# Project Title

# Data Cleaning & Preprocessing on Student Mental Health Dataset


## Author

**Md. Salman Farshe**



## Abstract:

This project focuses on cleaning, preprocessing, visualizing, and preparing a student mental health dataset for further analysis or modeling. The main objectives include handling missing values, detecting outliers, encoding categorical variables, normalization, duplicate removal, balancing the dataset, and splitting into training/testing sets.


## Objectives of the Project

- Identify and handle missing values
- Visualize missing values
- Detect and treat outliers using multiple approaches
- Identify invalid entries and correct them
- Convert categorical variables into numeric codes
- Normalize continuous variables
- Remove duplicate rows
- Apply filtering methods
- Fix class imbalance using oversampling
- Split the dataset into Training & Testing sets
- Perform descriptive statistical analysis
- Analyze sleep duration & study satisfaction patterns

# Dataset Description

This dataset is about 201 students and gives us a glimpse into their daily lives, challenges, and mental health. It covers simple things like age, gender, how many hours they usually sleep, their eating habits, and how much time they spend studying. It also looks at more personal aspects, such as whether they feel academic or financial pressure, if they've ever had suicidal thoughts, whether there's a family history of mental illness, and if they struggle with depression. Most of the information is filled in, but a few students skipped questions about their age, sleep, study hours, or mental health. Dataset presents a connection of how personal Information can be connected to students well-being.

***Upload Dataset and view:***

ds <- read.csv("E:/9th semester-Summer 2025/Data Science/Project1/Midterm_Dataset_Section(E).csv",na = c("NA", ""),stringsAsFactors = FALSE, fileEncoding = "UTF-8")

head(ds)

summary(ds)

```
> ds <- read.csv("E:/9th semester-Summer 2025/Data Science/Project1/Midterm_Dataset_Section(E).csv",na = c("NA", ""),stringsAsFactors
= FALSE, fileEncoding = "UTF-8")
> head(ds)
  Gender Age Academic.Pressure Study.Satisfaction    Sleep.Duration Dietary.Habits Have.you.ever.had.suicidal.thoughts..
1   Male  28                 2                   4         7-8 hours       Moderate                                    Yes
2   Male  28                 4                   5         5-6 hours        Healthy                                    Yes
3   Male  25                 1                   3         5-6 hours      Unhealthy                                    Yes
4   Male  23                 1                   4 More than 8 hours      Unhealthy                                    Yes
5   <NA>  31                 1                   5 More than 8 hours        Healthy                                    Yes
6   Male  19                 4                   4         5-6 hours      Unhealthy                                    Yes
  Study.Hours Financial.Stress Family.History.of.Mental.Illness Depression
1           9                2                              Yes         No
2           7                1                              Yes         No
3          10                4                               No        Yes
4           7                2                              Yes         No
5           4                2                              Yes         No
6           1                4                              Yes        Yes
> summary(ds)
    Gender               Age         Academic.Pressure Study.Satisfaction Sleep.Duration     Dietary.Habits
 Length:201         Min.   :-30.00   Min.   : 1.000    Min.   :1.000      Length:201         Length:201
 Class :character   1st Qu.: 22.00   1st Qu.: 2.000    1st Qu.:2.000      Class :character   Class :character
 Mode  :character   Median : 26.00   Median : 3.000    Median :3.000      Mode  :character   Mode  :character
                    Mean   : 27.75   Mean   : 3.154    Mean   :3.189
                    3rd Qu.: 30.00   3rd Qu.: 4.000    3rd Qu.:4.000
                    Max.   :230.00   Max.   :20.000    Max.   :5.000
                    NA's   :3
 Have.you.ever.had.suicidal.thoughts..  Study.Hours     Financial.Stress Family.History.of.Mental.Illness  Depression
 Length:201                            Min.   : 0.000   Min.   :1.00     Length:201                        Length:201
 Class :character                      1st Qu.: 3.000   1st Qu.:2.00     Class :character                  Class :character
 Mode  :character                      Median : 7.000   Median :3.00     Mode  :character                  Mode  :character
                                       Mean   : 6.332   Mean   :2.93
                                       3rd Qu.:10.000   3rd Qu.:4.00
                                       Max.   :12.000   Max.   :5.00
                                       NA's   :2
> str(ds)
'data.frame':   201 obs. of  11 variables:
 $ Gender                              : chr  "Male" "Male" "Male" "Male" ...
 $ Age                                 : int  28 28 25 23 31 19 34 20 NA 33 ...
 $ Academic.Pressure                   : int  2 4 1 1 1 4 4 4 1 4 ...
 $ Study.Satisfaction                  : int  4 5 3 4 5 4 2 1 4 3 ...
 $ Sleep.Duration                      : chr  "7-8 hours" "5-6 hours" "5-6 hours" "More than 8 hours" ...
 $ Dietary.Habits                      : chr  "Moderate" "Healthy" "Unhealthy" "Unhealthy" ...
 $ Have.you.ever.had.suicidal.thoughts..: chr  "Yes" "Yes" "Yes" "Yes" ...
 $ Study.Hours                         : int  9 7 10 7 4 1 6 3 10 10 ...
 $ Financial.Stress                    : int  2 1 4 2 2 4 2 4 3 1 ...
 $ Family.History.of.Mental.Illness    : chr  "Yes" "Yes" "No" "Yes" ...
 $ Depression                          : chr  "No" "No" "Yes" "No" ...
>
```

1. **If there are any missing values in the dataset, we should apply all applicable methods from the available options to handle the missing values.**

***Description of the task 1:***

In this task we looked for missing values in the dataset and handled them in different ways. First we identified which columns had missing values. Then we tried three approaches: one by removing rows with missing values, another by filling numeric values with the column mean, and another by filling categorical values with the most frequent value (mode). At the end we also saved the cleaned dataset so it can be used later.

***Code and output:***

***Missing values Check:***

colSums(is.na(ds))

```
> colSums(is.na(ds))
                                Gender                                    Age                       Academic.Pressure
                                     3                                      3                                       0
                    Study.Satisfaction                       Sleep.Duration                         Dietary.Habits
                                     0                                      3                                       0
Have.you.ever.had.suicidal.thoughts..                            Study.Hours                       Financial.Stress
                                     0                                      2                                       0
      Family.History.of.Mental.Illness                             Depression
                                     0                                      3
```

***Instances delete where missing values are available***

ds_delete <- na.omit(ds)
colSums(is.na(ds_delete))

```
> colSums(is.na(ds_delete))
                                Gender                                    Age                       Academic.Pressure
                                     0                                      0                                       0
                    Study.Satisfaction                       Sleep.Duration                         Dietary.Habits
                                     0                                      0                                       0
Have.you.ever.had.suicidal.thoughts..                            Study.Hours                       Financial.Stress
                                     0                                      0                                       0
      Family.History.of.Mental.Illness                             Depression
                                     0                                      0
>
```

***missing values replace by mean which column have numeric value and update dt:***

```
for (col in names(ds)) {
 if (is.numeric(ds[[col]])) {
  mean_val <- mean(ds[[col]], na.rm = TRUE)
  ds[[col]][is.na(ds[[col]])] <- mean_val
```

```
 }
}
```

colSums(is.na(ds))

```
> colSums(is.na(ds))
                                  Gender                                     Age                          Academic.Pressure
                                       3                                       0                                          0
                      Study.Satisfaction                          Sleep.Duration                             Dietary.Habits
                                       0                                       3                                          0
Have.you.ever.had.suicidal.thoughts..                             Study.Hours                           Financial.Stress
                                       0                                       0                                          0
     Family.History.of.Mental.Illness                              Depression
                                       0                                       3
```

***missing values replace by mean which column have numeric value and update dt and csv file:***

```
for (col in names(ds)) {
 if (is.character(ds[[col]]) || is.factor(ds[[col]]))
 {
  mode_val <- names(which.max(table(ds[[col]])))
  ds[[col]][is.na(ds[[col]])] <- mode_val
 }
}
colSums(is.na(ds))

write.csv(ds, "E:/9th semester-Summer 2025/Data Science/Project/Midterm_Dataset_Section(E).csv", row.names = FALSE)
```

```
> colSums(is.na(ds))
                                  Gender                                     Age                          Academic.Pressure
                                       0                                       0                                          0
                      Study.Satisfaction                          Sleep.Duration                             Dietary.Habits
                                       0                                       0                                          0
Have.you.ever.had.suicidal.thoughts..                             Study.Hours                           Financial.Stress
                                       0                                       0                                          0
     Family.History.of.Mental.Illness                              Depression
                                       0                                       0
> write.csv(ds, "E:/9th semester-Summer 2025/Data Science/Project/Midterm_Dataset_Section(E).csv", row.names = FALSE)
```

***Code description of task 1:***

The code first counted the number of missing values in each column. Then a version of the dataset was made by dropping rows with missing values. After that, numeric columns were fixed by replacing missing values with their mean, and categorical columns were fixed by replacing missing values with the most common category. Finally, the cleaned dataset was written into a new file for later use.

## 2. Missing values on a graph.

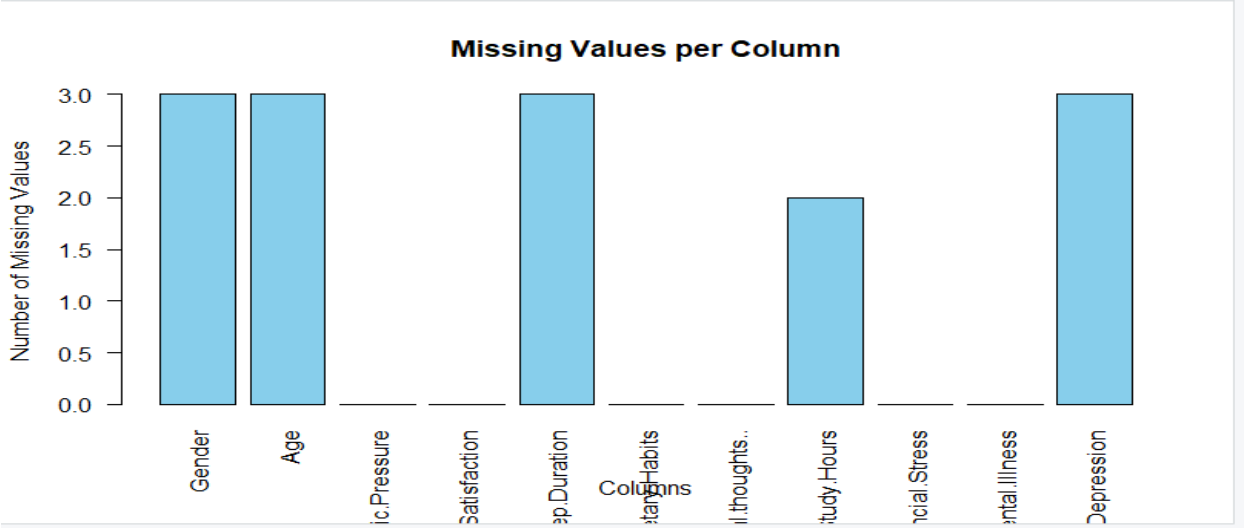
***Description of the task 2:***

In this task we tried to see missing values visually. Instead of only numbers, we wanted a simple bar chart to show how many missing values were in each column, so that it becomes easier to understand which columns need attention.

***Code and output:***

```
missing_counts <- colSums(is.na(ds))

barplot(missing_counts,
    main = "Missing Values per Column",
    xlab = "Columns",
    ylab = "Number of Missing Values",
    col = "skyblue",
    las = 2)
```




***Code description of task 3:***

The code calculated the number of missing values in each column and then used a barplot to display these counts. This gave a clear view of where the data problems exist and how serious they are.

## 3. Detect outliers in the data set and use the appropriate approach to handle those values.

***Description of the task 1:***

Here we focused on outliers in the numeric columns. Outliers were found using the IQR method, which checks values outside the normal range. We handled them in three different ways: removing rows with outliers, replacing them with the mean, and also replacing them with the median. This helped to reduce the effect of extreme values in the dataset.

***Code and output:***

***Outlier detection function (IQR method) and find outlier:***

```
detect_outliers <- function(x) {
 Q1 <- quantile(x, 0.25, na.rm = TRUE)
 Q3 <- quantile(x, 0.75, na.rm = TRUE)
 IQR_val <- Q3 - Q1
 lower_bound <- Q1 - 1.5 * IQR_val
 upper_bound <- Q3 + 1.5 * IQR_val
 return(x < lower_bound | x > upper_bound)
}

numeric_column <- names(ds)[sapply(ds, is.numeric)]
for(col in numeric_column) {
 outlier <- detect_outliers(ds[[col]])
 cat("Column:", col, "- Outliers found:", sum(outlier, na.rm = TRUE), "\n")
}
```

```
Column: Age - Outliers found: 4
Column: Academic.Pressure - Outliers found: 2
Column: Study.Satisfaction - Outliers found: 0
Column: Study.Hours - Outliers found: 0
Column: Financial.Stress - Outliers found: 0
```

***Outlier Handle by instances delete and outlier check:***

```
ds_outlier_delete <- ds

for(col in numeric_column) {
 outlier <- detect_outliers(ds_outlier_delete[[col]])
 ds_outlier_delete <- ds_outlier_delete[!outlier, ]
}

numeric_column <- names(ds_outlier_delete)[sapply(ds_outlier_delete, is.numeric)]
for(col in numeric_column) {
 outlier  <-  detect_outliers(ds_outlier_delete[[col]])
 cat("Column:", col, "- Outliers found:", sum(outlier, na.rm = TRUE), "\n")
}
```

```
Column: Age - Outliers found: 0
Column: Academic.Pressure - Outliers found: 0
Column: Study.Satisfaction - Outliers found: 0
Column: Study.Hours - Outliers found: 0
Column: Financial.Stress - Outliers found: 0
```

***Outlier Handle by replace with mean and outlier check:***

```
ds_outlier_mean <- ds

for(col in numeric_column) {
 outlier <- detect_outliers(ds_outlier_mean[[col]])
 mean_val <- mean(ds_outlier_mean[[col]], na.rm = TRUE)
 ds_outlier_mean[[col]][outlier] <- mean_val
}
for(col in numeric_column) {
 outlier  <-  detect_outliers(ds_outlier_mean[[col]])
 cat("Column:", col, "- Outliers found:", sum(outlier, na.rm = TRUE), "\n")
}
```

```
Column: Age - Outliers found: 0
Column: Academic.Pressure - Outliers found: 0
Column: Study.Satisfaction - Outliers found: 0
Column: Study.Hours - Outliers found: 0
Column: Financial.Stress - Outliers found: 0
```

***Outlier Handle by replace with median in original ds, outlier check and update csv file:***

```
for(col in numeric_column) {
 outlier <- detect_outliers(ds[[col]])
 median_val <- median(ds[[col]], na.rm = TRUE)
 ds[[col]][outlier] <- median_val
}

for(col in numeric_column) {
 outlier <- detect_outliers(ds[[col]])
 cat("Column:", col, "- Outliers found:", sum(outlier, na.rm = TRUE), "\n")
}

write.csv(ds, "E:/9th semester-Summer 2025/Data
Science/Project/Midterm_Dataset_Section(E).csv", row.names = FALSE)
```

```
> for(col in numeric_column) {
+   outlier <- detect_outliers(ds[[col]])
+   cat("Column:", col, "- Outliers found:", sum(outlier, na.rm = TRUE), "\n")
+ }
Column: Age - Outliers found: 0
Column: Academic.Pressure - Outliers found: 0
Column: Study.Satisfaction - Outliers found: 0
Column: Study.Hours - Outliers found: 0
Column: Financial.Stress - Outliers found: 0
>
> write.csv(ds, "E:/9th semester-Summer 2025/Data Science/Project/Midterm_Dataset_Section(E).csv", row.names = FALSE)
```

***Code description of task 3:***

The code defined a function to detect outliers based on Q1, Q3 and IQR. Then it went through numeric columns and counted how many outliers existed. Three versions were tried: one where the rows were dropped, another where outliers were replaced with the column mean, and the main dataset where outliers were replaced with the median. Each time, the number of remaining outliers was checked to see the improvement.

## 4. Detect invalid data in the data set and use the appropriate approach to handle those values.

***Description of the task 4:***

In this task we checked for invalid or wrong values in the dataset, like spelling mistakes or values outside the allowed set. After finding them, we counted how many invalid values were in each row, and then fixed the mistakes to make the data consistent.

***Code and output:***

```
valid_values <- list(
 Gender = c("Male", "Female"),
 Have.you.ever.had.suicidal.thoughts.. = c("Yes", "No"),
 Depression = c("Yes", "No"),
 Family.History.of.Mental.Illness = c("Yes", "No"),
 Dietary.Habits = c("Moderate", "Healthy", "Unhealthy"),
 Sleep.Duration = c("7-8 hours", "5-6 hours", "More than 8 hours", "Less than 5 hours")
)


count_invalid <- function(row) {
 sum(sapply(names(valid_values), function(col) {
  !(row[[col]] %in% valid_values[[col]])
 }))
}
```

```
dm$Invalid_Count <- apply(dm, 1, count_invalid)


invalid_rows <- dm[dm$Invalid_Count > 0, ]
invalid_rows
```

```
    Gender Age Academic.Pressure Study.Satisfaction Sleep.Duration Dietary.Habits Have.you.ever.had.suicidal.thoughts.. Study.Hours
33  Maleee  32                5                   2     5-6 hours       Moderate                                  Yes          12
40 Feemale  21                3                   3     7-8 hours       Moderate                                  Yes           8
   Financial.Stress Family.History.of.Mental.Illness Depression Invalid_Count
33                3                              No        Yes             1
40                5                             Yes        Yes             1
```

***Fixing invalid data:***

```
ds$Gender<-ifelse(ds$Gender %in% c("Feemale","Female"),"Female",
          ifelse(ds$Gender %in% c("Maleee","Male"),"Male",ds$Gender))


ds$Have.you.ever.had.suicidal.thoughts..<- ifelse(ds$Have.you.ever.had.suicidal.thoughts..
%in% c("Yes","Yess"),"Yes",
          ifelse(ds$Have.you.ever.had.suicidal.thoughts.. %in% c("No","Noo"),"No",
                 ds$Have.you.ever.had.suicidal.thoughts..))
```

***Code description of task 4:***

The code first created a list of valid values for each important column. Then it checked each row and counted mismatches, which was stored as an "Invalid Count". After that, the code corrected common spelling mistakes such as "Feemale" to "Female" or "Yess" to "Yes".

## 5. Convert attributes from numeric to categorical or categorical to numeric.

***Description of the task 5:***

In this task we converted text categories into numeric values so that later analysis and modeling becomes easier. This type of encoding makes it possible to use categorical columns in machine learning algorithms.

***Code and output:***

```
ds$Gender<-factor(ds$Gender,levels = c("Male","Female"),labels=c(1,0))

ds$Depression<-factor(ds$Depression,levels = c("Yes","No"),labels=c(1,0))

ds$Have.you.ever.had.suicidal.thoughts..<-
factor(ds$Have.you.ever.had.suicidal.thoughts..,levels = c("Yes","No"),labels=c(1,0))

ds$Family.History.of.Mental.Illness<-factor(ds$Family.History.of.Mental.Illness,levels =
c("Yes","No"),labels=c(1,0))

ds$Dietary.Habits<-factor(ds$Dietary.Habits,levels =
c("Moderate","Healthy","Unhealthy"),labels=c(0.5,1,0))

ds$Sleep.Duration<-factor(ds$Sleep.Duration,levels = c("7-8 hours","5-6 hours","More than
8 hours","Less than 5 hours"),labels=c(0.75,0.25,1,0))
```

```
head(ds, 10)
```

```
> head(ds, 10)
   Gender      Age Academic.Pressure Study.Satisfaction Sleep.Duration Dietary.Habits Have.you.ever.had.suicidal.thoughts..
1       1 28.00000                2                    4          0.75            0.5                                    1
2       1 28.00000                4                    5          0.25              1                                    1
3       1 25.00000                1                    3          0.25              0                                    1
4       1 23.00000                1                    4             1              0                                    1
5       1 31.00000                1                    5             1              1                                    1
6       1 19.00000                4                    4          0.25              0                                    1
7       0 34.00000                4                    2             1            0.5                                    1
8       0 20.00000                4                    1             1              1                                    1
9       0 27.74747                1                    4             1            0.5                                    0
10      1 33.00000                4                    3             0              0                                    1
   Study.Hours Financial.Stress Family.History.of.Mental.Illness Depression
1            9                2                                1          0
2            7                1                                1          0
3           10                4                                0          1
4            7                2                                1          0
5            4                2                                1          0
6            1                4                                1          1
7            6                2                                0          1
8            3                4                                1          1
9           10                3                                0          0
10          10                1                                0          1
```

***Code description of task 5:***

The code used factor conversion with specified levels and labels. For example, Gender was converted to 1 for Male and 0 for Female. Depression, Suicidal Thoughts and Family History were converted to 1 for Yes and 0 for No. Dietary habits were encoded with values for Healthy, Moderate and Unhealthy. Sleep duration categories were also mapped to numeric values according to their ranges.

## 6. Apply the normalization method for any continuous attribute.

***Description of the task C :***

In this task we normalized numeric columns so that all values fall between 0 and 1. This was done because numeric attributes may have very different ranges, and normalization makes them comparable.

***Code and output:***

```
normalize <- function(x) {
 return((x - min(x, na.rm = TRUE)) / (max(x, na.rm = TRUE) - min(x, na.rm = TRUE)))
}

ds[numeric_column] <- lapply(ds[numeric_column], normalize)
head(ds, 10)
```

```
> head(ds, 10)
   Gender       Age Academic.Pressure Study.Satisfaction Sleep.Duration Dietary.Habits Have.you.ever.had.suicidal.thoughts..
1       1 0.6250000              0.25                0.75           0.75            0.5                                    1
2       1 0.6250000              0.75                1.00           0.25              1                                    1
3       1 0.4375000              0.00                0.50           0.25              0                                    1
4       1 0.3125000              0.00                0.75              1              0                                    1
5       1 0.8125000              0.00                1.00              1              1                                    1
6       1 0.0625000              0.75                0.75           0.25              0                                    1
7       0 1.0000000              0.75                0.25              1            0.5                                    1
8       0 0.1250000              0.75                0.00              1              1                                    1
9       0 0.6092172              0.00                0.75              1            0.5                                    0
10      1 0.9375000              0.75                0.50              0              0                                    1
   Study.Hours Financial.Stress Family.History.of.Mental.Illness Depression
1   0.75000000             0.25                                1          0
2   0.58333333             0.00                                1          0
3   0.83333333             0.75                                0          1
4   0.58333333             0.25                                1          0
5   0.33333333             0.25                                1          0
6   0.08333333             0.75                                1          1
7   0.50000000             0.25                                0          1
8   0.25000000             0.75                                1          1
9   0.83333333             0.50                                0          0
10  0.83333333             0.00                                0          1
>
```

***Code description of task C:***

The code defined a normalize function which calculates (x - min) / (max - min). Then this function was applied to all numeric columns in the dataset. After applying, the first rows of the normalized dataset were checked to confirm the changes.

## 7. Find and remove duplicate rows:

***Description of the task 7:***

In this task we tried to identify duplicate rows and remove them. Duplicates can give wrong results and bias the analysis, so it was important to get rid of them.

***Code and output:***

```
duplicates <- duplicated(dm)
sum(duplicates)


ds <- ds[!duplicates, ]
sum(duplicated(ds))
```

```
87            8                4                                  0           1
88           10                3                                  0           0
89            5                5                                  0           1
90            0                2                                  0           0
 [ reached 'max' / getOption("max.print") -- omitted 102 rows ]
dt              192 obs. of 11 variables
```

***Code description of task 7:***

The code used the duplicated function to mark rows that were exactly the same. Then these rows were removed and only unique rows were kept. After removal, another check was done to make sure no duplicates remained.

## 8. Apply some filtering methods to filter the data.

***Description of the task 8:***

In this task we applied filtering to see smaller parts of the dataset. For example, we checked rows where students were depressed and male, or where the age was higher and diet was healthy. This helps us focus on specific groups of interest.

***Code and output:***

ds_filtered <- ds[ds$Gender == 1 C ds$Depression == 1, ]

head(ds_filtered, 5)

```
> ds_filtered <- ds[ds$Gender == 1 & ds$Depression == 1, ]
> head(ds_filtered, 5)
   Gender    Age Academic.Pressure Study.Satisfaction Sleep.Duration Dietary.Habits Have.you.ever.had.suicidal.thoughts..
3       1 0.4375             0.00               0.50          0.25              0                                    1
6       1 0.0625             0.75               0.75          0.25              0                                    1
10      1 0.9375             0.75               0.50             0              0                                    1
11      1 0.8125             1.00               0.75          0.25              1                                    1
12      1 0.3750             0.25               0.00          0.75              0                                    1
   Study.Hours Financial.Stress Family.History.of.Mental.Illness Depression
3   0.83333333             0.75                                0          1
6   0.08333333             0.75                                1          1
10  0.83333333             0.00                                0          1
11  0.52763819             0.75                                0          1
12  0.91666667             1.00                                0          1
```

ds_filtered <- ds %>% filter(Age > 0.5, Gender == 1)

head(ds_filtered, 5)

```
> head(ds_filtered, 5)
  Gender    Age Academic.Pressure Study.Satisfaction Sleep.Duration Dietary.Habits Have.you.ever.had.suicidal.thoughts..
1      1 0.6250             0.25               0.75          0.75             0.5                                    1
2      1 0.6250             0.75               1.00          0.25               1                                    1
3      1 0.8125             0.00               1.00             1               1                                    1
4      1 0.9375             0.75               0.50             0               0                                    1
5      1 0.8125             1.00               0.75          0.25               1                                    1
  Study.Hours Financial.Stress Family.History.of.Mental.Illness Depression
1   0.7500000             0.25                                1          0
2   0.5833333             0.00                                1          0
3   0.3333333             0.25                                1          0
4   0.8333333             0.00                                0          1
5   0.5276382             0.75                                0          1
```

ds_filtered <- subset(ds, Age > 0.5 C Dietary.Habits == 1)

head(ds_filtered, 5)

```
> head(ds_filtered, 5)
   Gender    Age Academic.Pressure Study.Satisfaction Sleep.Duration Dietary.Habits Have.you.ever.had.suicidal.thoughts..
2       1 0.6250             0.75               1.00          0.25              1                                    1
5       1 0.8125             0.00               1.00             1              1                                    1
11      1 0.8125             1.00               0.75          0.25              1                                    1
16      1 0.6250             1.00               0.50          0.25              1                                    1
26      1 0.8125             0.25               0.25          0.75              1                                    0
   Study.Hours Financial.Stress Family.History.of.Mental.Illness Depression
2    0.5833333             0.00                                1          0
5    0.3333333             0.25                                1          0
11   0.5276382             0.75                                0          1
16   0.6666667             0.50                                1          1
26   0.1666667             0.75                                1          0
```

***Code description of task 8:***

*The code used filtering conditions on the dataset. It selected rows based on logical conditions such as "Gender is 1 and Depression is 1" or "Age greater than a value and Diet is Healthy". The filtered results were then shown using the head function.*

## G. Convert the imbalanced data set into the balanced data set:

***Description of the task 3:***

This task was about class imbalance. We checked the target column Depression and found that one class had more rows than the other. To solve this, we oversampled the minority class so both classes have equal numbers.

### *Code and output:*

***Find imbalanced row:***

```
cat_cols <- names(ds)[sapply(dm, function(x) is.factor(x) || is.character(x))]

for (col in cat_cols) {
 counts <- table(ds[[col]])
 ratio <- ifelse(length(counts) > 1, max(counts)/min(counts), NA)
 cat("Column:", col, "\n")
 print(counts)
 cat("Imbalance ratio (max/min):", round(ratio,2), "\n\n")
}
```

```
Column: Gender

  1   0
113  88
Imbalance ratio (max/min): 1.28

Column: Sleep.Duration

0.75 0.25    1    0
  51   47   55   48
Imbalance ratio (max/min): 1.17

Column: Dietary.Habits

0.5   1   0
 66  66  69
Imbalance ratio (max/min): 1.05

Column: Have.you.ever.had.suicidal.thoughts..

  1   0
107  94
Imbalance ratio (max/min): 1.14

Column: Family.History.of.Mental.Illness

  1   0
 93 108
Imbalance ratio (max/min): 1.16

Column: Depression

  1   0
 83 118
Imbalance ratio (max/min): 1.42
```

Here **Depression** is imbalanced because the number of 1s (83) and 0s (118) differs significantly, giving an imbalance ratio of 1.42, whereas the other columns (are fairly balanced because their max/min ratios are close to 1 (1.05–1.28).

***Convert into imbalance to balance by over sampling:***

```
minority <- ds %>% filter(Depression == 1)
majority <- ds %>% filter(Depression == 0)

set.seed(123)
minority_oversampled <- minority %>%
 sample_n(size = nrow(majority), replace = TRUE)
ds_balanced <- bind_rows(majority, minority_oversampled)
ds_balanced <- ds_balanced[sample(nrow(ds_balanced)), ]
table(ds_balanced$Depression)
dim(ds_balanced)
```

```
> table(ds_balanced$Depression)

  1   0
118 118
> dim(ds_balanced)
[1] 236  11
```

***Code description of task 3:***

The code first counted how many rows were in each category of Depression. Then it separated the minority and majority classes. Using oversampling with replacement, the minority rows were repeated until they matched the majority. Finally, the two sets were combined and shuffled to get a balanced dataset.

## 10. Split the dataset for Training and Testing:

***Description of the task 10:***

In this task we divided the dataset into training and testing parts. This is necessary because we need one part to train models and another part to test their performance.

***Code and output:***

```
set.seed(123)
index <- createDataPartition(ds_balanced$Depression, p = 0.8, list = FALSE)
train_data <- ds_balanced[index, ]
test_data <- ds_balanced[-index, ]
cat("Training Data Rows:", nrow(train_data), "\n")
cat("Testing Data Rows:", nrow(test_data), "\n")
```

```
> cat("Training Data Rows:", nrow(train_data), "\n")
Training Data Rows: 190
> cat("Testing Data Rows:", nrow(test_data), "\n")
Testing Data Rows: 46
```

***Code description of task 10:***

The code set a seed to keep results consistent. Then it used the createDataPartition function to split the rows into 80% for training and 20% for testing while keeping the class balance the same. At the end, the size of both sets was shown.

11. **Calculate descriptive statistics and interpret the results for the following numerical variables between two target classes (depression = Yes and depression = No)**
    - ➢ **Study Hours**
    - ➢ **Age**

    ***Description of the task 7:***

    Here we compared Study Hours and Age for students with and without depression. The goal was to see if there are differences in these variables between the two groups.

    ***Code and output:***

```
install.packages("utf8")
library(dplyr)

study_stats <- ds_balanced %>%
 group_by(Depression) %>%
 summarise(
  Count = n(),
  Mean = mean(Study.Hours, na.rm = TRUE),
```

```
  Median = median(Study.Hours, na.rm = TRUE),
  SD = sd(Study.Hours, na.rm = TRUE),
  Min = min(Study.Hours, na.rm = TRUE),
  Max = max(Study.Hours, na.rm = TRUE),
  Q1 = quantile(Study.Hours, 0.25, na.rm = TRUE),
  Q3 = quantile(Study.Hours, 0.75, na.rm = TRUE)
 )

age_stats <- ds_balanced %>%
 group_by(Depression) %>%
 summarise(
  Count = n(),
  Mean = mean(Age, na.rm = TRUE),
  Median = median(Age, na.rm = TRUE),
  SD = sd(Age, na.rm = TRUE),
  Min = min(Age, na.rm = TRUE),
  Max = max(Age, na.rm = TRUE),
  Q1 = quantile(Age, 0.25, na.rm = TRUE),
  Q3 = quantile(Age, 0.75, na.rm = TRUE)
 )

study_stats
age_stats
```

```
> study_stats
# A tibble: 2 x 9
  Depression Count  Mean Median    SD   Min   Max    Q1    Q3
  <fct>      <int> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 1            118 0.557  0.583 0.322     0     1  0.25 0.833
2 0            118 0.506  0.555 0.315     0     1  0.25 0.75
> age_stats
# A tibble: 2 x 9
  Depression Count  Mean Median    SD   Min   Max    Q1    Q3
  <fct>      <int> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 1            118 0.438  0.375 0.293     0     1 0.188 0.688
2 0            118 0.561  0.562 0.285     0     1 0.375 0.812
```

***Code description of task 11:***

The code grouped the dataset by the Depression column and then calculated mean, median, standard deviation, minimum, maximum, and quartiles for both Study Hours and Age. The summaries were printed for easy comparison between groups.

12. **Compare average sleep duration between students with and without depression.**

***Description of the task 12:***

This task was about checking the relationship between sleep duration and depression. We compared the average sleep between depressed and non-depressed students.

***Code and output:***

```
sleep_stats <- ds_balanced %>%
 group_by(Depression) %>%
 summarise(
  Count = n(),
  Mean_Sleep = mean(as.numeric(Sleep.Duration), na.rm = TRUE),
  Median_Sleep = median(as.numeric(Sleep.Duration), na.rm = TRUE),
  SD_Sleep = sd(as.numeric(Sleep.Duration), na.rm = TRUE),
  Min_Sleep = min(as.numeric(Sleep.Duration), na.rm = TRUE),
  Max_Sleep = max(as.numeric(Sleep.Duration), na.rm = TRUE),
  Q1_Sleep = quantile(as.numeric(Sleep.Duration), 0.25, na.rm = TRUE),
  Q3_Sleep = quantile(as.numeric(Sleep.Duration), 0.75, na.rm = TRUE)
 )
 sleep_stats
```

```
> sleep_stats
# A tibble: 2 x 9
  Depression Count Mean_Sleep Median_Sleep SD_Sleep Min_Sleep Max_Sleep Q1_Sleep Q3_Sleep
  <fct>      <int>      <dbl>        <dbl>    <dbl>     <dbl>     <dbl>    <dbl>    <dbl>
1 1            118       2.34            2     1.06         1         4     1.25     3
2 0            118       2.53            3     1.14         1         4     1         3.75
>
```

***Code description of task 12:***

The code converted the sleep duration into numeric values. Then it grouped the dataset by Depression and calculated different summary values such as mean, median, standard deviation, minimum, maximum, and quartiles. These summaries showed the sleep patterns for each group.

## 13. Compare spread in study hours for students with different levels of study satisfaction.

***Description of the task 13:***

In the final task we studied how study hours are distributed across different levels of study satisfaction. This helped us understand how study satisfaction might be connected to study time.

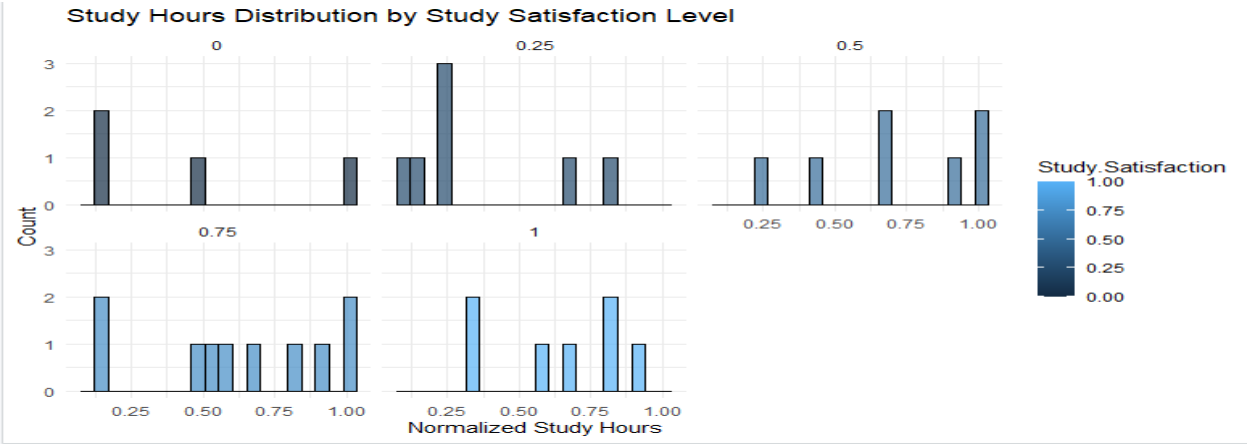
***Code and output:***

```
study_spread <- ds_balanced %>%
 group_by(Study.Satisfaction)   %>%
 summarise(
  Count = n(),
  Mean_Study_Hours = mean(Study.Hours, na.rm = TRUE),
  Median_Study_Hours = median(Study.Hours, na.rm = TRUE),
  SD_Study_Hours = sd(Study.Hours, na.rm = TRUE),
  Min_Study_Hours = min(Study.Hours, na.rm = TRUE),
  Max_Study_Hours = max(Study.Hours, na.rm = TRUE),
  Q1_Study_Hours = quantile(Study.Hours, 0.25, na.rm = TRUE),
  Q3_Study_Hours = quantile(Study.Hours, 0.75, na.rm = TRUE)
 )

study_spread
```

```
study_spread
A tibble: 5 x 9
Study.Satisfaction Count Mean_Study_Hours Median_Study_Hours SD_Study_Hours Min_Study_Hours Max_Study_Hours Q1_Study_Hours
             <dbl> <int>           <dbl>              <dbl>          <dbl>           <dbl>           <dbl>          <dbl>
              0       36           0.597              0.667          0.296               0               1          0.312
              0.25    58           0.488              0.417          0.355               0               1          0.167
              0.5     46           0.529              0.5            0.330               0               1          0.25
              0.75    53           0.598              0.667          0.306               0               1          0.417
              1       43           0.457              0.417          0.272               0               1          0.25
```

```
ggplot(ds_filtered, aes(x = Study.Hours, fill = Study.Satisfaction)) +
 geom_histogram(bins = 20, color = "black", alpha = 0.7) +
 facet_wrap(~Study.Satisfaction) +
 labs(title = "Study Hours Distribution by Study Satisfaction Level",
    x = "Normalized Study Hours",
    y = "Count") +
```




## ***Code description of task 13:***

The code grouped the dataset by Study Satisfaction and calculated summary values for Study Hours such as mean, median, and standard deviation. It also created histograms for each satisfaction level using ggplot. This gave both numerical and visual understanding of the spread of study hours.

## head(ds, 20)

```
> head(ds,20)
   Gender       Age Academic.Pressure Study.Satisfaction Sleep.Duration Dietary.Habits Have.you.ever.had.suicidal.thoughts..
1       1 0.6250000              0.25                0.75           0.75            0.5                                    1
2       1 0.6250000              0.75                1.00           0.25              1                                    1
3       1 0.4375000              0.00                0.50           0.25              0                                    1
4       1 0.3125000              0.00                0.75              1              0                                    1
5       1 0.8125000              0.00                1.00              1              1                                    1
6       1 0.0625000              0.75                0.75           0.25              0                                    1
7       0 1.0000000              0.75                0.25              1            0.5                                    1
8       0 0.1250000              0.75                0.00              1              1                                    1
9       0 0.6092172              0.00                0.75              1            0.5                                    0
10      1 0.9375000              0.75                0.50              0              0                                    1
11      1 0.8125000              1.00                0.75           0.25              1                                    1
12      1 0.3750000              0.25                0.00           0.75              0                                    1
13      0 0.3125000              1.00                1.00              1              0                                    1
14      1 0.4375000              0.00                0.00           0.25            0.5                                    1
15      1 0.1875000              1.00                0.00              1              0                                    1
16      1 0.6250000              1.00                0.50           0.25              1                                    1
17      1 0.3125000              1.00                0.25              1            0.5                                    0
18      0 0.3125000              0.00                0.50              1              1                                    1
19      0 0.1250000              1.00                1.00              1              0                                    1
20      1 0.6875000              0.75                0.50              1              0                                    1
   Study.Hours Financial.Stress Family.History.of.Mental.Illness Depression
1   0.75000000              0.25                                1          0
2   0.58333333              0.00                                1          0
3   0.83333333              0.75                                0          1
4   0.58333333              0.25                                1          0
5   0.33333333              0.25                                1          0
6   0.08333333              0.75                                1          1
7   0.50000000              0.25                                0          1
8   0.25000000              0.75                                1          1
9   0.83333333              0.50                                0          0
10  0.83333333              0.00                                0          1
11  0.52763819              0.75                                0          1
12  0.91666667              1.00                                0          1
13  0.16666667              0.00                                1          0
14  1.00000000              0.50                                1          1
15  0.25000000              1.00                                1          1
16  0.66666667              0.50                                1          1
17  0.52763819              0.75                                0          0
18  0.00000000              0.50                                0          0
19  0.16666667              1.00                                0          1
20  0.08333333              0.50                                0          1
```

## Update CSV file:

write.csv(ds, "E:/9th semester-Summer 2025/Data Science/Project1/Midterm_Dataset_Section(E).csv", row.names = FALSE)

| | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Gender | Age | Academic | Study.Sati | Sleep.Dur | Dietary.Ha | Have.you. | Study.Hou | Financial. | Family.His | Depression |
| 2 | 1 | 0.625 | 0.25 | 0.75 | 0.75 | 0.5 | 1 | 0.75 | 0.25 | 1 | 0 |
| 3 | 1 | 0.625 | 0.75 | 1 | 0.25 | 1 | 1 | 0.583333 | 0 | 1 | 0 |
| 4 | 1 | 0.4375 | 0 | 0.5 | 0.25 | 0 | 1 | 0.833333 | 0.75 | 0 | 1 |
| 5 | 1 | 0.3125 | 0 | 0.75 | 1 | 0 | 1 | 0.583333 | 0.25 | 1 | 0 |
| 6 | 1 | 0.8125 | 0 | 1 | 1 | 1 | 1 | 0.333333 | 0.25 | 1 | 0 |
| 7 | 1 | 0.0625 | 0.75 | 0.75 | 0.25 | 0 | 1 | 0.083333 | 0.75 | 1 | 1 |
| 8 | 0 | 1 | 0.75 | 0.25 | 1 | 0.5 | 1 | 0.5 | 0.25 | 0 | 1 |
| 9 | 0 | 0.125 | 0.75 | 0 | 1 | 1 | 1 | 0.25 | 0.75 | 1 | 1 |
| 10 | 0 | 0.609217 | 0 | 0.75 | 1 | 0.5 | 0 | 0.833333 | 0.5 | 0 | 0 |
| 11 | 1 | 0.9375 | 0.75 | 0.5 | 0 | 0 | 1 | 0.833333 | 0 | 0 | 1 |
| 12 | 1 | 0.8125 | 1 | 0.75 | 0.25 | 1 | 1 | 0.527638 | 0.75 | 0 | 1 |
| 13 | 1 | 0.375 | 0.25 | 0 | 0.75 | 0 | 1 | 0.916667 | 1 | 0 | 1 |
| 14 | 0 | 0.3125 | 1 | 1 | 1 | 0 | 1 | 0.166667 | 0 | 1 | 0 |
| 15 | 1 | 0.4375 | 0 | 0 | 0.25 | 0.5 | 1 | 1 | 0.5 | 1 | 1 |
| 16 | 1 | 0.1875 | 1 | 0 | 1 | 0 | 1 | 0.25 | 1 | 1 | 1 |
| 17 | 1 | 0.625 | 1 | 0.5 | 0.25 | 1 | 1 | 0.666667 | 0.5 | 1 | 1 |
| 18 | 1 | 0.3125 | 1 | 0.25 | 1 | 0.5 | 0 | 0.527638 | 0.75 | 0 | 0 |
| 19 | 0 | 0.3125 | 0 | 0.5 | 1 | 1 | 1 | 0 | 0.5 | 0 | 0 |
| 20 | 0 | 0.125 | 1 | 1 | 1 | 0 | 1 | 0.166667 | 1 | 0 | 1 |

# Project Code

```
install.packages("utf8")

library(dplyr)

library(caret)

library(ggplot2)


ds <- read.csv("E:/9th semester-Summer 2025/Data Science/Project1/Midterm_Dataset_Section(E).csv",na = c("NA", ""),stringsAsFactors = FALSE, fileEncoding = "UTF-8")

head(ds)

summary(ds)

str(ds)


colSums(is.na(ds))

ds_delete <- na.omit(ds)

colSums(is.na(ds_delete))

head(ds_delete, 20)



for (col in names(ds)) {
 if (is.numeric(ds[[col]])) {
  mean_val <- mean(ds[[col]], na.rm = TRUE)
  ds[[col]][is.na(ds[[col]])] <- mean_val
 }
}

colSums(is.na(ds))
```

```
for (col in names(ds)) {
 if (is.character(ds[[col]]) || is.factor(ds[[col]]))
 {
  mode_val <- names(which.max(table(ds[[col]])))
  ds[[col]][is.na(ds[[col]])] <- mode_val
 }
}
colSums(is.na(ds))

missing_counts <- colSums(is.na(ds))

barplot(missing_counts,
     main = "Missing Values per Column",
     xlab = "Columns",
     ylab = "Number of Missing Values",
     col = "skyblue",
     las = 2)



detect_outliers <- function(x) {
 Q1 <- quantile(x, 0.25, na.rm = TRUE)
 Q3 <- quantile(x, 0.75, na.rm = TRUE)
 IQR_val <- Q3 - Q1
 lower_bound <- Q1 - 1.5 * IQR_val
 upper_bound <- Q3 + 1.5 * IQR_val
 return(x < lower_bound | x > upper_bound)
}
```

```
numeric_column <- names(ds)[sapply(ds, is.numeric)]
for(col in numeric_column) {
 outlier <- detect_outliers(ds[[col]])
 cat("Column:", col, "- Outliers found:", sum(outlier, na.rm = TRUE), "\n")
}



ds_outlier_delete <- ds

for(col in numeric_column) {
 outlier <- detect_outliers(ds_outlier_delete[[col]])
 ds_outlier_delete <- ds_outlier_delete[!outlier, ]
}

numeric_column <- names(ds_outlier_delete)[sapply(ds_outlier_delete, is.numeric)]
for(col in numeric_column) {
 outlier  <-  detect_outliers(ds_outlier_delete[[col]])
 cat("Column:", col, "- Outliers found:", sum(outlier, na.rm = TRUE), "\n")
}


ds_outlier_mean <- ds

for(col in numeric_column) {
 outlier  <-  detect_outliers(ds_outlier_mean[[col]])
```

```
 mean_val <- mean(ds_outlier_mean[[col]], na.rm = TRUE)
 ds_outlier_mean[[col]][outlier] <- mean_val
}

for(col in numeric_column) {
 outlier  <-  detect_outliers(ds_outlier_mean[[col]])
 cat("Column:", col, "- Outliers found:", sum(outlier, na.rm = TRUE), "\n")
}



for(col in numeric_column) {
 outlier <- detect_outliers(ds[[col]])
 median_val <- median(ds[[col]], na.rm = TRUE)
 ds[[col]][outlier] <- median_val
}

for(col in numeric_column) {
 outlier <- detect_outliers(ds[[col]])
 cat("Column:", col, "- Outliers found:", sum(outlier, na.rm = TRUE), "\n")
}

valid_values <- list(
 Gender = c("Male", "Female"),
 Have.you.ever.had.suicidal.thoughts.. = c("Yes", "No"),
 Depression = c("Yes", "No"),
 Family.History.of.Mental.Illness = c("Yes", "No"),
 Dietary.Habits = c("Moderate", "Healthy", "Unhealthy"),
```

```
 Sleep.Duration = c("7-8 hours", "5-6 hours", "More than 8 hours", "Less than 5 hours")
)

count_invalid <- function(row) {
 sum(sapply(names(valid_values), function(col) {
  !(row[[col]] %in% valid_values[[col]])
 }))
}

ds$Invalid_Count <- apply(ds, 1, count_invalid)

invalid_rows <- ds[dm$Invalid_Count > 0, ]
invalid_rows

ds$Gender<-ifelse(ds$Gender %in% c("Feemale","Female"),"Female",
            ifelse(ds$Gender %in% c("Maleee","Male"),"Male",ds$Gender))

ds$Gender<-factor(ds$Gender,levels = c("Male","Female"),labels=c(1,0))
ds$Depression<-factor(ds$Depression,levels = c("Yes","No"),labels=c(1,0))
ds$Have.you.ever.had.suicidal.thoughts..<-factor(ds$Have.you.ever.had.suicidal.thoughts..,levels = c("Yes","No"),labels=c(1,0))
ds$Family.History.of.Mental.Illness<-factor(ds$Family.History.of.Mental.Illness,levels = c("Yes","No"),labels=c(1,0))
ds$Dietary.Habits<-factor(ds$Dietary.Habits,levels  = c("Moderate","Healthy","Unhealthy"),labels=c(0.5,1,0))
ds$Sleep.Duration<-factor(ds$Sleep.Duration,levels = c("7-8 hours","5-6 hours","More than 8 hours","Less than 5 hours"),labels=c(0.75,0.25,1,0))

head(ds, 10)
```

```
normalize <- function(x) {
 return((x - min(x, na.rm = TRUE)) / (max(x, na.rm = TRUE) - min(x, na.rm = TRUE)))
}

ds[numeric_column] <- lapply(ds[numeric_column], normalize)
head(ds, 10)

duplicates <- duplicated(dm)
sum(duplicates)

ds <- ds[!duplicates, ]
sum(duplicated(ds))

ds_filtered <- ds[ds$Gender == 1 C ds$Depression == 1, ]
head(ds_filtered, 5)


ds_filtered <- ds %>% filter(Age > 0.5, Gender == 1)
head(ds_filtered, 5)

ds_filtered <- subset(ds, Age > 0.5 C Dietary.Habits == 1)
head(ds_filtered, 5)
```

```
cat_cols <- names(ds)[sapply(dm, function(x) is.factor(x) || is.character(x))]

for (col in cat_cols) {
 counts <- table(ds[[col]])
 ratio <- ifelse(length(counts) > 1, max(counts)/min(counts), NA)
 cat("Column:", col, "\n")
 print(counts)
 cat("Imbalance ratio (max/min):", round(ratio,2), "\n)
}



minority <- ds %>% filter(Depression == 1)
majority <- ds %>% filter(Depression == 0)

set.seed(123)
minority_oversampled <- minority %>%
 sample_n(size = nrow(majority), replace = TRUE)
ds_balanced <- bind_rows(majority, minority_oversampled)
ds_balanced <- ds_balanced[sample(nrow(ds_balanced)), ]
table(ds_balanced$Depression)
dim(ds_balanced)
```

```
set.seed(123)

index <- createDataPartition(ds_balanced$Depression, p = 0.8, list = FALSE)

train_data <- ds_balanced[index, ]
test_data <- ds_balanced[-index, ]

cat("Training Data Rows:", nrow(train_data), "\n")
cat("Testing Data Rows:", nrow(test_data), "\n")



install.packages("utf8")
library(dplyr)

study_stats <- ds_balanced %>%
 group_by(Depression) %>%
 summarise(
  Count = n(),
  Mean = mean(Study.Hours, na.rm = TRUE),
  Median = median(Study.Hours, na.rm = TRUE),
  SD = sd(Study.Hours, na.rm = TRUE),
  Min = min(Study.Hours, na.rm = TRUE),
  Max = max(Study.Hours, na.rm = TRUE),
  Q1 = quantile(Study.Hours, 0.25, na.rm = TRUE),
  Q3 = quantile(Study.Hours, 0.75, na.rm = TRUE)
 )
```

```
age_stats <- ds_balanced %>%
 group_by(Depression) %>%
 summarise(
  Count = n(),
  Mean = mean(Age, na.rm = TRUE),
  Median = median(Age, na.rm = TRUE),
  SD = sd(Age, na.rm = TRUE),
  Min = min(Age, na.rm = TRUE),
  Max = max(Age, na.rm = TRUE),
  Q1 = quantile(Age, 0.25, na.rm = TRUE),
  Q3 = quantile(Age, 0.75, na.rm = TRUE)
 )


study_stats
age_stats

library(dplyr)


sleep_stats <- ds_balanced %>%
 group_by(Depression) %>%
 summarise(
  Count = n(),
  Mean_Sleep = mean(as.numeric(Sleep.Duration), na.rm = TRUE),
```

```
  Median_Sleep = median(as.numeric(Sleep.Duration), na.rm = TRUE),
  SD_Sleep = sd(as.numeric(Sleep.Duration), na.rm = TRUE),
  Min_Sleep = min(as.numeric(Sleep.Duration), na.rm = TRUE),
  Max_Sleep = max(as.numeric(Sleep.Duration), na.rm = TRUE),
  Q1_Sleep = quantile(as.numeric(Sleep.Duration), 0.25, na.rm = TRUE),
  Q3_Sleep = quantile(as.numeric(Sleep.Duration), 0.75, na.rm = TRUE)
 )

sleep_stats




library(dplyr)

study_spread <- ds_balanced %>%
 group_by(Study.Satisfaction)  %>%
 summarise(
  Count = n(),
  Mean_Study_Hours = mean(Study.Hours, na.rm = TRUE),
  Median_Study_Hours = median(Study.Hours, na.rm = TRUE),
  SD_Study_Hours = sd(Study.Hours, na.rm = TRUE),
  Min_Study_Hours = min(Study.Hours, na.rm = TRUE),
  Max_Study_Hours = max(Study.Hours, na.rm = TRUE),
  Q1_Study_Hours = quantile(Study.Hours, 0.25, na.rm = TRUE),
  Q3_Study_Hours = quantile(Study.Hours, 0.75, na.rm = TRUE)
 )
```

```
study_spread


library(ggplot2)
ggplot(ds_filtered, aes(x = Study.Hours, fill = Study.Satisfaction)) +
 geom_histogram(bins = 20, color = "black", alpha = 0.7) +
 facet_wrap(~Study.Satisfaction) +
 labs(title = "Study Hours Distribution by Study Satisfaction Level",
    x = "Normalized Study Hours",
    y = "Count") +
 theme_minimal()


head(ds,20)
 write.csv(ds, "E:/9th semester-Summer 2025/Data
Science/Project1/Midterm_Dataset_Section(E).csv", row.names = FALSE)
```