**LAB 02**

**DATA FRAME & VISUALIZATION**

**Aim: Data frame and visualization**

**New-Term used:**

1. **data(): This function is used to call data. In this code, it is not used explicitly.**
2. **c(): This function is used to create vectors by combining multiple values. It is used to create vectors for empid, age, gender, and status.**
3. **data.frame(): This function is used to create a data frame from vectors. It is used to create the data frame empinfo using the vectors empid, age, gender, and status.**
4. **factor(): This function is used to convert a variable into a factor variable. It is used to convert the gender and status variables in the empinfo data frame into factors with labels "male"/"female" and "staff"/"faculty", respectively.**
5. **subset(): This function is used to subset rows of a data frame based on specified conditions. It is used to extract subsets of empinfo data frame based on gender ("male" or "female") and status ("staff" or "faculty").**
6. **summary(): This function is used to compute summary statistics of a vector or data frame. It is used to calculate summary statistics for the subsets male, female, staff, and faculty.**
7. **table(): This function is used to create contingency tables or frequency tables. It is used to create one-way (table1 and table2) and two-way (table3) tables based on the gender and status variables in the empinfo data frame.**
8. **plot(): This function is used to create various types of plots. In this code, it is used to create a scatterplot of age against employee ID.**
9. **pie(): This function is used to create a pie chart. It is used to create a pie chart based on the frequencies in table1.**
10. **barplot(): This function is used to create bar plots. It is used to create grouped and stacked bar plots based on the frequencies in table3.**
11. **legend(): This function is used to add legends to plots. It is used to add legends to the bar plots.**
12. **boxplot(): This function is used to create box plots. It is used to create a box plot of age grouped by gender. The black line represents the mean value.**

**Input:**

**#data frame and visualization - title**

**#to call data - data()**

**#Computing summary statistics / plotting**

**#using tabulation and graphical representation**

**#creating a vector empid**

**#15 data points each**

**empid = c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15);**

**empid**

**age = c(13,24,35,46,56,34,23,12,12,23,34,45,67,78,12);**

**age**

**gender = c(1,1,1,0,1,1,0,1,1,0,1,0,1,1,0);**

**gender**

**status=c(1,2,2,1,2,2,1,1,2,1,2,1,2,1,2);**

**status**

**#data.frame() command**

**#make a table using data.frame**

**#table name is empinfo**

**empinfo = data.frame(empid, age, gender, status);**

**empinfo**

**#factor() command**

**#Tabelling character to numeric**

**#changing 0s to male and 1s to female**

**empinfo$gender = factor(empinfo$gender, labels = c("male", "female"))**

**empinfo**

**#changing 1s to staff and 2s to faculty**

**empinfo$status = factor(empinfo$status, labels = c("staff","faculty"))**

**empinfo**

**#Extract male data**

**male = subset(empinfo, empinfo$gender =="male")**

**male**

**#extracting female data**

**female = subset(empinfo, empinfo$gender == "female")**

**female**

**#extracting staff data**

**staff = subset(empinfo, empinfo$status == "staff")**

**staff**

**#extracting faculty data**

**faculty = subset(empinfo, empinfo$status == "faculty")**

**faculty**

**#summary statistics of male, female etc**

**summary(male)**

**summary(female)**

**summary(staff)**

**summary(faculty)**

**#To get the cross-section of the column data**

**#creating a table(1 - way)**

**table1 = table(empinfo$gender)**

**table1**

**table2 = table(empinfo$status)**

**table2**

**#creating a table(2 - way)**

**table3 = table(empinfo$gender, empinfo$status)**

**table3**

**#Graphical representation(scatterplot)**

**plot(empinfo$age, type="l", main="Age of employees",xlab="employee id",ylab="age in years",col="blue")**

**#Graphical representation(Pie Chart)**

**pie(table1)**

**#Grouped Bar plot - beside=T**

**barplot(table3, beside=T, xlim=c(1,15), ylim=c(0,5),col=c("purple", "red"))**

**#Stacked Bar Plot - beside=F**

**barplot(table3, beside=F, xlim=c(1,15), ylim=c(0,5),col=c("purple", "red"))**

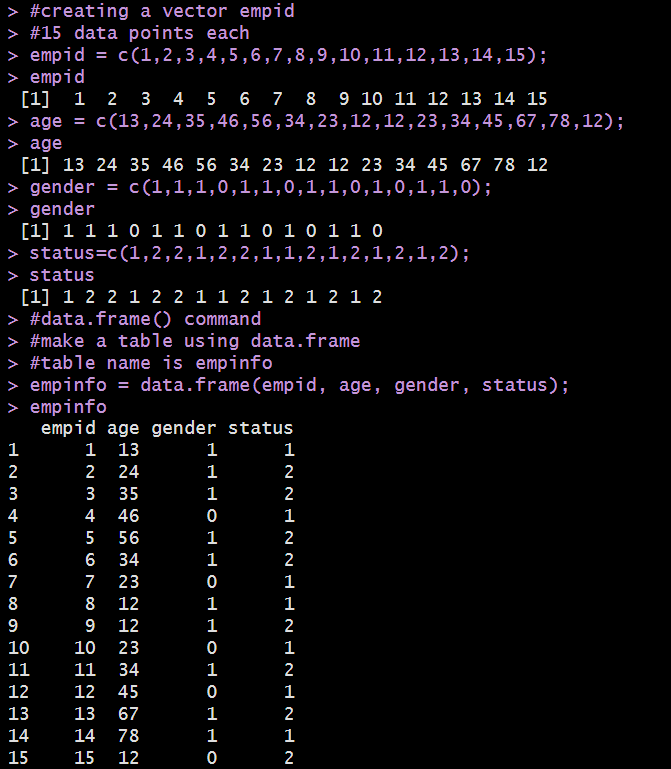
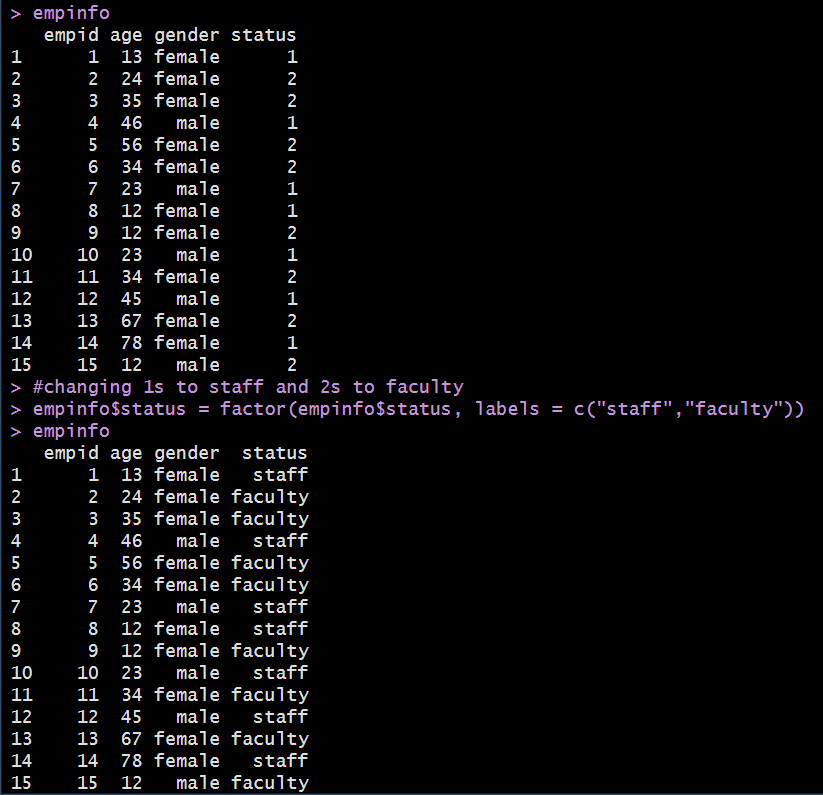
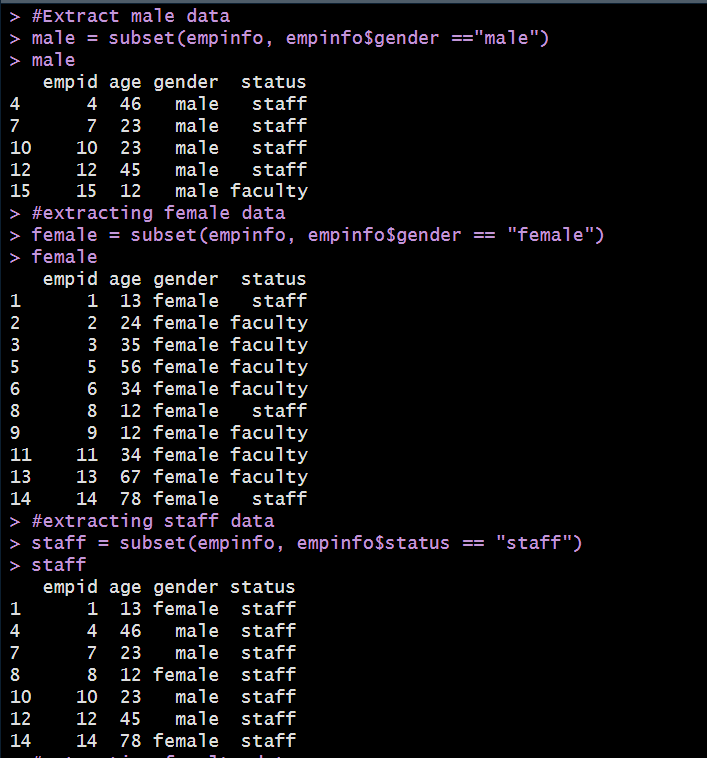
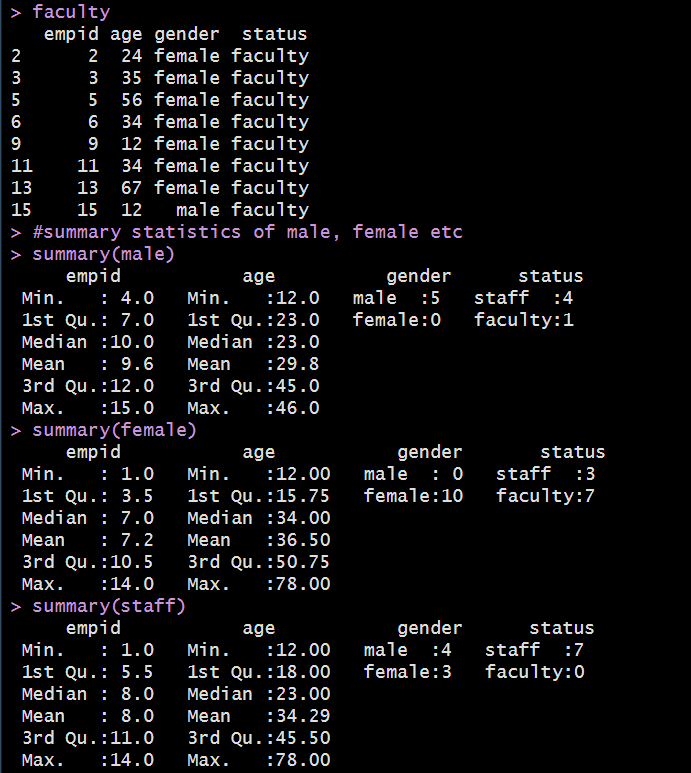
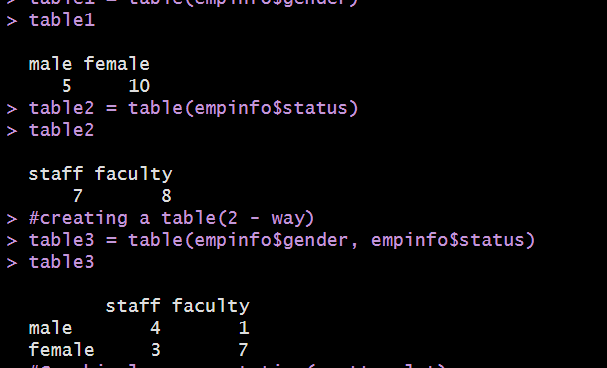
**legend("topright",legend=rownames(table3),fill=c("purple","red"),bty="n")**

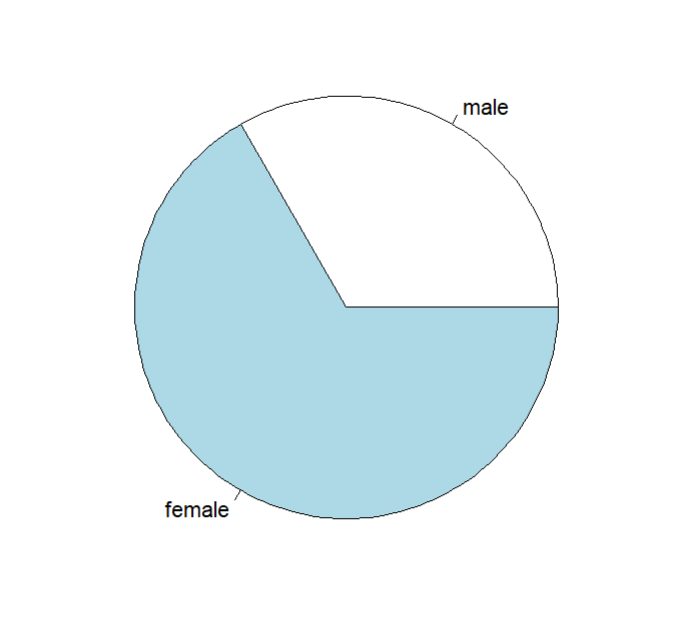
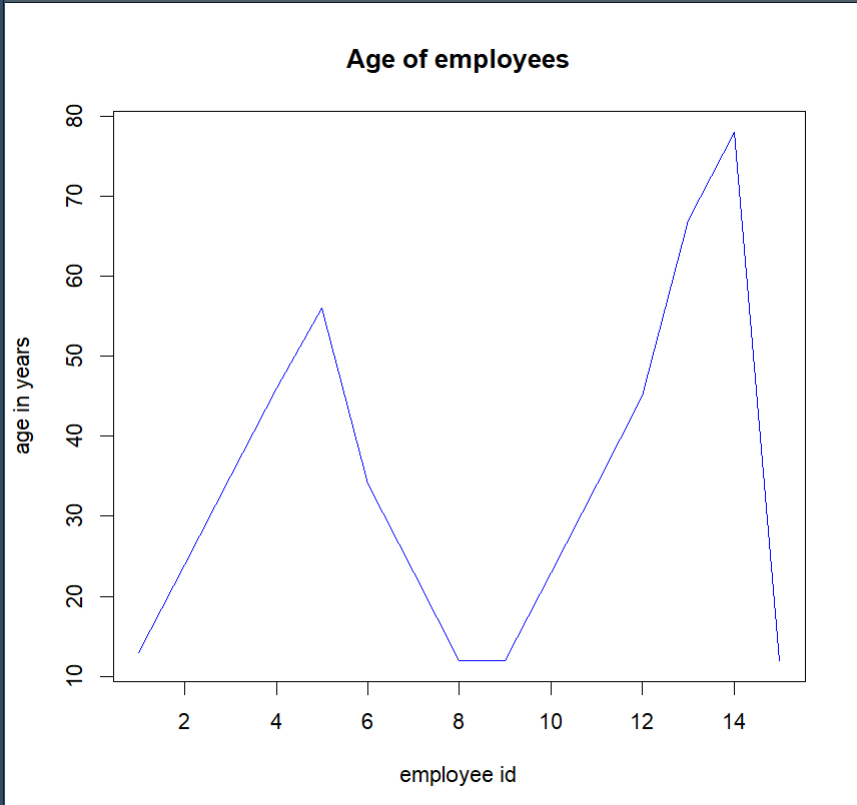
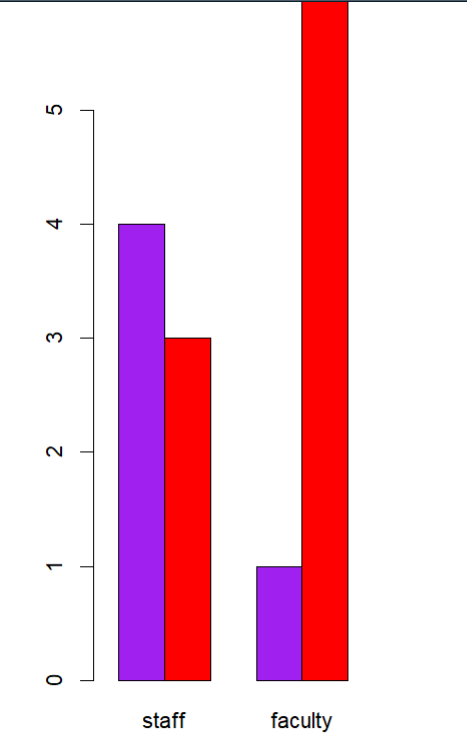
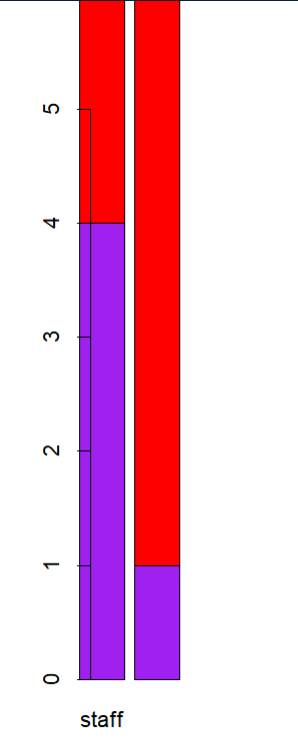
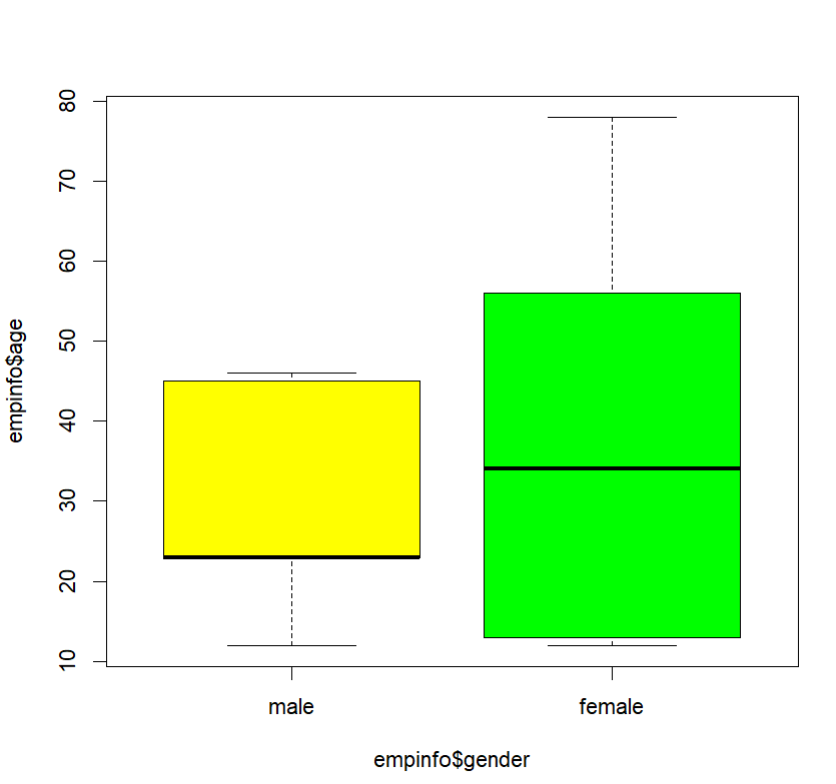
**legend("bottomright",legend=rownames(table3),fill=c("purple","red"),bty="s")**

**#Box plot**

**boxplot(empinfo$age~empinfo$gender,col=c("yellow","green"))**

**#black line - is the mean value**

**OUTPUT:**

**Graphs:**

**ASSIGNMENT**

* **Create a student data frame for student data sets with 5 variables and 10 rows.**
* **From the data frame remove a particular column**
* **Add one more variable**
* **Rename 1 of the columns**
* **Explore the various graphical representation for the above data frame(5 additional plots)**

**CODE:  
#1. Creating a student data frame for student data sets with 5 variables and 10 rows**

**names = c("Ram","Karan","Sita","Puneeth","Ajay","Lakshman","Rohan","Vijay","Arati","Ananya");**

**names**

**gender = c(1,1,0,1,1,1,1,1,0,0);**

**rollno = c(21:30)**

**age = c(17,18,19,17,16,18,19,17,16,19);**

**cgpa = c(9,8,9,7,6,7,7,9,9,10);**

**database = data.frame(names, rollno, gender, age, cgpa);**

**database**

**#0 - female ; 1 - male**

**database$gender = factor(database$gender, labels=c("female","male"));**

**database**

**#2. Removed Gender variable**

**DB1 <- subset(database, select = -c(age))**

**DB1**

**#3. Add one more variable**

**DB1$weight = c(45,34,34,56,47,34,23,34,36,40);**

**DB1**

**#4. Rename one of the columns**

**#changed rollno to regno**

**colnames(DB1)[2] = "regno"**

**#or**

**#colnames(DB1)[colnames(DB1)=="rollno"]="regno**

**DB1**

**#5. Graphical representation**

**#1. Scatterplot**

**plot(DB1$regno,DB1$weight, type="l", main="Weight of students", xlab="studentID", ylab="weight", col="red")**

**#2. Pie chart**

**table1= table(DB1$gender);**

**table1**

**pie(table1)**

**table3 = table(DB1$gender, DB1$cgpa);**

**table3**

**#3. bar plot**

**barplot(table3, beside=T, xlim=c(1,15), ylim=c(0,5),col=c("pink", "yellow"), xlab="cgpa", ylab="number of students")**

**legend("topright",legend=rownames(table3),fill=c("pink","yellow"),bty="n")**

**#4. stacked plot**

**barplot(table3, beside=F, xlim=c(0,15), ylim=c(0,5),col=c("pink", "yellow"), xlab="cgpa", ylab="number of students")**

**legend("topright",legend=rownames(table3),fill=c("pink","yellow"),bty="n")**

**#5. Box plot**

**boxplot(DB1$cgpa~DB1$gender,col=c("pink","yellow"))**

**legend("topright",legend=rownames(table3),fill=c("pink","yellow"),bty="n")**

**#6. Density Plot**

**hist(DB1$weight, breaks = 10, col = "violet",**

**main = "Histogram of weight of students",xlab = "Weight", prob = TRUE)**

**#7. Violin Chart**

**library(ggplot2)**

**# Load sample data**

**tips <- DB1**

**# Create violin plot**

**ggplot(tips, aes(x = DB1$cgpa, y = DB1$weight)) +**

**geom\_violin() +**

**# Set plot labels**

**ggtitle("Violin plot of weight vs cgpa") +**

**xlab("cgpa") +**

**ylab("weight of student")**

**#8. Line Chart**

**plot(DB1$regno,DB1$cgpa, type="o", main="Weight of students", xlab="studentID", ylab="weight", col="red")**

**#9.Dot chart**

**dotchart(DB1$cgpa, labels = DB1$name, cex = 0.75, main = "Dot chart of marks of each student", xlab = "cgpa")**

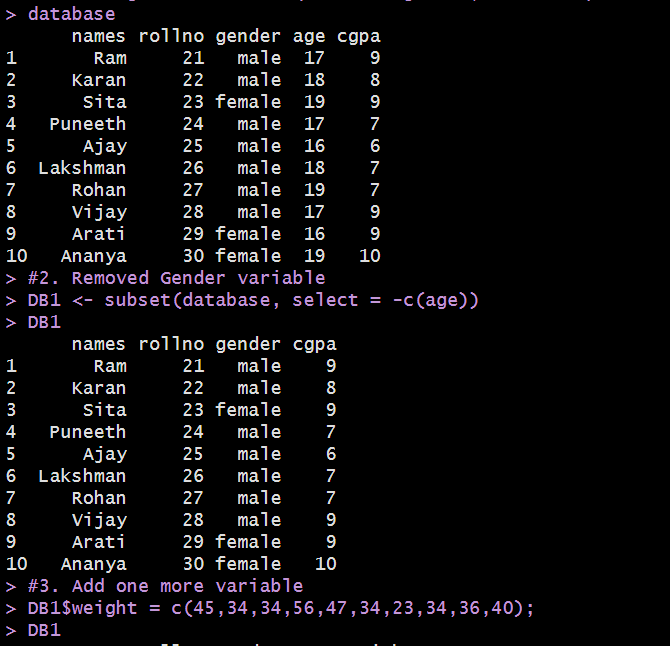
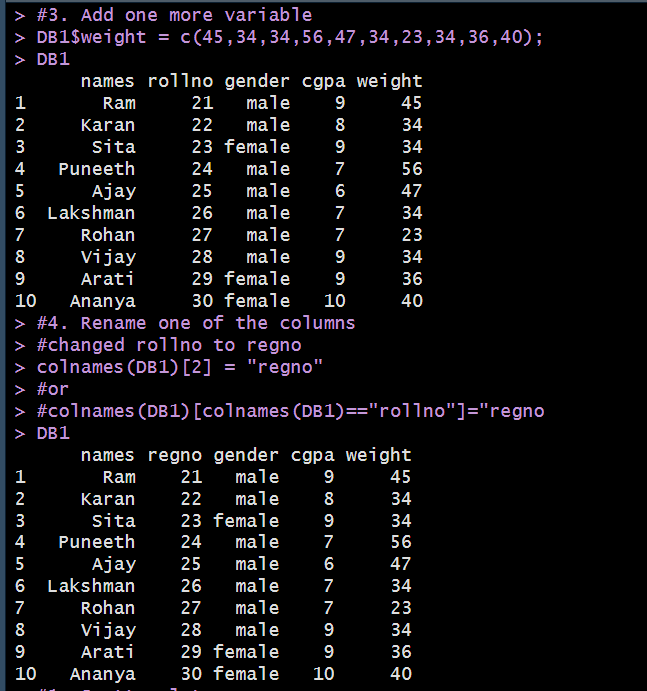
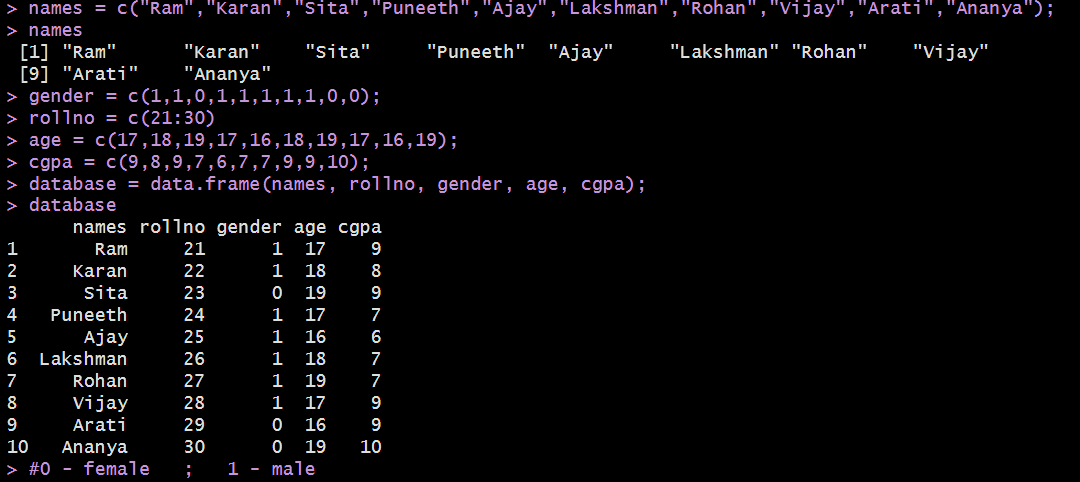
**#10. Heatmaps**

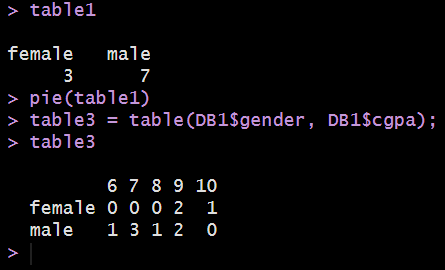
**data <- matrix(DB1$weight, nrow = 3, byrow = TRUE)**

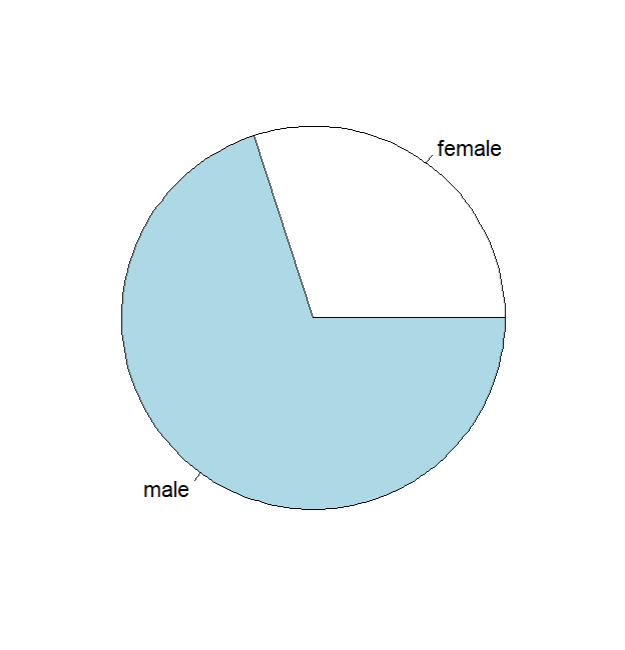
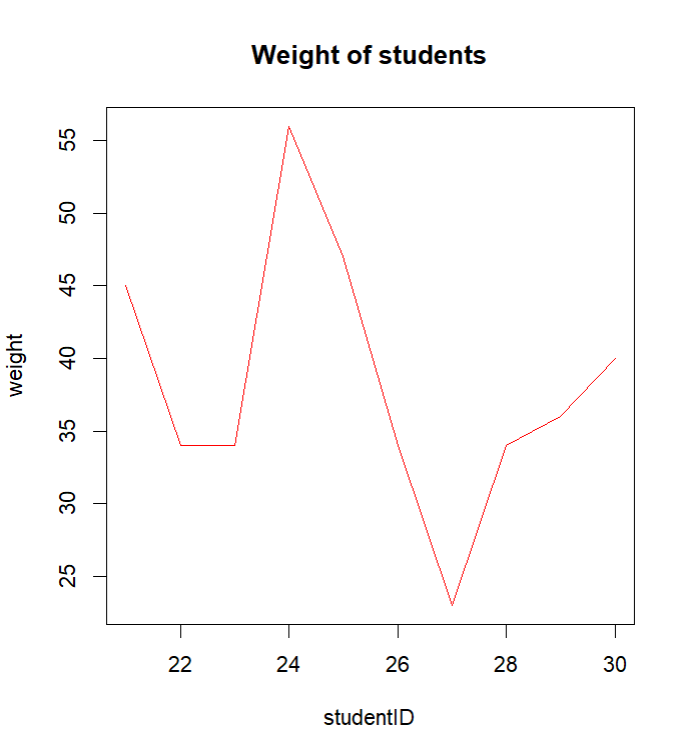
**rownames(data) <- c("Row 1", "Row 2", "Row 3")**

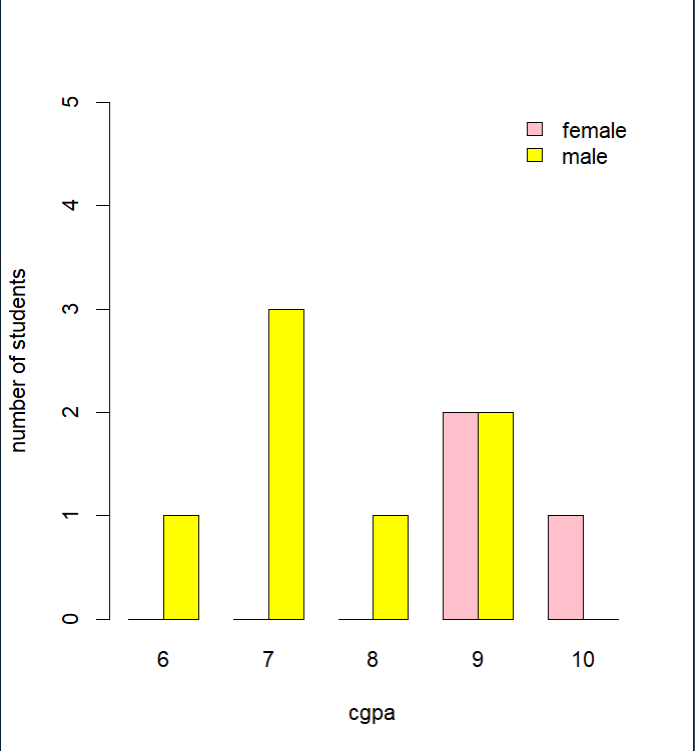
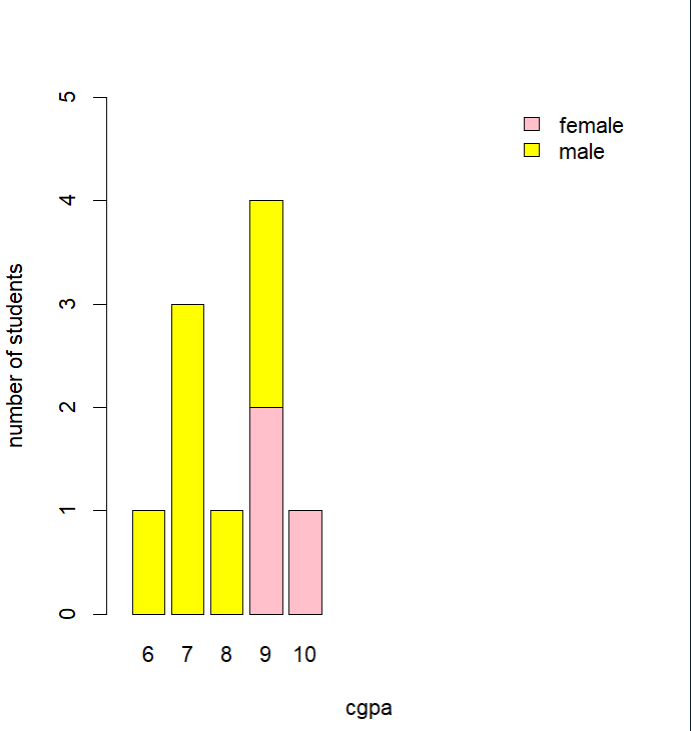
**colnames(data) <- c("Col 1", "Col 2", "Col 3")**

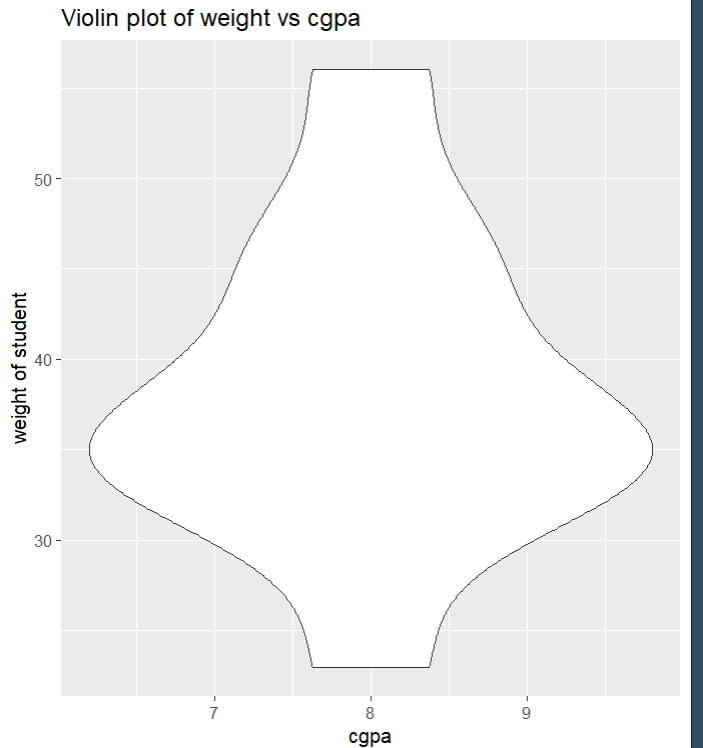
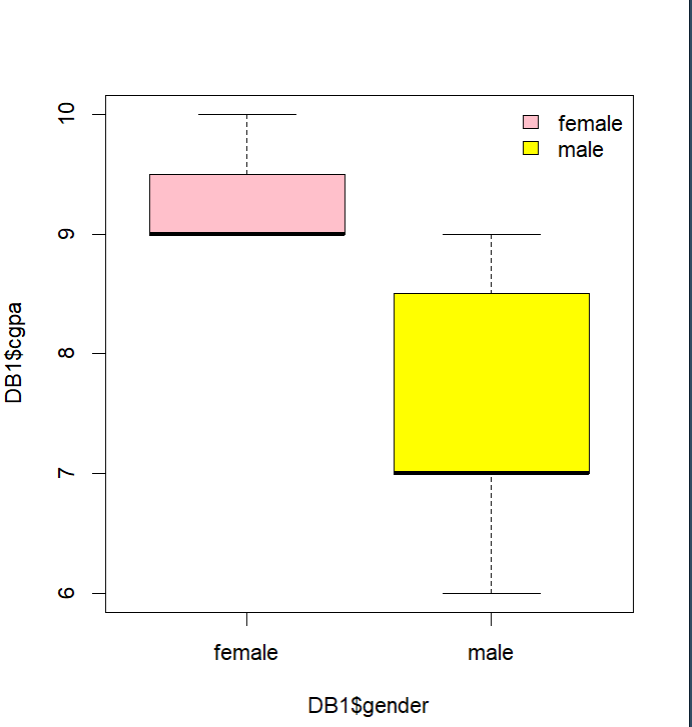
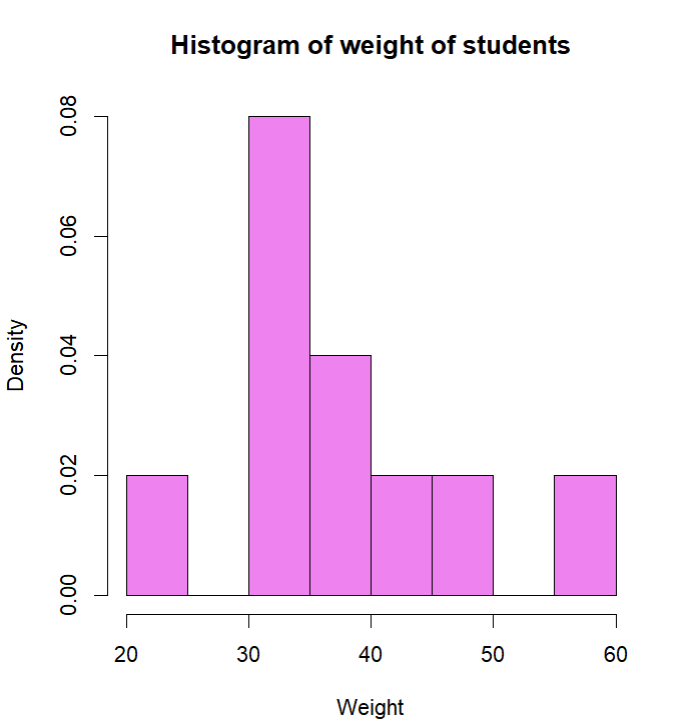
**heatmap(data, Rowv = NA, Colv = NA, col = heat.colors(256), margins = c(5, 10), main = "Heatmap Example")**

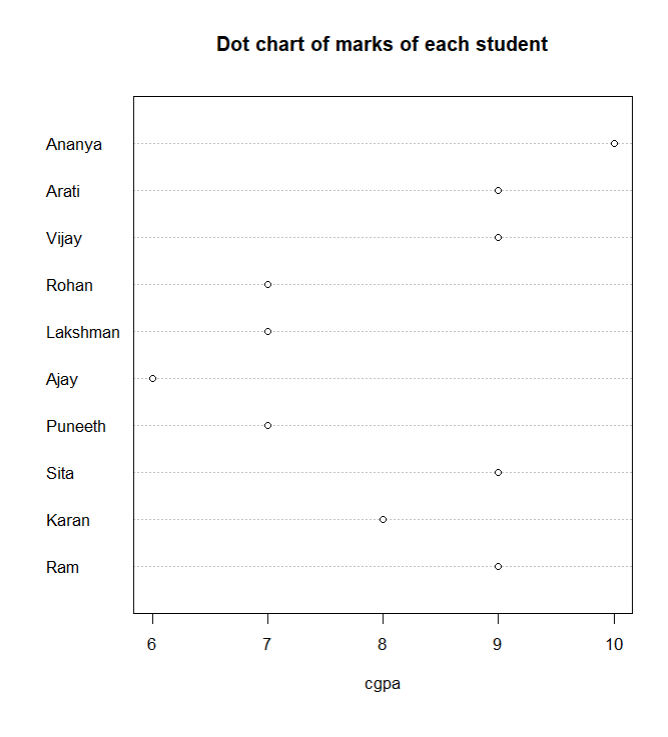
**OUTPUT**

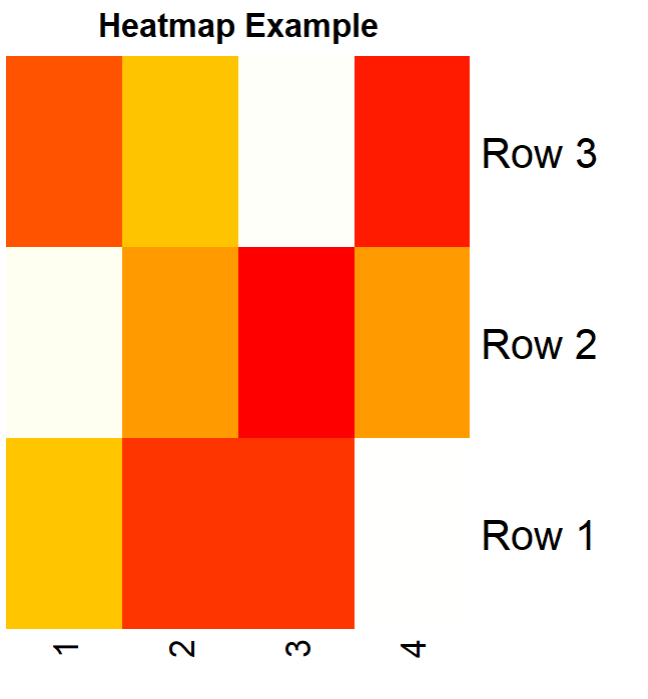
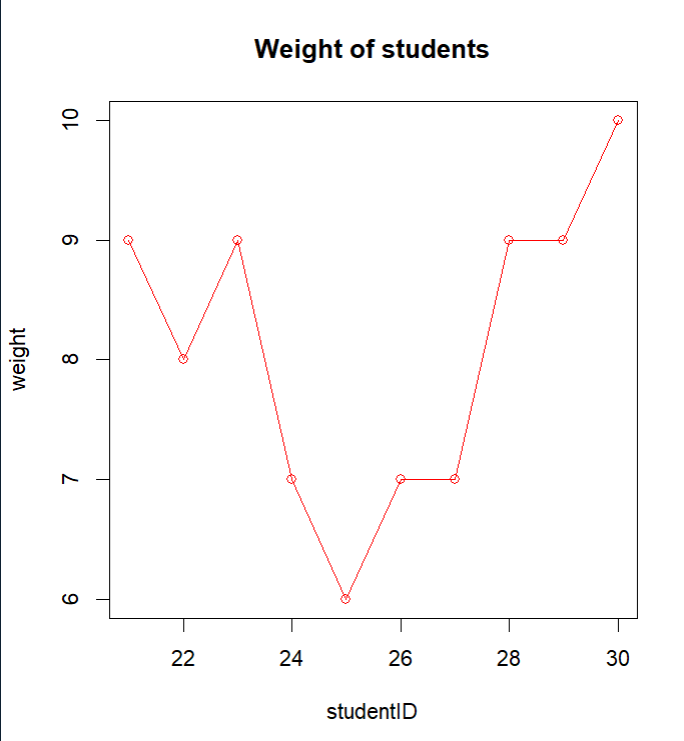
****

**Graphs**

****

****

****

****

**Inference:**

**Hence data can be plotted and visualised in R programming.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***