**Practical No. : 01**

**Aim: EDA & Data Visualization**

1. **To Read data from CSV file to R**

**Diabetes <- read.csv(file.choose(),header = TRUE,sep = ",")**

* Diabetes: is the variable we are creating to store the csv file in form of data frame
* Read.csv :is used as we are reading csv file
* File.choose () :wil open the browser to select desired csv file
* Header=TRUE: will treat first row as a header
* Sep: as we have csv that comma separated file , we use “,”

1. **To extract first few lines of data set**

**head(data\_set\_name)**

**head(eda\_data) #dataset name is eda\_data**

* This will produce first few , by default 6 lines of the dataset

1. **To check data type of every variable in dataset.**

It completely displays the internal structure of R object.

**str(data\_set\_name)**

* The term ‘category’ and ‘enumerated type’ are also used for factors

1. **To check summary of entire data frame object**

**summary(data\_set\_name)**

1. **To check first 10 rows of the dataset**

**data\_set\_name [row\_no:column\_no,]**

* **1st row and 10th column** **[1:10,]**

1. **To check only 2 columns of the dataset**

**data\_set\_name [,row\_no:column\_no]**

* **1st row and 2nd column [,1:2]**

1. **To display first 10 rows and only 2 columns of the dataset**

data\_set\_name [row, column]

* **1st to 10th row : 1st and 2nd column [1:10, 1:2]**

NOTE:

1. when we want to fetch rows we mention **datasetname rows [no\_of\_,]**
2. When we want to fetch columns we mention **datasetname[,no\_of\_cols]**
3. When we want to fetch both we write **datasetname[no\_of\_rows, no\_of\_cols]**
4. **To display observations having no of students who have done Graduation**

Syntax :

newdata1<-subset(datasetname, datasetname$column\_name=="value")

newdata1

**Example :**

**newdata1<-subset(EDA\_data,EDA\_data$Education == "Grad")**

**> newdata1**

* Here we created new variable called as newdata1 and we are storing the subset of EDA\_data data in newdata1. It is mandatory to write again newdata1 in order to view the output on console as we are creating new variable to store result of subsetting

1. **To display multiple conditions for subsetting**

**newdata2<-subset(EDA\_data, EDA\_data$Age=="51" & EDA\_data$Gender=="M")**

**>newdata2**

* Here we extracted details of students whose age is 51 and gender is Male.

1. **To sort the data of a column in ascending order**

* Sorting Data

To sort a data frame in R, use the order( ) function. By default, sorting is ASCENDING. Prepend the sorting variable by a minus sign to indicate DESCENDING order. Here are some examples.

Syntax:

newdata4 <-datasetname[order(datasetname$column\_name), ]

>newdata4

**Example :**

**i.) Newdata4 <- EDA\_data[order(EDA\_dataset&Name),]**

**>Newdata4**

**ii.) newdata5<-EDA\_data[order(EDA\_data$Education, EDA\_data$Salary),]**

**> newdata5**

* we are sorting on all rows hence we are not writing anything after ,

1. **To sort the data of a column in descending order**

**newdata5<-EDA\_data[order(-EDA\_data$Name),]**

**>newdata5**

**OR**

**Newdata5 <- EDA\_data[order(EDA\_data&Age, decreasing = TRUE),]**

* For Descending order we can use decreasing = TRUE

1. **To check if any column contains missing observation**

**colSums(is.na(datasetname))**

* NA is a logical constant of length 1 which contains a missing value indicator.

**Histogram, boxplot, scatterplot, barplot**

1. **To plot Histogram of a particular column in dataset**

**hist(datasetname$column\_name)**

1. **To plot boxplot of a particular column in dataset**

**boxplot(datasetname$column\_name)**

1. **To view properties of particular column of data**

**mean(datasetname$column\_name)**

**median(datasetname$column\_name)**

**max(datasetname$column\_name)**

**min(datasetname$column\_name)**

**mode:**

**y<-table(eda\_data$Baths)**

**names(y)[which(y==max(y))]**

**my\_mode <- function(x) { # Create mode function**

**unique\_x <- unique(x)**

**tabulate\_x <- tabulate(match(x, unique\_x))**

**unique\_x[tabulate\_x == max(tabulate\_x)]**

**}**

**my\_mode(eda\_data$Baths)**

**outlier:**

1)detection

2) Handle Outliers

#Capping outliers with the 95th/5th percentile (.95% 0.05%)

**missing value**

1. Skewness
2. Hist
3. Normally distributed – impute by mean
4. Skewed – median
5. Categorical – mode

**Conversion**

1. **As.factor()**
2. **As.numeric()**
3. **As.logical()**
4. **As.character()**

**EDA R PRACTICAL**

eda\_data<-read.csv(file.choose(), header=TRUE, sep=",")

eda\_data

head(eda\_data)

summary(eda\_data)

str(eda\_data)

eda\_data[1:8,]

head(eda\_data,3)

head(eda\_data,8)

tail(eda\_data,8)

eda\_data[1:8, c(1,5)]

eda\_data[,1:5]

newdata1<-subset(eda\_data,eda\_data$Education == "Grad")

newdata1

newdata2<-subset(eda\_data, eda\_data$Age=="51" & eda\_data$Gender=="M")

newdata2

a<-eda\_data[order(eda\_data$Name),]

a

a<-eda\_data[order(eda\_data$Education),]

a

a<-eda\_data[order(eda\_data$Education, decreasing = TRUE),]

a

a<-colSums(is.na(eda\_data))

a

hist(eda\_data$Age)

boxplot(eda\_data$Age)

mean(eda\_data$Age)

min(eda\_data$Age)

max(eda\_data$Age)

median(eda\_data$Age)

mode(eda\_data$Garage)

y<-table(eda\_data$Garage)

y

names(y)[which(y==max(y))]

ma<-max(y)

ma

whch<-which(y==ma)

whch

names(y)[whch]

x<-eda\_data$Garage

x

y<-unique(x)

y

mat<-match(x, y)

mat

tab<-tabulate(mat)

tab

m<-max(tab)

m

y[tab==m]

x<-eda\_data$Age

x

my\_mode <- function(x) { # Create mode function

unique\_x <- unique(x)

tabulate\_x <- tabulate(match(x, unique\_x))

unique\_x[tabulate\_x == max(tabulate\_x)]

}

my\_mode(x)

x<-c(0,0,0,1,1,1,1,2,2,2,2,4)

x

y<-table(x)

y

y[max(y)]

hist(eda\_data$Rooms)

hist(eda\_data$Salary)

#b<-skewness(eda\_data$Rooms)

#hist(b)

#Two-way table

#barplot

counts = table(eda\_data$Education,eda\_data$Gender)

counts

barplot(counts, main = "Data distribution by Education Vs Gender",col = c("blue","red"))

plot(eda\_data$Education,eda\_data$Gender, col = c("blue","red"))

#scatterplot

plot(eda\_data$Age, eda\_data$Salary)

library(PerformanceAnalytics)

a<-skewness(eda\_data$Rooms)

a

hist(eda\_data$Rooms)

#imputing missing values

library(e1071)

b<-skewness(eda\_data$Garage)

b

hist(eda\_data$Garage)

#library(ggplot2)

#ggplot(eda\_data$Rooms, x=returnsstat\_density(geom = "line"))

eda\_data$Garage[is.na(eda\_data$Garage)]<-mean(eda\_data$Garage, na.rm=TRUE)

View(eda\_data)

skewness(eda\_data$Rooms)

a

hist(a)

hist(eda\_data$Rooms)

b<-eda\_data$Rooms[is.na(eda\_data$Rooms)]<-median(eda\_data$Rooms, na.rm=TRUE)

b

hist(b)

View(eda\_data)

#mode

getmode <- function(v){

v=v[nchar(as.character(v))>0]

uniqv <- unique(v)

uniqv[which.max(tabulate(match(v, uniqv)))]

}

#Identifying duplicate data

data<-eda\_data[1:5, 3:4]

data

duplicated(data)

#removing duplicate data

a<-data[!duplicated(data),]

a

#removing an outlier

boxplot(eda\_data$AppraisedValue)

plot(eda\_data$AppraisedValue)

x<-eda\_data$AppraisedValue

x

out <- boxplot.stats(x)$out #identifying the outlier

out ## `boxplot.stats` has picked them out 1200 value

x<-x[!(x %in% out)]

x ## this removes 1200 from x

boxplot(x)

#imputing

q<-quantile(eda\_data$AppraisedValue, .95) #95th percentile

q #850

summary(eda\_data$AppraisedValue)

#ifelse(2==2, "equal", "not equal") #example of ifelse

app\_val<- ifelse(eda\_data$AppraisedValue >= 1000,850,eda\_data$AppraisedValue)

app\_val

boxplot(app\_val)

#conersion : character to numeric values

str<-eda\_data$Gender

str

str(eda\_data$Gender)

str(eda\_data$Education)

num<-as.numeric(str)

num

str(num)

typeof(num)

class(num)

num<-as.factor(num)

num

class(num)

num<-as.character(num)

num

class(num)

typeof(num)

#numeric to logical values

v<-c(0, 0, 1, 1)

v

logi<-as.logical(v)

logi

#logical to numeric

int<-as.integer(logi)

int

typeof(int)

fact<-as.factor(int)

fact

str(eda\_data$Name)

Practical No: 02 (Sequence to be followed)

**1)Heading**

**2)Question**

**3)Dataset – table format**

**4)Hypothesis – H0 & H1**

**5) type the command**

**6)Output – ss of command**

**7)Conclusion**