AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Faculty of Science and Technology

Project Cover Page

Assignment Title:	Introduction to Data Science Midterm Project		
Assignment No:	N/A	Date of Submission:	14 December 2024
Course Title:	INTRODUCTION TO DATA SCIENCE		
Course Code:	CSC4180	Section:	B
Semester:	Fall 2024-25	Course Teacher:	**TOHEDUL ISLAM**

Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgement is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaborationhas been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the Faculty for review and comparison, including review by external examiners.
7. I/we understand thatPlagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a formofcheatingandisaveryseriousacademicoffencethatmayleadtoexpulsionfromtheUniversity. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of them arterial used is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy my/our work.

* *Student(s) must complete all details except the faculty use part.*
** Please submit all assignments to your course teacher or the office of the concerned teacher.

Group Name/No.: 08

No	Name	ID	Program	Signature
1	DEVDOOT PARIAL	21-45061-2	BSc [CSE]	
2	ABDULLAH AL SHAHRIAR	21-44760-1	BSc [CSE]	
3	MD. SABBIR HOSSAIN KHAN	21-45256-2	BSc [CSE]	
4	GALIB HASAN ALVEE	21-44549-1	BSc [CSE]	

Faculty use only		
FACULTYCOMMENTS	**Marks Obtained**	
	Total Marks	

Dataset Description:

This dataset is called "Depression student dataset" as mentioned in kaggle. This dataset is of depressed students based on various factors. Those factors are demographic, academic, and lifestyle factors. It includes attributes such as gender, age, academic pressure, study satisfaction, sleep duration, dietary habits etc. It shows a connection between various factors and the mental wellbeing of students. Here the class variable is "depression". So, it can help us to detect depression status. Also, it can help identify mental health risks and suggest strategies to improve wellbeing based on those factors after analyzing the dataset.

Attributes:

Gender: Specifies the gender of the students. There are two genders in the dataset and they are male and female. It is a categorical attribute

Age: This indicates the age of the students. This is a numeric attribute

Academic Pressure: Levels of academic pressure experienced by the students. There are 5 levels, starting from 1 to maximum 5. This is a numeric attribute

Study Satisfaction: Satisfaction levels of the students with their studies. Here in the dataset, it is indicated in a range of 1 to 5. This is a numeric attribute

Sleep Duration: This indicates the hours of sleep students get per night. They are in 4 categories and those are "Less than 5 hours", "5-6 hours", "7-8 hours", "More than 8 hours". It is a categorical attribute.

Dietary Habits: Classification of dietary habits. They are "Unhealthy", "Moderate", "Healthy". It is a categorical attribute

Have you ever had suicidal thoughts?: Indicates whether students have had suicidal thoughts. The values are in Yes and No. It is a categorical attribute

Study Hours: This attribute indicates the time spent studying daily by the students. Value ranging from 0 to 12 in hours. This is a numeric attribute.

Financial Stress: Level of financial stress experienced. Ranging from 1 to 5. This is a numerical attribute.

Family History of Mental Illness: Indicates if there's a family history of mental illness of the students. The value is in Yes and No. This is a categorical attribute

Depression: Indicates whether the student is depressed or not and the value is in Yes and No. This a categorical column

Purpose:

The main goal of this dataset is to study the relationship between mental health and various demographic, academic, and lifestyle factors among students. By identifying trends and patterns, the dataset aims to help detect depression and assess mental health risks. Insights derived from this analysis can guide the development of preventive measures and interventions to promote student well-being.

Project Overview

This project explores the "Depression Student Dataset", analyzing how demographic, academic, and lifestyle factors impact student mental health. Key objectives include examining mental health patterns (e.g., links between sleep, diet, academic workload, and depression), identifying significant risk factors like family history of mental illness or suicidal thoughts, and proposing actionable interventions to improve mental well-being. The dataset supports descriptive and predictive analyses, using attributes such as gender, age, study hours, academic pressure, and dietary habits. Applications include tailoring preventive measures, providing academic support, and promoting healthy lifestyles to address mental health challenges and enhance student outcomes.

Data Preprocessing:

1. Importing Dataset:

Coding Part:

```
install.packages("openxlsx")
library(openxlsx)
mydata <- read.xlsx("C:/Users/sazin/Downloads/Midterm_Dataset.xlsx")
mydata
```

Output Part:

	Gender	Age	Academic_Pressure	Study_Satisfaction	Sleep_Duration	Dietary_Habits	Have_you_ever_had_suicidal_thoughts.?	Study_Hours	Financial_Stress	Family._History_of_Mental_Illness	Depression
1	Male	28	2	4	7-8 hours	Moderate	Yes	9	2	Yes	No
2	Male	28	4	5	5-6 hours	Healthy	Yes	7	1	Yes	No
3	Male	25	1	3	5-6 hours	Unhealthy	Yes	10	4	No	Yes
4	Male	23	1	4	More than 8 hours	Unhealthy	Yes	7	2	Yes	No
5	NA	31	1	5	More than 8 hours	Healthy	Yes	4	2	Yes	No
6	Male	19	4	4	5-6 hours	Unhealthy	Yes	1	4	Yes	Yes
7	Female	34	4	2	NA	Moderate	Yes	6	2	No	Yes
8	Female	20	4	1	More than 8 hours	Healthy	Yes	3	4	Yes	Yes
9	Female	NA	1	4	More than 8 hours	Moderate	No	10	3	No	No
10	Male	33	4	3	Less than 5 hours	Unhealthy	Yes	10	1	No	Yes
11	NA	31	5	4	5-6 hours	Healthy	Yes	NA	4	No	Yes
12	Male	24	2	1	7-8 hours	Unhealthy	Yes	11	5	No	Yes
13	Female	23	5	5	NA	Unhealthy	Yes	2	1	Yes	NA
14	Male	25	1	1	5-6 hours	Moderate	Yes	12	3	Yes	Yes
15	Male	21	5	1	More than 8 hours	Unhealthy	Yess	3	5	Yes	Yes
16	Male	28	5	3	5-6 hours	Healthy	Yes	8	3	Yes	Yes
17	Male	23	5	2	More than 8 hours	Moderate	No	NA	4	No	NA
18	Female	23	1	3	NA	Healthy	Yes	0	3	No	No
19	Female	20	5	5	More than 8 hours	Unhealthy	Yes	2	5	No	Yes
20	Male	29	4	3	More than 8 hours	Unhealthy	Yes	1	3	No	Yes
21	Male	31	2	3	More than 8 hours	Unhealthy	No	3	3	Yes	No
22	Male	24	3	4	More than 8 hours	Healthy	Yes	1	3	No	No
23	NA	31	2	4	More than 8 hours	Unhealthy	Noo	10	1	No	No
24	Female	33	3	2	7-8 hours	Moderate	No	11	5	Yes	No
25	Female	33	2	3	7-8 hours	Moderate	Yes	12	5	Yes	NA
26	Male	31	2	2	7-8 hours	Healthy	No	2	4	Yes	No
27	Male	30	3	4	7-8 hours	Moderate	Yes	0	2	Yes	No
28	Male	21	5	3	7-8 hours	Unhealthy	No	6	4	Yes	Yes
29	Female	29	3	5	Less than 5 hours	Moderate	Yes	4	3	Yes	No
30	Female	34	3	4	Less than 5 hours	Unhealthy	Yes	12	1	Yes	No
31	Female	20	3	2	More than 8 hours	Healthy	No	2	2	No	No

Showing 1 to 31 of 201 entries, 11 total columns

Explanation Part: The openxlsx package was used to import data from an Excel file into R. First, the package was installed and loaded. Then, the read.xlsx() function was used to read the file "Midterm_Dataset.xlsx" from the specified location into a data frame called mydata.

2. Structure of the Dataset:

Coding Part:

```
str(mydata)
```

Output Part:

```
> str(mydata)
'data.frame':	201 obs. of  11 variables:
 $ Gender                               : chr  "Male" "Male" "Male" "Male" ...
 $ Age                                  : num  28 28 25 23 31 19 34 20 NA 33 ...
 $ Academic_Pressure                    : num  2 4 1 1 1 4 4 4 1 4 ...
 $ Study_Satisfaction                   : num  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                          : num  9 7 10 7 4 1 6 3 10 10 ...
 $ Financial_Stress                     : num  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" ...
>
```

Explanation Part: The str(mydata) function provides a detailed structure of the mydata dataset. It displays the type of each column (e.g., numeric, factor, character), the number of observations (rows), and the first few values of each column. This helps in understanding the data types and gives an overview of the dataset's organization.

3. Descriptive Statistics using summary Function:

Coding Part:

```
summary(mydata)
```

Output Part:

```
> summary(mydata)
    Gender               Age         Academic_Pressure Study_Satisfaction Sleep_Duration      Dietary_Habits
 Length:201         Min.   : 18.00   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   : 28.25   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
>
```

Explanation Part: The summary(mydata) function generates a statistical summary of the dataset, providing key metrics for numeric columns, such as the minimum, median, mean, and maximum values, along with quartiles. For categorical columns, it shows the count of each category. This summary helps in quickly understanding the distribution and characteristics of the data.

<u>**4. Visualizing Missing Values on a Graph:**</u>

4.1 Detecting Missing Values Column Wise:

Code Part:

```
colSums(is.na(mydata))
sum(is.na(mydata))
```

Output Part:

```
> colSums(is.na(mydata))
                          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
> sum(is.na(mydata))
[1] 14
> 
```

Explanation Part: The colSums(is.na(mydata)) command calculates the number of missing values (NA) in each column of the dataset mydata**.** As we can see that there are total 5 attributes (Gender, Age, Sleep Duration, Study Hours, Depression) having missing values with the amount of missing values. sum(is.na(mydata) indicates the total number of missing values.

4.2 Detecting Row Indices of Missing Values in Each Column:

Code Part:

```
lapply(mydata, function(col) which(is.na(col)))
```

Output Part:

```
> lapply(mydata, function(col) which(is.na(col)))
$Gender
[1]  5 11 23

$Age
[1]  9 37 54

$Academic_Pressure
integer(0)

$Study_Satisfaction
integer(0)

$Sleep_Duration
[1]  7 13 18

$Dietary_Habits
integer(0)

$`Have_you_ever_had_suicidal_thoughts.?`
integer(0)

$Study_Hours
[1] 11 17

$Financial_Stress
integer(0)

$Family._History_of_Mental_Illness
integer(0)

$Depression
[1] 13 17 25

> |
```

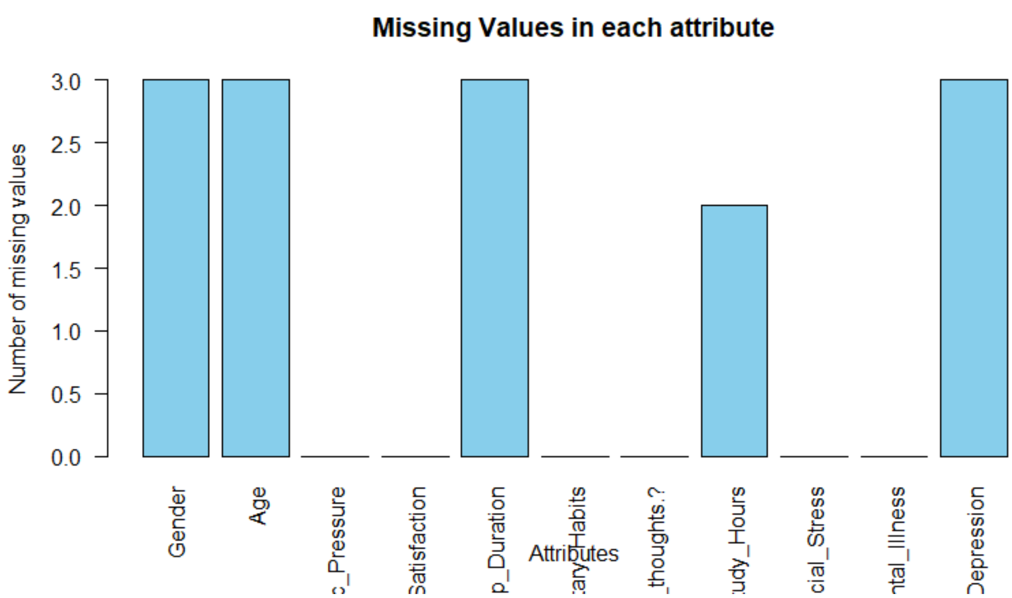
Explanation Part: The command lapply (mydata, function(col) which(is.na(col))) identifies the row indices of missing values (NA) in each column of the dataset mydata**.**

4.3 Bar Plot of Missing Value (Graph):

Code Part:

```
missing_counts <- colSums(is.na(mydata))
barplot(missing_counts,
        main = "Missing Values in each attribute",
        xlab = "Attributes",
        ylab = "Number of missing values",
        col = "skyblue",
        las = 2)
```

Output Part:



Explanation Part: The code generates a bar plot to visualize the number of missing values in each attribute of the dataset. It uses the missing_counts as parameter, which contains the count of missing values per column, and creates a bar plot with labels for the x-axis (attributes) and y-axis (number of missing values). The bars are colored sky blue, and the x-axis labels are rotated vertically for better readability. This plot helps identify which attributes have missing data.

4.4 Missing Values Visualization Gender Attribute:

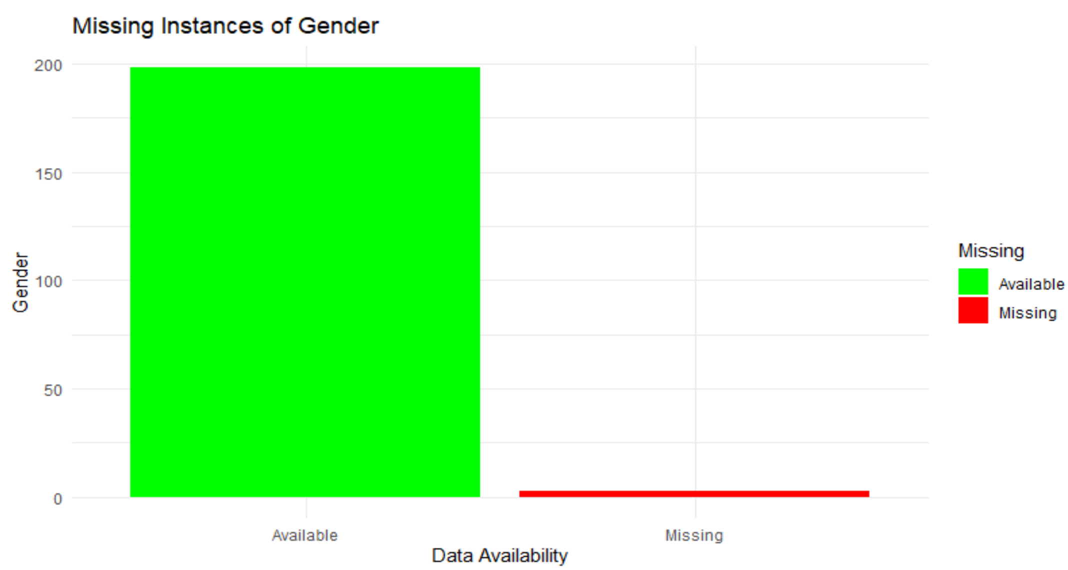
Code Part:

```
install.packages("ggplot2")
library(ggplot2)


missingData <- data.frame(
  Missing = c("Available", "Missing"),
  Gender  = c(sum(!is.na(mydata$Gender)), sum(is.na(mydata$Gender)))
)


ggplot(missingData, aes(x = Missing, y = Gender, fill = Missing)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("Available" = "green", "Missing" = "red")) +
  labs(
    title = "Missing Instances of Gender",
    x = "Data Availability",
    y = "Gender"
  ) +
  theme_minimal()
```

Output Part:



Explanation Part: The code creates a bar plot using ggplot2 to show the availability of data in the "Gender" column of the mydata dataset. It counts the number of available and missing values, then visualizes them with green bars for available data and red bars for missing data. The plot includes a title and axis labels, with a minimalistic theme for clarity. This helps in quickly assessing the completeness of the "Gender" data.

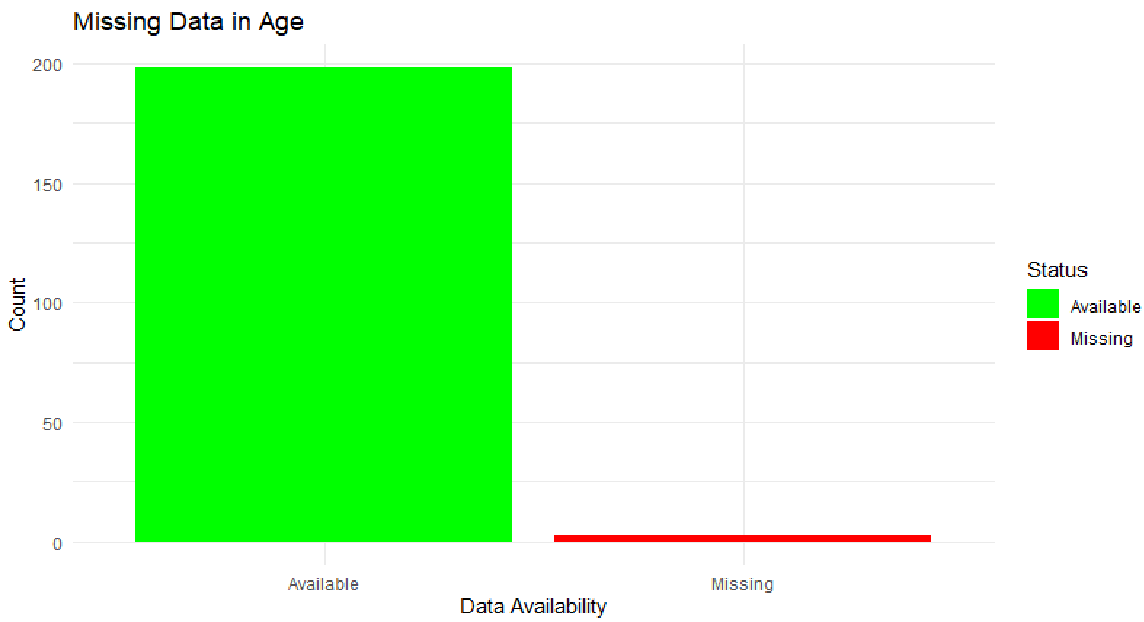
4.5 Missing Values Visualization Age Attribute:

Code Part:

```
missingData_Age <- data.frame(
  Status = c("Available", "Missing"),
  Count = c(sum(!is.na(mydata$Age)), sum(is.na(mydata$Age)))
)

ggplot(missingData_Age, aes(x = Status, y = Count, fill = Status)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("Available" = "green", "Missing" = "red")) +
  labs(
    title = "Missing Data in Age",
    x = "Data Availability",
    y = "Count"
  ) +
  theme_minimal()
```

Output Part:



Explanation Part: The code creates a bar plot using ggplot2 to visualize the availability of data in the "Age" column of the mydata dataset. It counts the available (non-missing) and missing values for the "Age" column, then plots them as bars. Green bars represent available data, and red bars represent missing data. The plot includes a title and axis labels, with a minimalistic theme for clarity. This helps in assessing how complete the "Age" data is in the dataset.

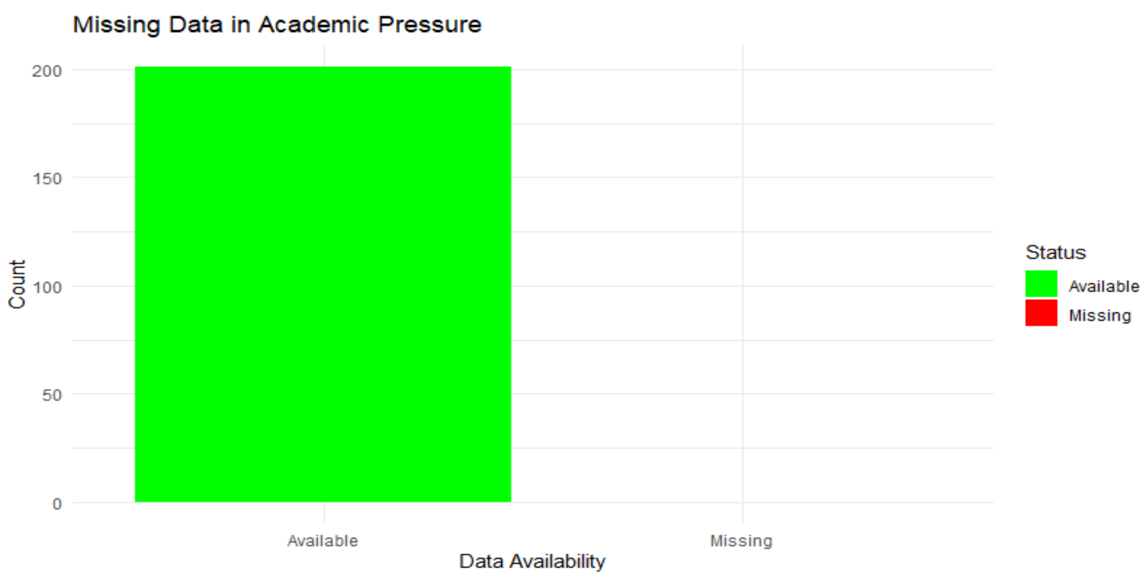
4.6 Missing Values Visualization Academic Pressure Attribute:

Code Part:

```
missingData_AcademicPressure <- data.frame(
  Status = c("Available", "Missing"),
  Count = c(sum(!is.na(mydata$Academic_Pressure)), sum(is.na(mydata$Academic_Pressure)))
)

ggplot(missingData_AcademicPressure, aes(x = Status, y = Count, fill = Status)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("Available" = "green", "Missing" = "red")) +
  labs(
    title = "Missing Data in Academic Pressure",
    x = "Data Availability",
    y = "Count"
  ) +
  theme_minimal()
```

Output Part:



Explanation Part: The code generates a bar plot using ggplot2 to visualize the availability of data in the "Academic Pressure" column of the mydata dataset. It counts the available (non-missing) and missing values for this column, then plots the results with green bars for available data and red bars for missing data. The plot includes a title and axis labels, with a minimalistic theme applied for simplicity. This visualization helps assess the completeness of the "Academic Pressure" data in the dataset.

4.7 Missing Values Visualization Study Satisfaction Attribute:

Code Part:

```
missingData_StudySatisfaction <- data.frame(
  Status = c("Available", "Missing"),
  Count = c(sum(!is.na(mydata$Study_Satisfaction)), sum(is.na(mydata$Study_Satisfaction)))
)

ggplot(missingData_StudySatisfaction, aes(x = Status, y = Count, fill = Status)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("Available" = "green", "Missing" = "red")) +
  labs(
    title = "Missing Data in Study Satisfaction",
    x = "Data Availability",
    y = "Count"
  ) +
  theme_minimal()
```

Output Part:

Explanation Part: The code creates a bar plot using ggplot2 to display the availability of data in the "Study_Satisfaction" column of the mydata dataset. It calculates the count of available (non-missing) and missing values for this column, then plots them with green bars for available data and red bars for missing data. The plot includes a title, axis labels, and uses a minimalistic theme for a clean and clear visualization. This helps in understanding the completeness of the "Study_Satisfaction" data in the dataset.

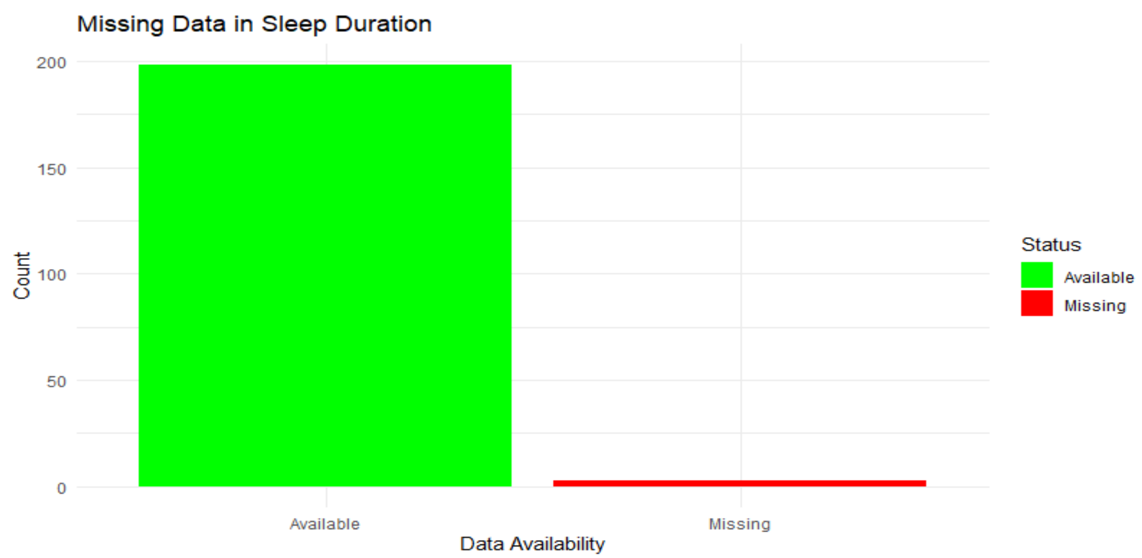
4.8 Missing Values Visualization Sleep Duration Attribute:

Code Part:

```
missingData_SleepDuration <- data.frame(
  Status = c("Available", "Missing"),
  Count = c(sum(!is.na(mydata$Sleep_Duration)), sum(is.na(mydata$Sleep_Duration)))
)

ggplot(missingData_SleepDuration, aes(x = Status, y = Count, fill = Status)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("Available" = "green", "Missing" = "red")) +
  labs(
    title = "Missing Data in Sleep Duration",
    x = "Data Availability",
    y = "Count"
  ) +
  theme_minimal()
```

Output Part:



Explanation Part: The code generates a bar plot using ggplot2 to visualize the availability of data in the "Sleep Duration" column of the mydata dataset. It counts the available (non-missing) and missing values for this column, then creates a plot with green bars representing available data and red bars for missing data. The plot includes a title, axis labels, and uses a minimalistic theme for a clean design. This allows for a clear understanding of the completeness of the "Sleep Duration" data in the dataset.

5. Handling Missing Values:

5.1 Detecting Missing values in Dataset:

	Gender	Age	Academic_Pressure	Study_Satisfaction	Sleep_Duration	Dietary_Habits	Have_you_ever_had_suicidal_thoughts.?	Study_Hours	Financial_Stress	Family._History_of_Mental_Illness	Depression
1	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
2	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
3	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
4	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
5	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
6	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
7	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
8	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
9	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
10	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
11	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE
12	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
13	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
14	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
15	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
16	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
17	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE
18	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE

5.2 Detecting Missing Values Row wise:

Code Part:

```
missing_rows <- which(rowSums(is.na(mydata)) > 0)
missing_rows
```

Output Part:

```
> missing_rows <- which(rowSums(is.na(mydata)) > 0)
> missing_rows
 5  7  9 11 13 17 18 23 25 37 54
 5  7  9 11 13 17 18 23 25 37 54
> 
```

Explanation Part: The code identifies the rows in the mydata dataset that contain at least one missing value (NA). It does this by checking each row for missing values and then returns the indices of the rows with any missing data. This helps in identifying which rows require attention for data handling.

5.3 Discarding Instances of Missing Values:

Code Part:

```r
mydata_remove_missing <- na.omit(mydata)
mydata_remove_missing
```

Initial Part:

	Gender	Age	Academic_Pressure	Study_Satisfaction	Sleep_Duration	Dietary_Habits	Have_you_ever_had_suicidal_thoughts.?	Study_Hours	Financial_Stress	Family_History_of_Mental_Illness	Depression
1	Male	28	2	4	7-8 hours	Moderate	Yes	9	2	Yes	No
2	Male	28	4	5	5-6 hours	Healthy	Yes	7	1	Yes	No
3	Male	25	1	3	5-6 hours	Unhealthy	Yes	10	4	No	Yes
4	Male	23	1	4	More than 8 hours	Unhealthy	Yes	7	2	Yes	No
5	NA	31	1	5	More than 8 hours	Healthy	Yes	4	2	Yes	No
6	Male	19	4	4	5-6 hours	Unhealthy	Yes	1	4	Yes	Yes
7	Female	34	4	2	NA	Moderate	Yes	6	2	No	Yes
8	Female	20	4	1	More than 8 hours	Healthy	Yes	3	4	Yes	Yes
9	Female	NA	1	4	More than 8 hours	Moderate	No	10	3	No	No
10	Male	33	4	3	Less than 5 hours	Unhealthy	Yes	10	1	No	Yes
11	NA	31	5	4	5-6 hours	Healthy	Yes	NA	4	No	Yes
12	Male	24	2	1	7-8 hours	Unhealthy	Yes	11	5	No	Yes
13	Female	23	5	5	NA	Unhealthy	Yes	2	1	Yes	NA
14	Male	25	1	1	5-6 hours	Moderate	Yes	12	3	Yes	Yes
15	Male	21	5	1	More than 8 hours	Unhealthy	Yess	3	5	Yes	Yes
16	Male	28	5	3	5-6 hours	Healthy	Yes	8	3	Yes	Yes
17	Male	23	5	2	More than 8 hours	Moderate	No	NA	4	No	NA
18	Female	23	1	3	NA	Healthy	Yes	0	3	No	No

Output Part:

	Gender	Age	Academic_Pressure	Study_Satisfaction	Sleep_Duration	Dietary_Habits	Have_you_ever_had_suicidal_thoughts.?	Study_Hours	Financial_Stress	Family_History_of_Mental_Illness	Depression
1	Male	28	2	4	7-8 hours	Moderate	Yes	9	2	Yes	No
2	Male	28	4	5	5-6 hours	Healthy	Yes	7	1	Yes	No
3	Male	25	1	3	5-6 hours	Unhealthy	Yes	10	4	No	Yes
4	Male	23	1	4	More than 8 hours	Unhealthy	Yes	7	2	Yes	No
6	Male	19	4	4	5-6 hours	Unhealthy	Yes	1	4	Yes	Yes
8	Female	20	4	1	More than 8 hours	Healthy	Yes	3	4	Yes	Yes
10	Male	33	4	3	Less than 5 hours	Unhealthy	Yes	10	1	No	Yes
12	Male	24	2	1	7-8 hours	Unhealthy	Yes	11	5	No	Yes
14	Male	25	1	1	5-6 hours	Moderate	Yes	12	3	Yes	Yes
15	Male	21	5	1	More than 8 hours	Unhealthy	Yess	3	5	Yes	Yes
16	Male	28	5	3	5-6 hours	Healthy	Yes	8	3	Yes	Yes
19	Female	20	5	5	More than 8 hours	Unhealthy	Yes	2	5	No	Yes
20	Male	29	4	3	More than 8 hours	Unhealthy	Yes	1	3	No	Yes
21	Male	31	2	3	More than 8 hours	Unhealthy	No	3	3	Yes	No
22	Male	24	3	4	More than 8 hours	Healthy	Yes	1	3	No	No
24	Female	33	3	2	7-8 hours	Moderate	No	11	5	Yes	No
26	Male	31	2	2	7-8 hours	Healthy	No	2	4	Yes	No
27	Male	30	3	4	7-8 hours	Moderate	Yes	0	2	Yes	No
28	Male	21	5	3	7-8 hours	Unhealthy	No	6	4	Yes	Yes
29	Female	29	3	5	Less than 5 hours	Moderate	Yes	4	3	Yes	No

Explanation Part: The code mydata_remove_missing <- na.omit(mydata) removes all rows from the mydata dataset that contain any missing values (NA). The na.omit() function creates a new dataset, mydata_remove_missing, which only includes complete rows (i.e., rows without any missing values). This is useful for cleaning the dataset before performing analysis or modeling, ensuring that only complete cases are considered.

5.4 Replacing Missing Values with Mean (Average) for Numeric Attribute:

Code Part:

```
mydata[] <- lapply(mydata, function(x) {
  if (is.numeric(x)) {
    x[is.na(x)] <- mean(x, na.rm = TRUE)
    x <- round(x)
  }
  return(x)
})
```

Initial Part:

	Gender	Age	Academic_Pressure	Study_Satisfaction	Sleep_Duration	Dietary_Habits	Have_you_ever_had_suicidal_thoughts.?	Study_Hours	Financial_Stress	Family_History_of_Mental_Illness	Depression
1	Male	28	2	4	7-8 hours	Moderate	Yes	9	2	Yes	No
2	Male	28	4	5	5-6 hours	Healthy	Yes	7	1	Yes	No
3	Male	25	1	3	5-6 hours	Unhealthy	Yes	10	4	No	Yes
4	Male	23	1	4	More than 8 hours	Unhealthy	Yes	7	2	Yes	No
5	NA	31	1	5	More than 8 hours	Healthy	Yes	4	2	Yes	No
6	Male	19	4	4	5-6 hours	Unhealthy	Yes	1	4	Yes	Yes
7	Female	34	4	2	NA	Moderate	Yes	6	2	No	Yes
8	Female	20	4	1	More than 8 hours	Healthy	Yes	3	4	Yes	Yes
9	Female	NA	1	4	More than 8 hours	Moderate	No	10	3	No	No
10	Male	33	4	3	Less than 5 hours	Unhealthy	Yes	10	1	No	Yes
11	NA	31	5	4	5-6 hours	Healthy	Yes	NA	4	No	Yes
12	Male	24	2	1	7-8 hours	Unhealthy	Yes	11	5	No	Yes
13	Female	23	5	5	NA	Unhealthy	Yes	2	1	Yes	NA
14	Male	25	1	1	5-6 hours	Moderate	Yes	12	3	Yes	Yes
15	Male	21	5	1	More than 8 hours	Unhealthy	Yess	3	5	Yes	Yes
16	Male	28	5	3	5-6 hours	Healthy	Yes	8	3	Yes	Yes
17	Male	23	5	2	More than 8 hours	Moderate	No	NA	4	No	NA
18	Female	23	1	3	NA	Healthy	Yes	0	3	No	No

Output Part:

	Gender	Age	Academic_Pressure	Study_Satisfaction	Sleep_Duration	Dietary_Habits	Have_you_ever_had_suicidal_thoughts.?	Study_Hours	Financial_Stress	Family_History_of_Mental_Illness	Depression
1	Male	28	2	4	7-8 hours	Moderate	Yes	9	2	Yes	No
2	Male	28	4	5	5-6 hours	Healthy	Yes	7	1	Yes	No
3	Male	25	1	3	5-6 hours	Unhealthy	Yes	10	4	No	Yes
4	Male	23	1	4	More than 8 hours	Unhealthy	Yes	7	2	Yes	No
5	NA	31	1	5	More than 8 hours	Healthy	Yes	4	2	Yes	No
6	Male	19	4	4	5-6 hours	Unhealthy	Yes	1	4	Yes	Yes
7	Female	34	4	2	NA	Moderate	Yes	6	2	No	Yes
8	Female	20	4	1	More than 8 hours	Healthy	Yes	3	4	Yes	Yes
9	Female	28	1	4	More than 8 hours	Moderate	No	10	3	No	No
10	Male	33	4	3	Less than 5 hours	Unhealthy	Yes	10	1	No	Yes
11	NA	31	5	4	5-6 hours	Healthy	Yes	6	4	No	Yes
12	Male	24	2	1	7-8 hours	Unhealthy	Yes	11	5	No	Yes
13	Female	23	5	5	NA	Unhealthy	Yes	2	1	Yes	NA
14	Male	25	1	1	5-6 hours	Moderate	Yes	12	3	Yes	Yes
15	Male	21	5	1	More than 8 hours	Unhealthy	Yess	3	5	Yes	Yes
16	Male	28	5	3	5-6 hours	Healthy	Yes	8	3	Yes	Yes
17	Male	23	5	2	More than 8 hours	Moderate	No	6	4	No	NA
18	Female	23	1	3	NA	Healthy	Yes	0	3	No	No
19	Female	20	5	5	More than 8 hours	Unhealthy	Yes	2	5	No	Yes
20	Male	29	4	3	More than 8 hours	Unhealthy	Yes	1	3	No	Yes

Explanation Part: The code processes the mydata dataset by replacing missing values (NA) in numeric columns with the mean of the respective column. It first checks if the column is numeric, then fills missing values with the mean of the non-missing values, ignoring any NAs. Afterward, it rounds the values in the column to the nearest integer. This approach helps handle missing data by imputing values with the mean and ensures that the data is rounded for consistency, making it ready for further analysis.

5.5 Replacing Missing Values with Mode (Most Frequent) for Categorical Attribute:

Code Part:

```
mydata[] <- lapply(mydata, function(x) {
  if (is.character(x) || is.factor(x)) {
    x[is.na(x)] <- names(which.max(table(x)))
  }
  return(x)
})
```

Initial Part:

	Gender	Age	Academic_Pressure	Study_Satisfaction	Sleep_Duration	Dietary_Habits	Have_you_ever_had_suicidal_thoughts.?	Study_Hours	Financial_Stress	Family_History_of_Mental_Illness	Depression
1	Male	28	2	4	7-8 hours	Moderate	Yes	9	2	Yes	No
2	Male	28	4	5	5-6 hours	Healthy	Yes	7	1	Yes	No
3	Male	25	1	3	5-6 hours	Unhealthy	Yes	10	4	No	Yes
4	Male	23	1	4	More than 8 hours	Unhealthy	Yes	7	2	Yes	No
5	NA	31	1	5	More than 8 hours	Healthy	Yes	4	2	Yes	No
6	Male	19	4	4	5-6 hours	Unhealthy	Yes	1	4	Yes	Yes
7	Female	34	4	2	NA	Moderate	Yes	6	2	No	Yes
8	Female	20	4	1	More than 8 hours	Healthy	Yes	3	4	Yes	Yes
9	Female	NA	1	4	More than 8 hours	Moderate	No	10	3	No	No
10	Male	33	4	3	Less than 5 hours	Unhealthy	Yes	10	1	No	Yes
11	NA	31	5	4	5-6 hours	Healthy	Yes	NA	4	No	Yes
12	Male	24	2	1	7-8 hours	Unhealthy	Yes	11	5	No	Yes
13	Female	23	5	5	NA	Unhealthy	Yes	2	1	Yes	NA
14	Male	25	1	1	5-6 hours	Moderate	Yes	12	3	Yes	Yes
15	Male	21	5	1	More than 8 hours	Unhealthy	Yess	3	5	Yes	Yes
16	Male	28	5	3	5-6 hours	Healthy	Yes	8	3	Yes	Yes
17	Male	23	5	2	More than 8 hours	Moderate	No	NA	4	No	NA
18	Female	23	1	3	NA	Healthy	Yes	0	3	No	No

Output Part:

Filter

	Gender	Age	Academic_Pressure	Study_Satisfaction	Sleep_Duration	Dietary_Habits	Have_you_ever_had_suicidal_thoughts.?	Study_Hours	Financial_Stress	Family_History_of_Mental_Illness	Depression
1	Male	28	2	4	7-8 hours	Moderate	Yes	9	2	Yes	No
2	Male	28	4	5	5-6 hours	Healthy	Yes	7	1	Yes	No
3	Male	25	1	3	5-6 hours	Unhealthy	Yes	10	4	No	Yes
4	Male	23	1	4	More than 8 hours	Unhealthy	Yes	7	2	Yes	No
5	Male	31	1	5	More than 8 hours	Healthy	Yes	4	2	Yes	No
6	Male	19	4	4	5-6 hours	Unhealthy	Yes	1	4	Yes	Yes
7	Female	34	4	2	More than 8 hours	Moderate	Yes	6	2	No	Yes
8	Female	20	4	1	More than 8 hours	Healthy	Yes	3	4	Yes	Yes
9	Female	28	1	4	More than 8 hours	Moderate	No	10	3	No	No
10	Male	33	4	3	Less than 5 hours	Unhealthy	Yes	10	1	No	Yes
11	Male	31	5	4	5-6 hours	Healthy	Yes	6	4	No	Yes
12	Male	24	2	1	7-8 hours	Unhealthy	Yes	11	5	No	Yes
13	Female	23	5	5	More than 8 hours	Unhealthy	Yes	2	1	Yes	No
14	Male	25	1	1	5-6 hours	Moderate	Yes	12	3	Yes	Yes
15	Male	21	5	1	More than 8 hours	Unhealthy	Yess	3	5	Yes	Yes
16	Male	28	5	3	5-6 hours	Healthy	Yes	8	3	Yes	Yes
17	Male	23	5	2	More than 8 hours	Moderate	No	6	4	No	No
18	Female	23	1	3	More than 8 hours	Healthy	Yes	0	3	No	No
19	Female	20	5	5	More than 8 hours	Unhealthy	Yes	2	5	No	Yes
20	Male	29	4	3	More than 8 hours	Unhealthy	Yes	1	3	No	Yes
21	Male	31	2	3	More than 8 hours	Unhealthy	No	3	3	Yes	No
22	Male	24	3	4	More than 8 hours	Healthy	Yes	1	3	No	No
23	Male	31	2	4	More than 8 hours	Unhealthy	Noo	10	1	No	No
24	Female	33	3	2	7-8 hours	Moderate	No	11	5	Yes	No

owing 1 to 25 of 201 entries, 11 total columns

Explanation Part: The code addresses missing values (NA) in character or factor columns of the mydata dataset by replacing them with the most frequent value in each respective column. It identifies whether a column is of character or factor type, and for those with missing values, it replaces NA with the most common value (mode). This method ensures that categorical columns retain a meaningful and representative value where data is missing, helping to maintain the integrity of the dataset while handling missing data.

5.6 Replacing Missing Values with Forward Fill:
Code Part:

```
forward_fill <- mydata
forward_fill[] <- lapply(forward_fill, function(column) {
  zoo::na.locf(column, na.rm = FALSE)
})
```

Initial Part:

	Gender	Age	Academic_Pressure	Study_Satisfaction	Sleep_Duration	Dietary_Habits	Have_you_ever_had_suicidal_thoughts.?	Study_Hours	Financial_Stress	Family_History_of_Mental_Illness	Depression
1	Male	28	2	4	7-8 hours	Moderate	Yes	9	2	Yes	No
2	Male	28	4	5	5-6 hours	Healthy	Yes	7	1	Yes	No
3	Male	25	1	3	5-6 hours	Unhealthy	Yes	10	4	No	Yes
4	Male	23	1	4	More than 8 hours	Unhealthy	Yes	7	2	Yes	No
5	NA	31	1	5	More than 8 hours	Healthy	Yes	4	2	Yes	No
6	Male	19	4	4	5-6 hours	Unhealthy	Yes	1	4	Yes	Yes
7	Female	34	4	2	NA	Moderate	Yes	6	2	No	Yes
8	Female	20	4	1	More than 8 hours	Healthy	Yes	3	4	Yes	Yes
9	Female	NA	1	4	More than 8 hours	Moderate	No	10	3	No	No
10	Male	33	4	3	Less than 5 hours	Unhealthy	Yes	10	1	No	Yes
11	NA	31	5	4	5-6 hours	Healthy	Yes	NA	4	No	Yes
12	Male	24	2	1	7-8 hours	Unhealthy	Yes	11	5	No	Yes
13	Female	23	5	5	NA	Unhealthy	Yes	2	1	Yes	NA
14	Male	25	1	1	5-6 hours	Moderate	Yes	12	3	Yes	Yes
15	Male	21	5	1	More than 8 hours	Unhealthy	Yess	3	5	Yes	Yes
16	Male	28	5	3	5-6 hours	Healthy	Yes	8	3	Yes	Yes
17	Male	23	5	2	More than 8 hours	Moderate	No	NA	4	No	NA
18	Female	23	1	3	NA	Healthy	Yes	0	3	No	No

Output Part:

	Gender	Age	Academic_Pressure	Study_Satisfaction	Sleep_Duration	Dietary_Habits	Have_you_ever_had_suicidal_thoughts.?	Study_Hours	Financial_Stress	Family_History_of_Mental_Illness	Depression
1	Male	28	2	4	7-8 hours	Moderate	Yes	9	2	Yes	No
2	Male	28	4	5	5-6 hours	Healthy	Yes	7	1	Yes	No
3	Male	25	1	3	5-6 hours	Unhealthy	Yes	10	4	No	Yes
4	Male	23	1	4	More than 8 hours	Unhealthy	Yes	7	2	Yes	No
5	Male	31	1	5	More than 8 hours	Healthy	Yes	4	2	Yes	No
6	Male	19	4	4	5-6 hours	Unhealthy	Yes	1	4	Yes	Yes
7	Female	34	4	2	5-6 hours	Moderate	Yes	6	2	No	Yes
8	Female	20	4	1	More than 8 hours	Healthy	Yes	3	4	Yes	Yes
9	Female	20	1	4	More than 8 hours	Moderate	No	10	3	No	No
10	Male	33	4	3	Less than 5 hours	Unhealthy	Yes	10	1	No	Yes
11	Male	31	5	4	5-6 hours	Healthy	Yes	10	4	No	Yes
12	Male	24	2	1	7-8 hours	Unhealthy	Yes	11	5	No	Yes
13	Female	23	5	5	7-8 hours	Unhealthy	Yes	2	1	Yes	Yes
14	Male	25	1	1	5-6 hours	Moderate	Yes	12	3	Yes	Yes
15	Male	21	5	1	More than 8 hours	Unhealthy	Yess	3	5	Yes	Yes
16	Male	28	5	3	5-6 hours	Healthy	Yes	8	3	Yes	Yes
17	Male	23	5	2	More than 8 hours	Moderate	No	8	4	No	Yes
18	Female	23	1	3	More than 8 hours	Healthy	Yes	0	3	No	No
19	Female	20	5	5	More than 8 hours	Unhealthy	Yes	2	5	No	Yes
20	Male	29	4	3	More than 8 hours	Unhealthy	Yes	1	3	No	Yes

Explanation Part: This code applies forward fill to all columns in the mydata data frame. A copy, forward_fill, is created, and lapply iterates through each column, using zoo::na.locf to replace NA values with the last observed value(previous value) while preserving NA at the start if no prior value exists

5.7 Replacing Missing Values with Backward Fill:

Code Part:

```r
backward_fill <- mydata
backward_fill [] <- lapply(backward_fill, function(column) {
  zoo::na.locf(column, na.rm = FALSE, fromLast = TRUE)
})
```

Initial Part:

	Gender	Age	Academic_Pressure	Study_Satisfaction	Sleep_Duration	Dietary_Habits	Have_you_ever_had_suicidal_thoughts.?	Study_Hours	Financial_Stress	Family_History_of_Mental_Illness	Depression
1	Male	28	2	4	7-8 hours	Moderate	Yes	9	2	Yes	No
2	Male	28	4	5	5-6 hours	Healthy	Yes	7	1	Yes	No
3	Male	25	1	3	5-6 hours	Unhealthy	Yes	10	4	No	Yes
4	Male	23	1	4	More than 8 hours	Unhealthy	Yes	7	2	Yes	No
5	NA	31	1	5	More than 8 hours	Healthy	Yes	4	2	Yes	No
6	Male	19	4	4	5-6 hours	Unhealthy	Yes	1	4	Yes	Yes
7	Female	34	4	2	NA	Moderate	Yes	6	2	No	Yes
8	Female	20	4	1	More than 8 hours	Healthy	Yes	3	4	Yes	Yes
9	Female	NA	1	4	More than 8 hours	Moderate	No	10	3	No	No
10	Male	33	4	3	Less than 5 hours	Unhealthy	Yes	10	1	No	Yes
11	NA	31	5	4	5-6 hours	Healthy	Yes	NA	4	No	Yes
12	Male	24	2	1	7-8 hours	Unhealthy	Yes	11	5	No	Yes
13	Female	23	5	5	NA	Unhealthy	Yes	2	1	Yes	NA
14	Male	25	1	1	5-6 hours	Moderate	Yes	12	3	Yes	Yes
15	Male	21	5	1	More than 8 hours	Unhealthy	Yess	3	5	Yes	Yes
16	Male	28	5	3	5-6 hours	Healthy	Yes	8	3	Yes	Yes
17	Male	23	5	2	More than 8 hours	Moderate	No	NA	4	No	NA
18	Female	23	1	3	NA	Healthy	Yes	0	3	No	No

Output Part:

	Gender	Age	Academic_Pressure	Study_Satisfaction	Sleep_Duration	Dietary_Habits	Have_you_ever_had_suicidal_thoughts.?	Study_Hours	Financial_Stress	Family._History_of_Mental_Illness	Depression
1	Male	28	2	4	7-8 hours	Moderate	Yes	9	2	Yes	No
2	Male	28	4	5	5-6 hours	Healthy	Yes	7	1	Yes	No
3	Male	25	1	3	5-6 hours	Unhealthy	Yes	10	4	No	Yes
4	Male	23	1	4	More than 8 hours	Unhealthy	Yes	7	2	Yes	No
5	Male	31	1	5	More than 8 hours	Healthy	Yes	4	2	Yes	No
6	Male	19	4	4	5-6 hours	Unhealthy	Yes	1	4	Yes	Yes
7	Female	34	4	2	More than 8 hours	Moderate	Yes	6	2	No	Yes
8	Female	20	4	1	More than 8 hours	Healthy	Yes	3	4	Yes	Yes
9	Female	33	1	4	More than 8 hours	Moderate	No	10	3	No	No
10	Male	33	4	3	Less than 5 hours	Unhealthy	Yes	10	1	No	Yes
11	Male	31	5	4	5-6 hours	Healthy	Yes	11	4	No	Yes
12	Male	24	2	1	7-8 hours	Unhealthy	Yes	11	5	No	Yes
13	Female	23	5	5	5-6 hours	Unhealthy	Yes	2	1	Yes	Yes
14	Male	25	1	1	5-6 hours	Moderate	Yes	12	3	Yes	Yes
15	Male	21	5	1	More than 8 hours	Unhealthy	Yess	3	5	Yes	Yes
16	Male	28	5	3	5-6 hours	Healthy	Yes	8	3	Yes	Yes
17	Male	23	5	2	More than 8 hours	Moderate	No	0	4	No	No
18	Female	23	1	3	More than 8 hours	Healthy	Yes	0	3	No	No
19	Female	20	5	5	More than 8 hours	Unhealthy	Yes	2	5	No	Yes
20	Male	29	4	3	More than 8 hours	Unhealthy	Yes	1	3	No	Yes

Explanation Part: This code applies backward fill to all columns in the mydata data frame. A copy, backward_fill, is created to preserve the original data. Using lapply, it iterates through each column, applying zoo::na.locf with fromLast = TRUE to replace NA values with the next observed value.

6. Mean, Median, Variance, SD:

6.1 Numeric Mean:

Code Part:

```
numeric_means <- sapply(mydata, function(col) {
  if (is.numeric(col)) {
    mean(col, na.rm = TRUE)
  } else {
    NA
  }
})
numeric_means
```

Output Part:

```
> numeric_means <- sapply(mydata, function(col) {
+   if (is.numeric(col)) {
+     mean(col, na.rm = TRUE)
+   } else {
+     NA
+   }
+ })
> numeric_means
                              Gender                                  Age                    Academic_Pressure
Study_Satisfaction
                                  NA                            28.233831                             3.154229
3.189055
                      Sleep_Duration                        Dietary_Habits Have_you_ever_had_suicidal_thoughts.?
Study_Hours
                                  NA                                   NA                                   NA
6.348259
                    Financial_Stress     Family._History_of_Mental_Illness                           Depression
                            2.930348                                   NA                                   NA
> |
```

Explanation Part: The code calculates the mean of each numeric column in the mydata dataset, ignoring any missing values (NA). It does this by applying a function to each column, checking if the column is numeric, and then computing the mean of the non-missing values using mean(col, na.rm = TRUE). If the column is not numeric, it returns NA. This provides a summary of the means for all numeric columns in the dataset.

6.2 Numeric Median:
Code Part:

```
numeric_median <- sapply(mydata, function(col) {
  if (is.numeric(col)) {
    median(col, na.rm = TRUE)
  } else {
    NA
  }
})
numeric_median
```

Output Part:

```
> numeric_median <- sapply(mydata, function(col) {
+   if (is.numeric(col)) {
+     median(col, na.rm = TRUE)
+   } else {
+     NA
+   }
+ })
> numeric_median
                              Gender                                 Age                   Academic_Pressure
                                  NA                                  26                                   3
                  Study_Satisfaction                      Sleep_Duration                      Dietary_Habits
                                   3                                  NA                                  NA
Have_you_ever_had_suicidal_thoughts.?                        Study_Hours                    Financial_Stress
                                  NA                                   7                                   3
    Family._History_of_Mental_Illness                          Depression
                                  NA                                  NA
> 
```

Explanation Part: The code calculates the median for each numeric column in the mydata dataset, ignoring any missing values (NA). It applies a function to each column, checks if the column is numeric, and then computes the median of the non-missing values using median (col, na.rm = TRUE). For non-numeric columns, it returns NA. This helps summarize the central tendency of all numeric columns in the dataset.

6.3 Numeric Variance:

Code Part:

```
numeric_var <-sapply(mydata, function(x) {
  if (is.numeric(x)) {
    var(x, na.rm = TRUE)
  } else {
    NA
  }
})
numeric_var
```

Output Part:

```
> numeric_var <-sapply(mydata, function(x) {
+   if (is.numeric(x)) {
+     var(x, na.rm = TRUE)
+   } else {
+     NA
+   }
+ })
> numeric_var
                               Gender                                   Age                      Academic_Pressure
                                   NA                            425.850050                               4.061095
                   Study_Satisfaction                        Sleep_Duration                         Dietary_Habits
                             1.754080                                    NA                                     NA
Have_you_ever_had_suicidal_thoughts.?                           Study_Hours                       Financial_Stress
                                   NA                             14.378109                               1.995124
    Family._History_of_Mental_Illness                            Depression
                                   NA                                    NA
>
```

Explanation Part: The code calculates the variance for each numeric column in the mydata dataset, ignoring any missing values (NA). It applies a function to each column, checks if the column is numeric, and then computes the variance using var(x, na.rm = TRUE), which removes NA values during the calculation. For non-numeric columns, it returns NA. This provides a measure of variability for all numeric columns in the dataset.

6.4 Numeric Standard Deviation:

Code Part:

```
numeric_sd <-sapply(mydata, function(x) {
  if (is.numeric(x)) {
    sd(x, na.rm = TRUE)
  } else {
    NA
  }
})
numeric_sd
```

Output Part:

```
                                  NA                                  NA
> numeric_sd <-sapply(mydata, function(x) {
+   if (is.numeric(x)) {
+     sd(x, na.rm = TRUE)
+   } else {
+     NA
+   }
+ })
> numeric_sd
                            Gender                                Age                 Academic_Pressure
                                NA                          20.636135                          2.015216
                Study_Satisfaction                     Sleep_Duration                    Dietary_Habits
                          1.324417                                 NA                                NA
Have_you_ever_had_suicidal_thoughts.?                     Study_Hours                  Financial_Stress
                                NA                           3.791848                          1.412489
    Family._History_of_Mental_Illness                      Depression
                                NA                                 NA
> |
```

Explanation Part: The code calculates the standard deviation for each numeric column in the mydata dataset, ignoring any missing values (NA). It applies a function to each column, checks if the column is numeric, and then computes the standard deviation using sd(x, na.rm = TRUE). For non-numeric columns, it returns NA. This helps measure the dispersion or spread of values for all numeric columns in the dataset.

7. Handling Duplicate Values:

7.1 Detecting Duplicate Values Row Wise:

Code Part:

```
duplicated_rows <- mydata[duplicated(mydata) | duplicated(mydata, fromLast = TRUE), ]
duplicated_rows
```

Output Part:

```
> duplicated_rows <- mydata[duplicated(mydata) | duplicated(mydata, fromLast = TRUE), ]
> duplicated_rows
   Gender Age Academic_Pressure Study_Satisfaction    Sleep_Duration Dietary_Habits
46   Male  20                 3                  5 More than 8 hours      Unhealthy
47   Male  20                 3                  5 More than 8 hours      Unhealthy
   Have_you_ever_had_suicidal_thoughts.? Study_Hours Financial_Stress Family._History_of_Mental_Illness Depression
46                                    No          11                4                                No         No
47                                    No          11                4                                No         No
> 
```

Explanation Part: The code identifies and returns all rows in the mydata dataset that are duplicated, either from the beginning or the end of the dataset. It uses the duplicated() function to detect duplicate rows and combines both directions of checking to capture all instances of duplication. This helps in identifying redundant data for further cleaning or analysis.

7.2 Removing Duplicated Values:

Code Part:

```
mydata <-distinct(mydata)
```

Output Part:

	Gender	Age	Academic_Pressure	Study_Satisfaction	Sleep_Duration	Dietary_Habits	Have_you_ever_had_suicidal_thoughts.?	Study_Hours	Financial_Stress	Family_History_of_Mental_Illness	Depression
40	Female	21	3	3	7-8 hours	Moderate	Yes	8	5	Yes	Yes
41	Male	29	1	1	7-8 hours	Healthy	No	6	1	Yes	No
42	Male	22	1	3	7-8 hours	Unhealthy	No	6	4	No	No
43	Male	21	3	2	7-8 hours	Unhealthy	Yes	1	5	No	Yes
44	Male	31	5	4	5-6 hours	Healthy	No	12	3	No	No
45	Female	24	1	3	5-6 hours	Moderate	No	3	5	No	No
46	Male	20	3	5	More than 8 hours	Unhealthy	No	11	4	No	No
47	Female	31	1	3	7-8 hours	Healthy	No	12	3	Yes	No
48	Male	21	1	5	5-6 hours	Unhealthy	Yes	1	1	No	No
49	Male	24	2	4	5-6 hours	Healthy	No	12	4	Yes	No
50	Male	34	3	4	7-8 hours	Healthy	No	8	3	No	No
51	Female	25	5	4	Less than 5 hours	Healthy	No	7	1	No	No
52	Male	27	2	5	5-6 hours	Healthy	Yes	10	3	No	No
53	Female	28	2	4	5-6 hours	Moderate	No	10	1	Yes	No
54	Male	26	4	4	5-6 hours	Unhealthy	No	9	1	Yes	No
55	Male	23	2	5	Less than 5 hours	Unhealthy	No	8	5	No	No

Explanation Part: The code mydata <- distinct(mydata) removes any duplicate rows from the mydata dataset. It keeps only the unique rows, ensuring that the dataset contains no redundancy. This operation helps in cleaning the data by eliminating repeated entries, making it ready for analysis**.**

8. Imbalanced to Balanced Dataset:

8.1 Detecting Imbalance Data:

Code Part:

```r
result <- lapply(mydata, function(x) {
  if (is.factor(x) | is.character(x)) {
    table(x)
  }
})
result
```

Output Part:

```
> mydata <-distinct(mydata)
> result <- lapply(mydata, function(x) {
+   if (is.factor(x) | is.character(x)) {
+     table(x)
+   }
+ })
> result
$Gender
x
Female   Male
    88    112

$Age
NULL

$Academic_Pressure
NULL

$Study_Satisfaction
NULL

$Sleep_Duration
x
        5-6 hours         7-8 hours Less than 5 hours More than 8 hours
               49                52                48                51

$Dietary_Habits
x
  Healthy  Moderate Unhealthy
       66        66        68

$`Have_you_ever_had_suicidal_thoughts.?`
x
  No  Noo  Yes Yess
  92    1  106    1

$Study_Hours
NULL

$Financial_Stress
NULL

$Family._History_of_Mental_Illness
x
 No Yes
107  93

$Depression
x
 No Yes
116  84

> |
```

Explanation Part: The code generates frequency tables for each factor or character column in the mydata dataset. It applies a function to each column, checking if the column is a factor or character type, and then counts the occurrences of each unique value using table(x). The result is a list of frequency tables, which helps to analyze the distribution of categorical data in the dataset and understand how often each value appears in the respective columns.

8.2 Under Sampling:

Code Part:

```
mydata$Gender <- as.factor(mydata$Gender)

undersampled_data <- ovun.sample(Gender ~ ., data = mydata, method = "under", seed = 123)$data
table(undersampled_data$Gender)
```

Output Part:

```
> mydata$Gender <- as.factor(mydata$Gender)
>
> undersampled_data <- ovun.sample(Gender ~ ., data = mydata, method = "under", seed = 123)$data
> table(undersampled_data$Gender)

  Male Female
    83     88
>
```

Explanation Part: The code converts the Gender column in the mydata dataset into a factor and then applies undersampling to balance the distribution of genders. The ovun.sample() function reduces the size of the majority class, ensuring that both classes in the Gender column have a similar number of instances.

8.3 Over Sampling:

Code Part:

```
oversampled_data <- ovun.sample(Gender ~ ., data = mydata, method = "over", seed = 123)$data
table(oversampled_data$Gender)
```

Output Part:

```
> oversampled_data <- ovun.sample(Gender ~ ., data = mydata, method = "over", seed = 123)$data
> table(oversampled_data$Gender)

  Male Female 
   112    108 
> |
```

Explanation Part: The code applies oversampling to balance the Gender column in the mydata dataset by increasing the instances of the minority class. This ensures both classes have an equal number of records, helping to address class imbalance. The table() function then displays the distribution of genders after oversampling.

9. Outlier/Invalid data:

9.1 Detecting Invalid data:

Code Part:

```
errors <- lapply(mydata, function(col) {
  unique(col)
})
errors
```

Output Part:

```
> errors <- lapply(mydata, function(col) {
+   unique(col)
+ })
> errors
$Gender
[1] "Male"   "Female"

$Age
 [1]  28  25  23  31  19  34  20  33  24  21  29  30  32  26  22  27  18 230 226

$Academic_Pressure
[1]  2  4  1  5  3 20 15

$Study_Satisfaction
[1] 4 5 3 2 1

$Sleep_Duration
[1] "7-8 hours"         "5-6 hours"         "More than 8 hours" "Less than 5 hours"

$Dietary_Habits
[1] "Moderate"  "Healthy"   "Unhealthy"

$`Have_you_ever_had_suicidal_thoughts.?`
[1] "Yes"  "No"   "Yess" "Noo"

$Study_Hours
 [1]  9  7 10  4  1  6  3 11  2 12  8  0  5

$Financial_Stress
[1] 2 1 4 3 5

$Family._History_of_Mental_Illness
[1] "Yes" "No"

$Depression
[1] "No"  "Yes"

> 
```

Explanation Part: The code retrieves the unique values from each of those columns. This helps in understanding the distinct categories/values present in the attributes, which is needed to detect invalid data

9.2 Handling Invalid Data:

Code Part:

```
mydata$Have_you_ever_had_suicidal_thoughts.. <- gsub("Yess", "Yes", mydata$Have_you_ever_had_suicidal_thoughts..)
mydata$Have_you_ever_had_suicidal_thoughts.. <- gsub("Noo", "No", mydata$Have_you_ever_had_suicidal_thoughts..)
unique(mydata$Have_you_ever_had_suicidal_thoughts..)
```

Output Part:

```
> mydata$Have_you_ever_had_suicidal_thoughts.. <- gsub("Yess", "Yes", mydata$Have_you_ever_had_suicidal_thoughts..)
> mydata$Have_you_ever_had_suicidal_thoughts.. <- gsub("Noo", "No", mydata$Have_you_ever_had_suicidal_thoughts..)
> unique(mydata$Have_you_ever_had_suicidal_thoughts..)
[1] "Yes" "No"
>
```

Explanation Part: The code corrects inconsistencies in the Have_you_ever_had_suicidal_thoughts.. column by replacing incorrect values ("Yess" with "Yes" and "Noo" with "No") using the gsub() function. It then displays the unique values in the column to ensure the data is consistent and ready for analysis.

9.3 Identify Outlier:

Code Part:

```
outliers_iqr <- lapply(mydata[sapply(mydata, is.numeric)], function(x) {
  Q1 <- quantile(x, 0.25, na.rm = TRUE)
  Q3 <- quantile(x, 0.75, na.rm = TRUE)
  IQR <- Q3 - Q1
  which(x < (Q1 - 1.5 * IQR) | x > (Q3 + 1.5 * IQR))
})
outliers_iqr
```

Output Part:

Name	Type	Value
outliers_iqr	list [5]	List of length 5
Age	integer [2]	115 121
Academic_Pressure	integer [2]	88 94
Study_Satisfaction	integer [0]	
Study_Hours	integer [0]	
Financial_Stress	integer [0]	

Explanation Part: The code detects outliers in the numeric columns of the mydata dataset using the Interquartile Range (IQR) method. It calculates the first and third quartiles (Q1 and Q3) for each column, computes the IQR, and identifies values that fall outside the range of 1.5 times the IQR from the quartiles. The result is a list of indices marking the outliers in each numeric column.

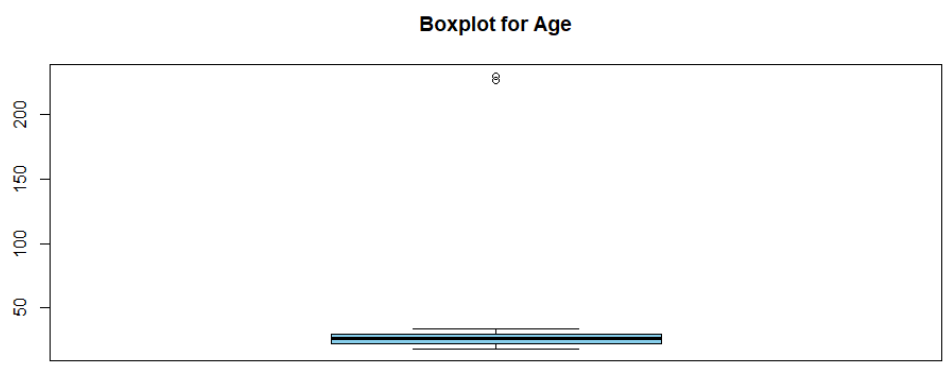
9.4 Outlier Visualization in Boxplot:

For Age Attribute:

Code Part:

```
boxplot(mydata$Age, main = "Boxplot for Age", col = "skyblue")
```

Output Part:



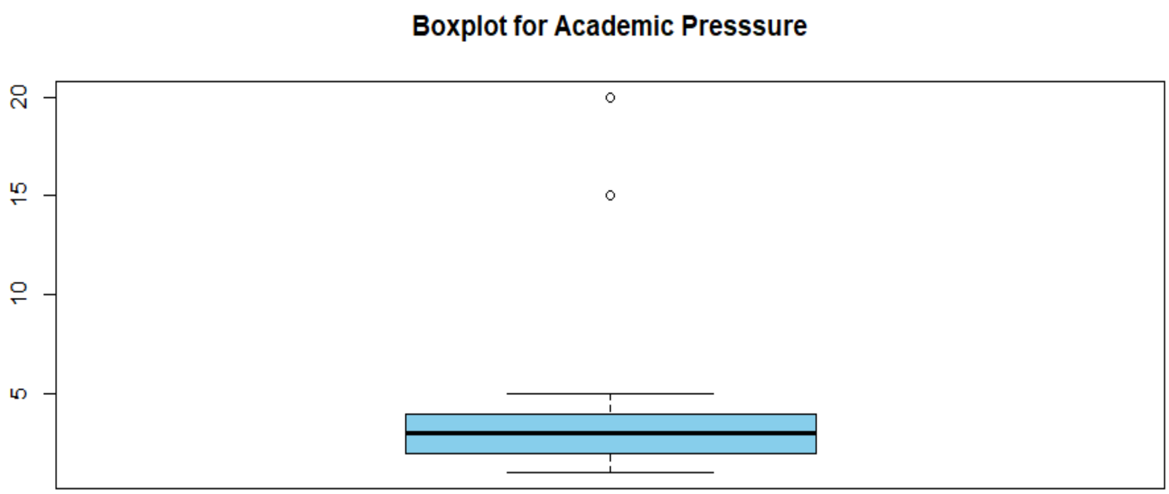
Explanation Part: The code creates a boxplot for the Age column in the mydata dataset. The boxplot() function visualizes the distribution of Age, showing the median, quartiles, and potential outliers. The main argument adds a title ("Boxplot for Age"), and the col argument colors the boxplot in "skyblue." This helps to understand the spread and potential outliers in the Age data.

Academic Pressure:

Code Part:

```
boxplot(mydata$Academic_Pressure, main = "Boxplot for Academic Presssure", col = "skyblue")
```

Output Part:



Explanation Part: The code generates a boxplot for the Academic_Pressure column in the mydata dataset. The boxplot() function displays the distribution of Academic_Pressure, including the median, quartiles, and any potential outliers. The main argument adds the title "Boxplot for Academic Pressure," and the col argument colors the boxplot in "skyblue." This provides a visual representation of how Academic_Pressure is distributed and helps identify any extreme values.

9.5 Replacing Outliers with Median:
Code Part:

```
mydata[sapply(mydata, is.numeric)] <- lapply(mydata[sapply(mydata, is.numeric)], function(x) {
  Q1 <- quantile(x, 0.25, na.rm = TRUE)
  Q3 <- quantile(x, 0.75, na.rm = TRUE)
  IQR <- Q3 - Q1
  x[x < (Q1 - 1.5 * IQR) | x > (Q3 + 1.5 * IQR)] <- median(x, na.rm = TRUE)
  return(x)
})
```

Output Part:

```
> mydata[sapply(mydata, is.numeric)] <- lapply(mydata[sapply(mydata, is.numeric)], function(x) {
+   Q1 <- quantile(x, 0.25, na.rm = TRUE)
+   Q3 <- quantile(x, 0.75, na.rm = TRUE)
+   IQR <- Q3 - Q1
+   x[x < (Q1 - 1.5 * IQR) | x > (Q3 + 1.5 * IQR)] <- median(x, na.rm = TRUE)
+   return(x)
+ })
> table(mydata$Age)

  18   19   20   21   22   23   24   25   26 26.5   27   28   29   30   31   32   33   34 
   8   10   16   10   10    6   14   13   13    2   11   16   15   10   12    7   16   11 
> table(mydata$Academic_Pressure)

 1  2  3  4  5 
36 43 44 37 40 
>
```

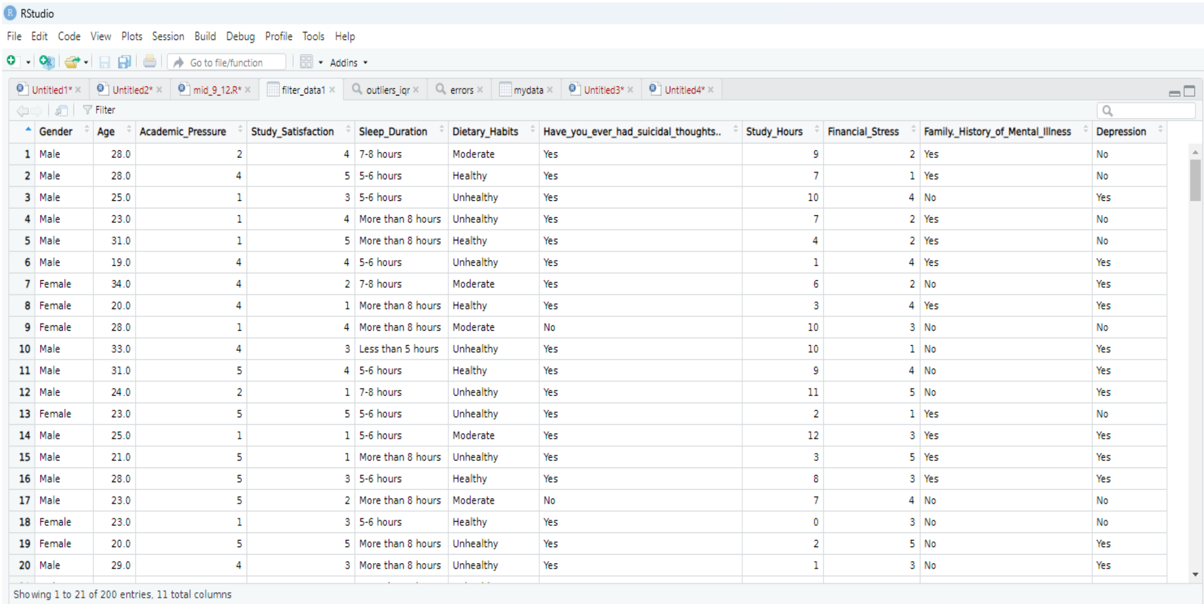
Explanation: The code processes the numeric columns in the mydata dataset by identifying and handling outliers. For each numeric column, it calculates the first and third quartiles (Q1 and Q3) and computes the Interquartile Range (IQR). Values that fall outside 1.5 times the IQR from the quartiles are considered outliers and are replaced with the median of the respective column. This approach ensures that extreme values do not skew the analysis, making the dataset more reliable and robust for further processing.

10. Filter Data:

10.1 Filter Data 1:

```
filter_data1 <- filter(mydata, Age >= 18 & Age <= 60)
```

Output Part:



	Gender	Age	Academic_Pressure	Study_Satisfaction	Sleep_Duration	Dietary_Habits	Have_you_ever_had_suicidal_thoughts..	Study_Hours	Financial_Stress	Family_History_of_Mental_Illness	Depression
1	Male	28.0	2	4	7-8 hours	Moderate	Yes	9	2	Yes	No
2	Male	28.0	4	5	5-6 hours	Healthy	Yes	7	1	Yes	No
3	Male	25.0	1	3	5-6 hours	Unhealthy	Yes	10	4	No	Yes
4	Male	23.0	1	4	More than 8 hours	Unhealthy	Yes	7	2	Yes	No
5	Male	31.0	1	5	More than 8 hours	Healthy	Yes	4	2	Yes	No
6	Male	19.0	4	4	5-6 hours	Unhealthy	Yes	1	4	Yes	Yes
7	Female	34.0	4	2	7-8 hours	Moderate	Yes	6	2	No	Yes
8	Female	20.0	4	1	More than 8 hours	Healthy	Yes	3	4	Yes	Yes
9	Female	28.0	1	4	More than 8 hours	Moderate	No	10	3	No	No
10	Male	33.0	4	3	Less than 5 hours	Unhealthy	Yes	10	1	No	Yes
11	Male	31.0	5	4	5-6 hours	Healthy	Yes	9	4	No	Yes
12	Male	24.0	2	1	7-8 hours	Unhealthy	Yes	11	5	No	Yes
13	Female	23.0	5	5	5-6 hours	Unhealthy	Yes	2	1	Yes	No
14	Male	25.0	1	1	5-6 hours	Moderate	Yes	12	3	Yes	Yes
15	Male	21.0	5	1	More than 8 hours	Unhealthy	Yes	3	5	Yes	Yes
16	Male	28.0	5	3	5-6 hours	Healthy	Yes	8	3	Yes	Yes
17	Male	23.0	5	2	More than 8 hours	Moderate	No	7	4	No	No
18	Female	23.0	1	3	5-6 hours	Healthy	Yes	0	3	No	No
19	Female	20.0	5	5	More than 8 hours	Unhealthy	Yes	2	5	No	Yes
20	Male	29.0	4	3	More than 8 hours	Unhealthy	Yes	1	3	No	Yes

Showing 1 to 21 of 200 entries, 11 total columns

Explanation: The code filters the mydata dataset to include only the rows where the Age column values are between 18 and 60, inclusive. The filter () function from the dplyr package is used to select the subset of data that meets the condition Age >= 18 & Age <= 60. This helps narrow down the dataset to individuals within the specified age range for further analysis.

10.2 Filter Data 2:

Code Part:

```
filter_data2 <- filter(mydata, Age >= 18 & Age <= 60 & Study_Hours > 5)
```

Output Part:

	Gender	Age	Academic_Pressure	Study_Satisfaction	Sleep_Duration	Dietary_Habits	Have_you_ever_had_suicidal_thoughts..	Study_Hours	Financial_Stress	Family_History_of_Mental_Illness	Depression
1	Male	28.0	2	4	7-8 hours	Moderate	Yes	9	2	Yes	No
2	Male	28.0	4	5	5-6 hours	Healthy	Yes	7	1	Yes	No
3	Male	25.0	1	3	5-6 hours	Unhealthy	Yes	10	4	No	Yes
4	Male	23.0	1	4	More than 8 hours	Unhealthy	Yes	7	2	Yes	No
5	Female	34.0	4	2	7-8 hours	Moderate	Yes	6	2	No	Yes
6	Female	28.0	1	4	More than 8 hours	Moderate	No	10	3	No	No
7	Male	33.0	4	3	Less than 5 hours	Unhealthy	Yes	10	1	No	Yes
8	Male	31.0	5	4	5-6 hours	Healthy	Yes	9	4	No	Yes
9	Male	24.0	2	1	7-8 hours	Unhealthy	Yes	11	5	No	Yes
10	Male	25.0	1	1	5-6 hours	Moderate	Yes	12	3	Yes	Yes
11	Male	28.0	5	3	5-6 hours	Healthy	Yes	8	3	Yes	Yes
12	Male	23.0	5	2	More than 8 hours	Moderate	No	7	4	No	No
13	Male	31.0	2	4	More than 8 hours	Unhealthy	No	10	1	No	No
14	Female	33.0	3	2	7-8 hours	Moderate	No	11	5	Yes	No
15	Female	33.0	2	3	7-8 hours	Moderate	Yes	12	5	Yes	Yes
16	Male	21.0	5	3	7-8 hours	Unhealthy	No	6	4	Yes	Yes
17	Female	34.0	3	4	Less than 5 hours	Unhealthy	Yes	12	1	Yes	No
18	Male	32.0	5	2	5-6 hours	Moderate	Yes	12	3	No	Yes
19	Female	21.0	5	3	7-8 hours	Moderate	No	9	2	No	Yes
20	Male	26.0	5	1	Less than 5 hours	Unhealthy	Yes	10	5	No	Yes

Showing 1 to 21 of 116 entries, 11 total columns

Explanation: The code filters the mydata dataset to include only the rows where the Age is between 18 and 60, and the Study Hours is greater than 5. The filter () function from the dplyr package is used with the condition Age >= 18 & Age <= 60 & study Hours > 5. This allows you to focus on individuals within the specified age range who also study more than 5 hours, refining the dataset for specific analysis.

10.3 Filter Data 3:

Code Part:

```
filter_data3 <- filter(mydata, Age >= 18 & Age <= 60 & Study_Hours < 5)
```

Output Part:

Untitled1* | Untitled2* | mid_9_12.R* | filter_data3 | filter_data2 | filter_data1 | outliers_iqr | errors | mydata | Untitled3* | Untitled4*

Filter

	Gender	Age	Academic_Pressure	Study_Satisfaction	Sleep_Duration	Dietary_Habits	Have_you_ever_had_suicidal_thoughts..	Study_Hours	Financial_Stress	Family_History_of_Mental_Illness	Depression
1	Male	31.0	1	5	More than 8 hours	Healthy	Yes	4	2	Yes	No
2	Male	19.0	4	4	5-6 hours	Unhealthy	Yes	1	4	Yes	Yes
3	Female	20.0	4	1	More than 8 hours	Healthy	Yes	3	4	Yes	Yes
4	Female	23.0	5	5	5-6 hours	Unhealthy	Yes	2	1	Yes	No
5	Male	21.0	5	1	More than 8 hours	Unhealthy	Yes	3	5	Yes	Yes
6	Female	23.0	1	3	5-6 hours	Healthy	Yes	0	3	No	No
7	Female	20.0	5	5	More than 8 hours	Unhealthy	Yes	2	5	No	Yes
8	Male	29.0	4	3	More than 8 hours	Unhealthy	Yes	1	3	No	Yes
9	Male	31.0	2	3	More than 8 hours	Unhealthy	No	3	3	Yes	No
10	Male	24.0	3	4	More than 8 hours	Healthy	Yes	1	3	No	No
11	Male	31.0	2	2	7-8 hours	Healthy	No	2	4	Yes	No
12	Male	30.0	3	4	7-8 hours	Moderate	Yes	0	2	Yes	No
13	Female	29.0	3	5	Less than 5 hours	Moderate	Yes	4	3	Yes	No
14	Female	20.0	3	2	More than 8 hours	Healthy	No	2	2	No	No
15	Female	33.0	2	5	Less than 5 hours	Moderate	Yes	3	3	Yes	No
16	Male	26.0	5	4	7-8 hours	Unhealthy	No	3	2	Yes	No
17	Female	28.0	1	1	Less than 5 hours	Healthy	No	2	2	No	No
18	Male	26.0	4	5	Less than 5 hours	Unhealthy	No	4	1	Yes	No
19	Male	21.0	3	2	7-8 hours	Unhealthy	Yes	1	5	No	Yes
20	Female	24.0	1	3	5-6 hours	Moderate	No	3	5	No	No

Showing 1 to 21 of 72 entries, 11 total columns

Explanation: The code filters the mydata dataset to include only the rows where the Age is between 18 and 60, and the study Hours is less than 5. The filter () function from the dplyr package is applied with the condition Age >= 18 & Age <= 60 & study Hours < 5. This creates a subset of the dataset focusing on individuals within the specified age range who study less than 5 hours, which can be used for further analysis.

11. Convert Attribute

11.1 Categorical to Numeric:

For Sleep Duration:

Code Part:

```
mydata$Sleep_Duration <- factor(mydata$Sleep_Duration,
                                levels = c("Less than 5 hours", "5-6 hours", "7-8 hours", "More than 8 hours"),labels = c(4,5.5,7.5,9))


table(mydata$Sleep_Duration)
```

Output Part:

```
> mydata$Sleep_Duration <- factor(mydata$Sleep_Duration,
+                              levels = c("Less than 5 hours", "5-6 hours", "7-8 hours", "More than 8 hours"),labels = c(4,5.5,7.5,9))
>
> table(mydata$Sleep_Duration)

  4 5.5 7.5   9 
 48  47  51  55 
>
```

	Gender	Age	Academic_Pressure	Study_Satisfaction	Sleep_Duration	Dietary_Habits	Have_you_ever_had_suicidal_thoughts.?
1	Male	28	2	4	7.5	Moderate	Yes
2	Male	28	4	5	5.5	Healthy	Yes
3	Male	25	1	3	5.5	Unhealthy	Yes
4	Male	23	1	4	9	Unhealthy	Yes
5	Male	31	1	5	9	Healthy	Yes
6	Male	19	4	4	5.5	Unhealthy	Yes
7	Female	34	4	2	9	Moderate	Yes
8	Female	20	4	1	9	Healthy	Yes
9	Female	28	1	4	9	Moderate	No
10	Male	33	4	3	4	Unhealthy	Yes
11	Male	31	5	4	5.5	Healthy	Yes
12	Male	24	2	1	7.5	Unhealthy	Yes
13	Female	23	5	5	9	Unhealthy	Yes
14	Male	25	1	1	5.5	Moderate	Yes
15	Male	21	5	1	9	Unhealthy	Yess
16	Male	28	5	3	5.5	Healthy	Yes
17	Male	23	5	2	9	Moderate	No
18	Female	23	1	3	9	Healthy	Yes

Explanation: This code converts the Sleep Duration column in mydata into a factor with specific levels mapped to numeric labels (4, 5.5, 7.5, 9). The table () function then summarizes the counts of each factor level, showing how many entries correspond to each sleep category.

For Gender:

Code Part:

```
mydata$Gender <- factor(mydata$Gender, levels = c("Male", "Female"),labels = c(1,2))
table(mydata$Gender)
```

Output Part:

```
> mydata$Gender <- factor(mydata$Gender, levels = c("Male", "Female"),labels = c(1,2))
> table(mydata$Gender)

  1   2 
112  88 
> 
```

Explanation:
The code converts the gender columns in the mydata dataset. For Gender, "Male" is labeled as 1 and "Female" as 2. The table () function is then used to display the frequency of each category in gender columns.

11.2 Numeric to Categorical:

Code Part:

```
mydata$Academic_Pressure <- factor(mydata$Academic_Pressure,
                                   levels = c(1, 2, 3, 4, 5),
                                   labels = c("Very Low", "Low", "Moderate", "High", "Very High"))
table(mydata$Academic_Pressure)
```

Output Part:

```
> mydata$Academic_Pressure <- factor(mydata$Academic_Pressure,
+                                    levels = c(1, 2, 3, 4, 5),
+                                    labels = c("Very Low", "Low", "Moderate", "High", "Very High"))
> table(mydata$Academic_Pressure)

 Very Low       Low  Moderate      High Very High 
       36        43        44        37        40 
> 
```

Untitled1* | Untitled2* | mid_9_12.R* | filter_data3 | filter_data2 | filter_data1 | outliers_iqr | errors | mydata | Untitled3* | Untitled4*

Filter

	Gender	Age	Academic_Pressure	Study_Satisfaction	Sleep_Duration	Dietary_Habits	Have_you_ever_had_suicidal_thoughts.?	Study_Hours	Financial_Stress	Family._History_of_Mental_Illness	Depression
1	1	28	Low	4	7-8 hours	2	Yes	9	2	Yes	No
2	1	28	High	5	5-6 hours	3	Yes	7	1	Yes	No
3	1	25	Very Low	3	5-6 hours	1	Yes	10	4	No	Yes
4	1	23	Very Low	4	More than 8 hours	1	Yes	7	2	Yes	No
5	1	31	Very Low	5	More than 8 hours	3	Yes	4	2	Yes	No
6	1	19	High	4	5-6 hours	1	Yes	1	4	Yes	Yes
7	2	34	High	2	More than 8 hours	2	Yes	6	2	No	Yes
8	2	20	High	1	More than 8 hours	3	Yes	3	4	Yes	Yes
9	2	28	Very Low	4	More than 8 hours	2	No	10	3	No	No
10	1	33	High	3	Less than 5 hours	1	Yes	10	1	No	Yes
11	1	31	Very High	4	5-6 hours	3	Yes	6	4	No	Yes
12	1	24	Low	1	7-8 hours	1	Yes	11	5	No	Yes
13	2	23	Very High	5	More than 8 hours	1	Yes	2	1	Yes	No
14	1	25	Very Low	1	5-6 hours	2	Yes	12	3	Yes	Yes
15	1	21	Very High	1	More than 8 hours	1	Yess	3	5	Yes	Yes
16	1	28	Very High	3	5-6 hours	3	Yes	8	3	Yes	Yes
17	1	23	Very High	2	More than 8 hours	2	No	6	4	No	No
18	2	23	Very Low	3	More than 8 hours	3	Yes	0	3	No	No
19	2	20	Very High	5	More than 8 hours	1	Yes	2	5	No	Yes
20	1	29	High	3	More than 8 hours	1	Yes	1	3	No	Yes

Showing 1 to 21 of 200 entries, 11 total columns

Explanation: The code converts the Academic_Pressure column in the mydata dataset from numeric values to descriptive labels. The numeric values 1 to 5 are mapped to "Very Low", "Low", "Moderate", "High", and "Very High". The table () function then shows the frequency of each category.

12. Normalization:

Code Part:

```
minMax <- function(x) {
  (x - min(x)) / (max(x) - min(x))
}
mydata$Gender <- as.numeric(as.factor(mydata$Gender))

mydata$Gender <- minMax(mydata$Gender)
```

Output Part:

	Gender	Age	Academic_Pressure	Study_Satisfaction	Sleep_Duration	Dietary_Habits	Have_you_ever_had_suicidal_thoughts.?	Study_Hours	Financial_Stress	Family_History_of_Mental_Illness	Depression
1	0	28	Low	4	7-8 hours	2	Yes	9	2	Yes	No
2	0	28	High	5	5-6 hours	3	Yes	7	1	Yes	No
3	0	25	Very Low	3	5-6 hours	1	Yes	10	4	No	Yes
4	0	23	Very Low	4	More than 8 hours	1	Yes	7	2	Yes	No
5	0	31	Very Low	5	More than 8 hours	3	Yes	4	2	Yes	No
6	0	19	High	4	5-6 hours	1	Yes	1	4	Yes	Yes
7	1	34	High	2	More than 8 hours	2	Yes	6	2	No	Yes
8	1	20	High	1	More than 8 hours	3	Yes	3	4	Yes	Yes
9	1	28	Very Low	4	More than 8 hours	2	No	10	3	No	No
10	0	33	High	3	Less than 5 hours	1	Yes	10	1	No	Yes
11	0	31	Very High	4	5-6 hours	3	Yes	6	4	No	Yes
12	0	24	Low	1	7-8 hours	1	Yes	11	5	No	Yes
13	1	23	Very High	5	More than 8 hours	1	Yes	2	1	Yes	No
14	0	25	Very Low	1	5-6 hours	2	Yes	12	3	Yes	Yes
15	0	21	Very High	1	More than 8 hours	1	Yess	3	5	Yes	Yes
16	0	28	Very High	3	5-6 hours	3	Yes	8	3	Yes	Yes
17	0	23	Very High	2	More than 8 hours	2	No	6	4	No	No
18	1	23	Very Low	3	More than 8 hours	3	Yes	0	3	No	No
19	1	20	Very High	5	More than 8 hours	1	Yes	2	5	No	Yes
20	0	29	High	3	More than 8 hours	1	Yes	1	3	No	Yes

Showing 1 to 21 of 200 entries, 11 total columns

Explanation: The code defines a minMax() function to normalize data between 0 and 1. It then converts the Gender column from categorical values ("Male" and "Female") into numeric codes. After that, the minMax() function is applied to the Gender column to scale the values between 0 and 1.

13. Conclusion:

In this dataset, initially there were missing values, outliers, invalid data and many more issues. So, the initial dataset is given below:

	Gender	Age	Academic_Pressure	Study_Satisfaction	Sleep_Duration	Dietary_Habits	Have_you_ever_had_suicidal_thoughts.?	Study_Hours	Financial_Stress	Family_History_of_Mental_Illness	Depression
1	Male	28	2	4	7-8 hours	Moderate	Yes	9	2	Yes	No
2	Male	28	4	5	5-6 hours	Healthy	Yes	7	1	Yes	No
3	Male	25	1	3	5-6 hours	Unhealthy	Yes	10	4	No	Yes
4	Male	23	1	4	More than 8 hours	Unhealthy	Yes	7	2	Yes	No
5	NA	31	1	5	More than 8 hours	Healthy	Yes	4	2	Yes	No
6	Male	19	4	4	5-6 hours	Unhealthy	Yes	1	4	Yes	Yes
7	Female	34	4	2	NA	Moderate	Yes	6	2	No	Yes
8	Female	20	4	1	More than 8 hours	Healthy	Yes	3	4	Yes	Yes
9	Female	NA	1	4	More than 8 hours	Moderate	No	10	3	No	No
10	Male	33	4	3	Less than 5 hours	Unhealthy	Yes	10	1	No	Yes
11	NA	31	5	4	5-6 hours	Healthy	Yes	NA	4	No	Yes
12	Male	24	2	1	7-8 hours	Unhealthy	Yes	11	5	No	Yes
13	Female	23	5	5	NA	Unhealthy	Yes	2	1	Yes	NA
14	Male	25	1	1	5-6 hours	Moderate	Yes	12	3	Yes	Yes
15	Male	21	5	1	More than 8 hours	Unhealthy	Yess	3	5	Yes	Yes
16	Male	28	5	3	5-6 hours	Healthy	Yes	8	3	Yes	Yes
17	Male	23	5	2	More than 8 hours	Moderate	No	NA	4	No	NA
18	Female	23	1	3	NA	Healthy	Yes	0	3	No	No

And this is after applying data preparation steps, our dataset looks like this:

	Gender	Age	Academic_Pressure	Study_Satisfaction	Sleep_Duration	Dietary_Habits	Have_you_ever_had_suicidal_thoughts..	Study_Hours	Financial_Stress	Family_History_of_Mental_Illness	Depression
1	1	28	Low	4	7.5	Moderate	Yes	9	2	Yes	No
2	1	28	High	5	5.5	Healthy	Yes	7	1	Yes	No
3	1	25	Very Low	3	5.5	Unhealthy	Yes	10	4	No	Yes
4	1	23	Very Low	4	9	Unhealthy	Yes	7	2	Yes	No
5	1	31	Very Low	5	9	Healthy	Yes	4	2	Yes	No
6	1	19	High	4	5.5	Unhealthy	Yes	1	4	Yes	Yes
7	0	34	High	2	9	Moderate	Yes	6	2	No	Yes
8	0	20	High	1	9	Healthy	Yes	3	4	Yes	Yes
9	0	28	Very Low	4	9	Moderate	No	10	3	No	No
10	1	33	High	3	4	Unhealthy	Yes	10	1	No	Yes
11	1	31	Very High	4	5.5	Healthy	Yes	6	4	No	Yes
12	1	24	Low	1	7.5	Unhealthy	Yes	11	5	No	Yes
13	0	23	Very High	5	9	Unhealthy	Yes	2	1	Yes	No
14	1	25	Very Low	1	5.5	Moderate	Yes	12	3	Yes	Yes
15	1	21	Very High	1	9	Unhealthy	Yes	3	5	Yes	Yes
16	1	28	Very High	3	5.5	Healthy	Yes	8	3	Yes	Yes
17	1	23	Very High	2	9	Moderate	No	6	4	No	No
18	0	23	Very Low	3	9	Healthy	Yes	0	3	No	No