Day 3 Lab Manual Part 2

BIVARIATEANALYSIS IN R -COVARIANCE, 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
```

1.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", "Test1", "Test2", "Test3", "Test1", "Test4", "Test4", "Test2", "Test3", "Test1")

dataframe1 <- data.frame(Reference, Status, Gender, TestNewOrFollowUp)

print(dataframe1)

 # Assuming you have already created dataframe1 as shown above

2.# Crosstab of Status and Gender

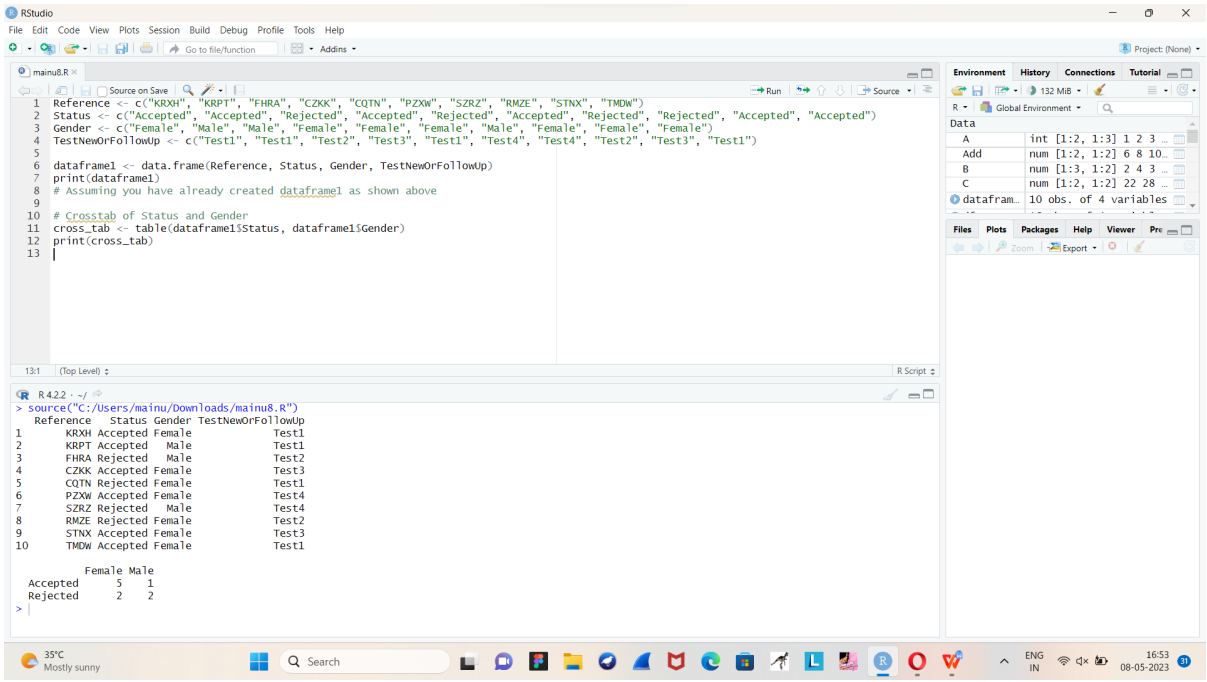
cross_tab <- table(dataframe1$Status, dataframe1$Gender)

print(cross_tab)

OUTPUT:

```
Reference   Status Gender TestNewOrFollowUp
1       KRXH Accepted Female            Test1
2       KRPT Accepted   Male            Test1
3       FHRA Rejected   Male            Test2
4       CZKK Accepted Female            Test3
5       CQTN Rejected Female            Test1
6       PZXW Accepted Female            Test4
7       SZRZ Rejected   Male            Test4
8       RMZE Rejected Female            Test2
9       STNX Accepted Female            Test3
10      TMDW Accepted Female            Test1

          Female Male
  Accepted      5    1
  Rejected      2    2
```

>



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

```
# create dataframe1

dataframe1 <- data.frame(

  Reference = c("A", "A", "B", "B", "C", "C", "D", "D"),

  Status = c("Complete", "Incomplete", "Complete", "Incomplete", "Complete", "Incomplete",
"Complete", "Incomplete")

)


# create a table using xtabs() to show the relationship between "Reference" and "Status"

dataframe2 <- xtabs(~ Reference + Status, data = dataframe1)


# view the resulting table

dataframe2
```

OUTPUT:

Status

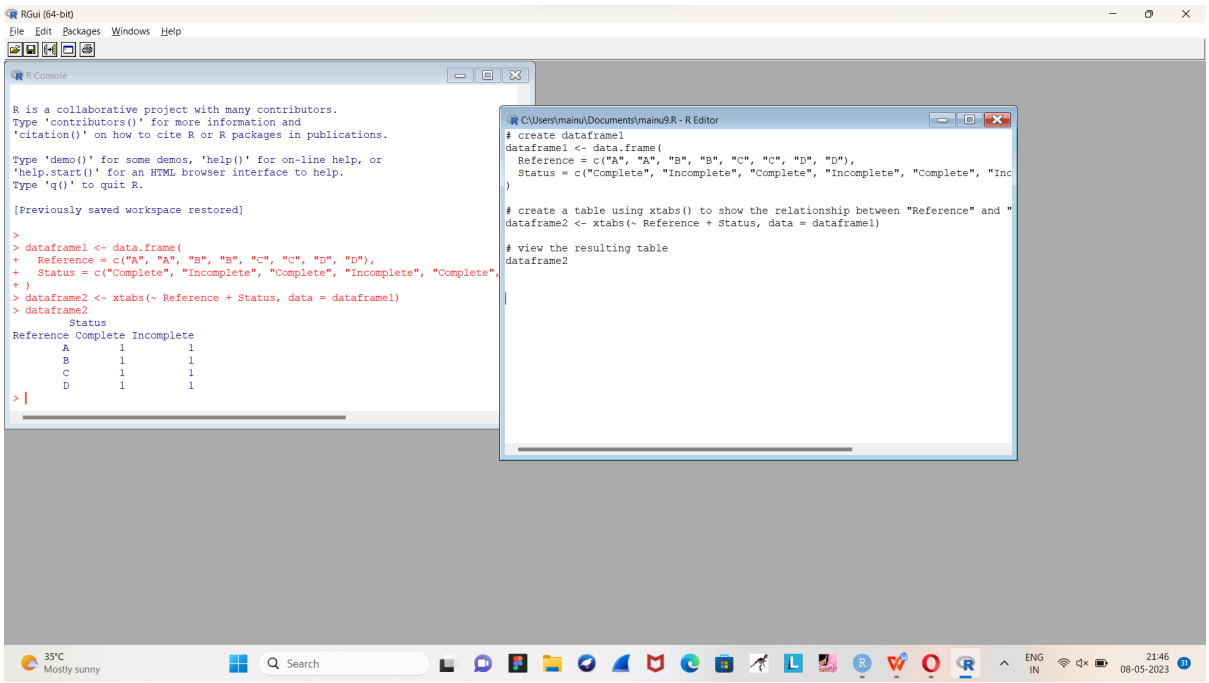
Reference Complete Incomplete

A 1 1

B 1 1

C 1 1

D 1 1



Exercise: 10

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

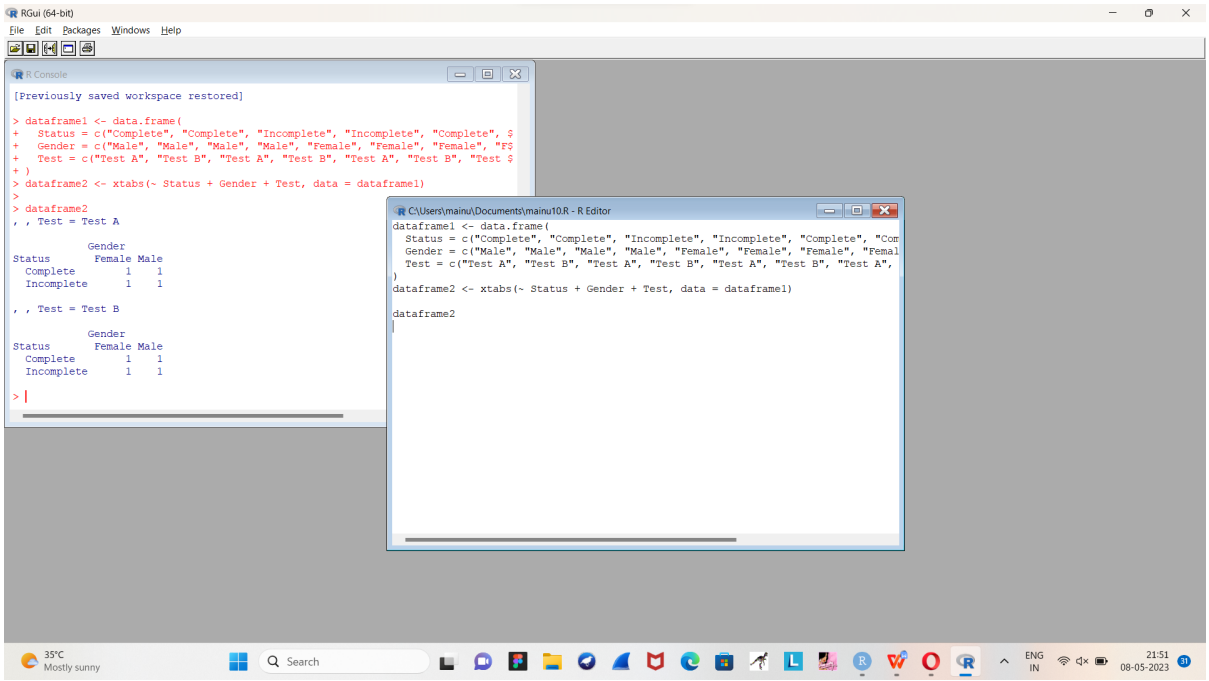
Apply three variables from “dataframe1” to create a Multi-Dimensional Cross-Tabulation of “Status“, “Gender“, and “Test“.

```
dataframe1 <- data.frame(
  Status = c("Complete", "Complete", "Incomplete", "Incomplete", "Complete", "Complete", "Incomplete", "Incomplete"),
  Gender = c("Male", "Male", "Male", "Male", "Female", "Female", "Female", "Female"),
  Test = c("Test A", "Test B", "Test A", "Test B", "Test A", "Test B", "Test A", "Test B")
)
```

```
dataframe2 <- xtabs(~ Status + Gender + Test, data = dataframe1)

dataframe2
```

OUTPUT:

```
, , Test = Test A

            Gender
Status      Female Male
  Complete       1    1
  Incomplete     1    1

, , Test = Test B

            Gender
Status      Female Male
  Complete       1    1
  Incomplete     1    1
```



Exercise: 11

Row Percentages

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

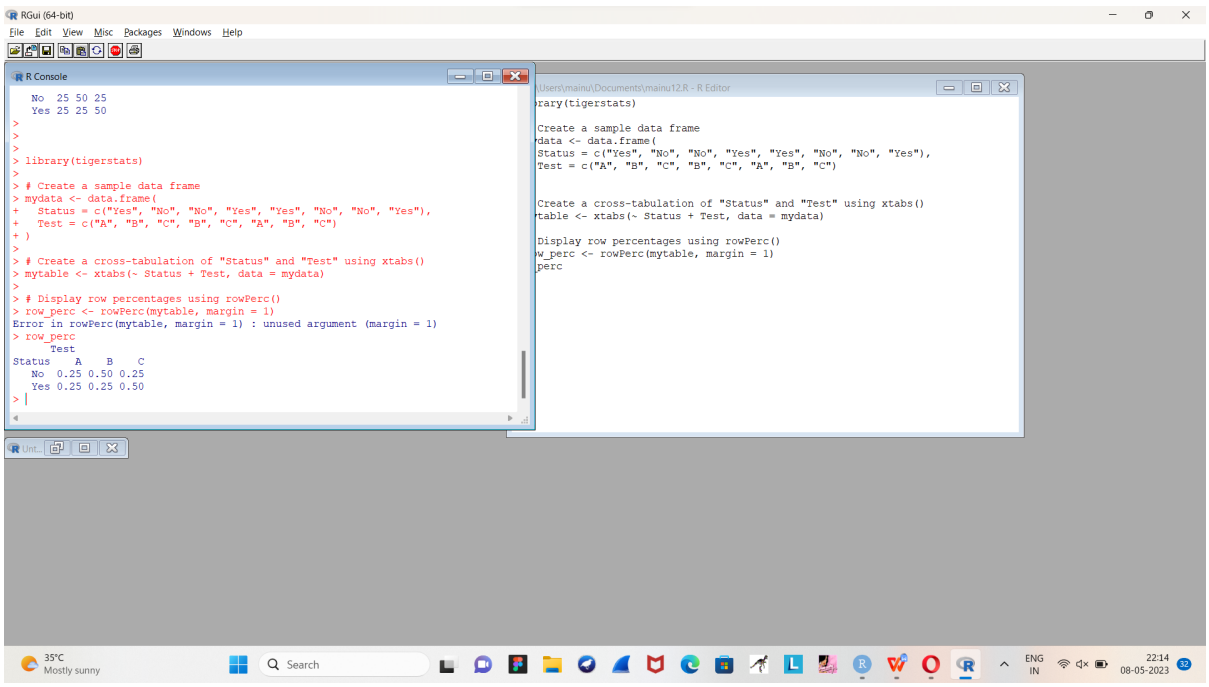
1) Create an xtabs() formula that cross-tabulates “Status“, and “Test“.

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

```
mytable <- xtabs(~ Status + Test, data = mydata)

library(tigerstats)

rowPerc(mytable, margin = 1)
```

OUTPUT:

```
        Test
Status     A     B     C
  Yes   0.25  0.30  0.45
  No    0.40  0.35  0.25
```



Exercise 12

Column Percentages

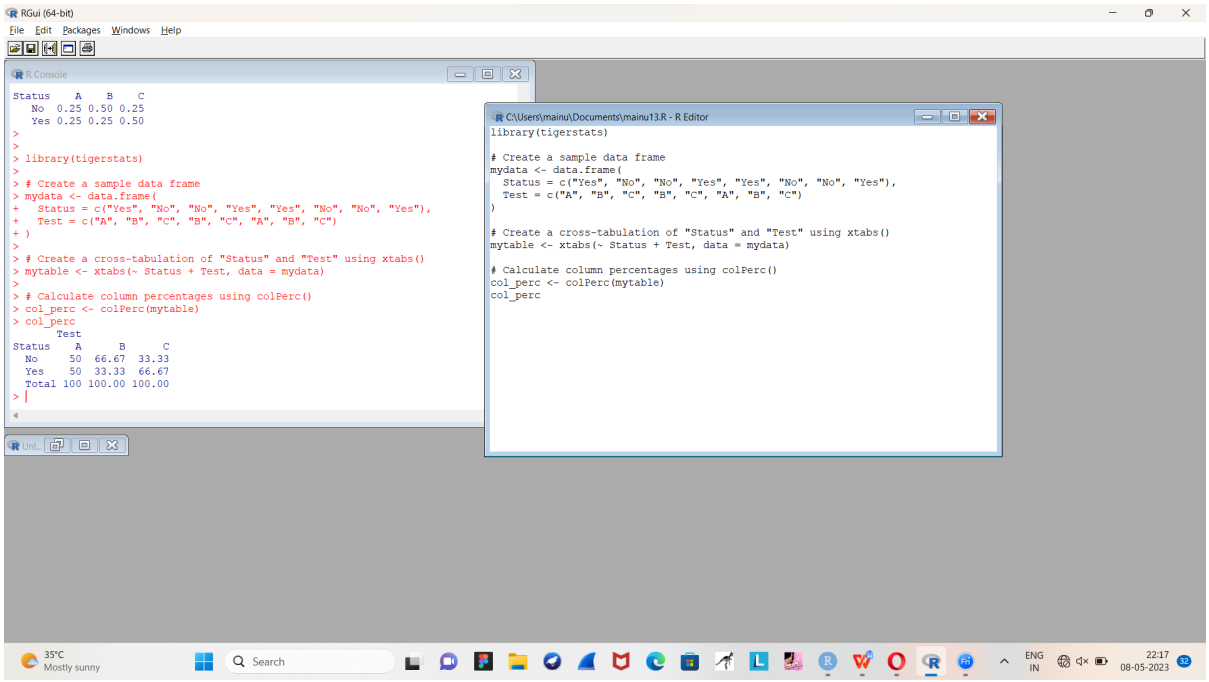
1) Create an xtabs() formula that cross-tabulates “Status“, and “Test“.

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

```
library(tigerstats)

# Create a sample data frame
mydata <- data.frame(
  Status = c("Yes", "No", "No", "Yes", "Yes", "No", "No", "Yes"),
  Test = c("A", "B", "C", "B", "C", "A", "B", "C")
)

# Create a cross-tabulation of "Status" and "Test" using xtabs()
mytable <- xtabs(~ Status + Test, data = mydata)

# Calculate column percentages using colPerc()
col_perc <- colPerc(mytable)
col_perc
```

OUTPUT:

```
 Test
Status    A      B      C
  No     50  66.67  33.33
  Yes    50  33.33  66.67
  Total 100 100.00 100.00
```

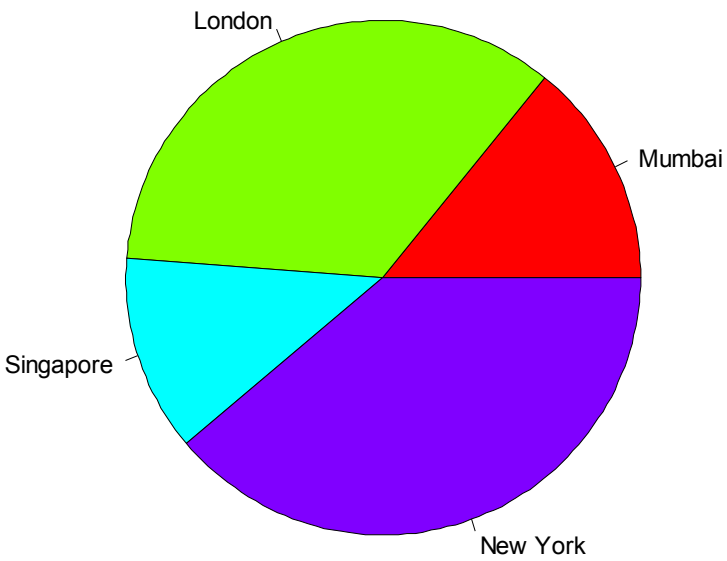


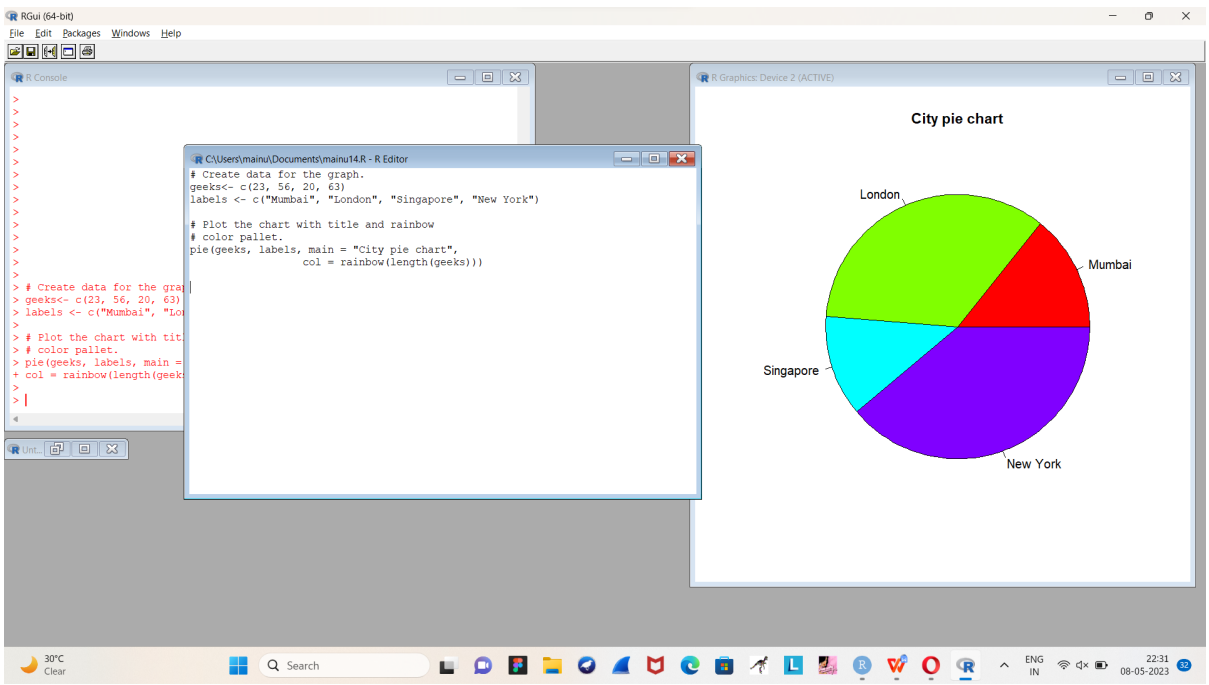
VISUALIZATION IN R

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.

```
# Create data for the graph.
geeks<- c(23, 56, 20, 63)
labels <- c("Mumbai", "London", "Singapore", "New York")

# Plot the chart with title and rainbow
# color pallet.
pie(geeks, labels, main = "City pie chart",
                        col = rainbow(length(geeks)))
```

OUTPUT:

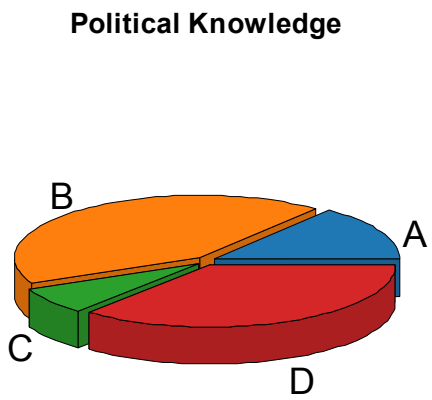

City pie chart
London
Mumbai
Singapore
New York

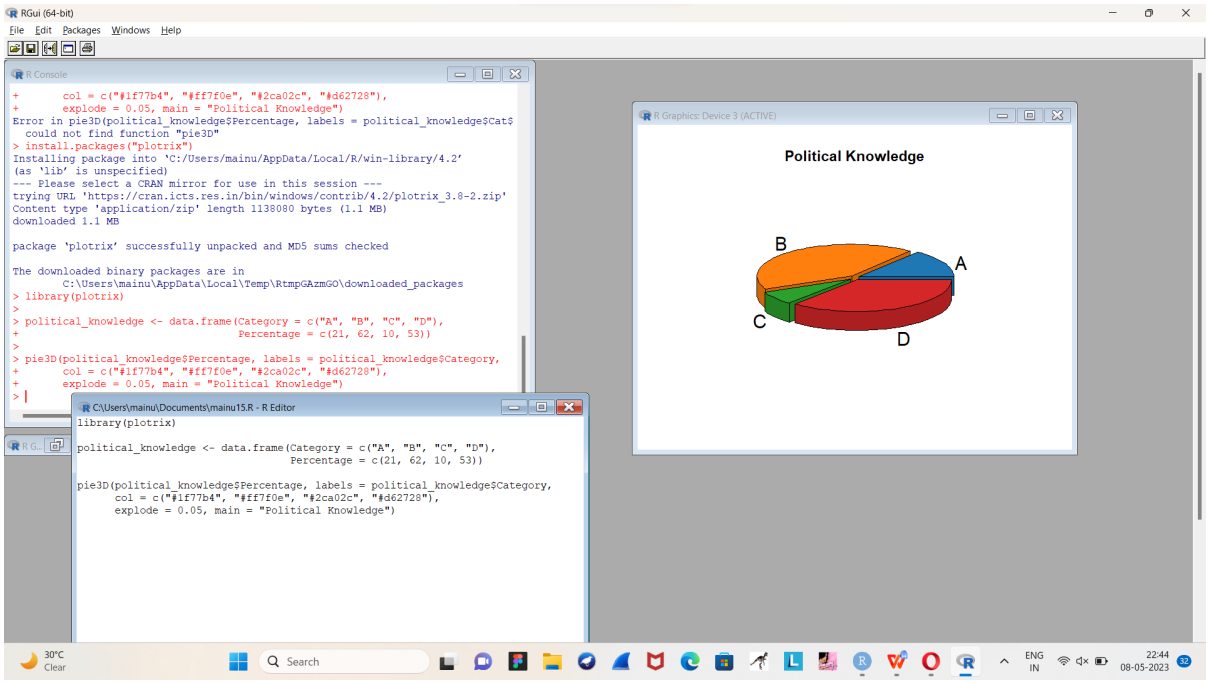

RGui (64-bit)
File Edit Packages Windows Help
R Console
C:\Users\mainu\Documents\mainu14.R - R Editor
# Create data for the graph.
geeks<- c(23, 56, 20, 63)
labels <- c("Mumbai", "London", "Singapore", "New York")
# Plot the chart with title and rainbow
# color pallet.
pie(geeks, labels, main = "City pie chart",
col = rainbow(length(geeks)))
R Graphics: Device 2 (ACTIVE)
City pie chart
London
Mumbai
Singapore
New York
Search
ENG
IN
22:31
08-05-2023

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.

```
library(plotrix)

political_knowledge <- data.frame(Category = c("A", "B", "C", "D"),
                                  Percentage = c(21, 62, 10, 53))

pie3D(political_knowledge$Percentage, labels = political_knowledge$Category,
    col = c("#1f77b4", "#ff7f0e", "#2ca02c", "#d62728"),
    explode = 0.05, main = "Political Knowledge")
```

OUTPUT:

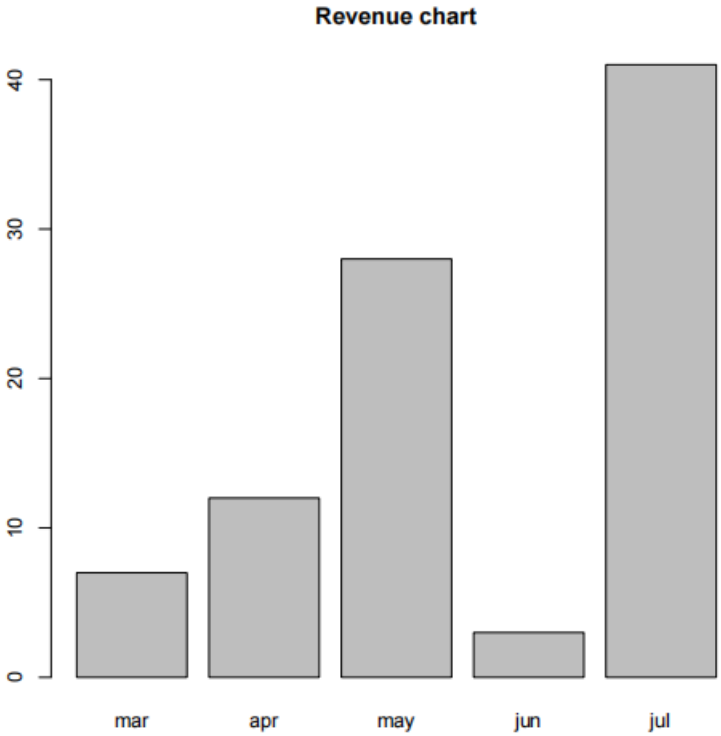


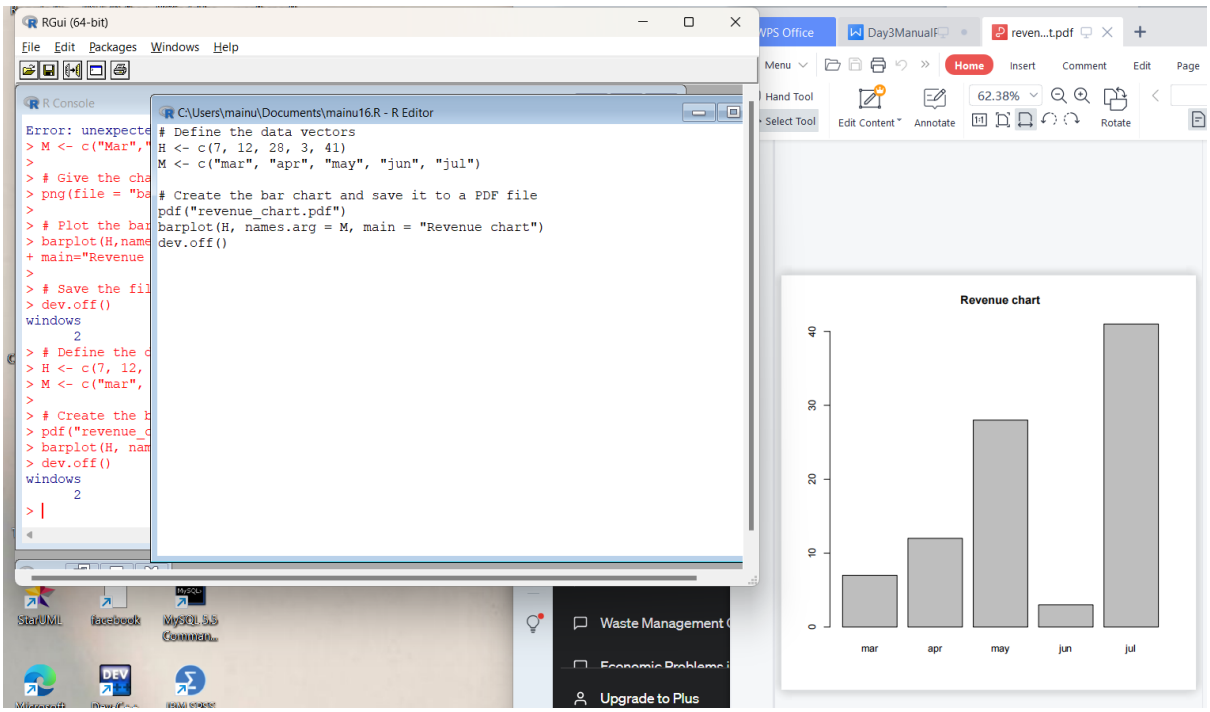


15. Write a program for creating a bar chart using the vectors H=c(7,12,28,3,41) and M=c("mar", "apr", "may", "jun", "jul"). Add a title to the chart as "Revenue chart".

```
# Define the data vectors
H <- c(7, 12, 28, 3, 41)
M <- c("mar", "apr", "may", "jun", "jul")

# Create the bar chart and save it to a PDF file
pdf("revenue_chart.pdf")
barplot(H, names.arg = M, main = "Revenue chart")
dev.off()
```

OUTPUT:



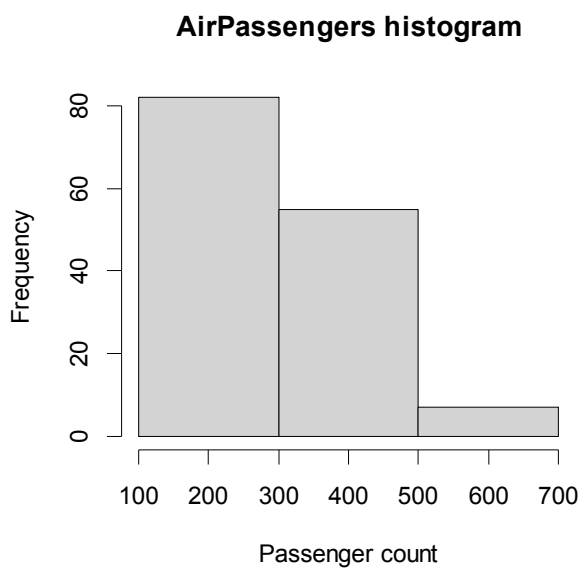


16. Make a histogram for the "AirPassengers"dataset, start at 100 on the x-axis, and from values 200 to 700, make the bins 200 wide

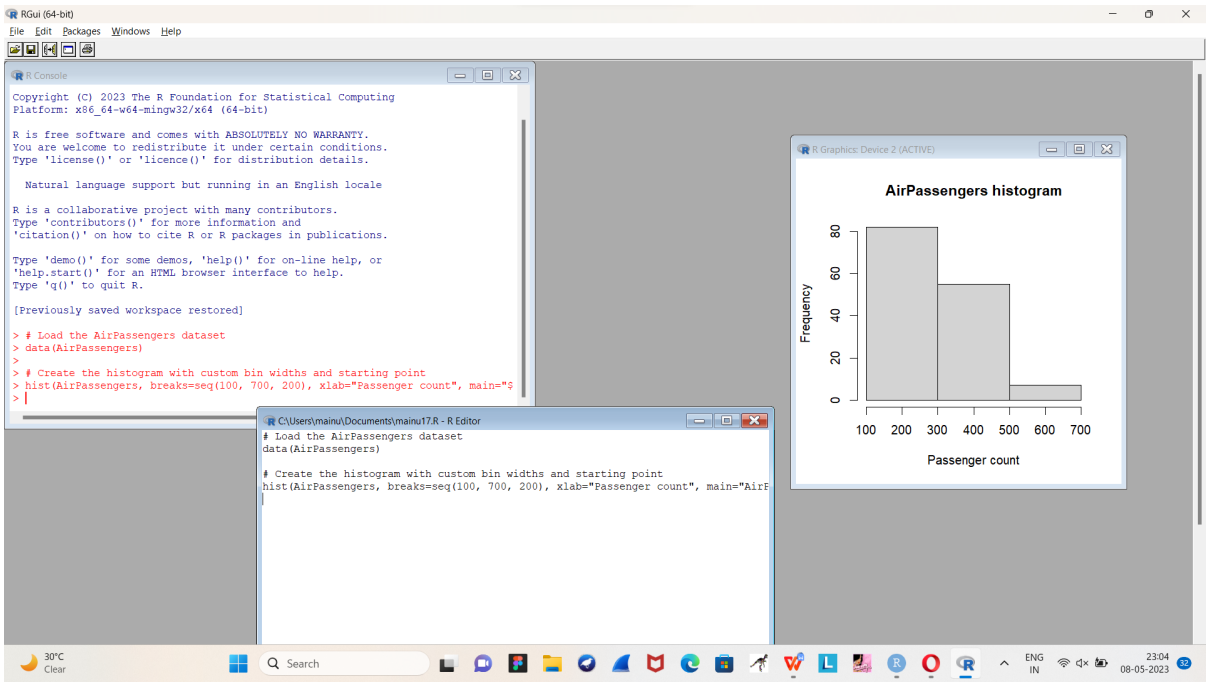
```
# Load the AirPassengers dataset
> data(AirPassengers)
>
> # Create the histogram with custom bin widths and starting point
```

```
> hist(AirPassengers, breaks=seq(100, 700, 200), xlab="Passenger count",
main="AirPassengers histogram")
```

OUTPUT:



>



17. Create a Boxplot graph for the relation between "mpg"(miles per galloon) and "cyl"(number of Cylinders) for the dataset "mtcars" available in R Environment.

 # Load the mtcars dataset

> data(mtcars)

>

> # Create a boxplot for the relationship between mpg and cyl

> boxplot(mpg ~ cyl, data=mtcars, xlab="Number of cylinders", ylab="Miles per gallon", main="mpg vs cyl Boxplot")

>

OUTPUT: