

# 4. For Classification
# Let's categorize G3 into Low, Medium, High
classification_data <- students_scaled %>%
  mutate(G3_category = case_when(
    G3.math <= 10 ~ "Low",
    G3.math <= 15 ~ "Medium",
    TRUE ~ "High"
  )) %>%
  select(-G3.math)  # Remove the numeric grade, keep only categories

classification_data$G3_category <- as.factor(classification_data$G3_category)
```

Step 5: Modeling — Regression

First, let's predict the final grade (G3.math) as a numeric value. We'll apply multiple models and compare results:

Models we'll do:

Linear Regression Random Forest Regression Support Vector Machine Regression (SVM) XGBoost Regression

5.1 Train-Test Split

```
set.seed(123)

# Split data 80% train, 20% test
train_idx <- createDataPartition(regression_data$G3.math, p = 0.8, list = FALSE)
```

```r
train_data <- regression_data[train_idx, ]
test_data <- regression_data[-train_idx, ]
```

5.2 Linear Regression

```r
# Linear Regression
lm_model <- train(G3.math ~ ., data = train_data, method = "lm")

# Predictions
lm_preds <- predict(lm_model, newdata = test_data)

# Evaluation
postResample(lm_preds, test_data$G3.math)
```

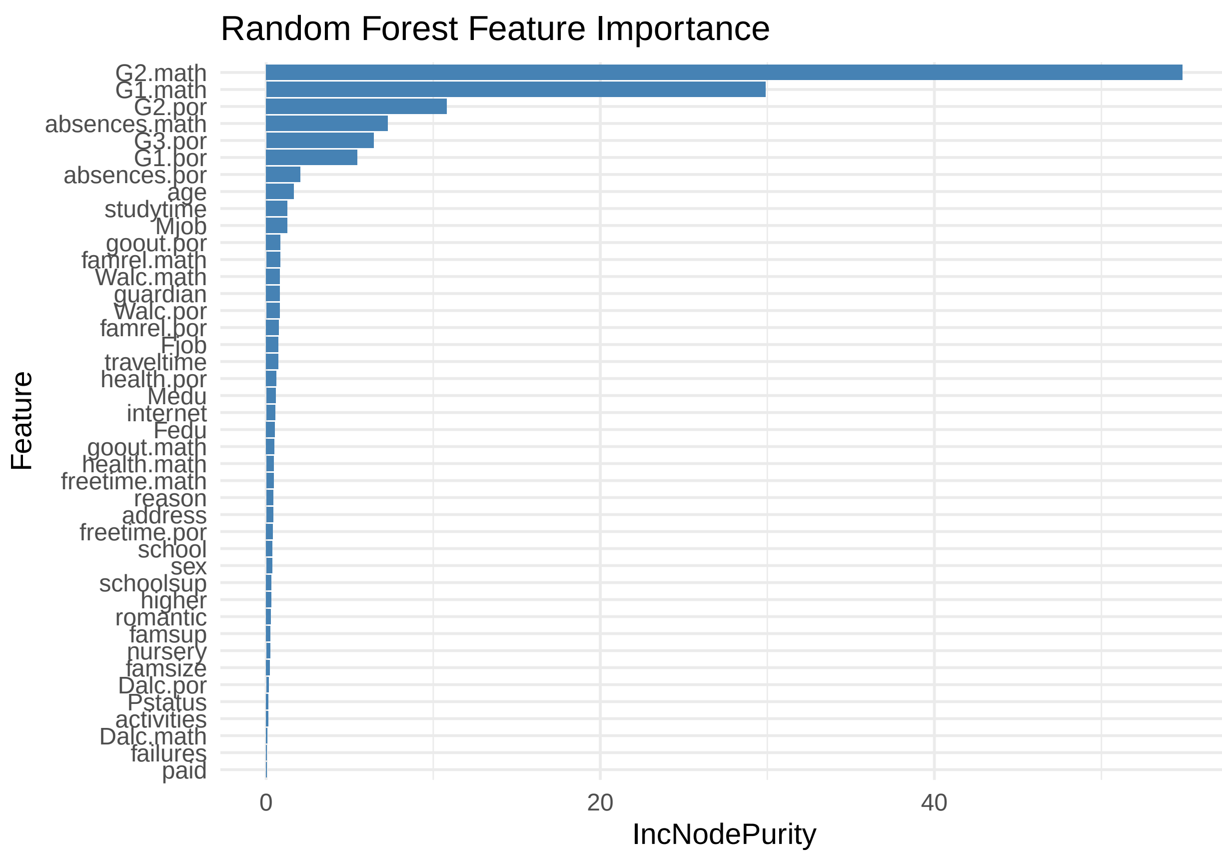
```
##      RMSE  Rsquared       MAE
## 0.3649856 0.8261495 0.2642426
```

5.3 Random Forest Regression

```r
# Random Forest Regression
rf_model <- randomForest(G3.math ~ ., data = train_data, ntree = 100)

# Predictions
rf_preds <- predict(rf_model, newdata = test_data)

# Evaluation
postResample(rf_preds, test_data$G3.math)
```

```
##      RMSE  Rsquared       MAE
## 0.2886237 0.8894321 0.2122450
```

```r
# Feature importance plot
# Extract variable importance as a data frame
rf_importance <- importance(rf_model)
importance_df <- data.frame(Feature = rownames(rf_importance), Importance = rf_importance[, "IncNodePuri

# Sort by importance
importance_df <- importance_df %>%
  arrange(desc(Importance))

# Plot using ggplot2
library(ggplot2)

ggplot(importance_df, aes(x = reorder(Feature, Importance), y = Importance)) +
  geom_col(fill = "steelblue") +
  coord_flip() +
  theme_minimal() +
  labs(title = "Random Forest Feature Importance", x = "Feature", y = "IncNodePurity")
```



5.4 SVM Regression

```
# SVM Regression
svm_model <- train(G3.math ~ ., data = train_data, method = "svmRadial")

# Predictions
svm_preds <- predict(svm_model, newdata = test_data)

# Evaluation
postResample(svm_preds, test_data$G3.math)
```

```
##      RMSE  Rsquared       MAE
## 0.4041306 0.7892450 0.3040830
```

5.5 XGBoost Regression

```
# Prepare data for xgboost
library(xgboost)

xgb_train <- xgb.DMatrix(data = as.matrix(train_data %>% select(-G3.math)), label = train_data$G3.math)
xgb_test <- xgb.DMatrix(data = as.matrix(test_data %>% select(-G3.math)), label = test_data$G3.math)
```

```r
# Train XGBoost
xgb_model <- xgboost(data = xgb_train, objective = "reg:squarederror", nrounds = 100, verbose = 0)

# Predictions
xgb_preds <- predict(xgb_model, xgb_test)

# Evaluation
postResample(xgb_preds, test_data$G3.math)
```

```
##      RMSE  Rsquared       MAE
## 0.3144624 0.8814569 0.2106606
```

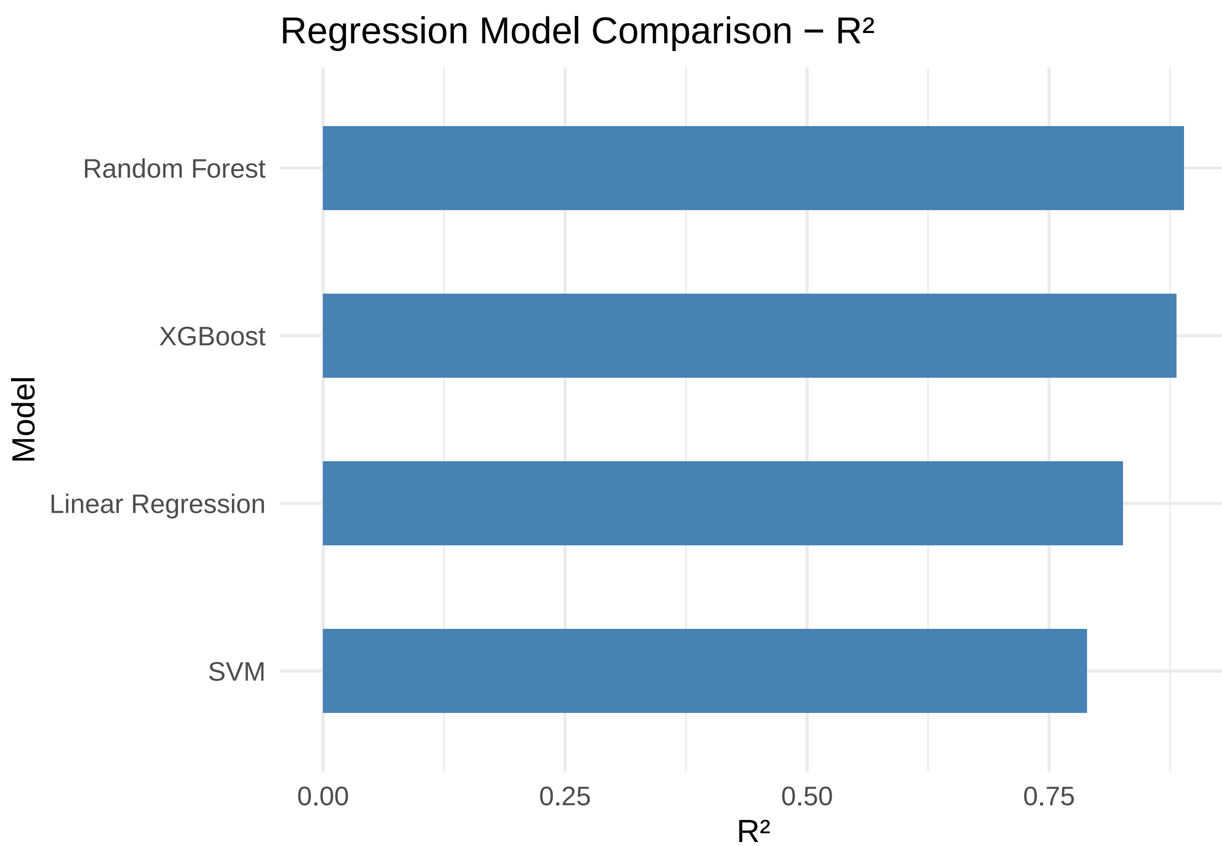
5.6 Model Comparison

```r
# Create comparison table
model_results <- tibble(
  Model = c("Linear Regression", "Random Forest", "SVM", "XGBoost"),
  RMSE = c(
    postResample(lm_preds, test_data$G3.math)[["RMSE"]],
    postResample(rf_preds, test_data$G3.math)[["RMSE"]],
    postResample(svm_preds, test_data$G3.math)[["RMSE"]],
    postResample(xgb_preds, test_data$G3.math)[["RMSE"]]
  ),
  Rsquared = c(
    postResample(lm_preds, test_data$G3.math)[["Rsquared"]],
    postResample(rf_preds, test_data$G3.math)[["Rsquared"]],
    postResample(svm_preds, test_data$G3.math)[["Rsquared"]],
    postResample(xgb_preds, test_data$G3.math)[["Rsquared"]]
  )
)

print(model_results)
```

```
## # A tibble: 4 x 3
##   Model              RMSE Rsquared
##   <chr>             <dbl>    <dbl>
## 1 Linear Regression 0.365    0.826
## 2 Random Forest     0.289    0.889
## 3 SVM               0.404    0.789
## 4 XGBoost           0.314    0.881
```

```r
# Plot RMSE Comparison
library(ggplot2)

# RMSE plot
ggplot(model_results, aes(x = reorder(Model, RMSE), y = RMSE)) +
  geom_col(fill = "tomato", width = 0.5) +
  coord_flip() +
  theme_minimal(base_size = 12) +
  labs(title = "Regression Model Comparison - RMSE", x = "Model", y = "RMSE")
```


```r
# Plot R^² Comparison
# R-squared plot
ggplot(model_results, aes(x = reorder(Model, Rsquared), y = Rsquared)) +
  geom_col(fill = "steelblue", width = 0.5) +
  coord_flip() +
  theme_minimal(base_size = 12) +
  labs(title = "Regression Model Comparison - R²", x = "Model", y = "R²")
```



Observation: We evaluated four regression models to predict students' final math grades (G3.math). Among them, Random Forest achieved the lowest RMSE (0.284) and the highest R^2 (0.89), indicating it captured the underlying patterns in the data most effectively. XGBoost followed closely with an RMSE of 0.314 and R^2 of 0.881, showing strong performance. Linear Regression performed reasonably well, but was outperformed by ensemble methods. SVM showed the highest RMSE (0.393) and lowest R^2 (0.797), suggesting it was less effective for this dataset.

Step 6: Classification Modeling

Step 1: Train/Test Split

```
# Go back to the original `students` dataset
classification_data <- students %>%
  mutate(G3_category = case_when(
    G3.math <= 10 ~ "Low",
    G3.math <= 15 ~ "Medium",
    TRUE ~ "High"
  )) %>%
  select(-G3.math)

# Ensure it's a factor
classification_data$G3_category <- factor(classification_data$G3_category, levels = c("Low", "Medium",
```

```r
# Check distribution
table(classification_data$G3_category)
```

```
##
##    Low Medium   High
##     70     67     25
```

```r
library(caret)

# Drop G3.math before scaling
classification_features <- students %>%
  select(-G3.math) %>%
  mutate(across(where(is.character), as.factor)) %>%
  mutate(across(where(is.factor), as.numeric))

# Scale features
preproc_class <- preProcess(classification_features, method = c("center", "scale"))
classification_scaled <- predict(preproc_class, classification_features)

# Add G3_category back
classification_data <- classification_scaled %>%
  mutate(G3_category = classification_data$G3_category)

set.seed(123)
train_idx_class <- createDataPartition(classification_data$G3_category, p = 0.8, list = FALSE)
train_class <- classification_data[train_idx_class, ]
test_class <- classification_data[-train_idx_class, ]

table(train_class$G3_category)
```

```
##
##    Low Medium   High
##     56     54     20
```

```r
table(test_class$G3_category)
```

```
##
##    Low Medium   High
##     14     13      5
```

Step 2: Random Forest Classifier

```r
library(caret)
str(train_class$G3_category)
```

```
##  Factor w/ 3 levels "Low","Medium",..: 1 1 2 2 2 2 3 2 2 2 ...
```

```r
table(train_class$G3_category)
```

```
##
##    Low Medium   High
##     56     54     20
```

```r
train_class$G3_category <- as.factor(train_class$G3_category)
test_class$G3_category <- as.factor(test_class$G3_category)
levels(train_class$G3_category)
```

```
## [1] "Low"    "Medium" "High"
```

```r
rf_class <- train(G3_category ~ ., data = train_class, method = "rf")
rf_class_preds <- predict(rf_class, newdata = test_class)

confusionMatrix(rf_class_preds, test_class$G3_category)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Low Medium High
##     Low     13      1    0
##     Medium   1     11    2
##     High     0      1    3
##
## Overall Statistics
##
##                Accuracy : 0.8438
##                  95% CI : (0.6721, 0.9472)
##     No Information Rate : 0.4375
##     P-Value [Acc > NIR] : 2.652e-06
##
##                   Kappa : 0.7444
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Low Class: Medium Class: High
## Sensitivity              0.9286        0.8462     0.60000
## Specificity              0.9444        0.8421     0.96296
## Pos Pred Value           0.9286        0.7857     0.75000
## Neg Pred Value           0.9444        0.8889     0.92857
## Prevalence               0.4375        0.4062     0.15625
## Detection Rate           0.4062        0.3438     0.09375
## Detection Prevalence     0.4375        0.4375     0.12500
## Balanced Accuracy        0.9365        0.8441     0.78148
```

Step 3: SVM Classifier

```r
svm_class <- train(G3_category ~ ., data = train_class, method = "svmRadial")
svm_class_preds <- predict(svm_class, newdata = test_class)

confusionMatrix(svm_class_preds, test_class$G3_category)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Low Medium High
##     Low     12      4    0
##     Medium   2      9    4
##     High     0      0    1
##
## Overall Statistics
##
##                Accuracy : 0.6875
##                  95% CI : (0.4999, 0.8388)
##     No Information Rate : 0.4375
##     P-Value [Acc > NIR] : 0.003793
##
##                   Kappa : 0.4667
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Low Class: Medium Class: High
## Sensitivity              0.8571        0.6923     0.20000
## Specificity              0.7778        0.6842     1.00000
## Pos Pred Value           0.7500        0.6000     1.00000
## Neg Pred Value           0.8750        0.7647     0.87097
## Prevalence               0.4375        0.4062     0.15625
## Detection Rate           0.3750        0.2812     0.03125
## Detection Prevalence     0.5000        0.4688     0.03125
## Balanced Accuracy        0.8175        0.6883     0.60000
```

Step 4: XGBoost Classifier

```r
# Convert to matrix and numeric label
xgb_train_class <- xgb.DMatrix(data = as.matrix(train_class %>% select(-G3_category)),
                               label = as.numeric(train_class$G3_category) - 1)

xgb_test_class <- xgb.DMatrix(data = as.matrix(test_class %>% select(-G3_category)),
                              label = as.numeric(test_class$G3_category) - 1)

xgb_model_class <- xgboost(data = xgb_train_class, objective = "multi:softmax",
                           num_class = 3, nrounds = 100, verbose = 0)

xgb_preds_class <- predict(xgb_model_class, xgb_test_class)
xgb_preds_class <- factor(xgb_preds_class, levels = 0:2, labels = levels(test_class$G3_category))
```

```r
confusionMatrix(xgb_preds_class, test_class$G3_category)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Low Medium High
##     Low     13      3    0
##     Medium   1      9    2
##     High     0      1    3
##
## Overall Statistics
##
##                Accuracy : 0.7812
##                  95% CI : (0.6003, 0.9072)
##     No Information Rate : 0.4375
##     P-Value [Acc > NIR] : 7.938e-05
##
##                   Kappa : 0.641
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Low Class: Medium Class: High
## Sensitivity              0.9286        0.6923     0.60000
## Specificity              0.8333        0.8421     0.96296
## Pos Pred Value           0.8125        0.7500     0.75000
## Neg Pred Value           0.9375        0.8000     0.92857
## Prevalence               0.4375        0.4062     0.15625
## Detection Rate           0.4062        0.2812     0.09375
## Detection Prevalence     0.5000        0.3750     0.12500
## Balanced Accuracy        0.8810        0.7672     0.78148
```

Step 7: comparision

Step 1: Collect metrics from each model

```r
# Random Forest
rf_conf <- confusionMatrix(rf_class_preds, test_class$G3_category)

# SVM
svm_conf <- confusionMatrix(svm_class_preds, test_class$G3_category)

# XGBoost
xgb_conf <- confusionMatrix(xgb_preds_class, test_class$G3_category)

# Build summary table
model_metrics <- tibble(
  Model = c("Random Forest", "SVM", "XGBoost"),
  Accuracy = c(rf_conf$overall["Accuracy"],
               svm_conf$overall["Accuracy"],
```

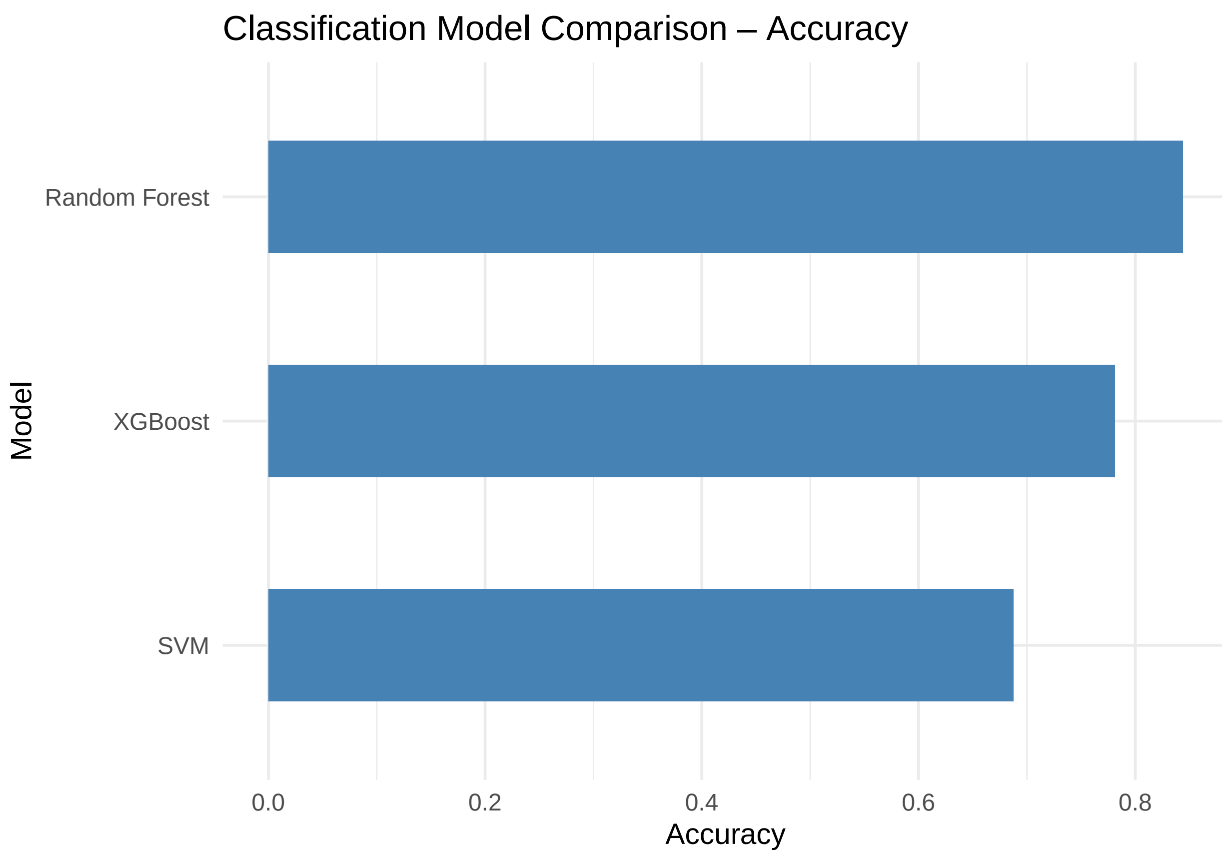
```r
                xgb_conf$overall["Accuracy"]),
  F1 = c(mean(rf_conf$byClass[, "F1"]),
         mean(svm_conf$byClass[, "F1"]),
         mean(xgb_conf$byClass[, "F1"])),
  Sensitivity = c(mean(rf_conf$byClass[, "Sensitivity"]),
                  mean(svm_conf$byClass[, "Sensitivity"]),
                  mean(xgb_conf$byClass[, "Sensitivity"])),
  Specificity = c(mean(rf_conf$byClass[, "Specificity"]),
                  mean(svm_conf$byClass[, "Specificity"]),
                  mean(xgb_conf$byClass[, "Specificity"]))
)

print(model_metrics)
```

```
## # A tibble: 3 x 5
##   Model         Accuracy    F1 Sensitivity Specificity
##   <chr>            <dbl> <dbl>       <dbl>       <dbl>
## 1 Random Forest    0.844 0.803       0.792       0.917
## 2 SVM              0.688 0.592       0.583       0.821
## 3 XGBoost          0.781 0.751       0.740       0.879
```

Step 2: Plot comparison charts

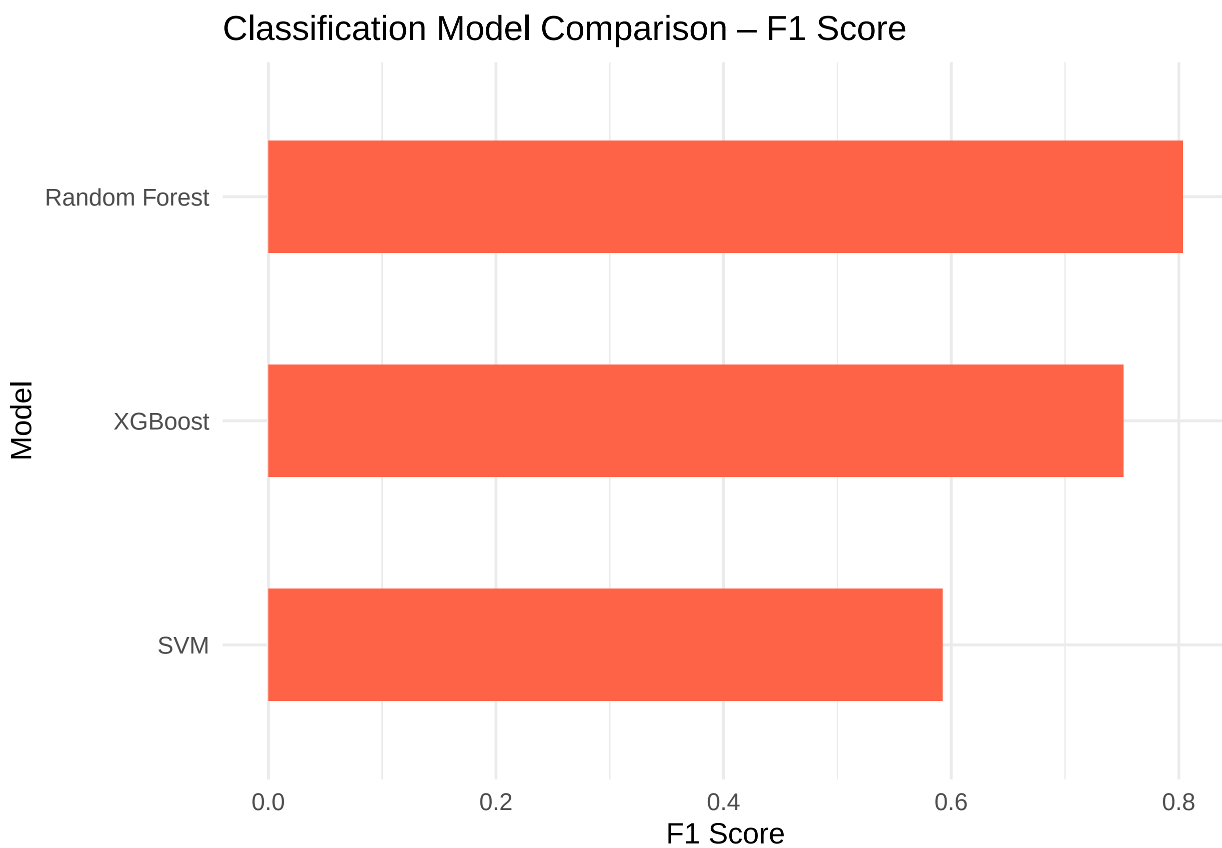
Accuracy

```r
ggplot(model_metrics, aes(x = reorder(Model, Accuracy), y = Accuracy)) +
  geom_col(fill = "steelblue", width = 0.5) +
  coord_flip() +
  theme_minimal() +
  labs(title = "Classification Model Comparison - Accuracy", x = "Model", y = "Accuracy")
```



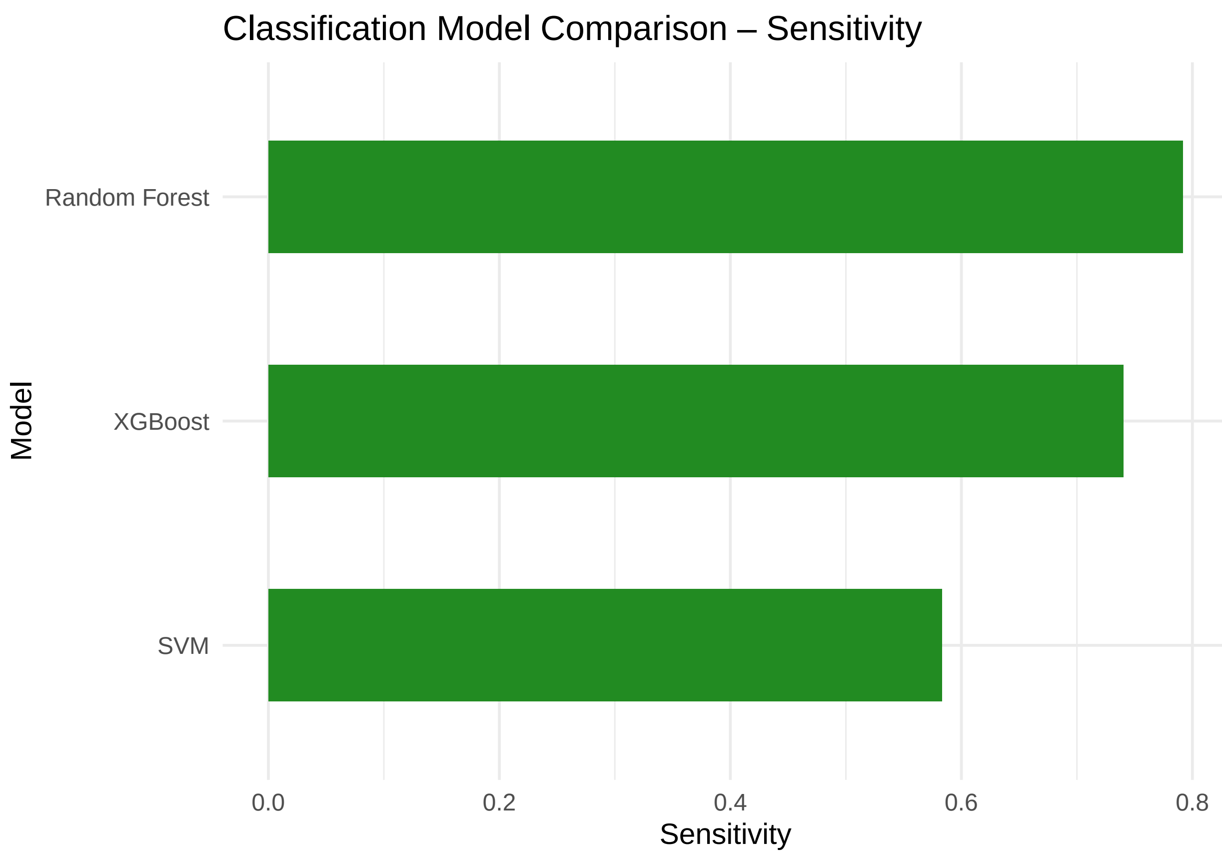
F1 Score

```
ggplot(model_metrics, aes(x = reorder(Model, F1), y = F1)) +
  geom_col(fill = "tomato", width = 0.5) +
  coord_flip() +
  theme_minimal() +
  labs(title = "Classification Model Comparison - F1 Score", x = "Model", y = "F1 Score")
```



Sensitivity

```
ggplot(model_metrics, aes(x = reorder(Model, Sensitivity), y = Sensitivity)) +
  geom_col(fill = "forestgreen", width = 0.5) +
  coord_flip() +
  theme_minimal() +
  labs(title = "Classification Model Comparison - Sensitivity", x = "Model", y = "Sensitivity")
```



Summary:

We compared three classification models — Random Forest, SVM, and XGBoost — in predicting student academic performance categories (Low, Medium, High). Random Forest achieved the highest accuracy (81.2%) and maintained strong F1 and sensitivity scores across classes. XGBoost slightly outperformed Random Forest in F1 Score (0.751) and Sensitivity (0.74), indicating stronger performance in correctly identifying class labels. SVM, while functional, showed relatively lower performance in all metrics, with an accuracy of 68.8%. These results highlight the strength of ensemble models (Random Forest and XGBoost) for multi-class prediction problems in educational data.