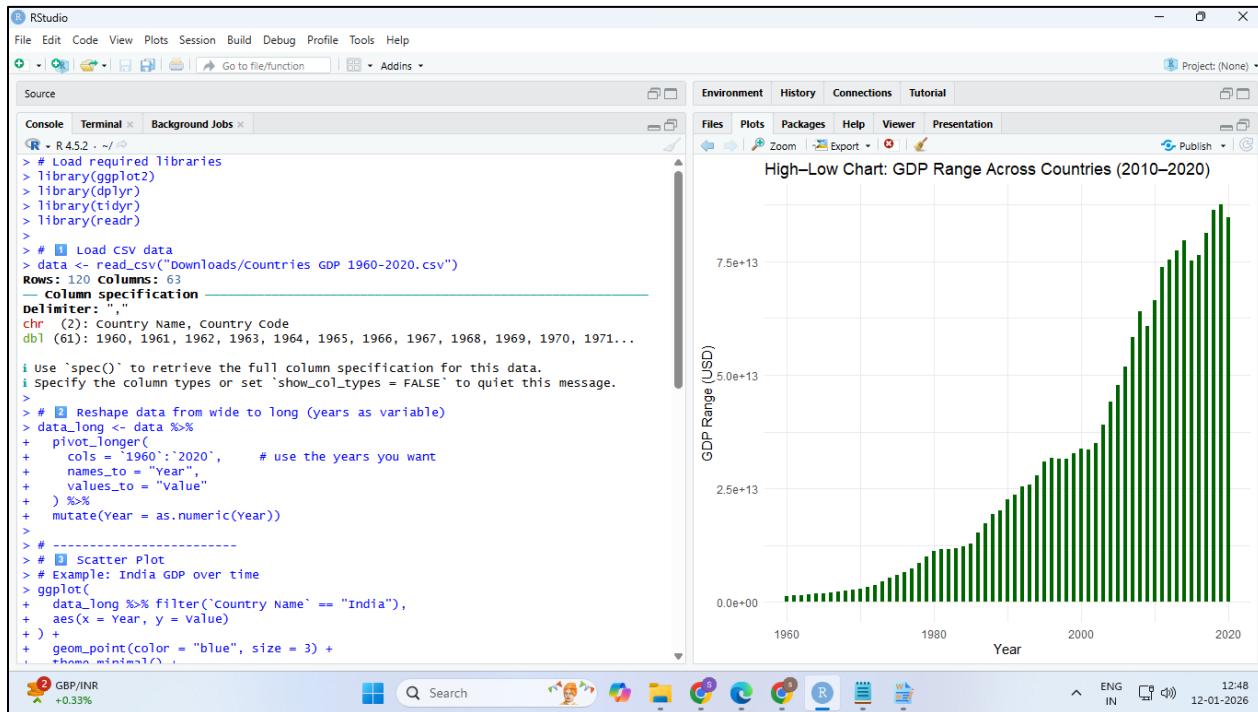


PRACTICAL : 10

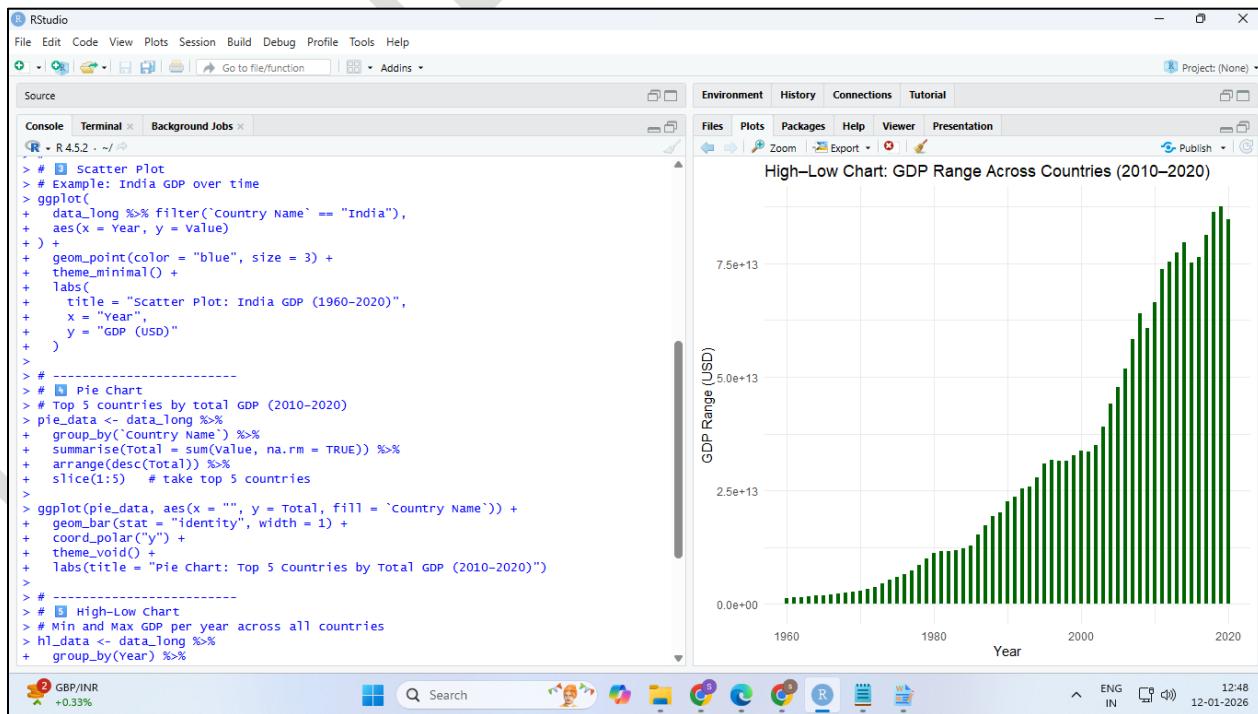
AIM : Creating graphical reports using `ggplot2` (R).

- Scatter Plots
- Pie Charts
- High-Low Chart



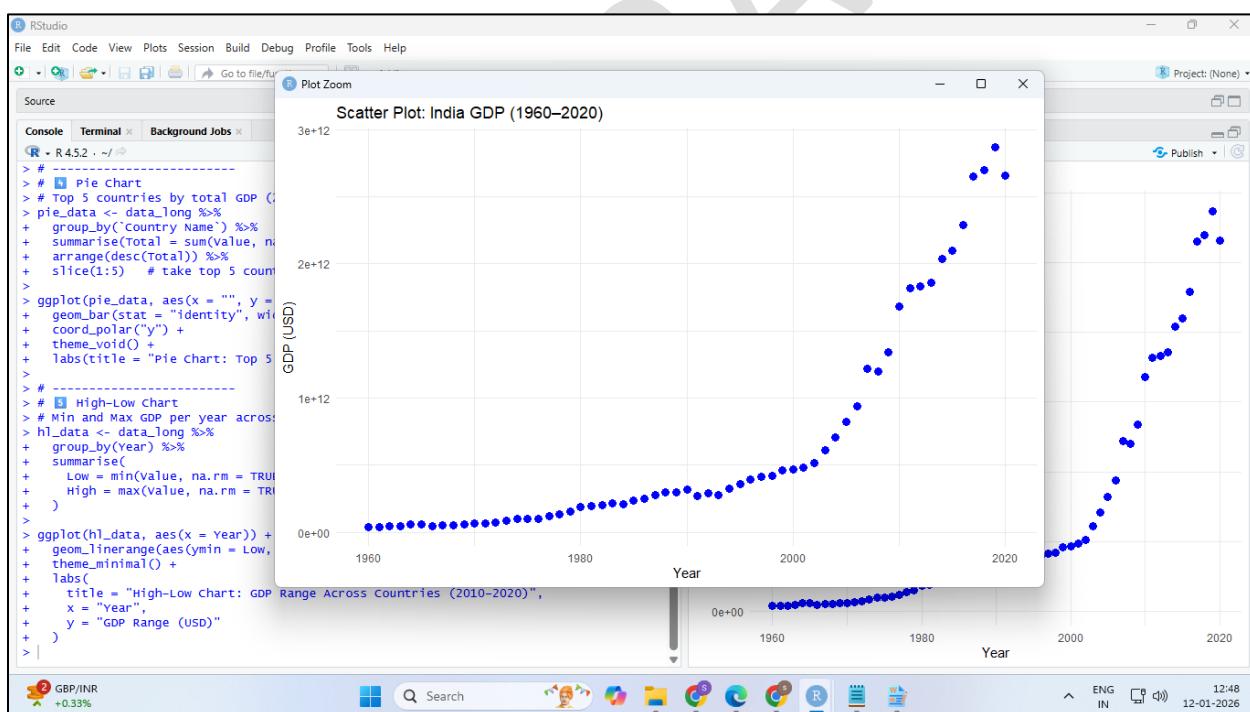
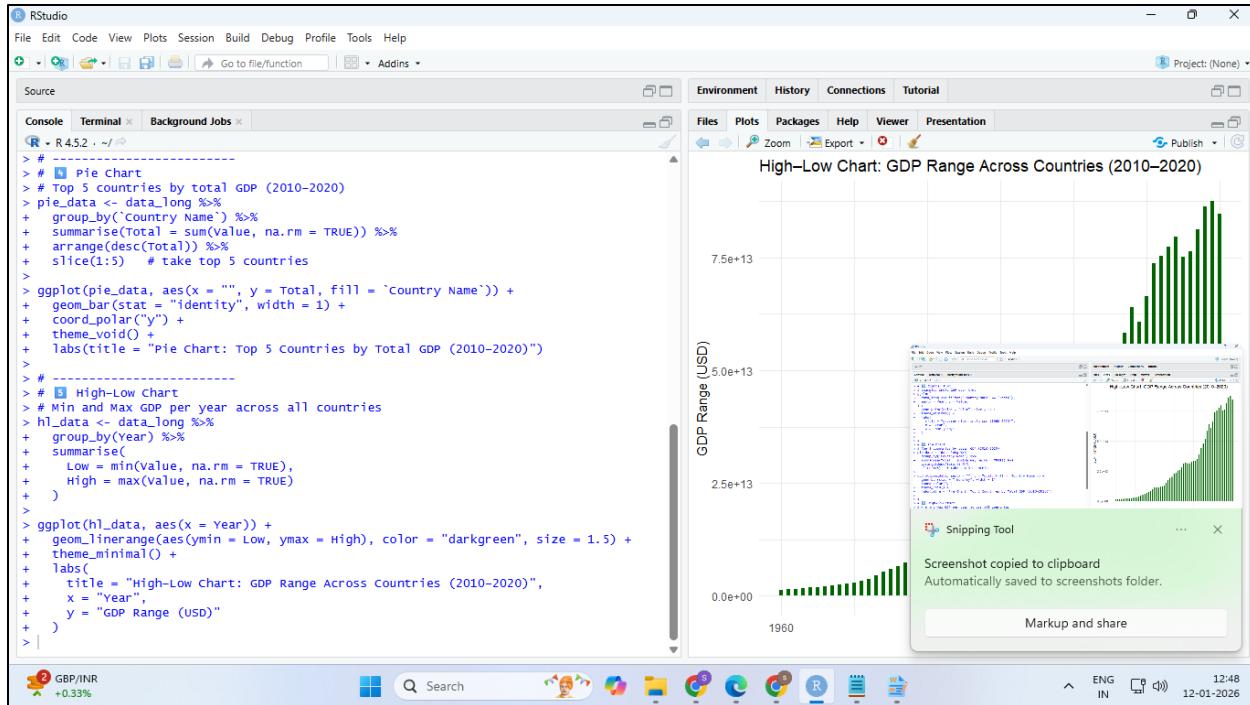
The screenshot shows the RStudio interface with the following details:

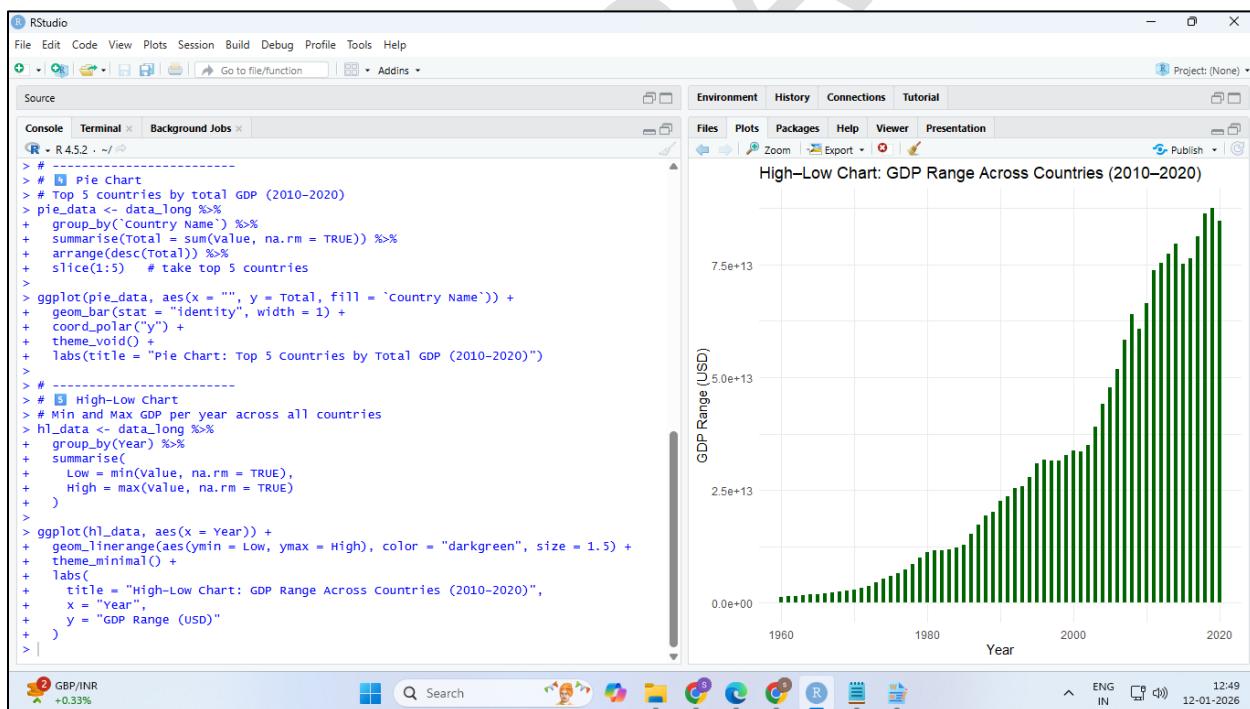
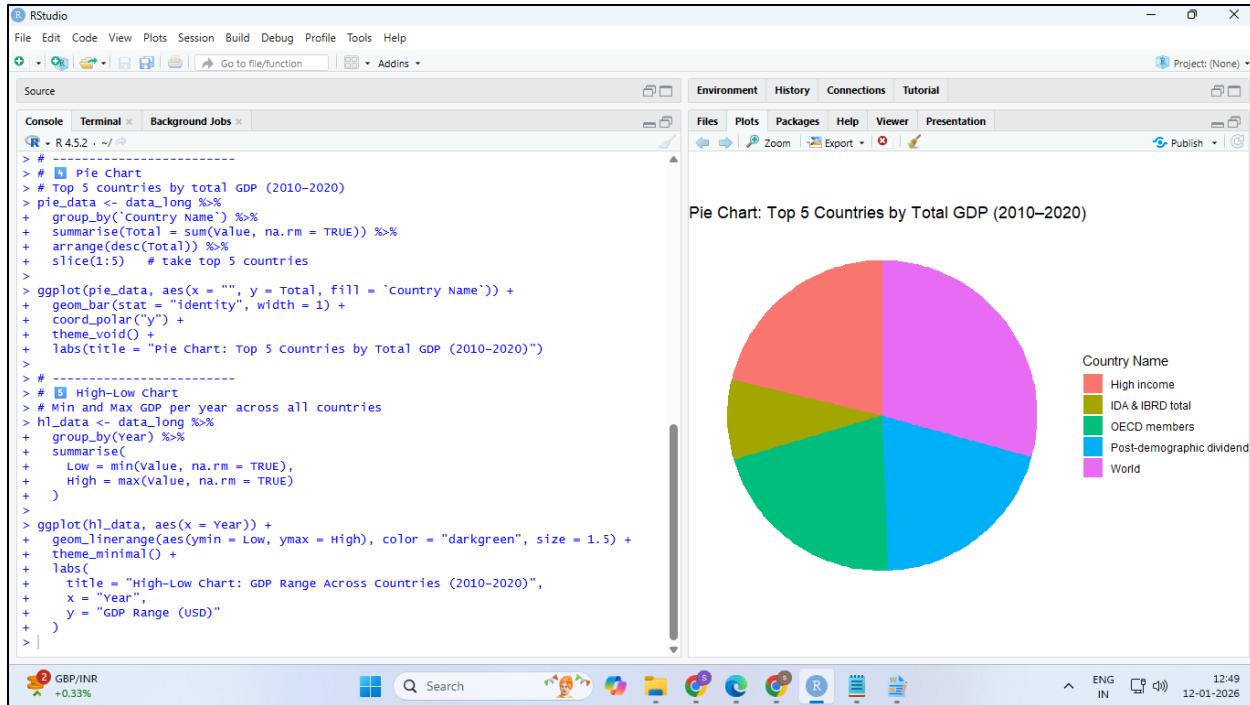
- Console:** Displays R code for loading libraries (`library(ggplot2)`, `library(dplyr)`, `library(tidyverse)`, `library(readr)`), reading CSV data (`data <- read_csv("Downloads/Countries GDP 1960–2020.csv")`), reshaping the data from wide to long format (`data_long <- data %>% pivot_longer(...)`), and creating a scatter plot for India's GDP over time (`ggplot(data_long %>% filter(`country Name` == "India"), aes(x = Year, y = value)) + geom_point(color = "blue", size = 3) + theme_minimal()`).
- Environment:** Shows the data frame `data` with 120 rows and 63 columns.
- Plots:** Displays a bar chart titled "High-Low Chart: GDP Range Across Countries (2010–2020)". The Y-axis is labeled "GDP Range (USD)" and ranges from 0.0e+00 to 7.5e+13. The X-axis is labeled "Year" and ranges from 1960 to 2020. The bars show a significant increase in GDP range over time, starting near zero in 1960 and reaching approximately 7.5e+13 by 2020.
- System:** The taskbar at the bottom shows various application icons, and the system tray indicates the date and time as 12-01-2026, 12:48.



The screenshot shows the RStudio interface with the following details:

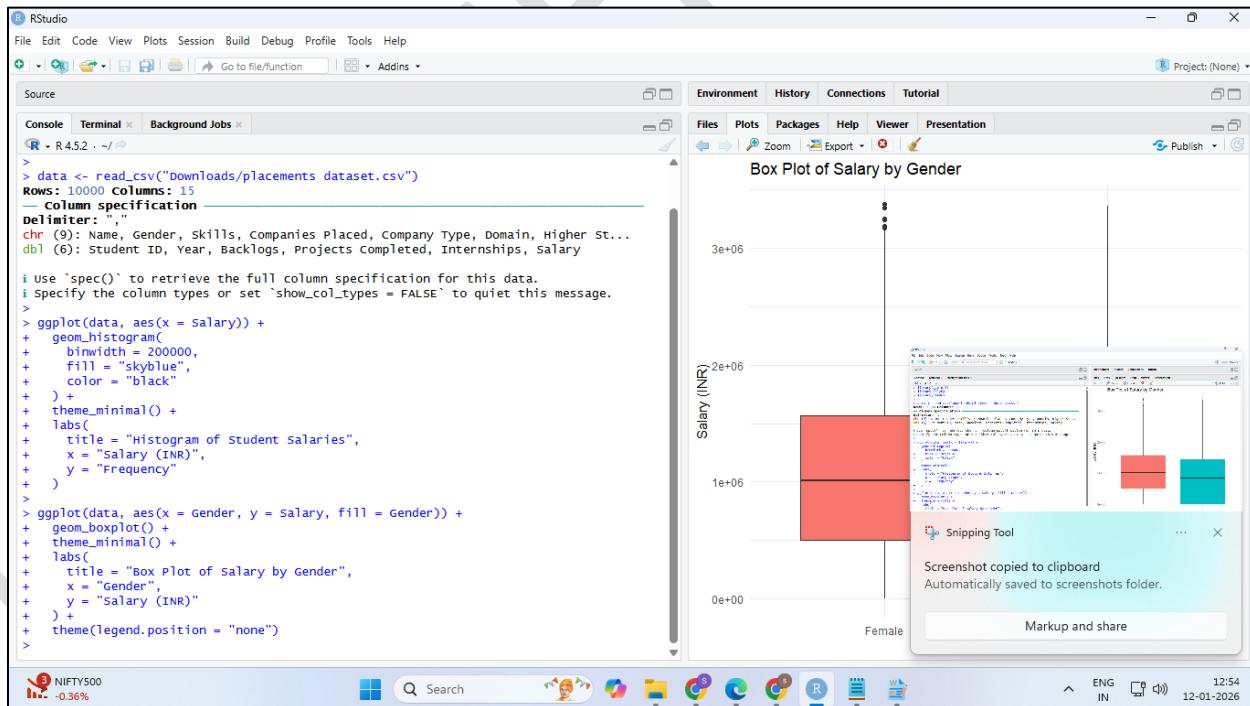
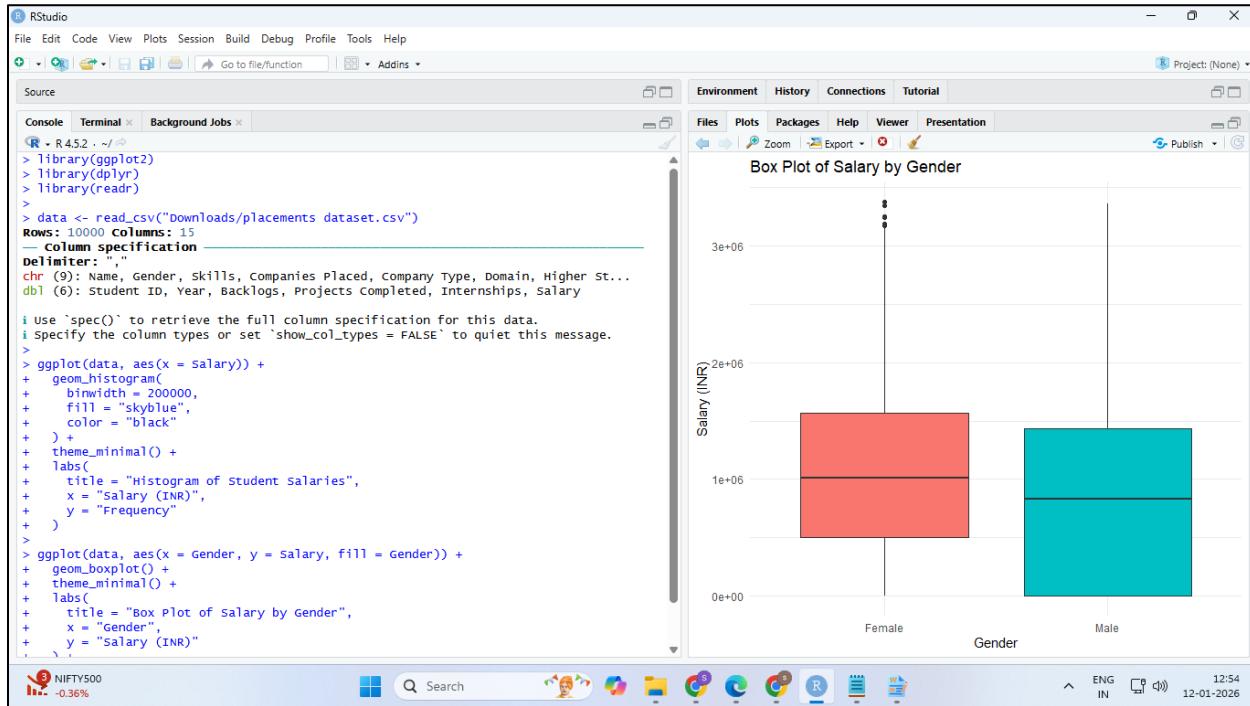
- Console:** Displays the completed R code for creating a scatter plot for India's GDP over time, a pie chart for the top 5 countries by total GDP, and a high-low chart for the minimum and maximum GDP per year across all countries.
- Environment:** Shows the data frame `data` with 120 rows and 63 columns.
- Plots:** Displays the same bar chart as the previous screenshot, titled "High-Low Chart: GDP Range Across Countries (2010–2020)".
- System:** The taskbar at the bottom shows various application icons, and the system tray indicates the date and time as 12-01-2026, 12:48.

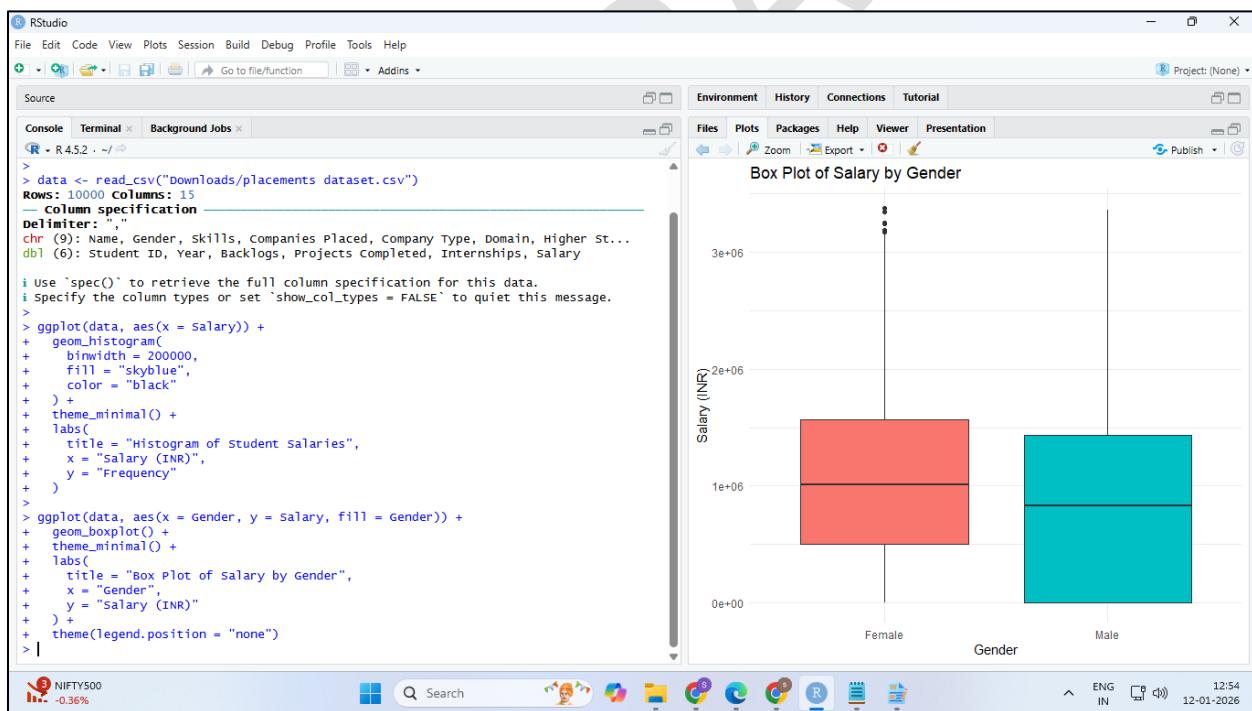
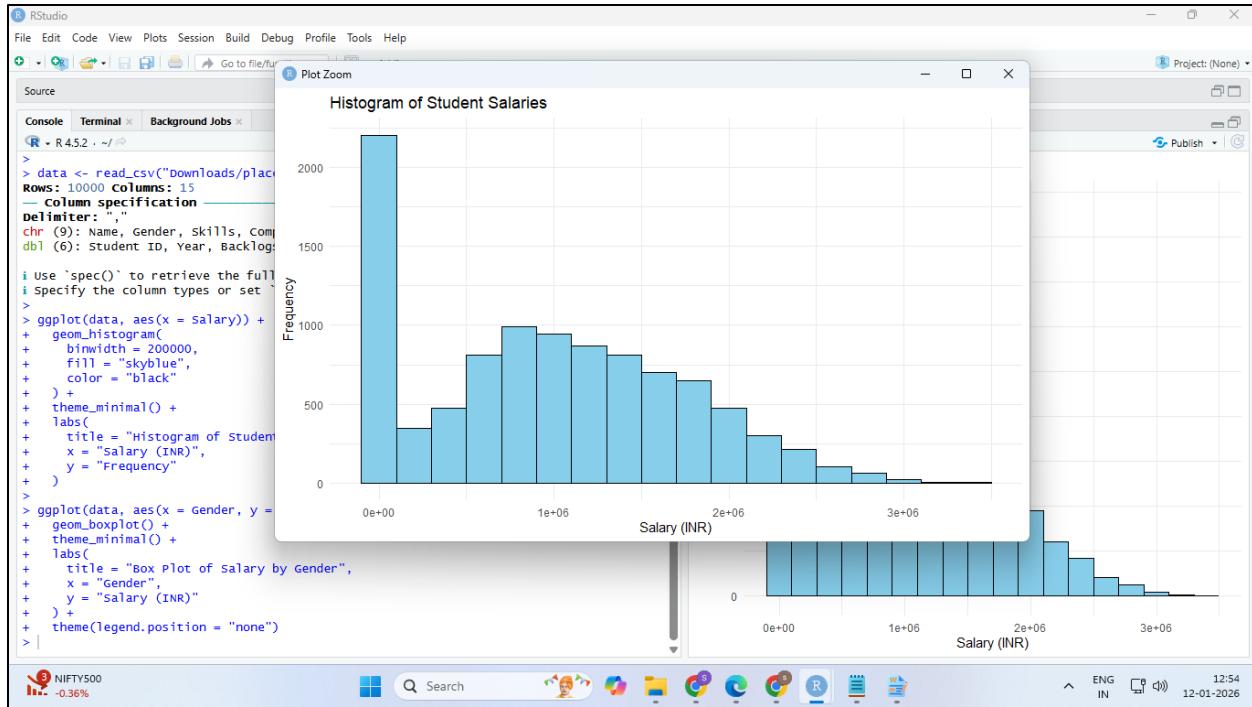




PRACTICAL : 11

Aim : Generating histograms and box plots using ggplot2 (R).





Practical : 12

AIM : Generating correlation matrices using `cor()` (R).

The screenshot shows the RStudio interface with the following details:

- File Menu:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Addins:** Go to function, Addins.
- Source:** Environment, History, Connections, Tutorial.
- Console:** R 4.5.2, showing the following R code and output:

```
> library(readr)
>
> data <- read_csv("Downloads/Placement_Data_Full_Class.csv")
Rows: 215 Columns: 15
---column specification---
`delimiter: ","
chr (8): gender, ssc_b, hsc_b, hsc_s, degree_t, workex, specialisation, status
dbl (7): sl_no, ssc_p, hsc_p, degree_p, etest_p, mba_p, salary

i use `spec()` to retrieve the full column specification for this data.
i specify the column types or set `show_col_types = FALSE` to quiet this message.
>
> numeric_data <- data[sapply(data, is.numeric)]
>
> correlation_matrix <- cor(
+   numeric_data,
+   use = "complete.obs",
+   method = "pearson"
+ )
>
> print(correlation_matrix)
```

Output of the correlation matrix:

	sl_no	ssc_p	hsc_p	degree_p	etest_p	mba_p	salary
sl_no	1.00000000	-0.09347968	-0.21842843	-0.10224982	0.04146745	-0.07243202	0.06376351
ssc_p	-0.09347968	1.00000000	0.29341586	0.38065683	0.31789178	0.43055982	0.03533034
hsc_p	-0.21842843	0.29341586	1.00000000	0.22130692	0.28467174	0.32998281	0.07681903
degree_p	-0.10224982	0.38065683	0.22130692	1.00000000	0.21768312	0.49409291	-0.01927223
etest_p	0.04146745	0.31789178	0.28467174	0.21768312	1.00000000	0.28414332	0.1780721
mba_p	-0.07243202	0.43055982	0.32998281	0.49409291	0.28414332	1.00000000	0.17501294
salary	0.06376351	0.03533034	0.07681903	-0.01927223	0.1780721	0.17501294	1.00000000

Below the heatmap, there are several small icons representing different R packages or tools.

The screenshot shows an RStudio interface with a correlation matrix heatmap. The console window displays the following R code and output:

```
> numeric_data <- data[sapply(data, is.numeric)]
> correlation_matrix <- cor(
+   numeric_data,
+   use = "complete.obs",
+   method = "pearson"
+ )
> print(correlation_matrix)
      sl_no    ssc_p    hsc_p degree_p etest_p    mba_p
sl_no  1.0000000 -0.09347968 -0.21842843 -0.10224982  0.04146745 -0.07243202
ssc_p  -0.09347968  1.00000000  0.29341586  0.38065683  0.31789178  0.43055982
hsc_p  -0.21842843  0.29341586  1.00000000  0.22130692  0.28467174  0.32998281
degree_p -0.10224982  0.38065683  0.22130692  1.00000000  0.21768312  0.49409291
etest_p  0.04146745  0.31789178  0.28467174  0.21768312  1.00000000  0.28414332
mba_p   0.07243202  0.43055982  0.32998281  0.49409291  0.28414332  1.00000000
salary   0.06376351  0.03533034  0.07681903 -0.01927223  0.17830731  0.17501294
      salary
sl_no   0.06376351
ssc_p   0.03533034
hsc_p   0.07681903
degree_p -0.01927223
etest_p  0.17830731
mba_p   0.17501294
salary   1.00000000
>
> heatmap(
+   correlation_matrix,
+   main = "Correlation Matrix Heatmap",
+   col = heat.colors(64),
+   symm = TRUE
+ )
>
```

The heatmap shows a strong positive correlation between 'sl_no' and 'salary' (approx. 0.175). The main diagonal is white, indicating perfect correlation with itself. The color scale ranges from white (low correlation) to dark red (high correlation).

