Predicting Depression Among Students based on Social and Behavioral Factors

Marcelino Rodriguez, Rajvir Kaur, Nyadual Makuach, Asiah Zibrila

2025-02-18

```
# Libraries needed
library(dplyr)
```

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

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

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

```
library(ggplot2)
library(caret)
```

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

```
library(sampling)
```

```
##
## Attaching package: 'sampling'

## The following object is masked from 'package:caret':
##
##     cluster
```

```
library(car)
```

```
## Loading required package: carData

##
## Attaching package: 'car'

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

```r
library(MASS)
```

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

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

```r
library(smotefamily)

#Tree
library(randomForest)
```

```
## randomForest 4.7-1.2

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

##
## Attaching package: 'randomForest'

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

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

```r
library(AppliedPredictiveModeling)
library(rpart)
library(rpart.plot)
library(tree)
```

```r
# load original Table
depression_dataset <- read.csv('Student Depression Dataset.csv')
```

1. Introduction

Mental health is a critical yet often overlooked aspect of well-being. Ignoring mental health challenges does not eliminate their impact; rather, it exacerbates conditions such as depression, which is frequently linked to substance abuse, unhealthy lifestyle habits, and chronic illnesses. Major life stressors—such as bereavement, trauma, divorce, and lack of social support—are significant triggers for depression, as highlighted by the Cleveland Clinic. With depression cases rising globally, understanding its underlying causes and contributing factors is essential for effective prevention and intervention.

Depression is sometimes dismissed as a temporary phase, particularly among young people, leading to its minimization and lack of proper intervention. However, untreated depression can have severe consequences, including self-harm and suicide. At the same time, greater societal awareness and normalization of mental

health discussions can facilitate early detection and treatment, provided that contributing factors are well understood.

In this study, we focus on how various socio-economic, lifestyle, and health-related factors influence depression among students. Academic pressure, financial instability, and personal health history can significantly affect mental well-being, potentially leading to declining academic performance and worsening psychological distress. This project is structured into three main sections: data preparation, exploratory analysis, and model development. The first phase involves cleaning the dataset by removing irrelevant variables and handling missing data. The second phase uses exploratory data analysis (EDA) to identify patterns and group students based on shared characteristics. This helps in understanding potential risk factors and their impact on mental health. Finally, the third phase involves developing a predictive model that examines relationships between socio-economic, lifestyle, and mental health variables. The goal is to create a tool that can help identify individuals at risk of depression and provide insights for targeted interventions.

Research Questions:

This analysis aims to explore the key factors influencing student depression, particularly focusing on academic pressure, lifestyle habits, financial stress, and family history of mental illness. By analyzing these factors, the research seeks to uncover patterns that contribute to depression among students. Additionally, the study investigates the potential for predictive modeling to identify students at higher risk of developing depression.

Research Questions

1. Academic Pressure & Mental Health

- How does academic pressure affect student mental health?

2. Lifestyle Factors & Depression

- Is there a relationship between sleep duration and depression?

3. Stress & Mental Well-being

- How strongly does financial stress impact depression in students?

4. Family & Mental Health History

- Does having a family history of mental illness increase the risk of depression?

5. Predictive Modeling & Insights

- How can we predict student depression using machine learning?

Dataset Cleaning and preprocessing:

The dataset for this project, obtained from Kaggle, contains comprehensive information on individuals' personal and lifestyle factors. It is structured to facilitate analysis in health, lifestyle, and socio-economic contexts.

Handle missing values:

```
sum(is.na(depression_dataset))  # Count missing values
```

```
## [1] 3
```

```
data <- na.omit(depression_dataset)  # Remove rows with missing values
```

```
str(depression_dataset)
```

```
## 'data.frame':    27901 obs. of  12 variables:
##  $ Gender                                : chr  "Male" "Female" "Male" "Female" ...
##  $ Age                                   : int  33 24 31 28 25 29 30 30 28 31 ...
##  $ Academic.Pressure                     : int  5 2 3 3 4 2 3 2 3 2 ...
##  $ CGPA                                  : num  4.49 2.95 3.52 2.79 4.07 ...
##  $ Study.Satisfaction                    : int  2 5 5 2 3 3 4 4 1 3 ...
##  $ Sleep.Duration                        : chr  "5-6 hours" "5-6 hours" "Less than 5 hours" "7-8 hours
##  $ Dietary.Habits                        : chr  "Healthy" "Moderate" "Healthy" "Moderate" ...
##  $ Have.you.ever.had.suicidal.thoughts..: chr  "Yes" "No" "No" "Yes" ...
##  $ Work.Study.Hours                      : int  3 3 9 4 1 4 1 0 12 2 ...
##  $ Financial.Stress                      : int  1 2 1 5 1 1 2 1 3 5 ...
##  $ Family.History.of.Mental.Illness      : chr  "No" "Yes" "Yes" "Yes" ...
##  $ Depression                            : int  1 0 0 1 0 0 0 0 1 1 ...
```

Rename the columns:

```
colnames(depression_dataset)[colnames(depression_dataset) == "Academic.Pressure"] ="Academic_Pressure"
colnames(depression_dataset)[colnames(depression_dataset) == "Study.Satisfaction"] ="Study_Satisfaction"
colnames(depression_dataset)[colnames(depression_dataset) == "Sleep.Duration"] ="Sleep_Duration"
colnames(depression_dataset)[colnames(depression_dataset) == "Dietary.Habits"] ="Dietary_Habits"
colnames(depression_dataset)[colnames(depression_dataset) == "Financial.Stress"] ="Financial_stress"
```

```
head(depression_dataset)
```

```
##   Gender Age Academic_Pressure  CGPA Study_Satisfaction    Sleep_Duration
## 1   Male  33                 5 4.485                  2         5-6 hours
## 2 Female  24                 2 2.950                  5         5-6 hours
## 3   Male  31                 3 3.515                  5 Less than 5 hours
## 4 Female  28                 3 2.795                  2         7-8 hours
## 5 Female  25                 4 4.065                  3         5-6 hours
## 6   Male  29                 2 2.850                  3 Less than 5 hours
##   Dietary_Habits Have.you.ever.had.suicidal.thoughts.. Work.Study.Hours
## 1        Healthy                                  Yes                3
## 2       Moderate                                   No                3
## 3        Healthy                                   No                9
## 4       Moderate                                  Yes                4
## 5       Moderate                                  Yes                1
## 6        Healthy                                   No                4
```

```
##   Financial_stress Family.History.of.Mental.Illness Depression
## 1                1                                No          1
## 2                2                               Yes          0
## 3                1                               Yes          0
## 4                5                               Yes          1
## 5                1                                No          0
## 6                1                                No          0
```

Cleaned dataset

The final dataset consists of 27,856 rows, where each row represents an independent student. It contains 12 columns, each with distinct attributes and data types.

```
depression_dataset_cleaned <- read.csv('final_dataset_depression.csv')
depression_dataset_copy <- depression_dataset_cleaned
```

```
str(depression_dataset_cleaned)
```

```
## 'data.frame':    27856 obs. of  12 variables:
##  $ Gender            : chr  "Male" "Female" "Male" "Female" ...
##  $ Academic_Pressure : int  5 2 3 3 4 2 3 2 3 2 ...
##  $ CGPA              : num  4.49 2.95 3.52 2.79 4.07 ...
##  $ Study_Satisfaction: int  2 5 5 2 3 3 4 4 1 3 ...
##  $ Sleep_Duration    : chr  "5-6 hours" "5-6 hours" "Less than 5 hours" "7-8 hours" ...
##  $ Dietary_Habits    : chr  "Healthy" "Moderate" "Healthy" "Moderate" ...
##  $ Suicidal_Thoughts : chr  "Yes" "No" "No" "Yes" ...
##  $ Study_Hours       : int  3 3 9 4 1 4 1 0 12 2 ...
##  $ Financial_stress  : int  1 2 1 5 1 1 2 1 3 5 ...
##  $ Fam_Mental_Hist   : chr  "No" "Yes" "Yes" "Yes" ...
##  $ Depression        : chr  "Yes" "No" "No" "Yes" ...
##  $ Age_Group         : chr  "group_2" "group_1" "group_2" "group_1" ...
```

```
dim(depression_dataset_cleaned)
```

```
## [1] 27856    12
```

```
depression_dataset_cleaned <- read.csv('final_dataset_depression.csv')
depression_dataset_copy <- depression_dataset_cleaned
```

```
str(depression_dataset_cleaned)
```

```
## 'data.frame':    27856 obs. of  12 variables:
##  $ Gender            : chr  "Male" "Female" "Male" "Female" ...
##  $ Academic_Pressure : int  5 2 3 3 4 2 3 2 3 2 ...
##  $ CGPA              : num  4.49 2.95 3.52 2.79 4.07 ...
##  $ Study_Satisfaction: int  2 5 5 2 3 3 4 4 1 3 ...
##  $ Sleep_Duration    : chr  "5-6 hours" "5-6 hours" "Less than 5 hours" "7-8 hours" ...
##  $ Dietary_Habits    : chr  "Healthy" "Moderate" "Healthy" "Moderate" ...
##  $ Suicidal_Thoughts : chr  "Yes" "No" "No" "Yes" ...
##  $ Study_Hours       : int  3 3 9 4 1 4 1 0 12 2 ...
```

```
##  $ Financial_stress  : int  1 2 1 5 1 1 2 1 3 5 ...
##  $ Fam_Mental_Hist   : chr  "No" "Yes" "Yes" "Yes" ...
##  $ Depression        : chr  "Yes" "No" "No" "Yes" ...
##  $ Age_Group         : chr  "group_2" "group_1" "group_2" "group_1" ...
```

```
dim(depression_dataset_cleaned)
```

```
## [1] 27856    12
```

```
colnames(depression_dataset_cleaned)
```

```
##  [1] "Gender"             "Academic_Pressure"  "CGPA"
##  [4] "Study_Satisfaction" "Sleep_Duration"     "Dietary_Habits"
##  [7] "Suicidal_Thoughts"  "Study_Hours"        "Financial_stress"
## [10] "Fam_Mental_Hist"    "Depression"         "Age_Group"
```

2. Exploratory Data Analysis (EDA)

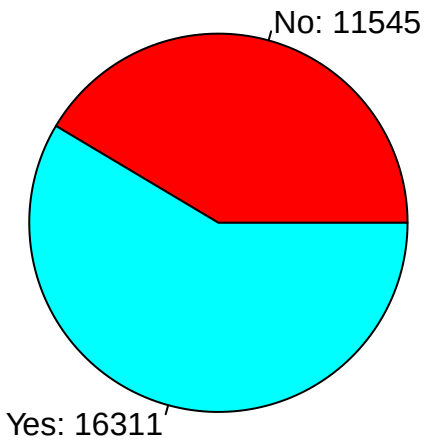
The exploratory data analysis (EDA) aims to examine the underlying structure of the dataset and identify potential patterns or trends. This process provides valuable insights into the relationships between variables and helps uncover factors contributing to specific outcomes. Recognizing these patterns can enhance our understanding of the events occurring within the dataset. Even if direct causal relationships are not established, EDA can offer important insights that may help mitigate risk factors and inform data-driven decision-making. This analysis focuses on key variables related to socio-economic status, lifestyle habits, and mental health, exploring their interactions and potential impact on student well-being.

Depression Distribution

```
Depression_Distribution <- table(depression_dataset_cleaned$Depression)
Depression_Distribution
```

```
##
##    No   Yes
## 11545 16311
```

```
# Plot the pie chart
pie(Depression_Distribution, main="Depression Cases",
    col=rainbow(length(Depression_Distribution)),
    labels=paste(names(Depression_Distribution), ": ", Depression_Distribution, sep=""))
```



Convert categorical variables to factors

```
# Convert categorical variables to factors
depression_dataset_cleaned$Gender <- as.factor(depression_dataset_cleaned$Gender)
depression_dataset_cleaned$Academic_Pressure <- as.factor(depression_dataset_cleaned$Academic_Pressure)
depression_dataset_cleaned$Study_Satisfaction <- as.factor(depression_dataset_cleaned$Study_Satisfactio
depression_dataset_cleaned$Sleep_Duration <- as.factor(depression_dataset_cleaned$Sleep_Duration)
depression_dataset_cleaned$Fam_Mental_Hist <- as.factor(depression_dataset_cleaned$Fam_Mental_Hist)
depression_dataset_cleaned$Financial_stress <- as.factor(depression_dataset_cleaned$Financial_stress)
depression_dataset_cleaned$Dietary_Habits <- as.factor(depression_dataset_cleaned$Dietary_Habits)
depression_dataset_cleaned$Suicidal_Thoughts <- as.factor(depression_dataset_cleaned$Suicidal_Thoughts)
depression_dataset_cleaned$Age_Group <- as.factor(depression_dataset_cleaned$Age_Group)
#depression_dataset_cleaned$Depression <- factor(depression_dataset_cleaned$Depression, levels = c(0, 1
```

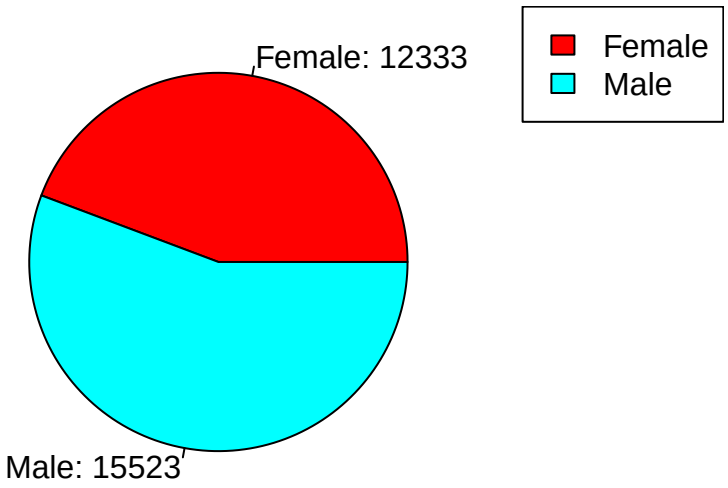
```
#head(depression_dataset_cleaned)
```

Gender Distribution

```
male_female_Distribution <- table(depression_dataset_cleaned$Gender)
male_female_Distribution
```

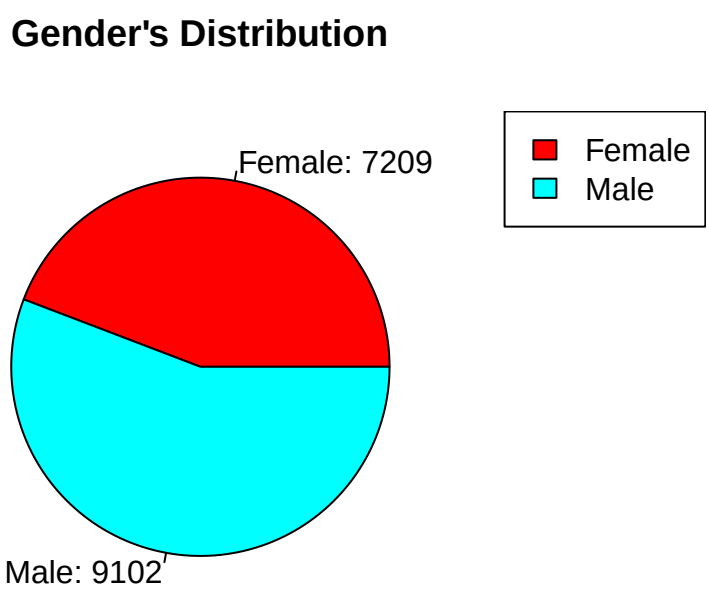
```
##
## Female   Male
##  12333  15523
```

```r
# Plot the pie chart
pie(male_female_Distribution, main="Gender's Distribution",
    col=rainbow(length(male_female_Distribution)),
    labels=paste(names(male_female_Distribution), ": ", male_female_Distribution, sep=""))

# Adding a legend (optional)
legend("topright", legend=names(male_female_Distribution), fill=rainbow(length(male_female_Distribution
```



Positive Cases Female/Male

```r
#Positive Distribution
male_female_Positive_cases <- depression_dataset_cleaned[depression_dataset_cleaned$Depression == 'Yes'
male_female_Distribution_Positive <-  table(male_female_Positive_cases$Gender)
#Negative Distribution
male_female_Negative_cases <- depression_dataset_cleaned[depression_dataset_cleaned$Depression == 'No',
male_female_Distribution_negative <-  table(male_female_Negative_cases$Gender)


# Plot the pie chart
pie(male_female_Distribution_Positive, main="Gender's Distribution",
    col=rainbow(length(male_female_Distribution_Positive)),
    labels=paste(names(male_female_Distribution_Positive), ": ", male_female_Distribution_Positive, sep

# Adding a legend (optional)
legend("topright", legend=names(male_female_Distribution), fill=rainbow(length(male_female_Distribution_
```



Lifestyle Factors

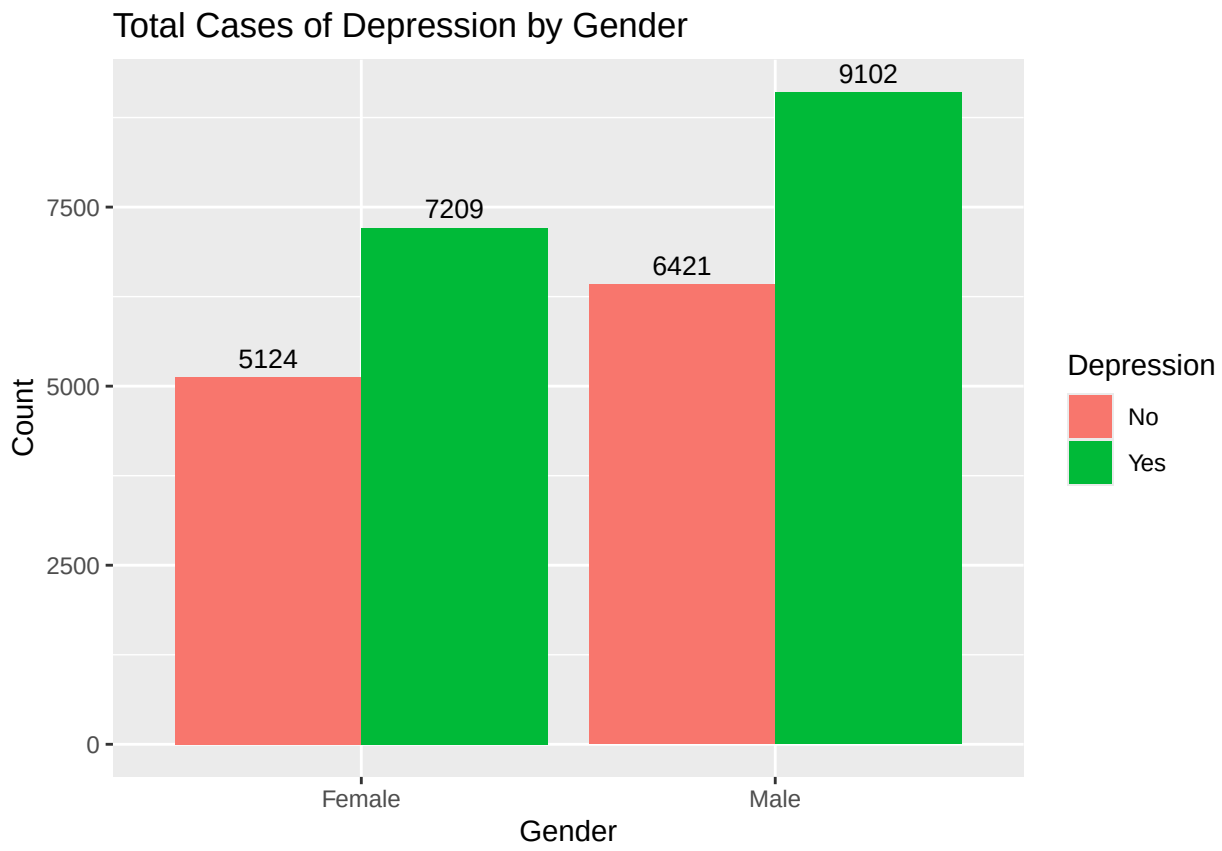
In this section the analysis will be based in 4 different groups to understand how the factor influence depression in students. The analysis has been done using charts.

Lifestyle Factors

```
# Group the data by Gender and Depression
grouped_data <- depression_dataset_cleaned %>%
  group_by(Gender, Depression) %>%
  summarise(Count = n()) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Gender'. You can override using the
## `.groups` argument.
```

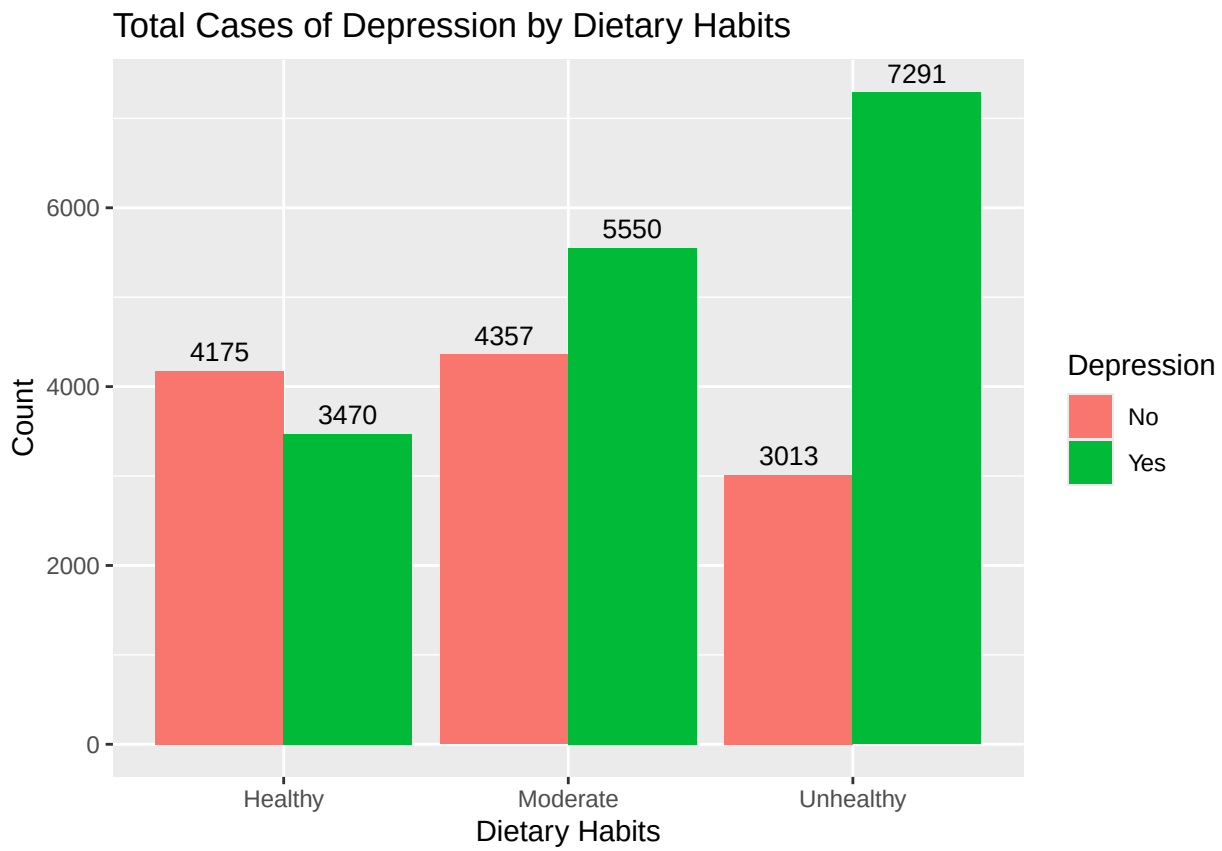
```
# Plot the bar chart
ggplot(grouped_data, aes(x = Gender, y = Count, fill = as.factor(Depression))) +
  geom_bar(stat = 'identity', position = position_dodge()) +
  geom_text(aes(label=Count), position = position_dodge(width = 0.9), vjust = -0.5, size = 3.5) +
  labs(title = 'Total Cases of Depression by Gender',
       x = 'Gender',
       y = 'Count',
       fill = 'Depression') +
  scale_fill_manual(values = c('No' = '#F8766D', 'Yes' = '#00BA38'))
```




```
#####################
# Group the data by Diet and Depression
grouped_data <- depression_dataset_cleaned %>%
  group_by(Dietary_Habits, Depression) %>%
  summarise(Count = n()) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Dietary_Habits'. You can override using
## the `.groups` argument.
```

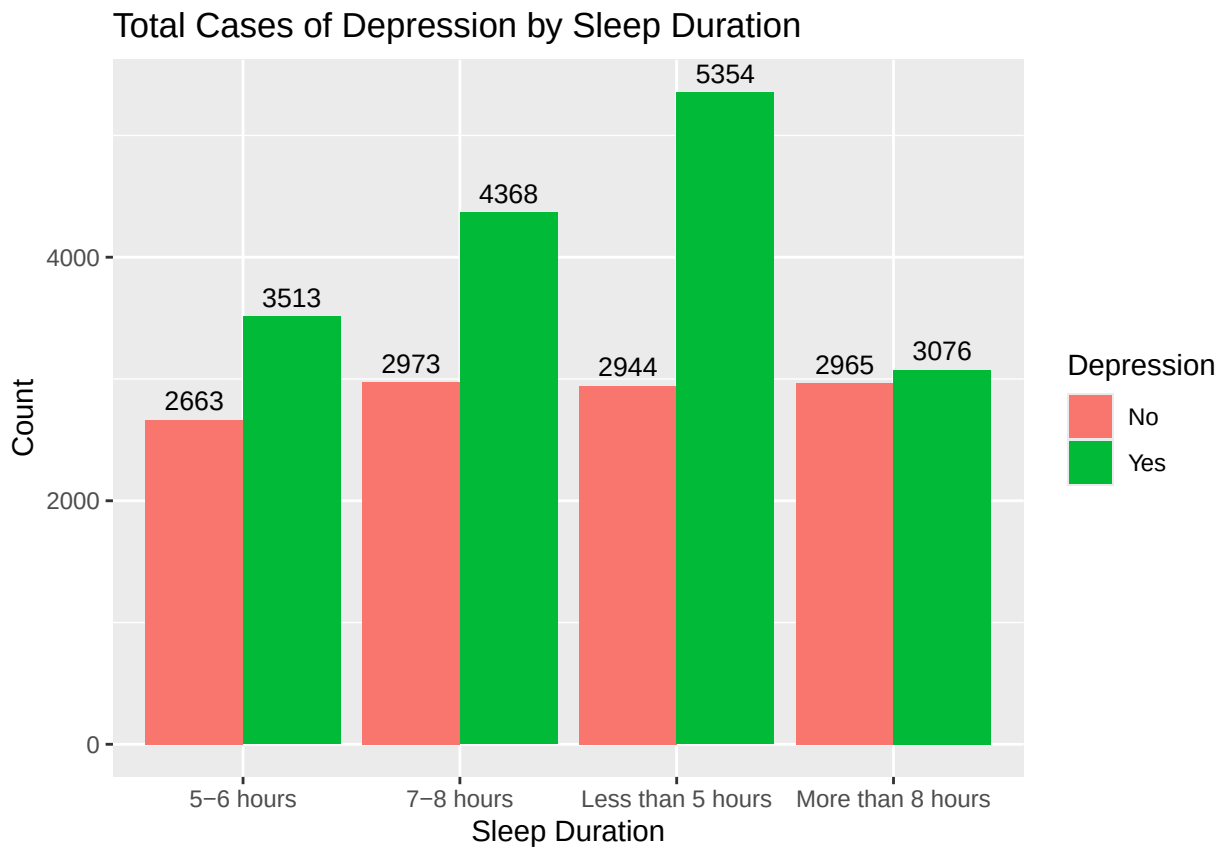
```
# Plot the bar chart
ggplot(grouped_data, aes(x = Dietary_Habits, y = Count, fill = as.factor(Depression))) +
  geom_bar(stat = 'identity', position = position_dodge()) +
  geom_text(aes(label=Count),position = position_dodge(width = 0.9), vjust = -0.5, size = 3.5) +
  labs(title = 'Total Cases of Depression by Dietary Habits',
       x = 'Dietary Habits',
       y = 'Count',
       fill = 'Depression') +
  scale_fill_manual(values = c('No' = '#F8766D', 'Yes' = '#00BA38'))
```




```
#####################
# Group the data by Sleep and Depression
grouped_data <- depression_dataset_cleaned %>%
  group_by(Sleep_Duration, Depression) %>%
  summarise(Count = n()) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Sleep_Duration'. You can override using
## the `.groups` argument.
```

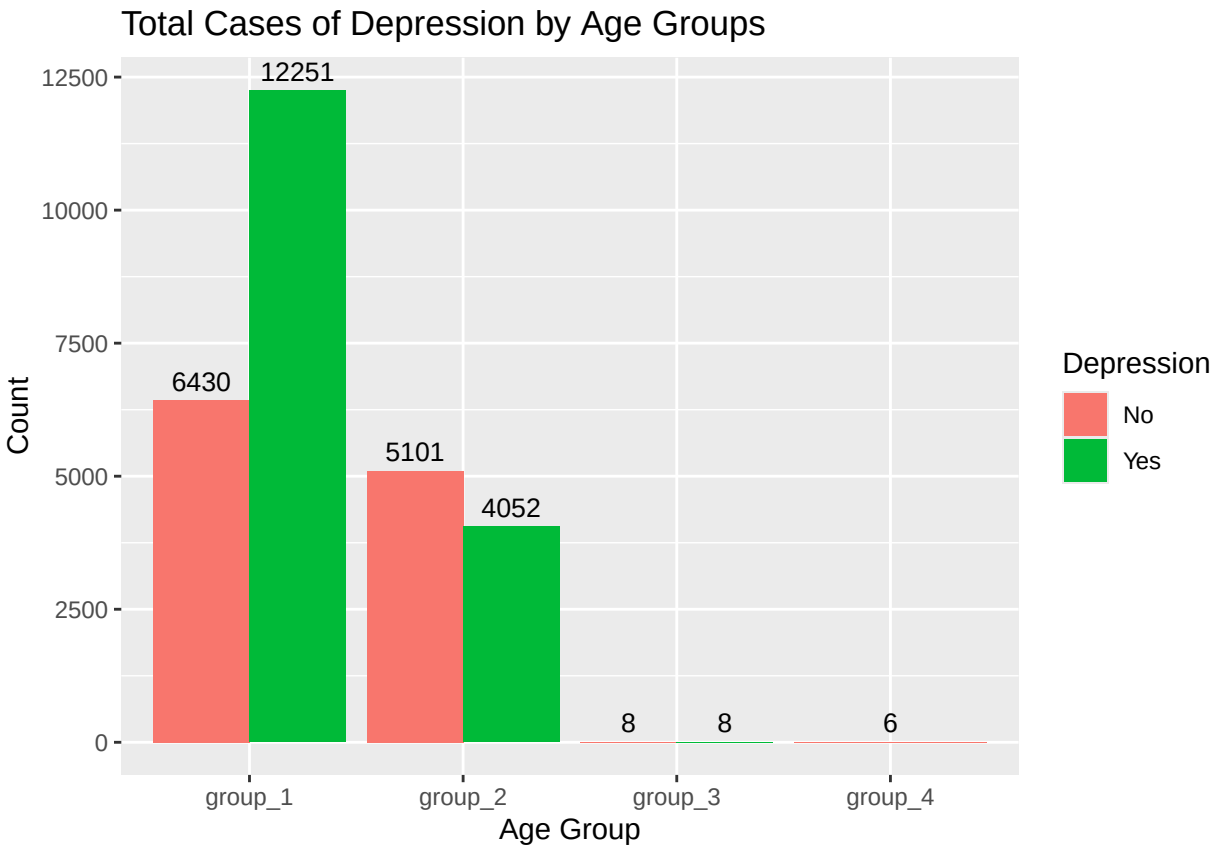
```
# Plot the bar chart
ggplot(grouped_data, aes(x = Sleep_Duration, y = Count, fill = as.factor(Depression))) +
  geom_bar(stat = 'identity', position = position_dodge()) +
  geom_text(aes(label=Count), position = position_dodge(width = 0.9), vjust = -0.5, size = 3.5) +
  labs(title = 'Total Cases of Depression by Sleep Duration',
       x = 'Sleep Duration',
       y = 'Count',
       fill = 'Depression') +
  scale_fill_manual(values = c('No' = '#F8766D', 'Yes' = '#00BA38'))
```




```
#####################
# Group the data by Age Group and Depression
grouped_data <- depression_dataset_cleaned %>%
  group_by(Age_Group, Depression) %>%
  summarise(Count = n()) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Age_Group'. You can override using the
## `.groups` argument.
```

```
# Plot the bar chart
ggplot(grouped_data, aes(x = Age_Group, y = Count, fill = as.factor(Depression))) +
  geom_bar(stat = 'identity', position = position_dodge()) +
  geom_text(aes(label=Count), position = position_dodge(width = 0.9), vjust = -0.5, size = 3.5) +
  labs(title = 'Total Cases of Depression by Age Groups',
       x = 'Age Group',
       y = 'Count',
       fill = 'Depression') +
  scale_fill_manual(values = c('No' = '#F8766D', 'Yes' = '#00BA38'))
```



The above results indicated that:

Diet and Depression

- Students with unhealthy dietary habits are more likely to be diagnosed with depression. This aligns with research suggesting that poor nutrition can negatively impact mental health.

Sleep Duration and Depression

- Students who sleep less than five hours per night have a higher likelihood of experiencing depression. Sleep deprivation has been widely linked to increased stress, anxiety, and mood disorders.

Age and Depression Risk

- Most students in the dataset are under the age of 38, and within this age group, depression is more prevalent. This may indicate that younger students, who often face significant academic and career pressures, are at greater risk.

Gender and Depression

- Since the dataset contains a higher proportion of male students, the findings suggest that males are more likely to be diagnosed with depression. This supports earlier observations regarding gender differences in emotional expression and coping mechanisms.

Stress & Mental Well-being Factors

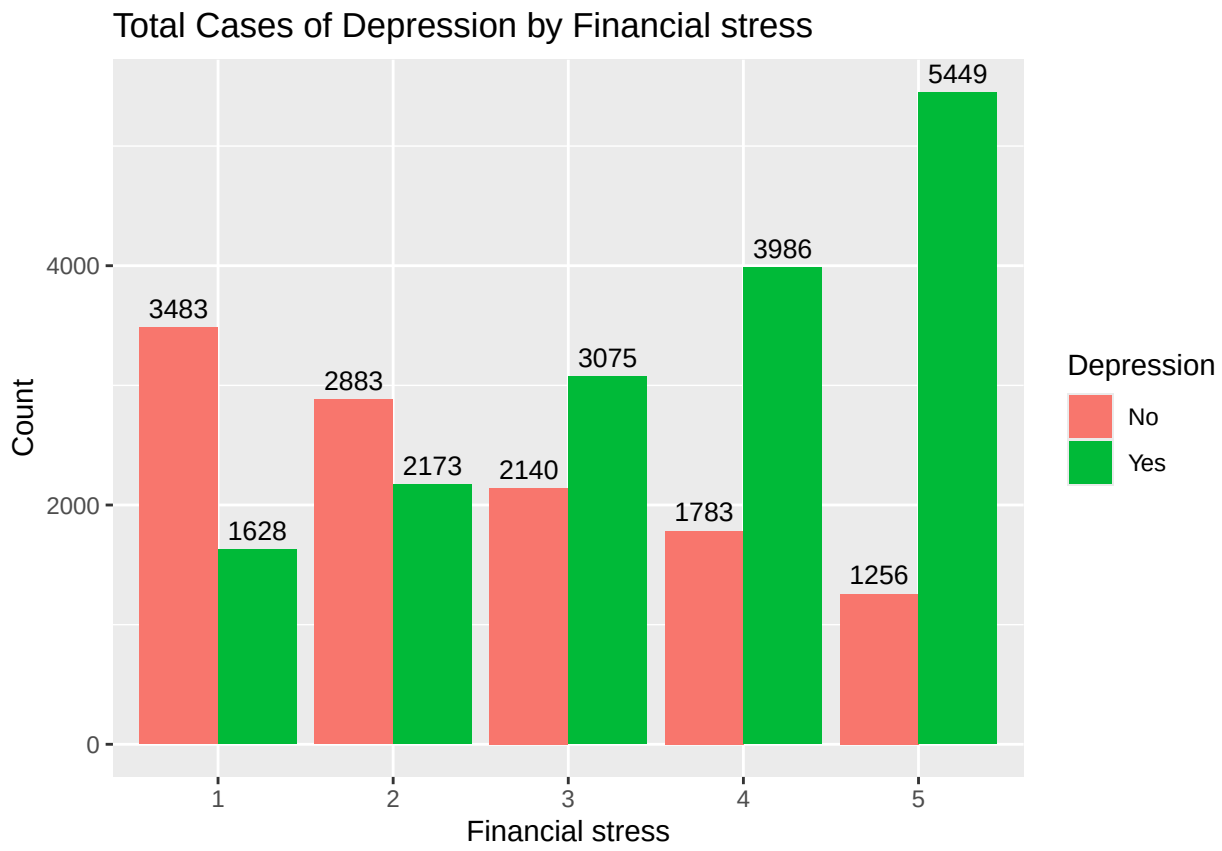
In this section the analysis will be based on 4 different groups to understand how the factor influence depression in students. The analysis has been done using charts.

Stress Factors

```
# Group the data by Gender and Depression
grouped_data <- depression_dataset_cleaned %>%
  group_by(Financial_stress, Depression) %>%
  summarise(Count = n()) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Financial_stress'. You can override using
## the `.groups` argument.
```

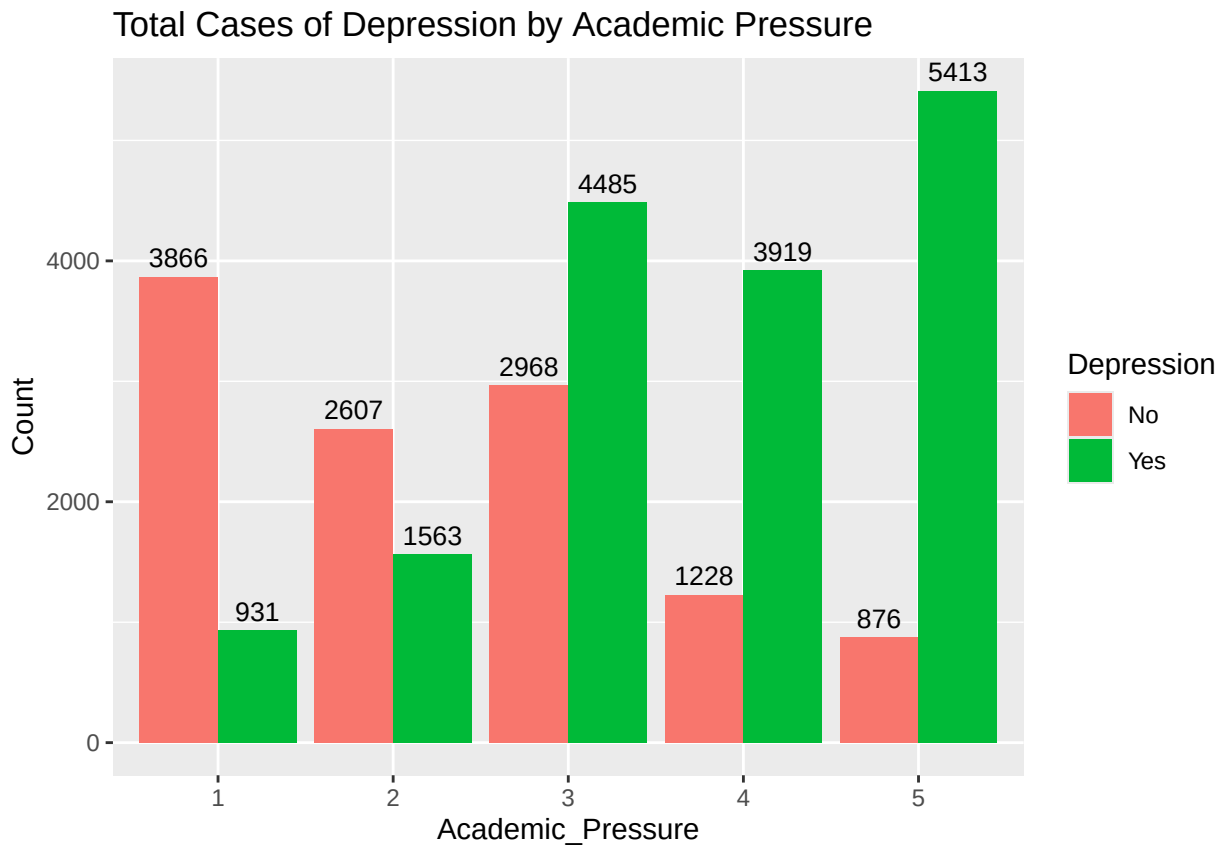
```
# Plot the bar chart
ggplot(grouped_data, aes(x = Financial_stress, y = Count, fill = as.factor(Depression))) +
  geom_bar(stat = 'identity', position = position_dodge()) +
  geom_text(aes(label=Count),position = position_dodge(width = 0.9), vjust = -0.5, size = 3.5) +
  labs(title = 'Total Cases of Depression by Financial stress',
       x = 'Financial stress',
       y = 'Count',
       fill = 'Depression') +
  scale_fill_manual(values = c('No' = '#F8766D', 'Yes' = '#00BA38'))
```



```
#####################
# Group the data by Academic Pres and Depression
grouped_data <- depression_dataset_cleaned %>%
  group_by(Academic_Pressure, Depression) %>%
  summarise(Count = n()) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Academic_Pressure'. You can override using
## the `.groups` argument.
```

```
# Plot the bar chart
ggplot(grouped_data, aes(x = Academic_Pressure, y = Count, fill = as.factor(Depression))) +
  geom_bar(stat = 'identity', position = position_dodge()) +
  geom_text(aes(label=Count), position = position_dodge(width = 0.9), vjust = -0.5, size = 3.5) +
  labs(title = 'Total Cases of Depression by Academic Pressure',
       x = 'Academic_Pressure',
       y = 'Count',
       fill = 'Depression') +
  scale_fill_manual(values = c('No' = '#F8766D', 'Yes' = '#00BA38'))
```

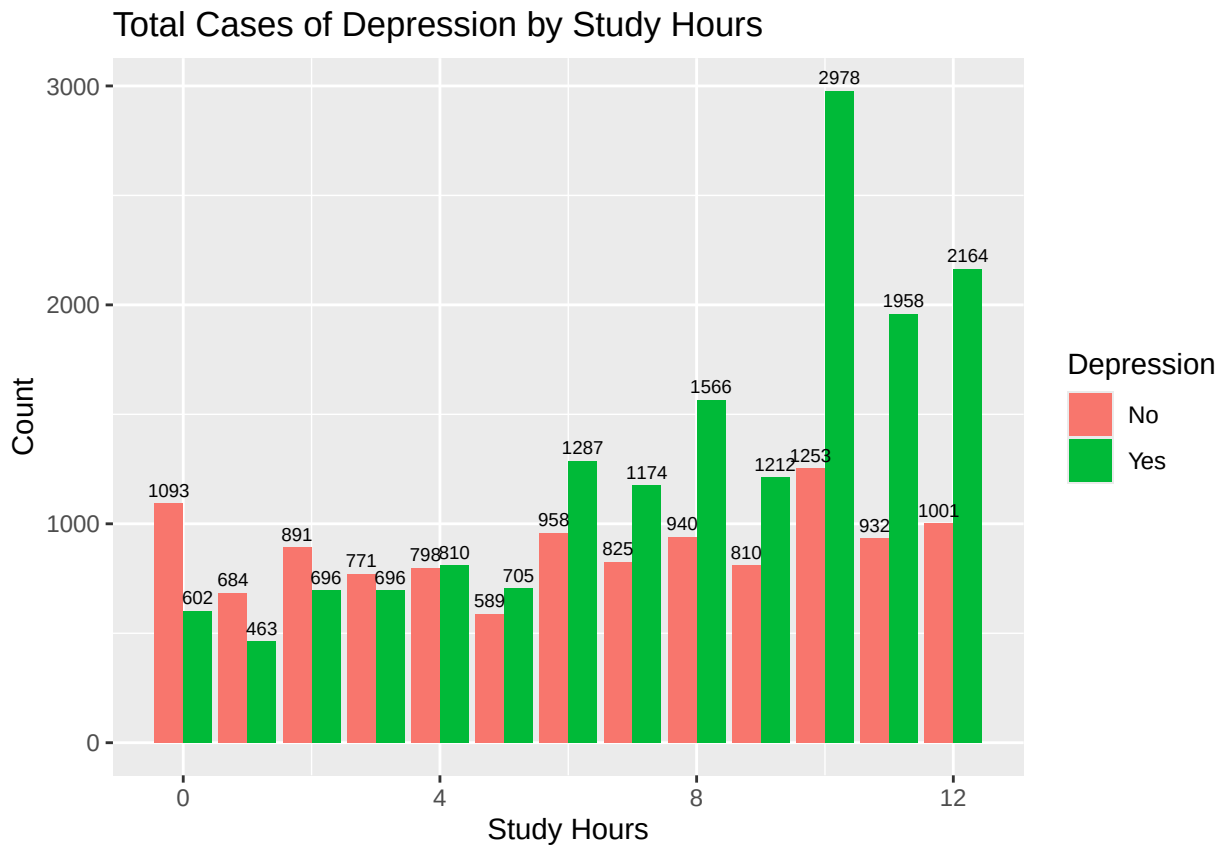



```
#####################
# Group the data by Study Hours  and Depression
grouped_data <- depression_dataset_cleaned %>%
  group_by(Study_Hours, Depression) %>%
```

```
  summarise(Count = n()) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Study_Hours'. You can override using the
## `.groups` argument.
```

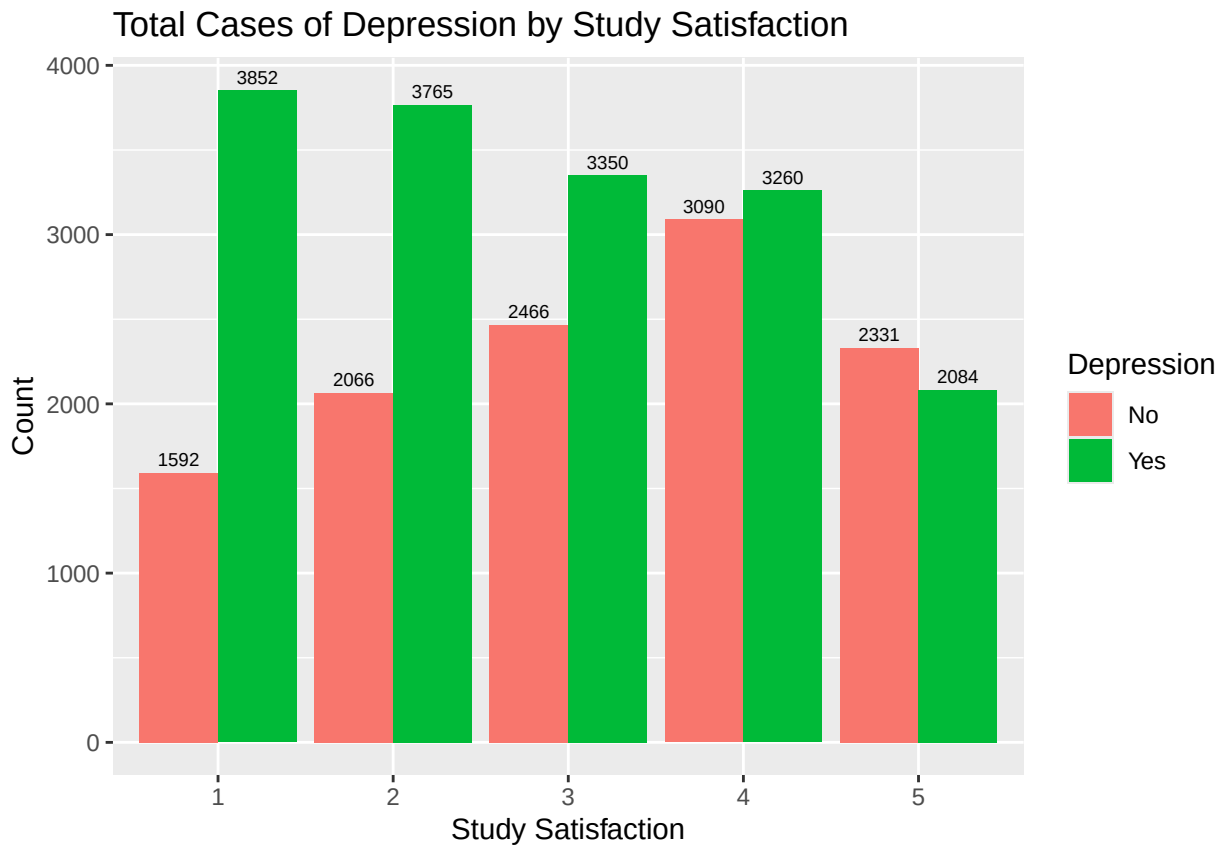
```
# Plot the bar chart
ggplot(grouped_data, aes(x = Study_Hours, y = Count, fill = as.factor(Depression))) +
  geom_bar(stat = 'identity', position = position_dodge()) +
  geom_text(aes(label=Count), position = position_dodge(width = 0.9), vjust = -0.5, size = 2.5) +
  labs(title = 'Total Cases of Depression by Study Hours',
       x = 'Study Hours',
       y = 'Count',
       fill = 'Depression') +
  scale_fill_manual(values = c('No' = '#F8766D', 'Yes' = '#00BA38'))
```




```
######################
# Group the data by StudySatisfaction  and Depression
grouped_data <- depression_dataset_cleaned %>%
  group_by(Study_Satisfaction, Depression) %>%
  summarise(Count = n()) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Study_Satisfaction'. You can override
## using the `.groups` argument.
```

```r
# Plot the bar chart
ggplot(grouped_data, aes(x = Study_Satisfaction, y = Count, fill = as.factor(Depression))) +
  geom_bar(stat = 'identity', position = position_dodge()) +
  geom_text(aes(label=Count), position = position_dodge(width = 0.9), vjust = -0.5, size = 2.5) +
  labs(title = 'Total Cases of Depression by Study Satisfaction',
       x = 'Study Satisfaction',
       y = 'Count',
       fill = 'Depression') +
  scale_fill_manual(values = c('No' = '#F8766D', 'Yes' = '#00BA38'))
```



The above results indicated that:

Academic Pressure and Depression

- Students who experience higher levels of academic pressure are more likely to be diagnosed with depression. This suggests that excessive academic demands can negatively impact mental health.

Financial Stress and Depression

- Students who face financial stress have a higher likelihood of depression. Financial instability can contribute to anxiety, uncertainty, and increased psychological distress.

Study Hours and Depression

- Students who dedicate longer hours to studying are more prone to depression. However, an interesting trend emerges after 10 hours of study per day, depression diagnoses decrease. This could be due to a smaller number of students studying for extended hours, or it may indicate that extremely dedicated students have better coping mechanisms.

Study Satisfaction and Depression

- Students with low study satisfaction are more likely to be diagnosed with depression. A negative academic experience, including dissatisfaction with coursework or career prospects, may contribute to mental health struggles.

Mental Health-Factors

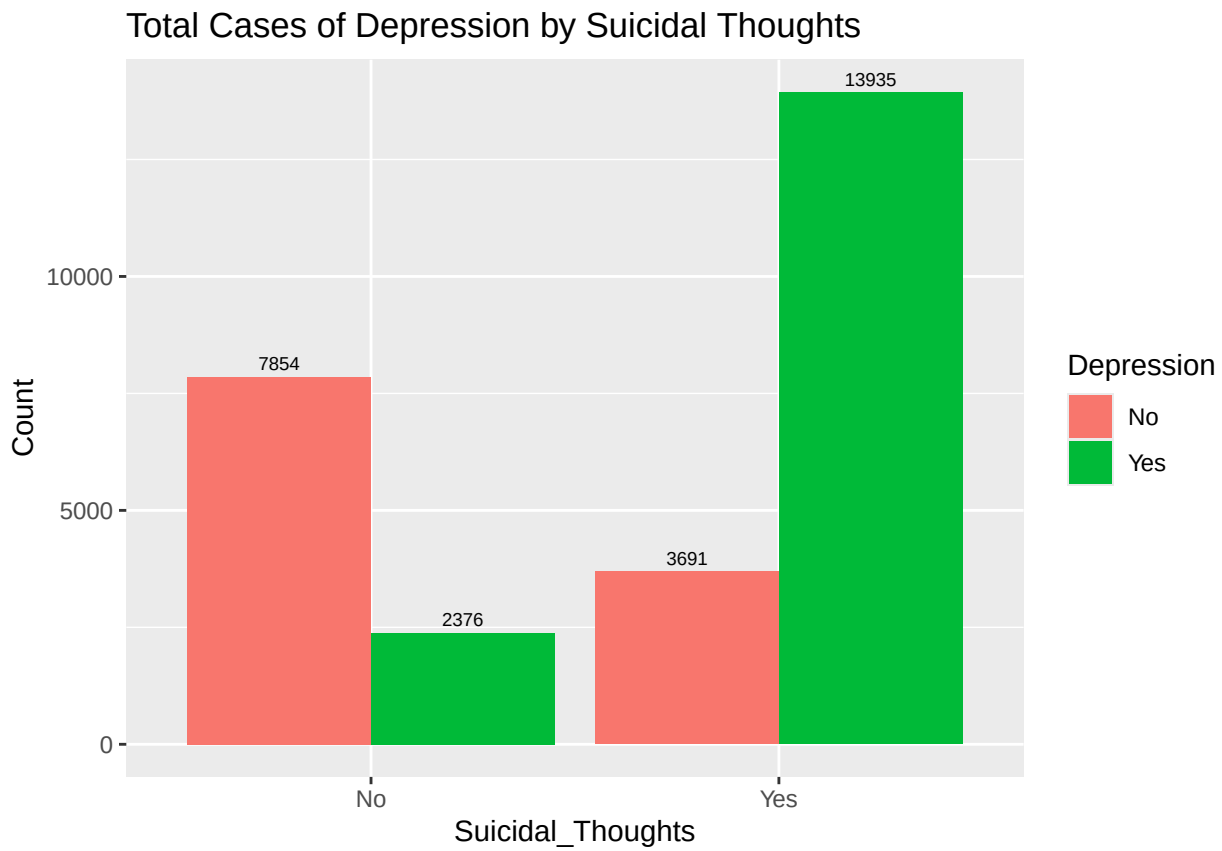
In this section the analysis will be based in 2 different groups to understand how the factor influence depression in students. The analysis has been done using charts.

Mental Health Factors

```
# Group the data by Suicidal and Depression
grouped_data <- depression_dataset_cleaned %>%
  group_by(Suicidal_Thoughts, Depression) %>%
  summarise(Count = n()) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Suicidal_Thoughts'. You can override using
## the `.groups` argument.
```

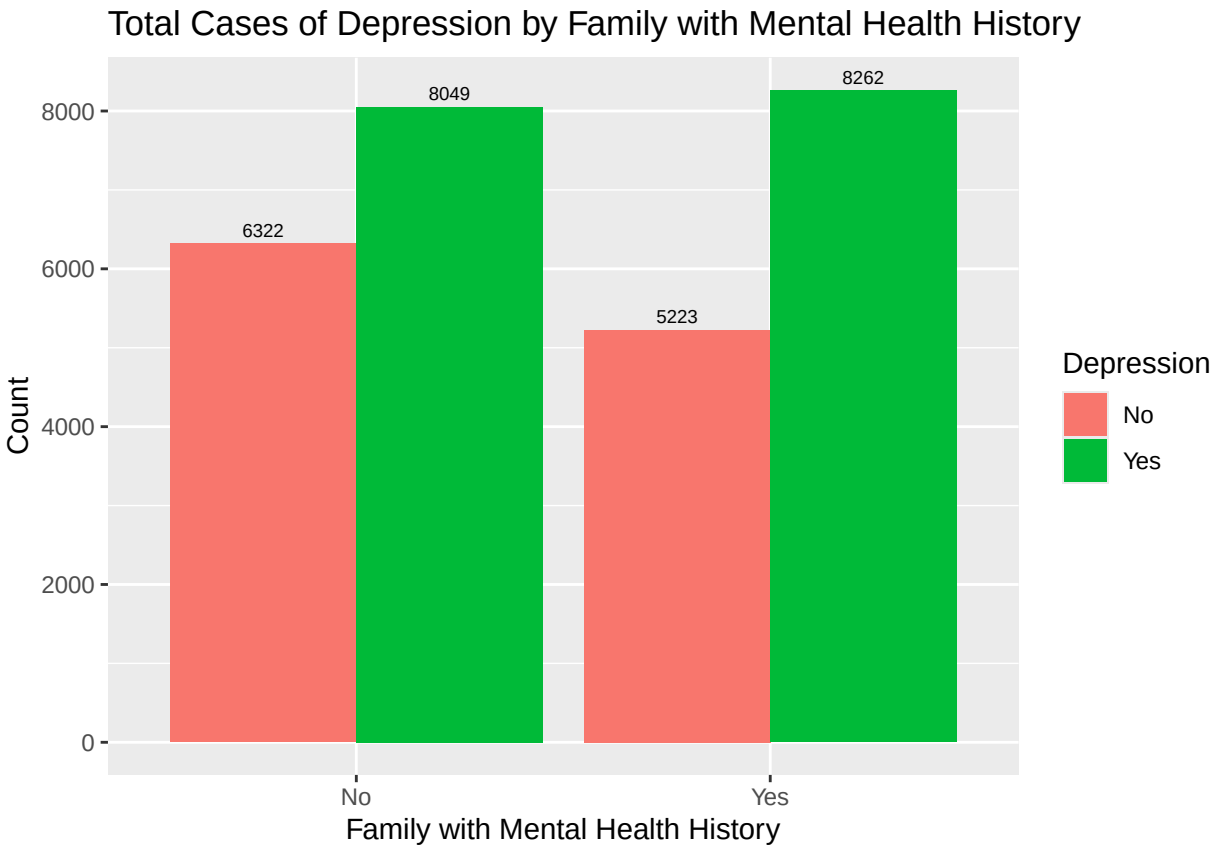
```
# Plot the bar chart
ggplot(grouped_data, aes(x = Suicidal_Thoughts, y = Count, fill = as.factor(Depression))) +
  geom_bar(stat = 'identity', position = position_dodge()) +
  geom_text(aes(label=Count), position = position_dodge(width = 0.9), vjust = -0.5, size = 2.5) +
  labs(title = 'Total Cases of Depression by Suicidal Thoughts',
       x = 'Suicidal_Thoughts',
       y = 'Count',
       fill = 'Depression') +
  scale_fill_manual(values = c('No' = '#F8766D', 'Yes' = '#00BA38'))
```




```
#####################
# Group the data by Family Mental and Depression
grouped_data <- depression_dataset_cleaned %>%
  group_by(Fam_Mental_Hist, Depression) %>%
  summarise(Count = n()) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Fam_Mental_Hist'. You can override using
## the `.groups` argument.
```

```
# Plot the bar chart
ggplot(grouped_data, aes(x = Fam_Mental_Hist, y = Count, fill = as.factor(Depression))) +
  geom_bar(stat = 'identity', position = position_dodge()) +
  geom_text(aes(label=Count), position = position_dodge(width = 0.9), vjust = -0.5, size = 2.5) +
  labs(title = 'Total Cases of Depression by Family with Mental Health History',
       x = 'Family with Mental Health History',
       y = 'Count',
       fill = 'Depression') +
  scale_fill_manual(values = c('No' = '#F8766D', 'Yes' = '#00BA38'))
```



The above results indicated that:

Suicidal Thoughts and Depression

- Students who report having suicidal thoughts are significantly more likely to be diagnosed with depression. This finding reinforces the strong link between depression and suicidal ideation, emphasizing the need for early mental health interventions.

Family History of Mental Health Disorders and Depression

- Students with a family history of mental illness are slightly more likely to be diagnosed with depression. This suggests a potential genetic or environmental influence, where individuals with close relatives experiencing mental health disorders may be at a higher risk themselves.

3. Statistical Analysis

In this section, the dataset has been structured and prepared for model selection. Key data elements were previously analyzed and are summarized below as a reference:

- **Total students:** 27,856
- **Total positive depression cases:** 16,311

- **Total negative depression cases:** 11,545
- **Total male students:** 15,523
  - **Positive cases (male):** 9,102
  - **Negative cases (male):** 6,421
- **Total female students:** 12,333
  - **Positive cases (female):** 7,209
  - **Negative cases (female):** 5,124

```
#total of Students Cases
total_of_students <- nrow(depression_dataset_cleaned)
total_of_positive_cases <- Depression_Distribution[2] #No
total_of_negative_cases <- Depression_Distribution[1] #Yes
## Total of Males Cases
total_of_Male_Students          <- male_female_Distribution['Male']
total_of_Positive_Male_Students <- male_female_Distribution_Positive['Male']
total_of_negative_Male_student  <- male_female_Distribution_negative['Male']
# Total of Females Cases
total_of_Female_Students           <- male_female_Distribution['Female']
total_of_Positive_Female_Students  <- male_female_Distribution_Positive['Female']
total_of_negative_Female_student   <- male_female_Distribution_negative['Female']
```

```
# Cases Summary
cases_data <- data.frame(
  Category = c("Total of Cases","Total Positive Cases", "Total Negative Cases",
               "Total Male Students", "Positive Male Students", "Negative Male Students",
               "Total Female Students", "Positive Female Students", "Negative Female Students"),
  Cases = c(total_of_students,total_of_positive_cases, total_of_negative_cases,
            total_of_Male_Students, total_of_Positive_Male_Students, total_of_negative_Male_student,
            total_of_Female_Students, total_of_Positive_Female_Students, total_of_negative_Female_stude
)

# Print
#print(cases_data)
cases_data
```

```
##                   Category Cases
## 1           Total of Cases 27856
## 2     Total Positive Cases 16311
## 3     Total Negative Cases 11545
## 4      Total Male Students 15523
## 5   Positive Male Students  9102
## 6   Negative Male Students  6421
## 7    Total Female Students 12333
## 8 Positive Female Students  7209
## 9 Negative Female Students  5124
```

From the summary listed above it is possible to continue and divide the dataset appropriately to generate random sample that is well distributed among male, female, positive and negative cases from the study. By

doing this we will are implementing the stratified random sampling method. To continue and do this, the next step is to define the proportions of each stratum.

Stratum Proportion:

```
# Get the Stratum Proportion per Gender and Positive or Negative Cases
#Negative
male_negative   <- total_of_negative_Male_student/total_of_students
female_negative <- total_of_negative_Female_student/total_of_students

#Positive
male_positive   <- total_of_Positive_Male_Students/total_of_students
female_positive <- total_of_Positive_Female_Students/total_of_students
```

```
# Cases Summary
stratum_distribution <- data.frame(
  Category = c("Positive Male", "Negative Male", "Positive Female","Negative Female"),
  Cases    = c( male_positive,   male_negative,   female_positive,  female_negative)
)

stratum_distribution
```

```
##          Category     Cases
## 1   Positive Male 0.3267519
## 2   Negative Male 0.2305069
## 3 Positive Female 0.2587952
## 4 Negative Female 0.1839460
```

Stratified Random Sampling:

Stratified simple sampling is a method of sampling that involves dividing a population into smaller groups, known as strata, that share similar characteristics. Within each stratum, a simple random sample is taken. This ensures that each subgroup is adequately represented in the overall sample, leading to more accurate and reliable results. It is particularly useful when there are distinct subgroups within a population that may have different behaviors or attributes. The main benefit of stratified sampling is that it increases the precision of the overall estimates by reducing sampling variability. This method also allows for more detailed subgroup analysis, providing insights into the characteristics and trends within each stratum.

```
# Create a stratified sample based on Gender and Depression
set.seed(2025)  # For reproducibility
strata_columns <- c("Gender", "Depression")
stratified_sample <- depression_dataset_cleaned %>%
  group_by_at(strata_columns) %>%
  sample_frac(size = 0.50)

# Check the stratified sample
#print(table(stratified_sample$Gender, stratified_sample$Depression))
print(nrow(stratified_sample))
```

```
## [1] 13927
```

The stratification is applied to the columns Depression and Gender. This ensures that the sample is filtered through these columns, resulting in values that are a combination of these attributes, as previously shown.

The sample size is half of the original dataset size. The decision to use 50% of the original size aims to have a sufficiently large dataset for further division between the training and testing sets for analysis. This way, we can still have a substantial amount of data to analyze, but it remains small enough to compute efficiently. Also, the code uses the "set.seed" command to set the create reproducibility allowing same results in different devices. The next step is to proceed to generate the two used dataset for training and for testing.

Split the dataset into training and testing:

Splitting data is the action of dividing the working data in two parts: * Training set * Test set Spliting data is a common practice in machine learning to evaluate the performance of a model. The training set is used to train the model, allowing it to learn patterns and relationships within the data. The test set, which is kept separate and not seen by the model during training, is used to assess the model's performance. This helps ensure that the model can generalize well to new, unseen data, and not just memorize the training data. The main benefit of this approach is that it provides a reliable estimate of the model's accuracy and robustness, helping to prevent overfitting and underfitting. The standard data split is 75% for training and 25% for testing.

```
# Split the stratified sample into training and testing sets (75% training, 25% testing)
trainIndex <- createDataPartition(stratified_sample$Depression, p = 0.75, list = FALSE)
train_data <- stratified_sample[trainIndex,]
test_data <- stratified_sample[-trainIndex,]

original_data_set_percentage75 <- nrow(stratified_sample)*0.75
original_data_set_percentage25 <- nrow(stratified_sample)*0.25

  # Check the splits
# 75% of the stratified sample
print (original_data_set_percentage75)
```

```
## [1] 10445.25
```

```
print(nrow(train_data))
```

```
## [1] 10446
```

```
# 25% of the stratified sample
print (original_data_set_percentage25)
```

```
## [1] 3481.75
```

```
print(nrow(test_data))
```

```
## [1] 3481
```

Logistic Regression Model

When using a logistic model, it is essential to consider a few key elements or conditions. The logistic model is appropriate when the response variable is binary, meaning it takes on values between 0 and 1. This is precisely what is needed for the analysis, as the response variable is 'Depression,' recorded as '0' and '1,' 'Yes'

and 'No,' or 'True' and 'False.' Therefore, treating the response variable as binary makes the logistic model a suitable choice for this analysis. Another important factor to consider is how we evaluate the response variable. Although the values between 0 and 1 are continuous, for the sake of this analysis all values greater than 0.5 were assigned as '1' and all values less than or equal to 0.5 as '0.' This binary classification allows to use the logistic model effectively to analyze the data.

One final element to consider is the appropriate classification of the variables (columns). It is crucial to treat these variables as intended. Specifically, columns that hold numeric values but represent categorical options should be converted to factors. Even though they contain numeric values, these numbers are merely grouped options, and thus, they need to be treated as factors for the model to accurately interpret and categorize them.

Full Logistice Model

1. Verify Columns Classification:

```
str(train_data)
```

```
## gropd_df [10,446 x 12] (S3: grouped_df/tbl_df/tbl/data.frame)
##  $ Gender            : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Academic_Pressure : Factor w/ 5 levels "1","2","3","4",..: 1 1 3 2 2 2 1 3 2 1 ...
##  $ CGPA              : num [1:10446] 3.75 2.93 4.02 4.02 3.69 ...
##  $ Study_Satisfaction: Factor w/ 5 levels "1","2","3","4",..: 4 2 5 4 4 1 1 4 5 3 ...
##  $ Sleep_Duration    : Factor w/ 4 levels "5-6 hours","7-8 hours",..: 1 3 1 1 4 1 3 4 1 2 ...
##  $ Dietary_Habits    : Factor w/ 3 levels "Healthy","Moderate",..: 2 1 1 2 1 1 2 1 1 2 ...
##  $ Suicidal_Thoughts : Factor w/ 2 levels "No","Yes": 2 2 1 2 1 1 2 1 1 1 ...
##  $ Study_Hours       : int [1:10446] 3 7 0 12 1 12 5 12 0 11 ...
##  $ Financial_stress  : Factor w/ 5 levels "1","2","3","4",..: 1 2 3 3 5 1 3 2 1 4 ...
##  $ Fam_Mental_Hist   : Factor w/ 2 levels "No","Yes": 2 1 2 2 1 1 1 2 2 2 ...
##  $ Depression        : chr [1:10446] "No" "No" "No" "No" ...
##  $ Age_Group         : Factor w/ 4 levels "group_1","group_2",..: 2 1 1 1 1 1 1 2 1 1 ...
##  - attr(*, "groups")= tibble [4 x 3] (S3: tbl_df/tbl/data.frame)
##   ..$ Gender    : Factor w/ 2 levels "Female","Male": 1 1 2 2
##   ..$ Depression: chr [1:4] "No" "Yes" "No" "Yes"
##   ..$ .rows     : list<int> [1:4]
##   .. ..$ : int [1:1911] 1 2 3 4 5 6 7 8 9 10 ...
##   .. ..$ : int [1:2680] 1912 1913 1914 1915 1916 1917 1918 1919 1920 1921 ...
##   .. ..$ : int [1:2418] 4592 4593 4594 4595 4596 4597 4598 4599 4600 4601 ...
##   .. ..$ : int [1:3437] 7010 7011 7012 7013 7014 7015 7016 7017 7018 7019 ...
##   .. ..@ ptype: int(0)
##   ..- attr(*, ".drop")= logi TRUE
```

2. Fit the model:

```
# Fit the logistic regression model

# Convert categorical variables to factors
train_data$Gender <- as.factor(train_data$Gender)
train_data$Academic_Pressure <- as.factor(train_data$Academic_Pressure)
train_data$Study_Satisfaction <- as.factor(train_data$Study_Satisfaction)
```

```r
train_data$Sleep_Duration <- as.factor(train_data$Sleep_Duration)
train_data$Fam_Mental_Hist <- as.factor(train_data$Fam_Mental_Hist)
train_data$Financial_stress <- as.factor(train_data$Financial_stress)
train_data$Dietary_Habits <- as.factor(train_data$Dietary_Habits)
train_data$Suicidal_Thoughts <- as.factor(train_data$Suicidal_Thoughts)
train_data$Age_Group <- as.factor(train_data$Age_Group)
train_data$Depression <- as.factor(train_data$Depression)

logistic_Model <- glm(Depression ~.,data = train_data, family = binomial)

# Summarize the model
summary(logistic_Model)
```

```
##
## Call:
## glm(formula = Depression ~ ., family = binomial, data = train_data)
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -5.077482   0.233021 -21.790  < 2e-16 ***
## GenderMale                      -0.026681   0.060324  -0.442 0.658280
## Academic_Pressure2               1.068665   0.102801  10.395  < 2e-16 ***
## Academic_Pressure3               1.897728   0.092522  20.511  < 2e-16 ***
## Academic_Pressure4               2.712256   0.104268  26.012  < 2e-16 ***
## Academic_Pressure5               3.478564   0.110740  31.412  < 2e-16 ***
## CGPA                             0.063501   0.040744   1.559 0.119108
## Study_Satisfaction2             -0.459661   0.098593  -4.662 3.13e-06 ***
## Study_Satisfaction3             -0.497498   0.097122  -5.122 3.02e-07 ***
## Study_Satisfaction4             -0.805318   0.095169  -8.462  < 2e-16 ***
## Study_Satisfaction5             -1.072992   0.102911 -10.426  < 2e-16 ***
## Sleep_Duration7-8 hours          0.063094   0.085391   0.739 0.459974
## Sleep_DurationLess than 5 hours  0.292438   0.084317   3.468 0.000524 ***
## Sleep_DurationMore than 8 hours -0.355809   0.089151  -3.991 6.58e-05 ***
## Dietary_HabitsModerate           0.473761   0.073275   6.466 1.01e-10 ***
## Dietary_HabitsUnhealthy          1.180351   0.076332  15.463  < 2e-16 ***
## Suicidal_ThoughtsYes             2.496880   0.064177  38.906  < 2e-16 ***
## Study_Hours                      0.122943   0.008179  15.031  < 2e-16 ***
## Financial_stress2                0.451806   0.095585   4.727 2.28e-06 ***
## Financial_stress3                1.026654   0.093945  10.928  < 2e-16 ***
## Financial_stress4                1.486964   0.095078  15.639  < 2e-16 ***
## Financial_stress5                2.242554   0.098192  22.839  < 2e-16 ***
## Fam_Mental_HistYes               0.314024   0.059913   5.241 1.59e-07 ***
## Age_Groupgroup_2                -0.962288   0.063613 -15.127  < 2e-16 ***
## Age_Groupgroup_3                -0.556789   1.570074  -0.355 0.722870
## Age_Groupgroup_4               -11.469077 139.130423  -0.082 0.934302
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 14173.7  on 10445  degrees of freedom
## Residual deviance:  7245.8  on 10420  degrees of freedom
## AIC: 7297.8
```

```
##
## Number of Fisher Scoring iterations: 10
```

3. Verify multicollinearity:

Multicollinearity occurs when predictors in a model are highly correlated with each other, which can lead to instability in the coefficient estimates and make it difficult to determine the individual effect of each predictor. That is why is important to keep record of it. The list below shows a brief interpretation of the relevance on multicollinearity.

• VIF = 1: No correlation between the predictor and other predictors. • 1 < VIF < 5: Moderate correlation, typically not a cause for concern. • VIF 5: High correlation, indicating significant multicollinearity. • VIF > 10: Very high correlation, often considered a serious problem that requires correction.

Multicollinearity

```
library(car)
vif(logistic_Model)
```

```
##                       GVIF Df GVIF^(1/(2*Df))
## Gender            1.011975  1        1.005970
## Academic_Pressure 1.125542  4        1.014893
## CGPA              1.011823  1        1.005894
## Study_Satisfaction 1.048684  4        1.005960
## Sleep_Duration    1.022292  3        1.003681
## Dietary_Habits    1.041278  2        1.010163
## Suicidal_Thoughts 1.082380  1        1.040375
## Study_Hours       1.024995  1        1.012421
## Financial_stress  1.069461  4        1.008430
## Fam_Mental_Hist   1.006080  1        1.003035
## Age_Group         1.029187  3        1.004806
```

From the modeling fitting we notice that there are two predictors that are insignificant for the model. Therefore, we could potentially have two logistic models. • Original Model: that includes all the predictors • Reduce Model: that holds only the predictors that are significant The correspondent process for the original model had been done. The reduced model needs to also be loaded and verify its multicollinearity.

Reduced Logistic Model

1. Fit the model without the CGPA Predictor / Gender

```
# Fit the logistic regression model
reduce_Logistic_Model <- glm(Depression ~ Academic_Pressure +  Study_Satisfaction +
                              Sleep_Duration + Dietary_Habits + Suicidal_Thoughts +
                             Study_Hours + Financial_stress + Fam_Mental_Hist +
                              Age_Group, data = train_data, family = binomial)

# Summarize the model
summary(reduce_Logistic_Model)
```

```
##
## Call:
## glm(formula = Depression ~ Academic_Pressure + Study_Satisfaction +
##     Sleep_Duration + Dietary_Habits + Suicidal_Thoughts + Study_Hours +
##     Financial_stress + Fam_Mental_Hist + Age_Group, family = binomial,
##     data = train_data)
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    -4.842664   0.166377 -29.107  < 2e-16 ***
## Academic_Pressure2              1.067847   0.102737  10.394  < 2e-16 ***
## Academic_Pressure3              1.895354   0.092459  20.499  < 2e-16 ***
## Academic_Pressure4              2.707400   0.104132  26.000  < 2e-16 ***
## Academic_Pressure5              3.473145   0.110597  31.404  < 2e-16 ***
## Study_Satisfaction2            -0.460333   0.098574  -4.670 3.01e-06 ***
## Study_Satisfaction3            -0.502662   0.097045  -5.180 2.22e-07 ***
## Study_Satisfaction4            -0.807683   0.095137  -8.490  < 2e-16 ***
## Study_Satisfaction5            -1.081502   0.102684 -10.532  < 2e-16 ***
## Sleep_Duration7-8 hours         0.064075   0.085370   0.751 0.452924
## Sleep_DurationLess than 5 hours 0.292580   0.084276   3.472 0.000517 ***
## Sleep_DurationMore than 8 hours -0.359162   0.089101  -4.031 5.55e-05 ***
## Dietary_HabitsModerate          0.473115   0.073258   6.458 1.06e-10 ***
## Dietary_HabitsUnhealthy         1.178791   0.076143  15.481  < 2e-16 ***
## Suicidal_ThoughtsYes            2.497810   0.064161  38.930  < 2e-16 ***
## Study_Hours                     0.122726   0.008172  15.018  < 2e-16 ***
## Financial_stress2               0.450775   0.095564   4.717 2.39e-06 ***
## Financial_stress3               1.028015   0.093934  10.944  < 2e-16 ***
## Financial_stress4               1.490379   0.095031  15.683  < 2e-16 ***
## Financial_stress5               2.242510   0.098150  22.848  < 2e-16 ***
## Fam_Mental_HistYes              0.314267   0.059903   5.246 1.55e-07 ***
## Age_Groupgroup_2               -0.962550   0.063602 -15.134  < 2e-16 ***
## Age_Groupgroup_3               -0.615111   1.576099  -0.390 0.696333
## Age_Groupgroup_4              -11.434775 139.146444  -0.082 0.934505
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 14173.7  on 10445  degrees of freedom
## Residual deviance:  7248.4  on 10422  degrees of freedom
## AIC: 7296.4
##
## Number of Fisher Scoring iterations: 10
```

2. Verify multicollinearity

```
library(car)
vif(reduce_Logistic_Model)
```

```
##                       GVIF Df GVIF^(1/(2*Df))
## Academic_Pressure  1.121936  4        1.014486
## Study_Satisfaction 1.042953  4        1.005271
```

```
## Sleep_Duration     1.020801  3        1.003437
## Dietary_Habits     1.034161  2        1.008433
## Suicidal_Thoughts  1.082465  1        1.040416
## Study_Hours        1.023700  1        1.011781
## Financial_stress   1.067063  4        1.008147
## Fam_Mental_Hist    1.006141  1        1.003066
## Age_Group          1.028476  3        1.004691
```

From figure 18, we confirm that the predictors are not high correlated.

Logistic Regression Model Testing

There are two available options: the original model, which includes the insignificant predictors Gender and CGPA, and a reduced model where these insignificant predictors had been removed. For the sake of the analysis, the next step involves applying both models and comparing their respective behaviors to highlight any interesting results.

Full Logistic model Model Testing

Using the created full logistic model, the test dataset previously created its going to be used for this test.

Applying Logistic Model to Test Dataset

```
predictions <- predict(logistic_Model, newdata=test_data, type='response')
pred_class <- ifelse(predictions >= 0.5, 1, 0)
```

Create a confusion matrix

```
logistic_conf_matrix <- table(Predicted = pred_class, Actual = test_data$Depression)
print(logistic_conf_matrix)
```

```
##          Actual
## Predicted   No  Yes
##         0 1143  243
##         1  300 1795
```

```
# Extract values from Confusion Matrix Logistic
TN_logistic <- logistic_conf_matrix[1, 1]  # True Negatives
FP_logistic <- logistic_conf_matrix[1, 2]  # False Positives
FN_logistic <- logistic_conf_matrix[2, 1]  # False Negatives
TP_logistic <- logistic_conf_matrix[2, 2]  # True Positives

# Compute Performance Metrics
accuracy_logistic <- round((TP_logistic + TN_logistic) / (TP_logistic + TN_logistic + FP_logistic + FN_l
precision_logistic <- round(TP_logistic / (TP_logistic + FP_logistic), 4)
recall_logistic <- round(TP_logistic / (TP_logistic + FN_logistic), 4)
f1_score_logistic <- round(2 * ((precision_logistic * recall_logistic) / (precision_logistic + recall_lo

# Print Metrics
cat("Logistic Model Performance Metrics:\n")
```

```
## Logistic Model Performance Metrics:
```

```
cat("Model Accuracy:", accuracy_logistic, "\n")
```

```
## Model Accuracy: 0.844
```

```
cat("Precision:", precision_logistic, "\n")
```

```
## Precision: 0.8808
```

```
cat("Recall:", recall_logistic, "\n")
```

```
## Recall: 0.8568
```

```
cat("F1 Score:", f1_score_logistic,  "\n")
```

```
## F1 Score: 0.8686
```

Percentage of correct predictions

```
logistic_correct_predictions <- sum(diag(logistic_conf_matrix)) / sum(logistic_conf_matrix)
print(paste("Overall Correct Predictions Percentages: ", logistic_correct_predictions))
```

```
## [1] "Overall Correct Predictions Percentages:  0.844010341855789"
```

The above results indicates thst out of the 3481 rows in the test dataset the model successfully predicted 2931 depression cases. Therefore 550 depression cases are false positive and false negative. The accuracy of this model is about 84.4%. In other words, 84.4% of the times, it predicts correctly.

Reduce Logistic Model Testing

Following the same steps from the previous testing but in this case using the Reduce Logistic Model.

Applying Logistic Model to Test Dataset

```
predictions_reduce <- predict(reduce_Logistic_Model, newdata=test_data, type='response')
pred_class_reduce <- ifelse(predictions_reduce >= 0.5, 1, 0)
```

Create a confusion matrix

```
Predicted = pred_class_reduce
Actual = test_data$Depression
Rlogistic_conf_matrix <- table(Predicted,Actual )
print(Rlogistic_conf_matrix)
```

```
##          Actual
## Predicted   No  Yes
##         0 1144  248
##         1  299 1790
```

```r
# Extract values from Confusion Matrix
TN_Rlogistic <- Rlogistic_conf_matrix[1, 1]  # True Negatives
FP_Rlogistic <- Rlogistic_conf_matrix[1, 2]  # False Positives
FN_Rlogistic <- Rlogistic_conf_matrix[2, 1]  # False Negatives
TP_Rlogistic <- Rlogistic_conf_matrix[2, 2]  # True Positives

# Compute Performance Metrics
accuracy_Rlogistic <- round((TP_Rlogistic + TN_Rlogistic) / (TP_Rlogistic + TN_Rlogistic + FP_Rlogistic
precision_Rlogistic <- round(TP_Rlogistic / (TP_Rlogistic + FP_Rlogistic), 4)
recall_Rlogistic <- round(TP_Rlogistic / (TP_Rlogistic + FN_Rlogistic), 4)
f1_score_Rlogistic <- round(2 * ((precision_Rlogistic * recall_Rlogistic) / (precision_Rlogistic + recal

# Print Metrics
cat("Reduce Logistic Model Performance Metrics:\n")
```

```
## Reduce Logistic Model Performance Metrics:
```

```r
cat("Model Accuracy:", accuracy_Rlogistic, "\n")
```

```
## Model Accuracy: 0.8429
```

```r
cat("Precision:", precision_Rlogistic, "\n")
```

```
## Precision: 0.8783
```

```r
cat("Recall:", recall_Rlogistic, "\n")
```

```
## Recall: 0.8569
```

```r
cat("F1 Score:", f1_score_Rlogistic, "\n")
```

```
## F1 Score: 0.8675
```

Percentage of correct predictions

```r
correct_predictions_reduce <- sum(diag(Rlogistic_conf_matrix)) / sum(Rlogistic_conf_matrix)
print(paste("Overall Fraction of Correct Predictions: ", correct_predictions_reduce))
```

```
## [1] "Overall Fraction of Correct Predictions:  0.84286124676817"
```

From the above results, we conclude that:

- Out of the 3481 rows in the test dataset the model successfully predicted 2934 depression cases. Therefore 547 depression cases are false positive and false negative.
- The accuracy of this model is about 84.3%. In other words, 84.3% of the times, it predicts correctly.

Accuracy for both Logistic and Reduced Logistic Models:

After running the same test in both logistic models, it is easier to compare them and display an objective result: Figure 21 shows the comparison between the two logistic models. It is evident that the reduced model behaves better than the full model when evaluated or being used for prediction purposes. In addition, that, the reduce model is less complex having two less predictors making the reduced logistic model a better option for usage.

Compare Logistic Models:

```
# Create a data frame to store model performance
model_results <- data.frame(
  Model = c("Logistic Regression", "Reduce Logistic Model"),
  Correct_rate_Predictors = c(
    logistic_correct_predictions,
    correct_predictions_reduce
  )
)

# Display the results
print(model_results)
```

```
##                   Model Correct_rate_Predictors
## 1   Logistic Regression               0.8440103
## 2 Reduce Logistic Model               0.8428612
```

The comparison between the two logistic models. It is evident that the original model behaves better than the reduced model when evaluated or being used for prediction purposes. However, the reduce model is less complex having two less predictors. According to the needs of the user and the actual application of the model the decision of which model is better depended on the mentioned factor.

Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) is a supervised machine learning technique used for classification and dimensionality reduction. It identifies a linear combination of features that optimally separates multiple classes by maximizing the ratio of between-class variance to within-class variance. LDA constructs the within-class scatter matrix, representing variations within each class, and the between-class scatter matrix, capturing differences between class means, to determine the most effective projection direction. Unlike Principal Component Analysis (PCA), which prioritizes variance, LDA preserves information crucial for distinguishing classes. It is widely applied in pattern recognition, medical diagnosis, and face recognition, though its effectiveness depends on the assumption that data is normally distributed with equal covariance across classes, which should be validated before use.LDA Requires some assumptions to meet:

1. Normality:

Plot

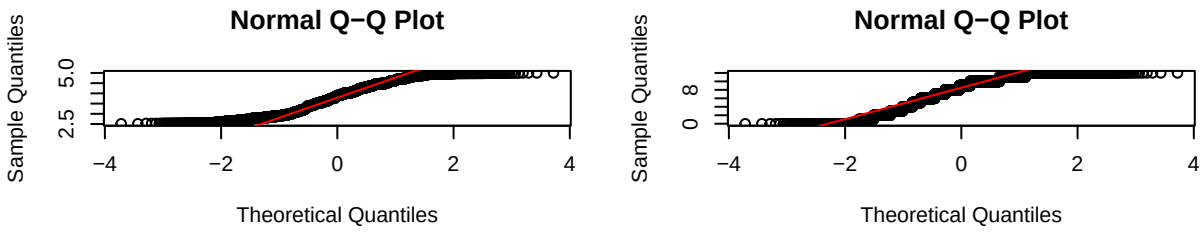
```
## This Variable is the Sample
stratified_sample_positive <- stratified_sample[stratified_sample$Depression == 'Yes',]
stratified_sample_negative <- stratified_sample[stratified_sample$Depression == 'No',]

# Limit the size to a maximum of 5000 values
```

```r
stratified_sample_positive <- stratified_sample_positive[1:min(5000, nrow(stratified_sample_positive)),
stratified_sample_negative <- stratified_sample_negative[1:min(5000, nrow(stratified_sample_negative)),
```

```r
variables <- c("Gender",  "Academic_Pressure",  "CGPA", "Study_Satisfaction", "Sleep_Duration", "Dietary
               "Financial_stress",  "Fam_Mental_Hist",  "Depression",  "Age_Group")

# Ensure variables are numeric before performing the Normality tests
numeric_variables <- sapply(stratified_sample_positive[, variables], is.numeric)
numeric_vars <- variables[numeric_variables]
```

Q-Q Plots for Positive Values

```r
par(mfrow = c(3,2))

for (i in numeric_vars) {
  if (is.numeric(stratified_sample_positive[[i]])) {
    qqnorm(stratified_sample_positive[[i]])
    qqline(stratified_sample_positive[[i]], col = "Red")
  } else {
    cat("Skipping non-numeric variable:", i, "\n")
  }
}
```



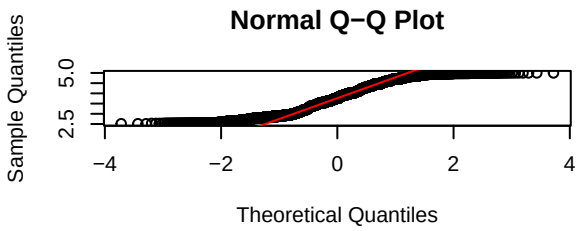
Shapiro Test for Positive Values

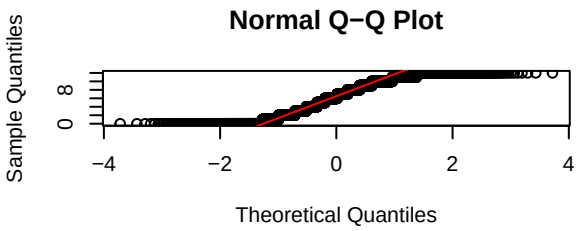
```r
# Perform Shapiro-Wilk test
shapiro_results <- list()

for (i in numeric_vars) {
  shapiro_results[[i]] <- shapiro.test(stratified_sample_positive[[i]])
  cat("Shapiro-Wilk test for", i, ":\n")
  print(shapiro_results[[i]])
  cat("\n")
}
```

```
## Shapiro-Wilk test for CGPA :
##
##  Shapiro-Wilk normality test
##
## data:  stratified_sample_positive[[i]]
## W = 0.9478, p-value < 2.2e-16
##
##
## Shapiro-Wilk test for Study_Hours :
##
##  Shapiro-Wilk normality test
##
## data:  stratified_sample_positive[[i]]
## W = 0.90744, p-value < 2.2e-16
```

Q-Q Plots for Negative Values

```r
par(mfrow = c(3,2))

for (i in numeric_vars) {
  if (is.numeric(stratified_sample_negative[[i]])) {
    qqnorm(stratified_sample_negative[[i]])
    qqline(stratified_sample_negative[[i]], col = "Red")
  } else {
    cat("Skipping non-numeric variable:", i, "\n")
  }
}
```





Shapiro Test for Negative Values

```
# Perform Shapiro-Wilk test
shapiro_results <- list()

for (i in numeric_vars) {
  shapiro_results[[i]] <- shapiro.test(stratified_sample_negative[[i]])
  cat("Shapiro-Wilk test for", i, ":\n")
  print(shapiro_results[[i]])
  cat("\n")
}
```

```
## Shapiro-Wilk test for CGPA :
##
##  Shapiro-Wilk normality test
##
## data:  stratified_sample_negative[[i]]
## W = 0.9428, p-value < 2.2e-16
##
##
## Shapiro-Wilk test for Study_Hours :
##
##  Shapiro-Wilk normality test
##
## data:  stratified_sample_negative[[i]]
## W = 0.93109, p-value < 2.2e-16
```

From the result of both the QQ plot and the shapiro test we can see that the data is not normally distributed, Normality can be treated, it is possible to used log transformation. However we are analyzing how the models behabe first and in case that the LDA model performs best then the data tranformation will take place other wise it will be ignored.

2. **Fit LDA Model**

```
lda_model <- lda(Depression ~., data = train_data)
print(lda_model)
```

```
## Call:
## lda(Depression ~ ., data = train_data)
##
## Prior probabilities of groups:
##       No      Yes
## 0.414417 0.585583
##
## Group means:
##     GenderMale Academic_Pressure2 Academic_Pressure3 Academic_Pressure4
## No   0.5585586          0.2245322          0.2554863          0.1081081
## Yes  0.5618767          0.0980873          0.2726827          0.2412948
##     Academic_Pressure5     CGPA Study_Satisfaction2 Study_Satisfaction3
## No          0.06791407 3.797021           0.1769462           0.2157542
## Yes         0.33006376 3.825401           0.2277260           0.2133399
##     Study_Satisfaction4 Study_Satisfaction5 Sleep_Duration7-8 hours
## No            0.2647263           0.2025872               0.2510973
## Yes           0.1989537           0.1317639               0.2728462
##     Sleep_DurationLess than 5 hours Sleep_DurationMore than 8 hours
## No                        0.2547933                       0.2661123
## Yes                       0.3320255                       0.1835867
##     Dietary_HabitsModerate Dietary_HabitsUnhealthy Suicidal_ThoughtsYes
## No               0.3869254              0.2499422            0.3229383
## Yes              0.3406899              0.4451529            0.8589178
##     Study_Hours Financial_stress2 Financial_stress3 Financial_stress4
## No     6.215292         0.2413952         0.1917302         0.1559252
## Yes    7.841426         0.1355239         0.1922511         0.2411313
##     Financial_stress5 Fam_Mental_HistYes Age_Groupgroup_2 Age_Groupgroup_3
## No          0.1101871          0.4398244        0.4492954     0.0002310002
## Yes         0.3323525          0.5041687        0.2448913     0.0001634788
##     Age_Groupgroup_4
## No      0.0004620005
## Yes     0.0000000000
##
## Coefficients of linear discriminants:
##                                       LD1
## GenderMale                     -0.00290920
## Academic_Pressure2              0.53942600
## Academic_Pressure3              1.06916057
## Academic_Pressure4              1.50656790
## Academic_Pressure5              1.79057885
## CGPA                            0.03329860
## Study_Satisfaction2            -0.19122843
## Study_Satisfaction3            -0.21070748
```

```
## Study_Satisfaction4             -0.37154287
## Study_Satisfaction5             -0.50877698
## Sleep_Duration7-8 hours          0.04145317
## Sleep_DurationLess than 5 hours  0.13265340
## Sleep_DurationMore than 8 hours -0.18029830
## Dietary_HabitsModerate           0.25307649
## Dietary_HabitsUnhealthy          0.56580601
## Suicidal_ThoughtsYes             1.53573984
## Study_Hours                      0.05928446
## Financial_stress2                0.26290682
## Financial_stress3                0.57618746
## Financial_stress4                0.80444412
## Financial_stress5                1.12768897
## Fam_Mental_HistYes               0.15589561
## Age_Groupgroup_2                -0.48285394
## Age_Groupgroup_3                -0.14953266
## Age_Groupgroup_4                -1.90882562
```

```
lda_predictions <- predict(lda_model, newdata = test_data)$class
```

```
lda_conf_matrix <-table( Predicted = lda_predictions, Actual =test_data$Depression)
lda_conf_matrix
```

```
##           Actual
## Predicted   No  Yes
##       No  1123  231
##       Yes  320 1807
```

Percentage of correct predictions

```
lda_correct_predictions <- sum(diag(lda_conf_matrix)) / sum(lda_conf_matrix)
print(paste("Overall Correct Predictions Percentages: ", lda_correct_predictions))
```

```
## [1] "Overall Correct Predictions Percentages:  0.841712151680552"
```

Model Performance

```
# Extract values from Confusion Matrix Lineal LDA
TN_LDA <- lda_conf_matrix[1, 1]  # True Negatives
FP_LDA <- lda_conf_matrix[1, 2]  # False Positives
FN_LDA <- lda_conf_matrix[2, 1]  # False Negatives
TP_LDA <- lda_conf_matrix[2, 2]  # True Positives

# Compute Performance Metrics
accuracy_LDA <- round((TP_LDA + TN_LDA) / (TP_LDA + TN_LDA + FP_LDA + FN_LDA), 4)
precision_LDA <- round(TP_LDA / (TP_LDA + FP_LDA), 4)
recall_LDA <- round(TP_LDA / (TP_LDA + FN_LDA), 4)
f1_score_LDA <- round(2 * ((precision_LDA * recall_LDA) / (precision_LDA + recall_LDA)), 4)

# Print Metrics
cat("LDA Model Performance Metrics:\n")
```

```
## LDA Model Performance Metrics:
```

```
cat("Model Accuracy:", accuracy_LDA, "\n")
```

```
## Model Accuracy: 0.8417
```

```
cat("Precision:", precision_LDA, "\n")
```

```
## Precision: 0.8867
```

```
cat("Recall:", recall_LDA, "\n")
```

```
## Recall: 0.8496
```

```
cat("F1 Score:", f1_score_LDA, "\n")
```

```
## F1 Score: 0.8678
```

From the results, we conclude that: • Out of the 3481 rows in the test dataset the model successfully predicted 2930 depression cases. Therefore 551 depression cases are false positive and false negative. • The accuracy of this model is about 84.17%. In other words, 84.2% of the times, it predicts correctly.

Quadratic Discriminant Analysis (QDA)

Quadratic Discriminant Analysis (QDA) is a classification technique that assumes different covariance matrices for each class, allowing for more flexibility compared to Linear Discriminant Analysis (LDA). QDA models the probability distribution of each class using a multivariate normal distribution and then classifies new observations based on the likelihood of belonging to each class. This method is particularly useful when the decision boundaries between classes are nonlinear. The primary advantage of QDA is its ability to model complex relationships between features, but it requires a larger dataset to estimate the covariance matrices accurately and may be less stable with small sample sizes or highly collinear features. QDA does not care about the Factors elements all contrary it might affect the behavior of the model, therefore we can used a copy of the original clean data set that has not the factors changed integrated.

```
str(depression_dataset_copy)
```

```
## 'data.frame':    27856 obs. of  12 variables:
##  $ Gender            : chr  "Male" "Female" "Male" "Female" ...
##  $ Academic_Pressure : int  5 2 3 3 4 2 3 2 3 2 ...
##  $ CGPA              : num  4.49 2.95 3.52 2.79 4.07 ...
##  $ Study_Satisfaction: int  2 5 5 2 3 3 4 4 1 3 ...
##  $ Sleep_Duration    : chr  "5-6 hours" "5-6 hours" "Less than 5 hours" "7-8 hours" ...
##  $ Dietary_Habits    : chr  "Healthy" "Moderate" "Healthy" "Moderate" ...
##  $ Suicidal_Thoughts : chr  "Yes" "No" "No" "Yes" ...
##  $ Study_Hours       : int  3 3 9 4 1 4 1 0 12 2 ...
##  $ Financial_stress  : int  1 2 1 5 1 1 2 1 3 5 ...
##  $ Fam_Mental_Hist   : chr  "No" "Yes" "Yes" "Yes" ...
##  $ Depression        : chr  "Yes" "No" "No" "Yes" ...
##  $ Age_Group         : chr  "group_2" "group_1" "group_2" "group_1" ...
```

Important note:

Quadratic Discriminant Analysis (QDA) is a classification technique that assumes different covariance matrices for each class, allowing for more flexibility compared to Linear Discriminant Analysis (LDA). QDA models the probability distribution of each class using a multivariate normal distribution and then classifies new observations based on the likelihood of belonging to each class. This method is particularly useful when the decision boundaries between classes are nonlinear. The primary advantage of QDA is its ability to model complex relationships between features, but it requires a larger dataset to estimate the covariance matrices accurately and may be less stable with small sample sizes or highly collinear features.

Same as LDA, QDA needs to meet some data assumption from the model to perform accurate. These assumptions were examined during the LDA analysis.

- Normality
- Linear independency
- Sufficient data

From the LDA analysis all assumptions were meet. Moving on, for Quadratic Discriminant Analysis (QDA), the balance between variables is crucial. From the previous Exploratory Data Analysis (EDA), we gained insights into the data distribution and identified some important considerations regarding *Age Grouping*. Specifically, out of the four groups, two have minimal representation and hold almost no values compared to the other two groups. To maintain the integrity of the dataset, we will remove these two underrepresented groups. Additionally, as these groups are considered outliers, it is a good practice to eliminate them at this stage.

Drop the Age Group 3 and 4

```
depression_dataset_copy <- depression_dataset_copy[depression_dataset_copy$Age_Group != "group_3", ]
depression_dataset_copy <- depression_dataset_copy[depression_dataset_copy$Age_Group != "group_4", ]
```

Age Group Confirmation

```
age_Groups <- unique(depression_dataset_copy$Age_Group)
print(age_Groups)
```

```
## [1] "group_2" "group_1"
```

Display the Dimensions of the update dataset

```
dim(depression_dataset_copy)
```

```
## [1] 27834    12
```

Picture XX shows that new dataset dimensions being reduced after removing the students from the group ages 3 and 4. The picture also shows that there are only 2 groups in the age grouping column.

The next step for the QDA is to create a new dataset using stratification method this because QDA is sensible to the distribution, so it is important to keep in check the correct distribution of values for positive and negative cases.

Get the stratification Sample

```r
#Total size 27834, 75% = 20875 (10438, 10437) , 25% = 6958
set.seed(2025)
index_quadratic <- sampling:::strata(depression_dataset_copy, stratanames = c('Depression'), size =c(10
```

Separate and set the Stratification in TEsting and Training

```r
training_quadratic  <- depression_dataset_copy[index_quadratic$ID_unit,]
testing_quadratic <- depression_dataset_copy[- index_quadratic$ID_unit,]
```

New Datasets Dimensions

```r
# Training
dim(training_quadratic)
```

```
## [1] 20875    12
```

```r
table(training_quadratic$Depression)
```

```
##
##    No   Yes
## 10437 10438
```

```r
# Testing
#dim(testing_quadratic)
#table(testing_quadratic$Depression)
```

The next step then is to load the QDA model with the training dataset:

1. Fit the QDA Model

```r
quadratic_model<- qda(Depression ~. , data = training_quadratic)
```

After having the QDA model it is ready to test with test data:

2. Create a confusion matrix

```r
qda_predictions <- predict(quadratic_model, newdata = testing_quadratic)$class
qda_conf_matrix <-table( Predicted = qda_predictions, Actual =testing_quadratic$Depression)
qda_conf_matrix
```

```
##           Actual
## Predicted   No  Yes
##       No   916  974
##       Yes  178 4891
```

```r
# Extract values from Confusion Matrix
TN_QDA <- qda_conf_matrix[1, 1]  # True Negatives
FP_QDA <- qda_conf_matrix[1, 2]  # False Positives
FN_QDA <- qda_conf_matrix[2, 1]  # False Negatives
```

```r
TP_QDA <- qda_conf_matrix[2, 2]  # True Positives

# Compute Performance Metrics
accuracy_QDA <- round((TP_QDA + TN_QDA) / (TP_QDA + TN_QDA + FP_QDA + FN_QDA), 4)
precision_QDA <- round(TP_QDA / (TP_QDA + FP_QDA), 4)
recall_QDA <- round(TP_QDA / (TP_QDA + FN_QDA), 4)
f1_score_QDA <- round(2 * ((precision_QDA * recall_QDA) / (precision_QDA + recall_QDA)), 4)

# Print Metrics
cat("QDA Model Performance Metrics:\n")
```

```
## QDA Model Performance Metrics:
```

```r
cat("Model Accuracy:", accuracy_QDA, "\n")
```

```
## Model Accuracy: 0.8345
```

```r
cat("Precision:", precision_QDA, "\n")
```

```
## Precision: 0.8339
```

```r
cat("Recall:", recall_QDA, "\n")
```

```
## Recall: 0.9649
```

```r
cat("F1 Score:", f1_score_QDA, "\n")
```

```
## F1 Score: 0.8946
```

3. Percentage of correct predictions

```r
qda_correct_predictions <- sum(diag(qda_conf_matrix)) / sum(qda_conf_matrix)
print(paste("Overall Correct Predictions Percentages: ", qda_correct_predictions))
```

```
## [1] "Overall Correct Predictions Percentages:  0.834458973990516"
```

From the results, we conclude that:

- Out of the 6959 rows in the test dataset the model successfully predicted 5807 depression cases. Therefore 1152 depression cases are false positive and false negative.
- The accuracy of this model is about 83.44%. In other words, 83.4% of the times, it predicts correctly.

Regression Tree Model

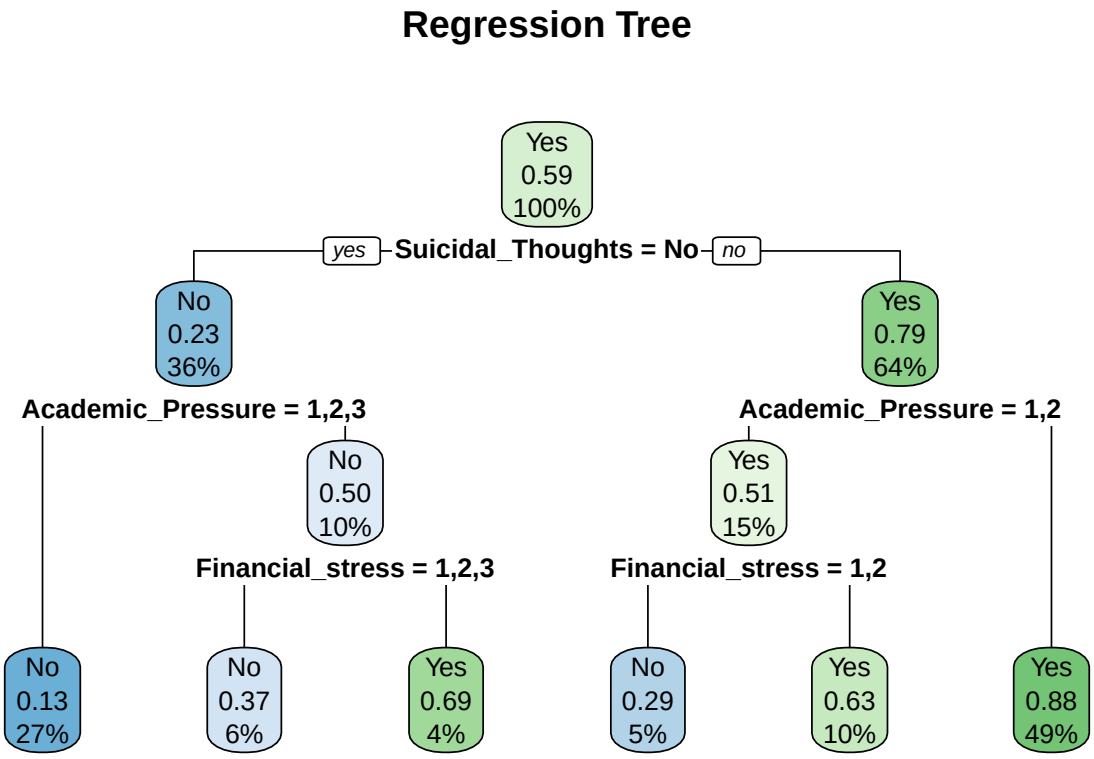
A tree model, also known as a decision tree, is a machine learning algorithm used for classification and regression tasks. It works by recursively splitting the data into subsets based on the values of input features, creating a tree-like structure of decisions. Each internal node represents a decision based on a feature, each branch represents the outcome of the decision, and each leaf node represents a predicted outcome or class.

The main advantages of tree models are their interpretability and simplicity, as they mimic human decision-making processes. They are also non-parametric, meaning they do not assume any underlying distribution of the data, making them versatile and effective for various types of data.

Same as previous models we load the model using the training data to verify its accuaracy percantage.

1. Applying Tree Model to Test Dataset

```
tree_Model <- rpart(Depression ~ ., data = train_data, method = "class")
rpart.plot(tree_Model, main = "Regression Tree")
```



From the model representation it is possible to notice that there are 3 main categories that the model uses to generate the dessicions. * Suicidal Thoughts * Academic_preasure * Financial Stress

2. Create a confusion matrix

```
tree_predictions<-predict(tree_Model,test_data,type = "class")# class means classification
tree_conf_matrix <-table(tree_predictions,test_data$Depression)
tree_conf_matrix
```

```
##
## tree_predictions   No  Yes
##              No  1072  264
##              Yes  371 1774
```

```
# Extract values from Confusion Matrix
TN_tree <- tree_conf_matrix[1, 1]  # True Negatives
```

```r
FP_tree <- tree_conf_matrix[1, 2]  # False Positives
FN_tree <- tree_conf_matrix[2, 1]  # False Negatives
TP_tree <- tree_conf_matrix[2, 2]  # True Positives

# Compute Performance Metrics
accuracy_tree <- round((TP_tree + TN_tree) / (TP_tree + TN_tree + FP_tree + FN_tree), 4)
precision_tree <- round(TP_tree / (TP_tree + FP_tree), 4)
recall_tree <- round(TP_tree / (TP_tree + FN_tree), 4)
f1_score_tree <- round(2 * ((precision_tree * recall_tree) / (precision_tree + recall_tree)), 4)

# Print Metrics
cat("Decision Tree Model Performance Metrics:\n")
```

```
## Decision Tree Model Performance Metrics:
```

```r
cat("Model Accuracy:", accuracy_tree, "\n")
```

```
## Model Accuracy: 0.8176
```

```r
cat("Precision:", precision_tree, "\n")
```

```
## Precision: 0.8705
```

```r
cat("Recall:", recall_tree, "\n")
```

```
## Recall: 0.827
```

```r
cat("F1 Score:", f1_score_tree, "\n")
```

```
## F1 Score: 0.8482
```

3. Percentage of correct predictions

```r
tree_correct_predictions <- sum(diag(tree_conf_matrix)) / sum(tree_conf_matrix)
print(paste("Overall Correct Predictions Percentages: ", tree_correct_predictions))
```

```
## [1] "Overall Correct Predictions Percentages:  0.817581154840563"
```

From the results, we conclude that:

- Out of the 3481 rows in the test dataset the model successfully predicted 2805 depression cases. Therefore 676 depression cases are false positive and false negative.
- The accuracy of this model is about 80.58%. In other words, 80.6% of the times, it predicts correctly.

Random Forest Model

A Random Forest is an ensemble learning method used for classification and regression tasks. It operates by constructing multiple decision trees during training and outputting the mode of the classes (classification) or mean prediction (regression) of the individual trees. This approach enhances the accuracy and robustness of the model by reducing overfitting.

The output of a Random Forest model includes the predicted values and an assessment of variable importance. The variable importance plot shows which features contribute the most to the prediction, providing insights into the underlying patterns in the data. In addition to that, same as Tree model, the do not require stringent assumptions about the data, such as normality or linear relationships. It can handle both numerical and categorical features, manage missing values, and is resistant to overfitting due to its ensemble nature.

Splitting the data

```
# Ensure 'Depression' is a factor
stratified_sample$Depression <- as.factor(stratified_sample$Depression)

# Split data using stratified sampling (75% training, 25% testing)
set.seed(2025)
trainIndex <- createDataPartition(stratified_sample$Depression, p = 0.75, list = FALSE)
train_data <- stratified_sample[trainIndex, ]
test_data <- stratified_sample[-trainIndex, ]
```

Fit model

```
# Train Random Forest Model
set.seed(2025)
random_model <- randomForest(Depression ~ ., data = train_data, ntree = 500, mtry = 3, importance = TRU

# Print Model Summary
print(random_model)
```

```
##
## Call:
##  randomForest(formula = Depression ~ ., data = train_data, ntree = 500,      mtry = 3, importance = 
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 16.09%
## Confusion matrix:
##       No  Yes class.error
## No  3372  957   0.2210672
## Yes  724 5393   0.1183587
```

```
# Make Predictions
random_predictions <- predict(random_model, newdata = test_data)

# Ensure factor levels match
random_predictions <- factor(random_predictions, levels = levels(test_data$Depression))

# Compute Confusion Matrix
random_conf_matrix <- confusionMatrix(random_predictions, test_data$Depression)
print(random_conf_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1159  275
##        Yes  284 1763
##
##                Accuracy : 0.8394
##                  95% CI : (0.8268, 0.8515)
##     No Information Rate : 0.5855
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6689
##
##  Mcnemar's Test P-Value : 0.7351
##
##             Sensitivity : 0.8032
##             Specificity : 0.8651
##          Pos Pred Value : 0.8082
##          Neg Pred Value : 0.8613
##              Prevalence : 0.4145
##          Detection Rate : 0.3330
##    Detection Prevalence : 0.4120
##       Balanced Accuracy : 0.8341
##
##        'Positive' Class : No
##
```

```
# Extract values from Confusion Matrix
TN_random <- random_conf_matrix$table[1, 1]  # True Negatives
FP_random <- random_conf_matrix$table[1, 2]  # False Positives
FN_random <- random_conf_matrix$table[2, 1]  # False Negatives
TP_random <- random_conf_matrix$table[2, 2]  # True Positives

# Compute Performance Metrics
accuracy_random <- round((TP_random + TN_random) / (TP_random + TN_random + FP_random + FN_random),4)
precision_random <- round(TP_random / (TP_random + FP_random), 4)
recall_random <- round(TP_random / (TP_random + FN_random), 4)
f1_score_random <- round(2 * ((precision_random * recall_random) / (precision_random + recall_random)),

# Print Metrics
cat("Random Forest Model Performance Metrics:\n")
```

```
## Random Forest Model Performance Metrics:
```

```
cat("Model Accuracy:", accuracy_random, "\n")
```

```
## Model Accuracy: 0.8394
```

```
cat("Precision:", precision_random, "\n")
```

```
## Precision: 0.8651
```

```r
cat("Recall:", recall_random, "\n")
```

```
## Recall: 0.8613
```

```r
cat("F1 Score:",f1_score_random,"\n")
```

```
## F1 Score: 0.8632
```

From the results, we conclude that: • Out of the 3481 rows in the test dataset the model successfully predicted 2920 depression cases. Therefore 561 depression cases are false positive and false negative. • The accuracy of this model is about 83.88%. In other words, 83.9% of the times, it predicts correctly.

Models Performance

In the field of mental health analytics, predicting depression based on various psychological, social, and lifestyle factors is a crucial challenge. Machine learning models provide a powerful approach to analyzing complex patterns in depression-related data. However, selecting the best model requires evaluating multiple algorithms based on key performance metrics such as accuracy, precision, recall, and F1-score.

This study compares five machine learning models—Logistic Regression, Random Forest, Decision Tree, Linear Discriminant Analysis (LDA), and Quadratic Discriminant Analysis (QDA)—to determine which model is best suited for depression prediction. Each model is trained and tested using a structured dataset containing features such as academic pressure, sleep duration, dietary habits, suicidal thoughts, study hours, financial stress, and family mental health history.

The performance of each model is evaluated using four critical metrics:

- Accuracy: Measures the overall correctness of the model.
- Precision: Evaluates how many predicted positive cases are actually positive.
- Recall (Sensitivity): Measures the model's ability to identify true positive cases.
- F1-Score: Balances precision and recall to assess the model's reliability.

By comparing these metrics, we aim to determine which model provides the most reliable predictions for identifying individuals at risk of depression. The findings from this analysis will help in selecting an optimal model for real-world applications in mental health screening and early intervention.

```r
# Create dataframes to hold confusion matrices
confusion_matrices <- data.frame(
  Model = c("Logistic", "Reduce Logistic", "LDA", "QDA", "Random Forest", "Decision Tree"),
  TN = c(TN_logistic, TN_Rlogistic, TN_LDA, TN_QDA, TN_random, TN_tree),
  FP = c(FP_logistic, FP_Rlogistic, FP_LDA, FP_QDA, FP_random, FP_tree),
  FN = c(FN_logistic, FN_Rlogistic, FN_LDA, FN_QDA, FN_random, FN_tree),
  TP = c(TP_logistic, TP_Rlogistic, TP_LDA, TP_QDA, TP_random, TP_tree)
)

# Create dataframes to hold performance metrics
performance_metrics <- data.frame(
  Model = c("Logistic", "Reduce Logistic", "LDA", "QDA", "Random Forest", "Decision Tree"),
  Accuracy = c(accuracy_logistic, accuracy_Rlogistic, accuracy_LDA, accuracy_QDA, accuracy_random, accu
  Precision = c(precision_logistic, precision_Rlogistic, precision_LDA, precision_QDA, precision_random
  Recall = c(recall_logistic, recall_Rlogistic, recall_LDA, recall_QDA, recall_random, recall_tree),
```

```
  F1_Score = c(f1_score_logistic, f1_score_Rlogistic, f1_score_LDA, f1_score_QDA, f1_score_random, f1_s
)

# Print dataframes
print(confusion_matrices)
```

```
##             Model   TN  FP  FN   TP
## 1        Logistic 1143 243 300 1795
## 2 Reduce Logistic 1144 248 299 1790
## 3             LDA 1123 231 320 1807
## 4             QDA  916 974 178 4891
## 5   Random Forest 1159 275 284 1763
## 6   Decision Tree 1072 264 371 1774
```

```
print(performance_metrics)
```

```
##             Model Accuracy Precision Recall F1_Score
## 1        Logistic   0.8440    0.8808 0.8568   0.8686
## 2 Reduce Logistic   0.8429    0.8783 0.8569   0.8675
## 3             LDA   0.8417    0.8867 0.8496   0.8678
## 4             QDA   0.8345    0.8339 0.9649   0.8946
## 5   Random Forest   0.8394    0.8651 0.8613   0.8632
## 6   Decision Tree   0.8176    0.8705 0.8270   0.8482
```

Overall, the Logistic Model is the most balanced and accurate, but the Reduce Logistic Model and LDA are strong alternatives. The QDA model excels in recall, making it a good choice if the priority is to minimize false negatives. Quadratic Discriminant Analysis (QDA) shows good recall, it struggles with precision, making it less optimal for ensuring accurate positive classifications. Decision Tree, on the other hand, performs the worst due to its low recall and F1-score, leading to a higher rate of misclassification. These findings suggest that LDA is the most suitable model for practical applications in mental health prediction, providing a strong foundation for decision-making in early detection and intervention strategies. However, further improvements such as feature selection, hyperparameter tuning, or ensemble methods could enhance model performance, leading to more refined and accurate depression predictions.

Conclusion:

In conclusion the dataset yields several key insights into the factors influencing student mental health. Among the predictive models tested, the Logistic Model, Reduced Logistic Model, and Linear Discriminant Analysis (LDA) demonstrated strong predictive performance, with the Logistic Model achieving the highest accuracy. The Quadratic Discriminant Analysis (QDA) model was particularly effective in identifying true cases of depression due to its high recall rate. Our analysis highlighted three primary factors that significantly contributed to depression predictions: suicidal thoughts, academic pressure, and financial stress. Students exhibiting suicidal ideation, experiencing high academic pressure, or facing financial instability were at the highest risk of expressing depression. Additionally, other variables, including unhealthy dietary habits, insufficient sleep, age, gender, study hours, and study satisfaction, also played a role in predicting depression. The data indicated that younger students, male students, and those experiencing high academic or financial stress were more vulnerable to depression. However, a key limitation of this study is that the linearity assumption was not fully met, which may affect the interpretation of some model results. Future research should explore non-linear modeling approaches to enhance predictive accuracy. Despite this limitation, our findings provide valuable insights into the multifaceted nature of student mental health and underscore the

importance of targeted interventions to mitigate depression risks. While further research is necessary for a more comprehensive understanding, these findings suggest that mental health interventions tailored to high-risk groups particularly those struggling with financial stress, academic pressure, or suicidal ideation could be instrumental in effectively managing and reducing depression among students.