**12. Import XLSX File Into DataFrame to Filter the data**

**Aim**

To write a R program with XLSX file having empid,name,age,gender, salary,basic, DA.Provide atleast 20 datasets.

Read the following file and filter the data as follows:

a. Genderwise

b Age>40 and gender=male

c Salary >600 for different genders mentioned in the xlsx file

d. find out the difference between salary and Basic+DA

e. Salary >600 and Basic >300 and DA < 200

**Algorithm**

**Step 1:** Start the process to read and analyze Excel data in R.

**Step 2:** Set and check the working directory using setwd() and getwd().

**Step 3:** Install and load the readxl package (installation is required only once).

**Step 4:** Read the Excel file using read\_excel() and assign it to a variable (e.g., gender\_data).

**Step 5:** Print the dataset to verify the data has been read correctly.

**Step 6**: Use the subset() function to extract and print:

All records where Gender = "Female".

All records where Gender = "Male".

All male employees older than 40 years.

All employees with Salary greater than 600.

All employees with Salary > 600, Basic > 300, and DA < 200.

**Step 7:** Calculate and print the difference between Salary and the sum of (Basic + DA).

**Step 8:** End the program.

**Program :**

getwd()

setwd("D:/ThamilMani/Learning-Programming-/R Programming/12. XLSX File Handling")

getwd()

install.packages("readxl")

library(readxl)

#Tools -> Install Packages

data <- read\_excel("gender.xlsx")

print(data)

retval <- subset(data,data$Gender=="Female")

print(retval)

retval <- subset(data,data$Gender=="Male")

print(retval)

retval <- subset(data,data$Age>40 & data$Gender=="Male")

print(retval)

retval <- subset(data,data$Salary>600)

print(retval)

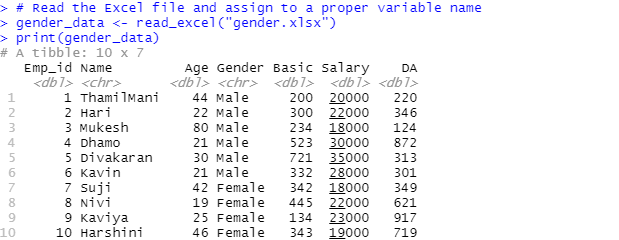
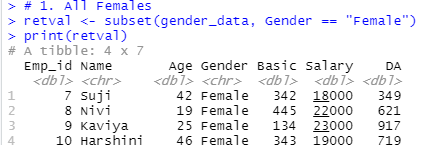
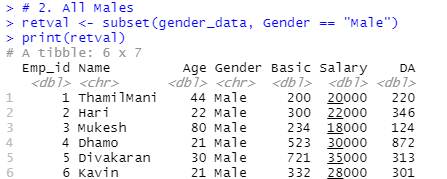
retval <- subset(data,data$Salary>600 & data$Basic>300 & data$DA<200)

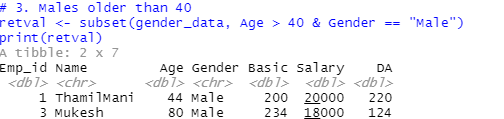
print(retval)

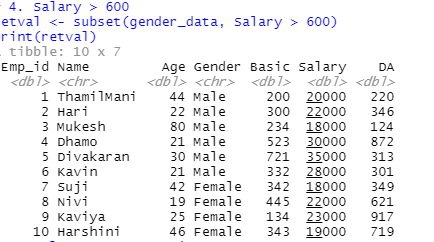
retval <- data$Salary - (data$Basic + data$DA)

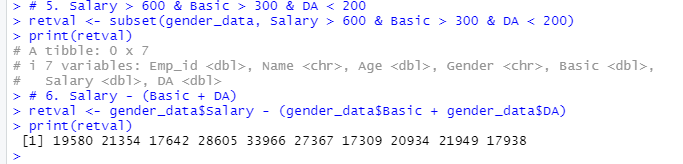
print(retval)

**OUTPUT:**









**RESULT**

Thus, our program has been successfully saved and executed.

**13. Merge XML and Json into DataFrame to Filter the data**

**Aim**

To write a R program with read employee data from an XLSX file and a JSON file, merge them into a single DataFrame, and apply various filters to analyze employee details based on gender, age, and salary-related conditions.

Read the following file and filter the data as follows:

a. Genderwise

b Age>40 and gender=male

c Salary >600 for different genders mentioned in the xlsx file

d. find out the difference between salary and Basic+DA

e. Salary >600 and Basic >300 and DA < 200

**Algorithm**

**Step 1:** Start the process to handle XML and JSON file conversion in R.

**Step 2:** Open RStudio and write the program using required packages (xml2, jsonlite).

**Step 3:** Read the XML file (e.g., gender.xml) using read\_xml().

**Step 4:** Extract all records from the XML using xml\_find\_all().

**Step 5:** Convert the extracted XML nodes into a data frame with proper column names and values.

**Step 6:** Convert necessary columns (Emp\_id, Age, Basic, Salary, DA) into integer type for further processing.

**Step 7:** Convert the data frame into JSON format using toJSON() and save it into a JSON file (e.g., gender.json).

**Step 8:** Read the JSON file back into R using fromJSON().

**Step 9:** Perform filtering and subsetting operations (e.g., Female employees, Male employees, Salary > 600, etc.).

**Step 10:** Perform calculations such as salary difference (Salary – (Basic + DA)).

**Step 11:** Print the results.

**Step 12:** End the program.

**Program :**

getwd()

setwd("D:/ThamilMani/Learning-Programming-/R Programming/13. XML File To JSON File")

getwd()

install.packages("xml2")

install.packages("jsonlite")

library(xml2)

library(jsonlite)

doc <- read\_xml("gender.xml")

records <- xml\_find\_all(doc, ".//Record")

data <- as.data.frame(

t(sapply(records, function(node) {

setNames(xml\_text(xml\_children(node)), xml\_name(xml\_children(node)))

})),

stringsAsFactors = FALSE

)

data$Emp\_id <- as.integer(data$Emp\_id)

data$Age <- as.integer(data$Age)

data$Basic <- as.integer(data$Basic)

data$Salary <- as.integer(data$Salary)

data$DA <- as.integer(data$DA)

json\_text <- toJSON(data, pretty = TRUE, auto\_unbox = TRUE)

write(json\_text, file = "gender.json")

data <- fromJSON("gender.json")

print(data)

retval <- subset(data, Gender == "Female")

print(retval)

retval <- subset(data, Gender == "Male")

print(retval)

retval <- subset(data, Age > 40 & Gender == "Male")

print(retval)

retval <- subset(data, Salary > 600)

print(retval)

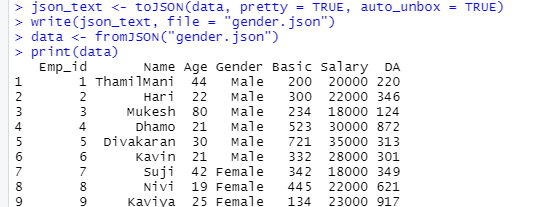
retval <- subset(data, Salary > 600 & Basic > 300 & DA < 200)

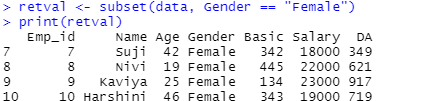
print(retval)

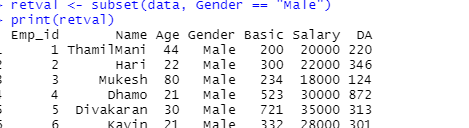
retval <- data$Salary - (data$Basic + data$DA)

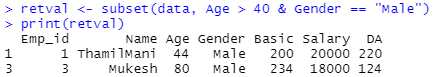
print(retval)

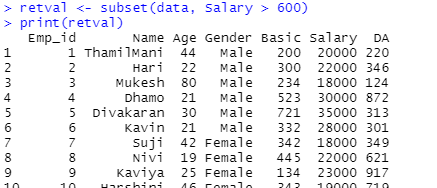
**OUTPUT:**

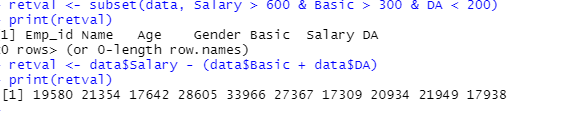
****











**RESULT**

Thus, our program has been successfully saved and executed.

**14. Generate PieChart**

**AIM:**

To write a R Program create and display a bar chart showing the monthly revenue distribution.

**ALGORITHM:**

**Step 1:** Start the process to visualize monthly revenue using a bar chart.

**Step 2:** Open RStudio and write the program using vectors and the barplot() function.

**Step 3:** Create a vector months to store the month names (e.g., months <- c("Mar", "Apr", "May", "Jun", "Jul")).

**Step 4:** Create another vector revenue to store revenue values corresponding to each month (e.g., revenue <- c(7, 8, 9, 10, 11)).

**Step 5:** Use the barplot() function to display the revenue values as vertical bars.

**Step 6:** Add labels for the x-axis (xlab = "Month") and y-axis (ylab = "Revenue").

**Step 7:** Add the main title (main = "Revenue Chart") to the bar chart.

**Step 8:** Enhance the chart with colors (e.g., col = "blue") and bar borders (e.g., border = "red").

**Step 9:** Run the program and view the bar chart output.

**Step 10:** End the process.

**PROGRAM:**

months <- c("Mar", "Apr", "May", "Jun", "Jul")

revenue <- c(7, 8, 9, 10, 11)

pie(revenue,

labels = paste(months, "\n", revenue, " units"),

main = "Monthly Revenue Distribution",

col = rainbow(length(months)),

border = "white")

x <- c(21, 62, 10, 53, 76)

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

library(plotrix)

pie3D(x,

labels = labels,

explode = 0.1,

main = "3D Pie Chart of Countries")

legend("topright",

labels,

cex = 0.6,

fill = rainbow(length(x)))

#Work with CSV Files

setwd("D:/24PCA014/Practical/Pie chart")

df <- read.csv("Combined.csv")

print(df)

v <- df[, c("Basic")]

lbl <- v # using values as labels

pie(v,labels = lbl,main="Basic",col=rainbow(length(v)))

legend("topleft",

legend = v,

cex = 0.7,

fill = rainbow(length(v)))

pie(v,

labels = lbl,

main = "Basic Pie Chart",

col = rainbow(length(v)))

legend("topleft",

legend = v,

cex = 0.7,

fill = rainbow(length(v)))

library(plotrix)

pie3D(v,

labels = lbl,

explode = 0.1,

main = "3D Pie Chart - Basic",

col = rainbow(length(v)))

legend("topright",

legend = lbl,

cex = 0.5,

fill = rainbow(length(v)))

v <- df[, c("Salary")]

print(v)

lbl <- c("1","2","3","4","5","6","7","8")

pie(v,

labels = lbl,

main = "Salary Pie Chart",

col = rainbow(length(v)))

pie3D(v,

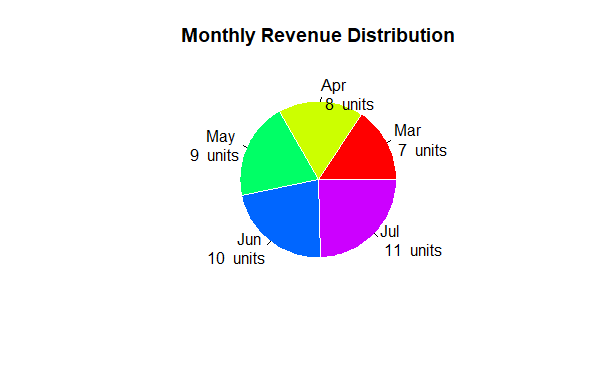
labels = lbl,

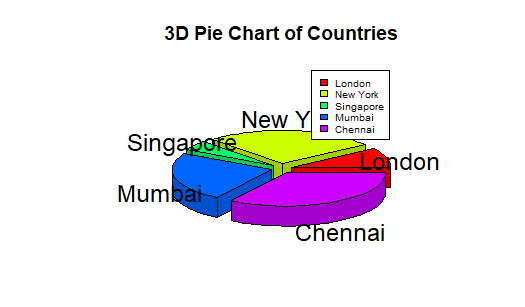
explode = 0.1,

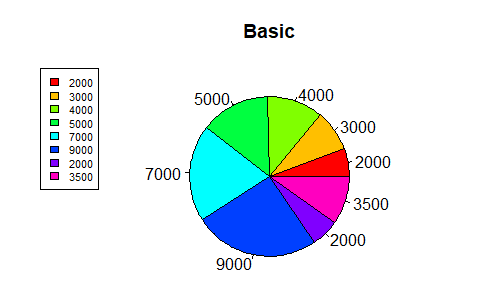
main = "3D Pie Chart - Salary",

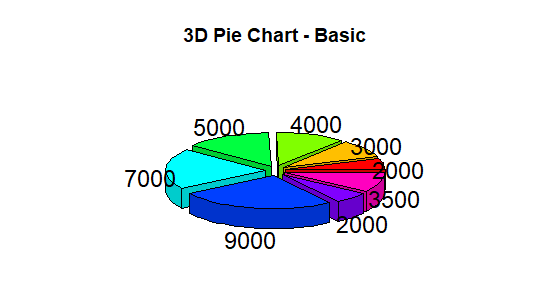
col = rainbow(length(v)))

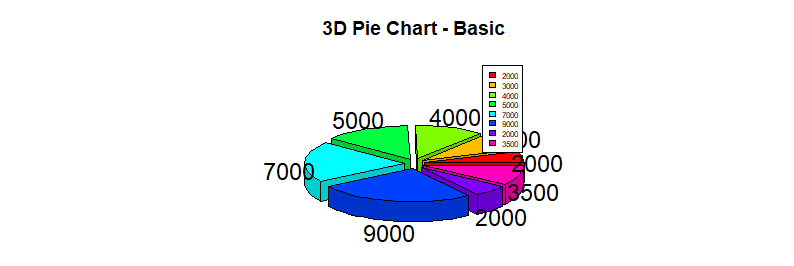
**OUTPUT:**

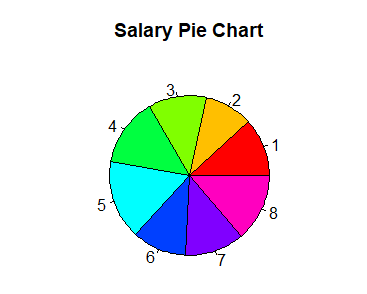


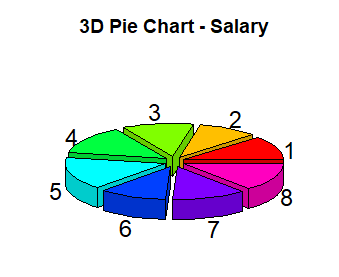












**RESULT**

Thus, our program has been successfully saved and executed.

**15. Generate  BarChart**

**Aim:**

To write a R Program to create a Bar Chart in R for displaying revenue values of different months.

**Algorithm:**

**Step 1:** Start the process to create a bar chart using R.

**Step 2:** Open RStudio and write the program.

**Step 3:** Create a numeric vector containing the revenue values.

**Step 4:** Create another vector containing the month names as labels.

**Step 5:** Use the barplot() function with the following arguments:

height → revenue values

names.arg → months

xlab → label for x-axis

ylab → label for y-axis

main → title of the chart

col → bar color

border → border color

**Step 6:** Execute the program to display the bar chart.

**Step 7:** End the program.

**Program :**

h<-c(7,8,9,10,11)

#png(file="bar")

barplot(h)

months <- c("Mar", "Apr", "May", "Jun", "Jul")

revenue <- c(7, 8, 9, 10, 11)

barplot(revenue,

names.arg = months,

xlab = "Month",

ylab = "Revenue",

col = "blue",

main = "Revenue Chart",

border = "red")

#group and stacked Bar chart

colors<-c("green","orange","brown")

months<-c("Mar","Apr","Jun","Jul")

regions<-c("East","west","North")

values<-matrix(c(2,9,3,11,9,4,8,7,3,12,5,2,8,10,11),nrow=3,ncol=5,byrow=TRUE)

barplot(values,main="Total Forecats",names.arg=months,xlab="months",ylab="Forecast",col=colors)

legend("topleft",regions,cex=1.2,fill=colors)

#CSV File

setwd("D:/24PCA014/Practical/Barplot")

df<-read.csv("Combined.csv")

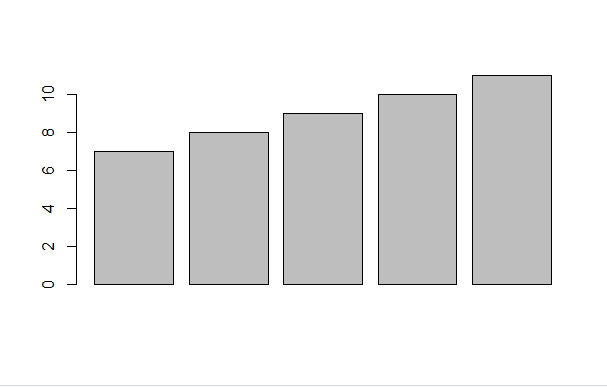
print(df)

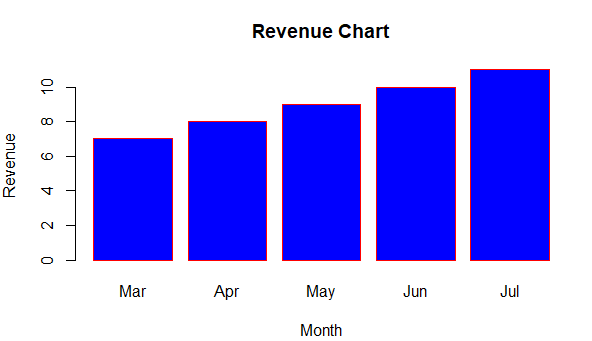
v<-df[,c("Basic")]

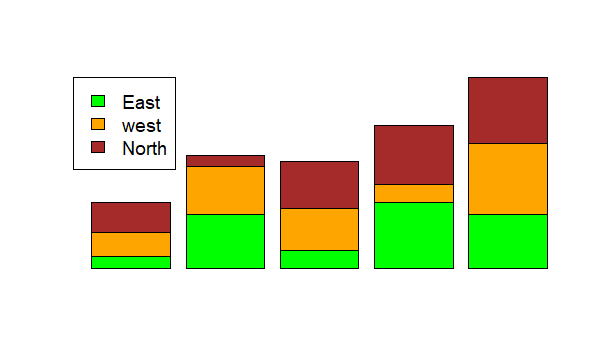
print(v)

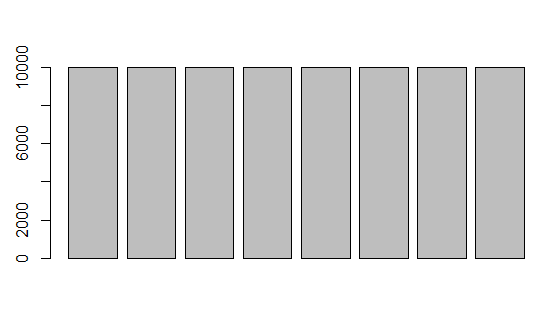
barplot(v)

**OUTPUT:**









**RESULT**

Thus, our program has been successfully saved and executed.