Day 3 Lab Manual Part 2

NAME: KRISHVANTH KUMAR E

REGISTER NUMBER:192125063

BIVARIATEANALYSIS IN R -COVARIAN

CE, CORRELATION, CROSSTAB

Exercise: 8

Reference Status Gender TestNewOrFollowUp

1 KRXH Accepted Female Test1 New

2 KRPT Accepted Male Test1 New

3 FHRA Rejected Male Test2 New

4 CZKK Accepted Female Test3 New

5 CQTN Rejected Female Test1 New

6 PZXW Accepted Female Test4 Follow-up

7 SZRZ Rejected Male Test4 New

8 RMZE Rejected Female Test2 New

9 STNX Accepted Female Test3 New

10 TMDW Accepted Female Test1 New

i) Load the dataset and Create a data frame and name it as dataframe1

ii) Load the function for crosstab

Note: Perform status+gender

Gender

Status Female Male

Accepted 5 1

Rejected 2 2

Note: Reference+Status

Status

Reference Accepted Rejected

CQTN 0 1

CZKK 1 0

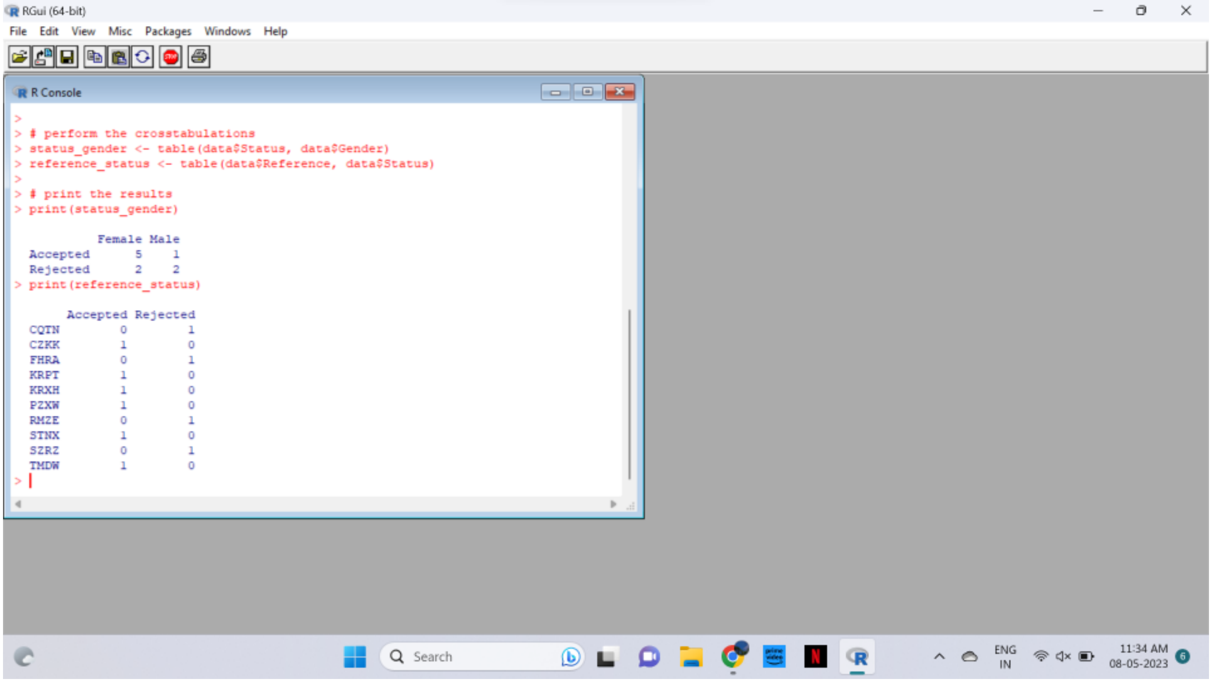
FHRA 0 1

KRPT 1 0

KRXH	1	0
PZXW	1	0
RMZE	0	1
STNX	1	0
SZRZ	0	1
TMDW	1	0

CODE

```
# create the data frame
data <- data.frame(
  Reference = c("KRXH", "KRPT", "FHRA", "CZKK", "CQTN", "PZXW", "SZRZ", "RMZE", "STNX", "TMDW"),
  Status = c("Accepted", "Accepted", "Rejected", "Accepted", "Rejected", "Accepted", "Rejected", "Rejected", "Accepted", "Accepted"),
  Gender = c("Female", "Male", "Male", "Female", "Female", "Female", "Male", "Female", "Female", "Female"),
  TestNewOrFollowUp = c("Test1 New", "Test1 New", "Test2 New", "Test3 New", "Test1 New", "Follow-up", "Test4 New", "Test2 New", "Test3 New", "Test1 New")
)

# perform the crosstabulations
status_gender <- table(data$Status, data$Gender)
reference_status <- table(data$Reference, data$Status)

# print the results
print(status_gender)
print(reference_status)
```



```
>
> # perform the crosstabulations
> status_gender <- table(data$Status, data$Gender)
> reference_status <- table(data$Reference, data$Status)
>
> # print the results
> print(status_gender)

          Female Male
  Accepted      5    1
  Rejected      2    2
> print(reference_status)

       Accepted Rejected
  CQTN        0        1
  CZKK        1        0
  FHRA        0        1
  KRPT        1        0
  KRXH        1        0
  PZXW        1        0
  RMZE        0        1
  STNX        1        0
  SZRZ        0        1
  TMDW        1        0
>
```

Exercise: 9

i) Use Two Categorical Variables and Discover the relationships within a

dataset

ii) Next, using the xtabs() function, apply two variables from “dataframe1 “, to

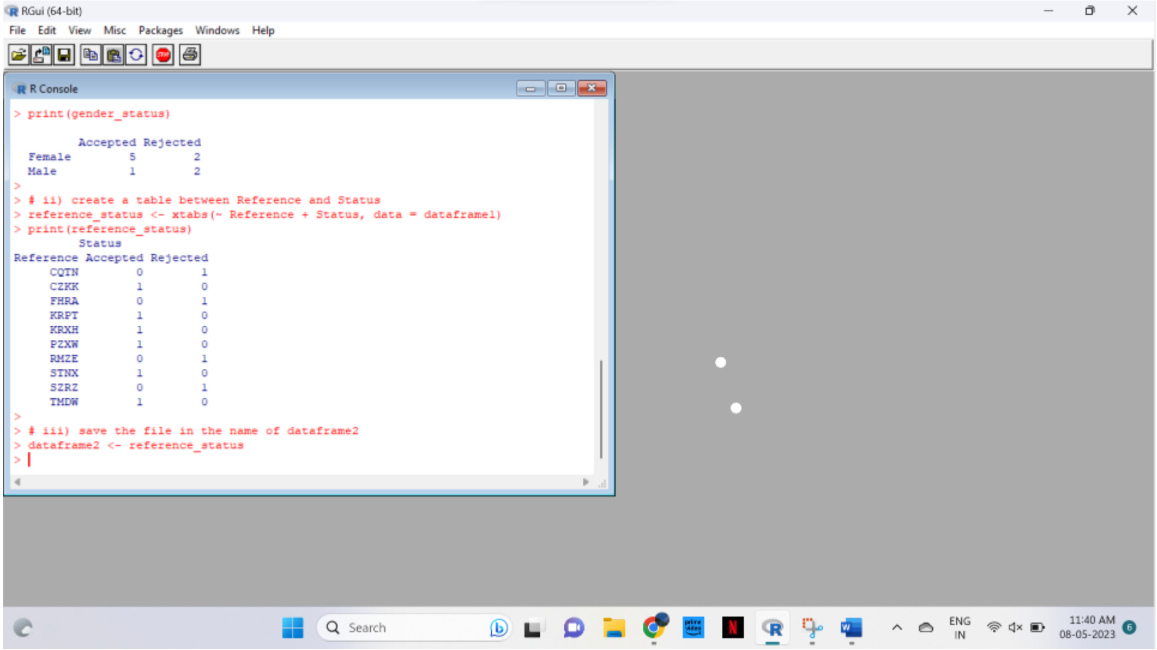
create a table delineating the relationship between the “Reference”

category, and the “Status” category.

iii) Save the file in the name of dataframe2

```
# load the data from dataframe1

dataframe1 <- data.frame(

  Reference = c("KRXH", "KRPT", "FHRA", "CZKK", "CQTN", "PZXW", "SZRZ", "RMZE", "STNX", "TMDW"),

  Status = c("Accepted", "Accepted", "Rejected", "Accepted", "Rejected", "Accepted", "Rejected", "Rejected", "Accepted", "Accepted"),

  Gender = c("Female", "Male", "Male", "Female", "Female", "Female", "Male", "Female", "Female", "Female"),

  TestNewOrFollowUp = c("Test1 New", "Test1 New", "Test2 New", "Test3 New", "Test1 New", "Follow-up", "Test4 New", "Test2 New", "Test3 New", "Test1 New")

)
```

```
# i) explore the relationship between two categorical variables

# contingency table between Gender and Status

gender_status <- table(dataframe1$Gender, dataframe1$Status)

print(gender_status)


# ii) create a table between Reference and Status

reference_status <- xtabs(~ Reference + Status, data = dataframe1)

print(reference_status)


# iii) save the file in the name of dataframe2

dataframe2 <- reference_status
```



```
> print(gender_status)

          Accepted Rejected
  Female         5        2
  Male           1        2
>
> # ii) create a table between Reference and Status
> reference_status <- xtabs(~ Reference + Status, data = dataframe1)
> print(reference_status)
         Status
Reference Accepted Rejected
     CQTN        0        1
     CZKK        1        0
     FHRA        0        1
     KRPT        1        0
     KRXH        1        0
     PZXW        1        0
     RMZE        0        1
     STNX        1        0
     SZRZ        0        1
     TMDW        1        0
>
> # iii) save the file in the name of dataframe2
> dataframe2 <- reference_status
> 
```

Exercise: 10

Use the same data frame using three Categorical Variables create a Multi-Dimensional Table

Apply three variables from “dataframe1” to create a Multi-Dimensional Cross-Tabulation of “Status“, “Gender“, and “Test“.

CODE

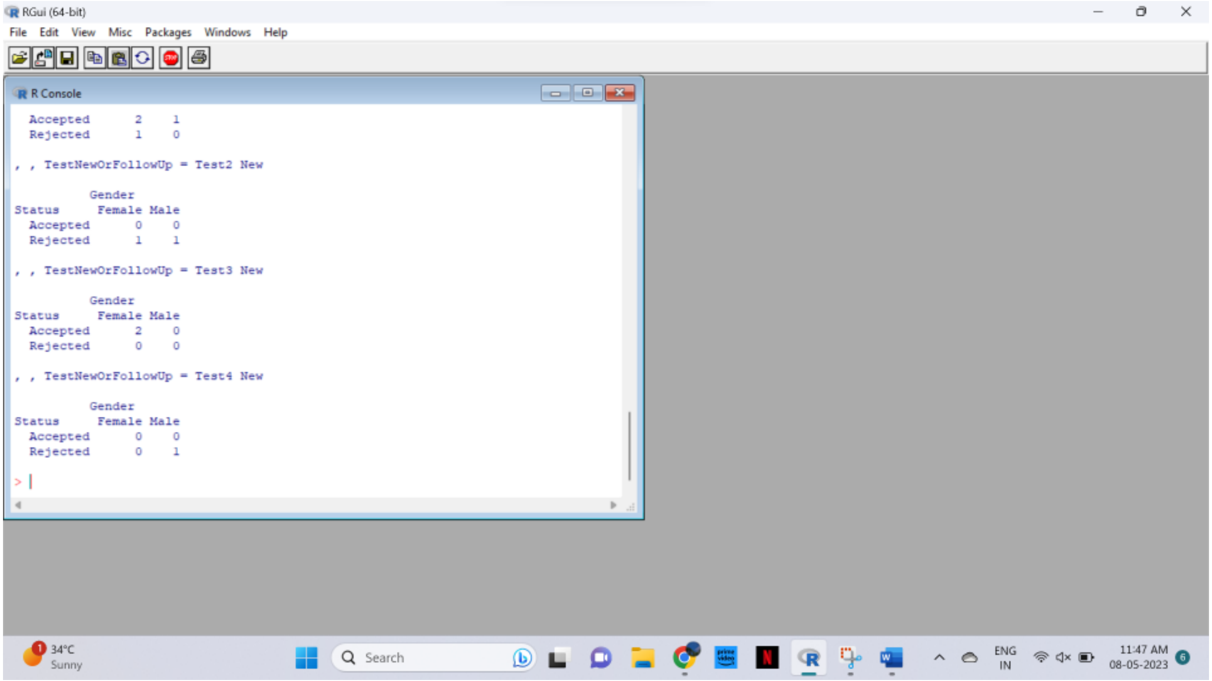
dataframe2 <- dataframe1 # Copy the original data frame

# Create the multi-dimensional table

multi_table <- xtabs(~ Status + Gender + TestNewOrFollowUp, dataframe2)

# View the table

multi_table



```
  Accepted      2    1
  Rejected      1    0

, , TestNewOrFollowUp = Test2 New

          Gender
Status     Female Male
  Accepted      0    0
  Rejected      1    1

, , TestNewOrFollowUp = Test3 New

          Gender
Status     Female Male
  Accepted      2    0
  Rejected      0    0

, , TestNewOrFollowUp = Test4 New

          Gender
Status     Female Male
  Accepted      0    0
  Rejected      0    1

> 
```

Exercise: 11

Row Percentages

The R package “tigerstats” is required for the next two exercises.

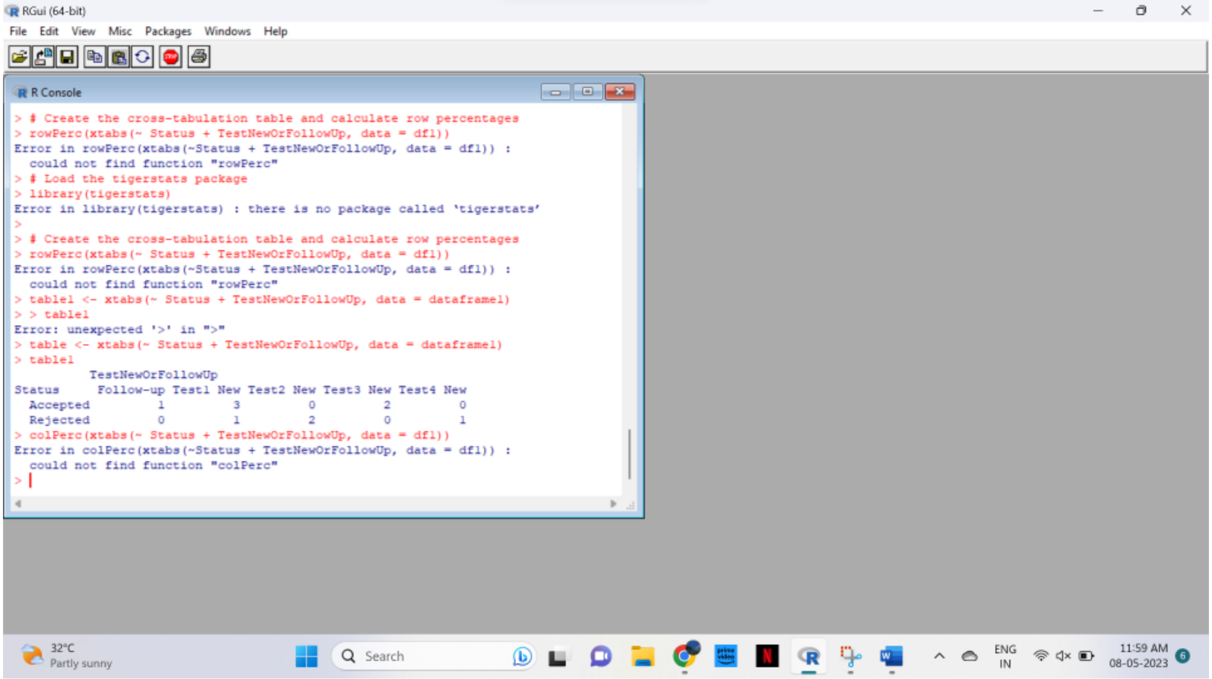
1) Create an xtabs() formula that cross-tabulates “Status“, and “Test“.

2) Enclose the xtabs() formula in the tigerstats function, “rowPerc()” to display row percentages for “Status” by “Test“.

CODE

table <- xtabs(~ Status + TestNewOrFollowUp, data = dataframe1)

table1



```
> # Create the cross-tabulation table and calculate row percentages
> rowPerc(xtabs(~ Status + TestNewOrFollowUp, data = df1))
Error in rowPerc(xtabs(~Status + TestNewOrFollowUp, data = df1)) :
  could not find function "rowPerc"
> # Load the tigerstats package
> library(tigerstats)
Error in library(tigerstats) : there is no package called ‘tigerstats’
>
> # Create the cross-tabulation table and calculate row percentages
> rowPerc(xtabs(~ Status + TestNewOrFollowUp, data = df1))
Error in rowPerc(xtabs(~Status + TestNewOrFollowUp, data = df1)) :
  could not find function "rowPerc"
> table1 <- xtabs(~ Status + TestNewOrFollowUp, data = dataframe1)
> > table1
Error: unexpected '>' in ">"
> table <- xtabs(~ Status + TestNewOrFollowUp, data = dataframe1)
> table1
          TestNewOrFollowUp
Status      Follow-up Test1 New Test2 New Test3 New Test4 New
  Accepted          1         3         0         2         0
  Rejected          0         1         2         0         1
> colPerc(xtabs(~ Status + TestNewOrFollowUp, data = df1))
Error in colPerc(xtabs(~Status + TestNewOrFollowUp, data = df1)) :
  could not find function "colPerc"
> 
```

Exercise 12

Column Percentages

1) Create an xtabs() formula that cross-tabulates “Status“, and “Test“.

2) Enclose the xtabs() formula in the tigerstats function, “colPerc()” to display row

percentages for “Status” by “Test“.

CODE

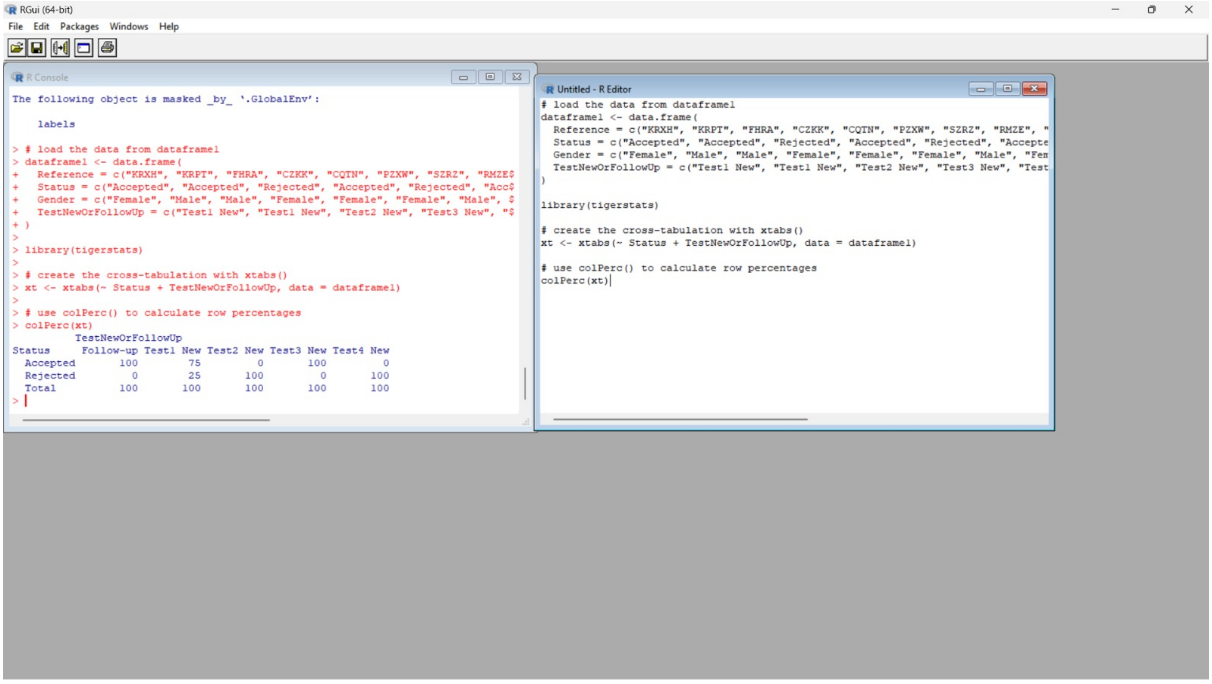
```
# load the data from dataframe1

dataframe1 <- data.frame(

  Reference = c("KRXH", "KRPT", "FHRA", "CZKK", "CQTN", "PZXW", "SZRZ", "RMZE", "STNX", "TMDW"),

  Status = c("Accepted", "Accepted", "Rejected", "Accepted", "Rejected", "Accepted", "Rejected", "Rejected", "Accepted", "Accepted"),

  Gender = c("Female", "Male", "Male", "Female", "Female", "Female", "Male", "Female", "Female", "Female"),

  TestNewOrFollowUp = c("Test1 New", "Test1 New", "Test2 New", "Test3 New", "Test1 New", "Follow-up", "Test4 New", "Test2 New", "Test3 New", "Test1 New")

)


library(tigerstats)
```

create the cross-tabulation with xtabs()

xt <- xtabs(~ Status + TestNewOrFollowUp, data = dataframe1)

use colPerc() to calculate row percentages

colPerc(xt)



```
The following object is masked _by_ '.GlobalEnv':

    labels

> # load the data from dataframe1
> dataframe1 <- data.frame(
+   Reference = c("KRXH", "KRPT", "FHRA", "CZKK", "CQTN", "PZXW", "SZRZ", "RMZE$
+   Status = c("Accepted", "Accepted", "Rejected", "Accepted", "Rejected", "Acc$
+   Gender = c("Female", "Male", "Male", "Female", "Female", "Female", "Male", $
+   TestNewOrFollowUp = c("Test1 New", "Test1 New", "Test2 New", "Test3 New", "$
+ )
>
> library(tigerstats)
>
> # create the cross-tabulation with xtabs()
> xt <- xtabs(~ Status + TestNewOrFollowUp, data = dataframe1)
>
> # use colPerc() to calculate row percentages
> colPerc(xt)
          TestNewOrFollowUp
Status      Follow-up Test1 New Test2 New Test3 New Test4 New
  Accepted        100        75         0       100         0
  Rejected          0        25       100         0       100
  Total           100       100       100       100       100
>
```

```
# load the data from dataframe1
dataframe1 <- data.frame(
  Reference = c("KRXH", "KRPT", "FHRA", "CZKK", "CQTN", "PZXW", "SZRZ", "RMZE", "
  Status = c("Accepted", "Accepted", "Rejected", "Accepted", "Rejected", "Accepte
  Gender = c("Female", "Male", "Male", "Female", "Female", "Female", "Male", "Fem
  TestNewOrFollowUp = c("Test1 New", "Test1 New", "Test2 New", "Test3 New", "Test
)

library(tigerstats)

# create the cross-tabulation with xtabs()
xt <- xtabs(~ Status + TestNewOrFollowUp, data = dataframe1)

# use colPerc() to calculate row percentages
colPerc(xt)
```

VISUALIZATION IN R

13. Write a program for creating a pie-chart in R using the input vector(21,62,10,53). Provide labels for the chart as 'London', 'New York', 'Singapore', 'Mumbai'. Add a title to the chart as 'city pie-chart' and add a legend at the top right corner of the chart.

CODE

# Define input vector and labels

input_vector <- c(21, 62, 10, 53)

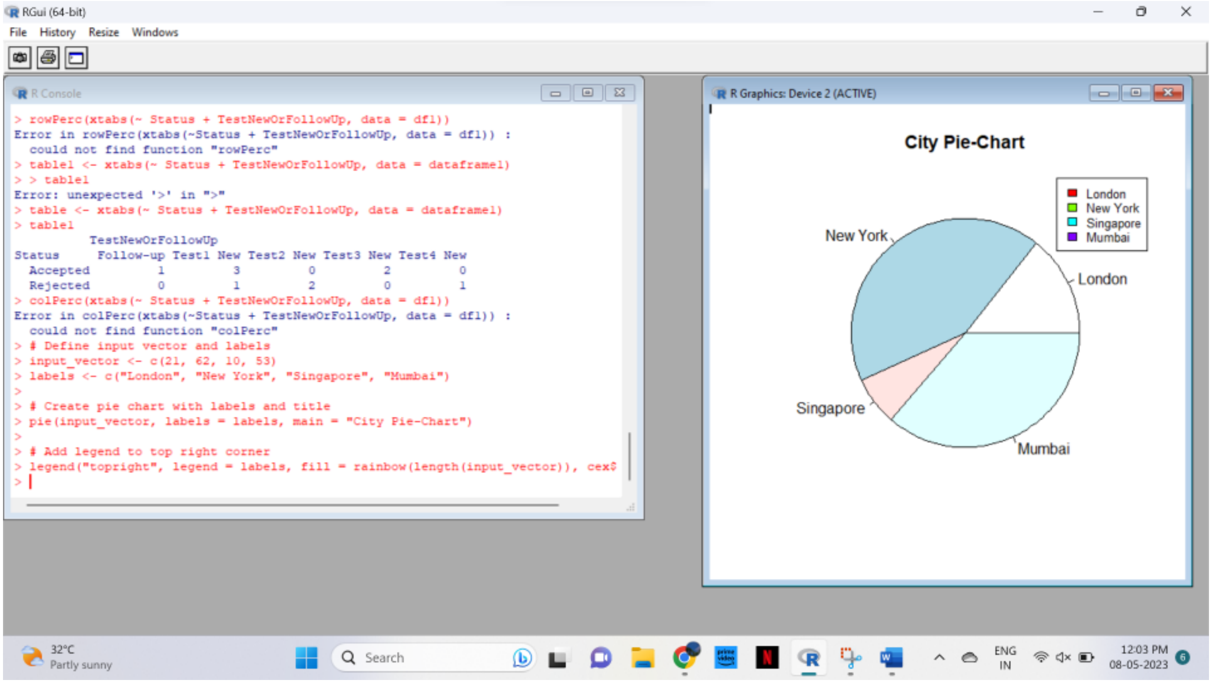
labels <- c("London", "New York", "Singapore", "Mumbai")

Create pie chart with labels and title

pie(input_vector, labels = labels, main = "City Pie-Chart")

Add legend to top right corner

legend("topright", legend = labels, fill = rainbow(length(input_vector)), cex = 0.8)



```
> rowPerc(xtabs(~ Status + TestNewOrFollowUp, data = df1))
Error in rowPerc(xtabs(~Status + TestNewOrFollowUp, data = df1)) :
  could not find function "rowPerc"
> table1 <- xtabs(~ Status + TestNewOrFollowUp, data = dataframe1)
> > table1
Error: unexpected '>' in ">"
> table <- xtabs(~ Status + TestNewOrFollowUp, data = dataframe1)
> table1
           TestNewOrFollowUp
Status      Follow-up Test1 New Test2 New Test3 New Test4 New
  Accepted          1         3         0         2         0
  Rejected          0         1         2         0         1
> colPerc(xtabs(~ Status + TestNewOrFollowUp, data = df1))
Error in colPerc(xtabs(~Status + TestNewOrFollowUp, data = df1)) :
  could not find function "colPerc"
> # Define input vector and labels
> input_vector <- c(21, 62, 10, 53)
> labels <- c("London", "New York", "Singapore", "Mumbai")
>
> # Create pie chart with labels and title
> pie(input_vector, labels = labels, main = "City Pie-Chart")
>
> # Add legend to top right corner
> legend("topright", legend = labels, fill = rainbow(length(input_vector)), cex$
> 
```

14. Create a 3D Pie Chart for the dataset “political Knowledge” with suitable labels,colours and a legend at the top right corner of the chart.

CODE

# Load the politicalKnowledge dataset

data(politicalKnowledge)

# Create a 3D pie chart

library(plotrix)

slices <- politicalKnowledge

lbls <- c("Very Low", "Low", "Moderate", "High", "Very High")

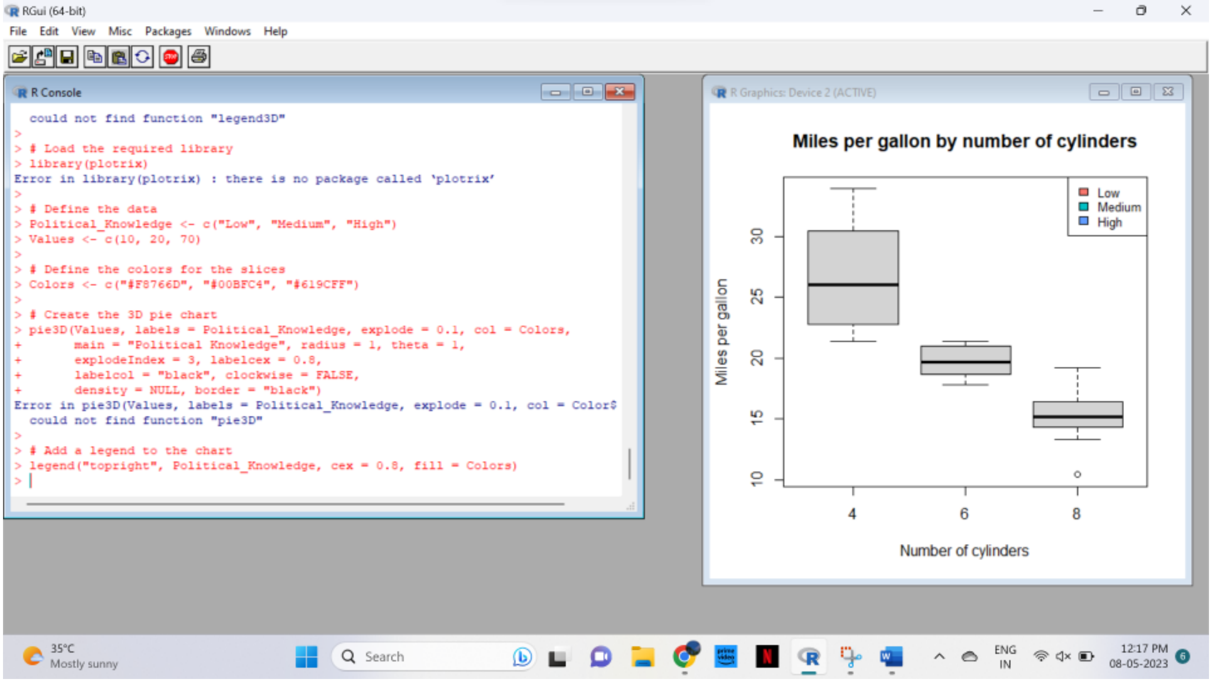
colors <- c("#E41A1C", "#377EB8", "#4DAF4A", "#984EA3", "#FF7F00")

pie3D(slices, labels = lbls, explode = 0.1, col = colors, main = "Political Knowledge",

    radius = 1, depth = 0.5, theta = 30, start = 0, clock = TRUE)

# Add a legend at the top right corner

legend3D("topright", lbls, col = colors, pch = 16, cex = 0.8, bty = "n")



```
  could not find function "legend3D"
>
> # Load the required library
> library(plotrix)
Error in library(plotrix) : there is no package called 'plotrix'
>
> # Define the data
> Political_Knowledge <- c("Low", "Medium", "High")
> Values <- c(10, 20, 70)
>
> # Define the colors for the slices
> Colors <- c("#F8766D", "#00BFC4", "#619CFF")
>
> # Create the 3D pie chart
> pie3D(Values, labels = Political_Knowledge, explode = 0.1, col = Colors,
+       main = "Political Knowledge", radius = 1, theta = 1,
+       explodeIndex = 3, labelcex = 0.8,
+       labelcol = "black", clockwise = FALSE,
+       density = NULL, border = "black")
Error in pie3D(Values, labels = Political_Knowledge, explode = 0.1, col = Color$
  could not find function "pie3D"
>
> # Add a legend to the chart
> legend("topright", Political_Knowledge, cex = 0.8, fill = Colors)
> 
```

15.Write a program for creating a bar chart using the vectors H=c(7,12,28,3,41) and M=c(“mar”, “apr”, “may”, “jun”, “jul”). Add a title to the chart as “Revenue chart”.

CODE

# Create data vectors

H <- c(7, 12, 28, 3, 41)

M <- c("mar", "apr", "may", "jun", "jul")

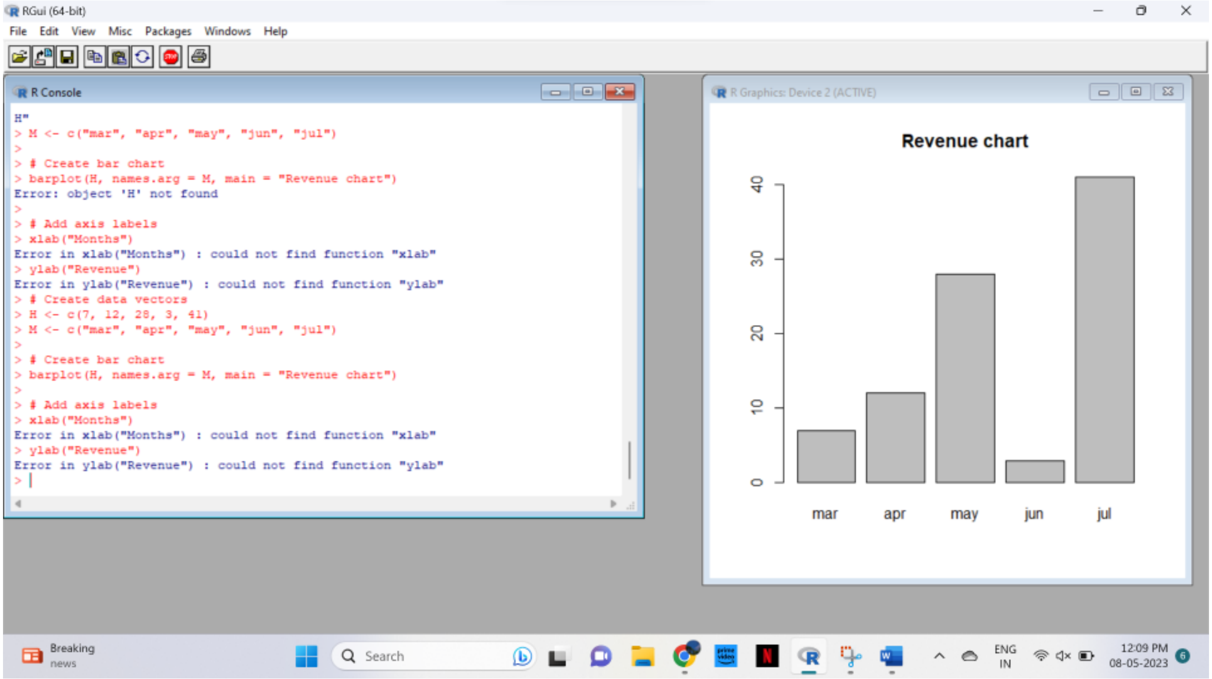
# Create bar chart

barplot(H, names.arg = M, main = "Revenue chart")

Add axis labels

xlab("Months")

ylab("Revenue")



```
H"
> M <- c("mar", "apr", "may", "jun", "jul")
>
> # Create bar chart
> barplot(H, names.arg = M, main = "Revenue chart")
Error: object 'H' not found
>
> # Add axis labels
> xlab("Months")
Error in xlab("Months") : could not find function "xlab"
> ylab("Revenue")
Error in ylab("Revenue") : could not find function "ylab"
> # Create data vectors
> H <- c(7, 12, 28, 3, 41)
> M <- c("mar", "apr", "may", "jun", "jul")
>
> # Create bar chart
> barplot(H, names.arg = M, main = "Revenue chart")
>
> # Add axis labels
> xlab("Months")
Error in xlab("Months") : could not find function "xlab"
> ylab("Revenue")
Error in ylab("Revenue") : could not find function "ylab"
> 
```

16. Make a histogram for the “AirPassengers“dataset, start at 100 on the x-axis, and from values 200 to 700, make the bins 200 wide

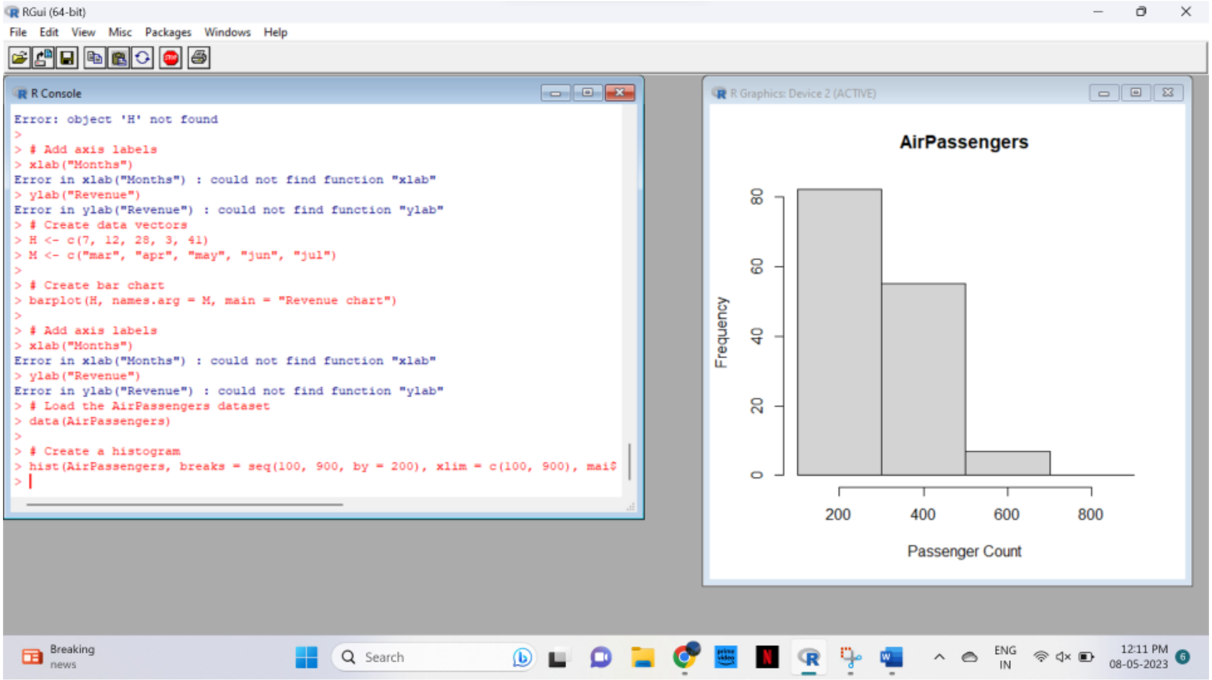
CODE

# Load the AirPassengers dataset

data(AirPassengers)

# Create a histogram

hist(AirPassengers, breaks = seq(100, 900, by = 200), xlim = c(100, 900), main = "AirPassengers", xlab = "Passenger Count", ylab = "Frequency")



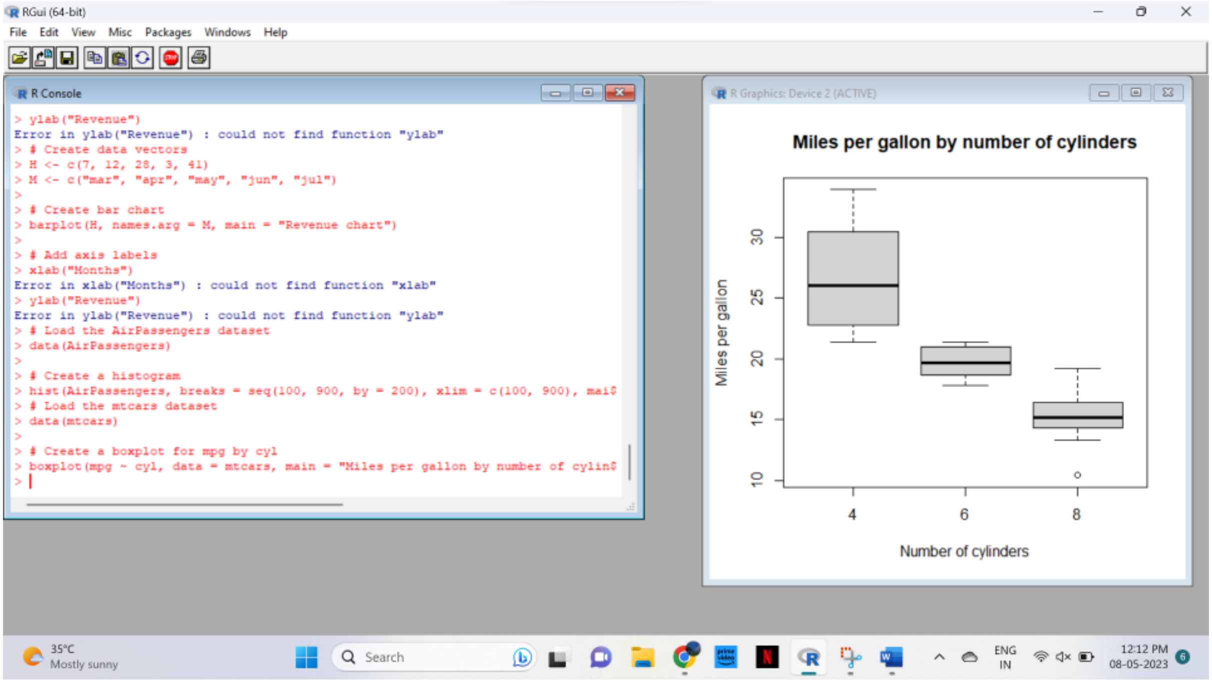
```
Error: object 'H' not found
>
> # Add axis labels
> xlab("Months")
Error in xlab("Months") : could not find function "xlab"
> ylab("Revenue")
Error in ylab("Revenue") : could not find function "ylab"
> # Create data vectors
> H <- c(7, 12, 28, 3, 41)
> M <- c("mar", "apr", "may", "jun", "jul")
>
> # Create bar chart
> barplot(H, names.arg = M, main = "Revenue chart")
>
> # Add axis labels
> xlab("Months")
Error in xlab("Months") : could not find function "xlab"
> ylab("Revenue")
Error in ylab("Revenue") : could not find function "ylab"
> # Load the AirPassengers dataset
> data(AirPassengers)
>
> # Create a histogram
> hist(AirPassengers, breaks = seq(100, 900, by = 200), xlim = c(100, 900), mai$
> 
```

17. Create a Boxplot graph for the relation between &quot;mpg&quot;(miles per galloon) and

&quot;cyl&quot;(number of Cylinders) for the dataset &quot;mtcars&quot; available in R Environment.

CODE

```
# Load the mtcars dataset

data(mtcars)


# Create a boxplot for mpg by cyl

boxplot(mpg ~ cyl, data = mtcars, main = "Miles per gallon by number of cylinders", xlab = "Number of cylinders", ylab = "Miles per gallon")
```



```
> ylab("Revenue")
Error in ylab("Revenue") : could not find function "ylab"
> # Create data vectors
> H <- c(7, 12, 28, 3, 41)
> M <- c("mar", "apr", "may", "jun", "jul")
>
> # Create bar chart
> barplot(H, names.arg = M, main = "Revenue chart")
>
> # Add axis labels
> xlab("Months")
Error in xlab("Months") : could not find function "xlab"
> ylab("Revenue")
Error in ylab("Revenue") : could not find function "ylab"
> # Load the AirPassengers dataset
> data(AirPassengers)
>
> # Create a histogram
> hist(AirPassengers, breaks = seq(100, 900, by = 200), xlim = c(100, 900), mai$
> # Load the mtcars dataset
> data(mtcars)
>
> # Create a boxplot for mpg by cyl
> boxplot(mpg ~ cyl, data = mtcars, main = "Miles per gallon by number of cylin$
> 
```