```
#control statements in R

#if statement
a=20
b=34
if(a<b){
  "A is less than B"
}
x=20
y=50
if(x!=y)
{
  "X not equal to y"
}

#if else statement
p=250
q=100
if(p==q){
  print("p and q are equal")
}else{
  print("p and q are not equal")
}

a1=345
b1=890
if(a1!=b1){
  print("they are not equal")
}else{
  print("they are equal")
}

#working with csv files
getwd()
#read csv from default directory
```

```
rdcsv=read.csv("Instagram_Analytics.csv")
View(rdcsv) #entire dataset
head(rdcsv) #first 6 rows
rd=read.csv("API_AG.LND.TOTL.K2_DS2_en_csv_v2_511817.csv")
head(rd)
tail(rd)
View(rd)
summary(rd)
csv=read.csv(file="C:\\Users\\Charitha K\\OneDrive\\Documents\\mba3\\API_AG.LND.TOTL.K2_DS2_en_csv_v2_511817.csv")
head(csv)

#data frames
df=data.frame(
  st_name=c("chari","pari","siri"),
  st_sub=c("BA","FM","HR"),
  st_marks=c(82,78,95)
)
print(df)

#extracting column data
df$st_name
df$st_sub
df$st_marks
df[1]
df[1,1]
df[3,3]

df1=data.frame(
  st_name=c("chari","pari","siri"),
  st_sub=c("BA","FM","HR"),
  st_marks=c(82,78)
) #error

#add new column
```

```
df$st_result=c("FC","SC","DIST")
print(df)
df[1,]

#product table
df_product=data.frame(
  prod_id=c(1001,1002,1003,1004),
  prod_name=c("iphone","accessories","clothes","watch"),
  prod_price=c(100000,25000,1000,5000),
  e_platform=c("amazon","flipkart","flipkart","myntra")
)
print(df_product)

#factors
ex_gen=factor(c("male","female","male","male"))
print(ex_gen)
#Data Structures
#vector
ex_vector=c(12,34.6,89,999) #c is concat
print(ex_vector)
#To access particular value from the vector
ex_vector[2]
ex_vector[3]
ex_vector[5]
#character values
ex_char=c("R","BI","IAPM","CVFM","SMB","AI","CT")
print(ex_char)
ex_char[2]
ex_char[4]
ex_char[-1]
ex_char[-2]

#LIST-hetrogenous
ex_list=c(67,"MBA",89,90.2,"BI","CVFM",678)
print(ex_list)
print(ex_list[4])
ex_list[5]



a1=134 #variable
b1=456
c1=a1+b1
print(c1)

a2="mba" #character data
print(a2)

x1=45.6
y1=30.2
print(x1+y1)

print(a1-b1)
print(a1*b1)
print(a1/b1)
print(b1%%a1)



#for loop-repeat task
#example 1
for(i in 1:4)
{
  print(i)
}
print("for loop is over")
#example 2
for (x in seq(1:10))
{
  print(x)
}
#example-3
```

```
p=c(10,12,-25,70)
for (x in p){
  print(x)
}
#example 4
prg=c("HR","fin","bi","ba")
for (a in prg)
{
  print(a)
}
#example 5
list_for=list("HR",1234,"BA",FALSE,7878)
for (i in list_for)
{
  print(i)
}

#example-6
for(i in 1:5)
{
  print(i^2)
}

#break statement
subjects<-list("HR","BI","Marketing","R","FN")
for(x in subjects){
  if(x=="Marketing")
  {
    break  #stop the loop
  }
  print(x)
}

10%%3 #gives the remainder after division
10/2
```

```
for(i in 1:10){
  if(i%%2==0){
    print(paste(i,"is even")) #combining with a text
  }
}

for(j in 1:10){
  if(j%%3==0){
    print(paste(j,"is odd"))
  }
}


#packages
#ggplots example

#Install packages only once
install.packages("ggplot2")

#load the package execute every time
library(ggplot2)

#daiamonds datasets
datadim=diamonds
View(datadim)
summary(diamonds)

#example1
colnames(datadim)
row.names(datadim)

ggplot(data=datadim, aes(x=carat, y=cut, col=color))+
  labs(title = "Diamonds Data Plot")
#Geometric layer
#point plots
```

```
ggplot(data=datadim, aes(x=carat, y=cut, col=price))+
  geom_point()

ggplot(data=datadim, aes(x=carat, y=cut, col=price))+
  geom_point()
+
  labs(title = "diamonds data plot", x= "dimondas carat",y="dimondas cut")

ggplot(data=datadim, aes(x=carat, y=cut, col=color))+
  geom_point(color="pink")+
  labs(title = "diamonds data plot",
       x="diamonds carat",
       y="diamonds cut")

# color support by R language
#color()

# different colors
ggplot(data = diamonds , aes(x=cut))+
  geom_bar(color="green",fill="yellow")

ggplot(data = diamonds , aes(x=color))+
  geom_bar(color="purple",fill="lightblue")

ggplot(data = diamonds, aes(x=cut,y=carat))+
  geom_line(color="red")

#use mtcars dataset
#comparing values across categories
dt=mtcars
View(dt)
#line plot
ggplot(dt,aes(wt,mpg))+
  geom_line(color="red")
```

```r
#Bar Chart
ggplot(dt, aes(x=factor(cyl)))+
  geom_bar(color="black",fill="gray")+
  labs(title="count of cars by cylinders",x="cylinder",y="count")

#histogram
#use for showing the distribution of a continuons variable,
ggplot(dt,aes(x=mpg))+
  geom_histogram(binwidth = 2,fill="orange",color="red")+
  labs(title="histogram of mpg",x="miles per gallon",
       y="frequency")



# list example in R
list_ex=list("siri",123,c(12,56,89)) # list is function
print(list_ex)
list_ex[1]

# example-2
list_ex2= list(name="sony",branch="MBA",marks=c(78,90,99))
list_ex2[2]

# extract by variable name
list_ex2$branch
list_ex2$marks

#matrix function
mat_ex=matrix(c(1:12),nrow=3,ncol=4)
print(mat_ex)
print(mat_ex[2,4])
print(mat_ex[0,0])
print(mat_ex[1,1])

mat_ex2=matrix(seq(1:16),nrow=4,ncol=4,byrow=TRUE)
print(mat_ex2)
print(mat_ex2[3,4])
typeof(mat_ex2) # shows data type of variable
class(mat_ex2) # shows data type of structure

#data frames in R
df_employee=data.frame(emp_id=c(101,102,103,104,105,106),
            emp_name=c("spandu","satwick","manoj","thejus","sonu","shree"),
            emp_salary=c(90000,80000,90000,100000,80000,100000),
            emp_dept=c("HR","MAR","BA","SC","sales","FIN"))
print(df_employee)
df_employee$emp_salary  # $ is for extracting col name
df_employee[1]  # 1 indicates index number
df_employee$emp_desig=c("SM","DA","BAny","MarkAnay","HRAnay","BAnay")
print(df_employee)

#ex 2

customer_info=data.frame(cust_id=c(101,102,103,104,105,106),
            cust_name=c("spandu","satwick","manoj","thejus","sonu","shree"),
           product_name=c("iphone","laptop","cloths","books","pen","cap"),
            quantity=c(1,2,3,4,5,6),
           price=c(34,67,23,12,56,78))
print(customer_info)
customer_info$cust_id  # $ is for extracting col name
customer_info$product_cost=customer_info$quantity*customer_info$price
print(customer_info)

#apply function in R
#dataframe for apply function
df_apply = data.frame(
  x = 1:4,
  y = 5:8,
  z = 10:13
)
```

```
print(df_apply)


#ex 2
#column wise
apply(df_apply,2,sum)
apply(df_apply,2,max)
apply(df_apply,2,min)
apply(df_apply,2,mean)
#row wise
apply(df_apply,1,sum)
apply(df_apply,1,max)
apply(df_apply,1,min)
apply(df_apply,1,mean)
mat_app=matrix(seq(1:16),nrow=4)
mat_app=matrix(1:16,nrow = 4)
print(mat_app)



sp=read.csv("C:\\Users\\Charitha K\\OneDrive\\Documents\\mba3\\StudentsPerformance.csv")
View(sp)
summary(sp)

#while loop
number=1 #variable to store current number
sum=0 #to store current sum
while(number<=10){
  sum=sum+number
  print(sum)
  number=number+2
}
print(sum)



#apply functions
# apply - used for matrix and df
```

```
df_apply=data.frame(x=1:11,y=20:30,z=40:50)
print(df_apply)
apply(df_apply,1,sum) #1 indicate row wise
apply(df_apply,2,sum) # 2 indicate col wise
apply(df_apply,2,max)
apply(df_apply,1,max)
apply(df_apply,1,min)
apply(df_apply,2,min)
apply(df_apply,2,mean)
apply(df_apply,1,mean)
apply(df_apply,2,median)

mat_ex=matrix(c(1:16),nrow=4,ncol=4)
print(mat_ex)
apply(mat_ex,1,sum)
apply(mat_ex,2,sum)
apply(mat_ex,1,max)
apply(mat_ex,2,median)
print(apply(mat_ex,2,sum))

#lapply l stands for list
#used for list and data frames
my_list=list(1:5,seq(1:15),c(-12,78,45,1,2))
my_list
lapply(my_list,sum)
lapply(my_list,max)
lapply(my_list,mean)
lapply(my_list,median)


data=read.csv("C:\\Users\\Charitha K\\OneDrive\\Documents\\mba3\\Air Quality Missing Data.csv")
View(data)
summary(data)
head(data)
tail(data)
```

```
is.na(data) #gives true false for every value
sum(is.na(data)) #to find NA in the dataset
data1=na.omit(data)
summary(data1)

#monday class
aircsv=read.csv("C:\\Users\\Charitha K\\OneDrive\\Documents\\mba3\\Air Quality Missing Data.csv")
is.na(aircsv)
sum(is.na(aircsv)) #To find NA in the data set
na.omit(aircsv)
View(aircsv)
colSums(is.na(aircsv)) #columnwise NAs

#to draw charts
install.packages("naniar")
library(naniar)
gg_miss_var(aircsv)

# 1.Remove rows with NA
#Use only if missing values are few

dim(aircsv) #Total no of rows and columns

aircsv_clean = na.omit(aircsv)
View(aircsv_clean)

# 2.Replace with Mean
# ozone column
mean_aircsv=mean(aircsv$Ozone,na.rm = TRUE)

aircsv$Ozone[is.na(aircsv$Ozone)]=mean_aircsv
sum(is.na(aircsv$Ozone)) # Check the column
summary(aircsv)

#median
```

```
median_aircsv=median(aircsv$Solar,na.rm = TRUE)
aircsv$Solar[is.na(aircsv$Solar)]=median_aircsv
sum(is.na(aircsv$Solar)) # Check the column
summary(aircsv)

#tuesday-outliers
summary(aircsv$Ozone)
x=aircsv$Ozone
boxplot(x,main="ozone before outlier treatment",col="blue")

#calculate IQR,Lower bound,upper bound
q1=quantile(x,0.25,na.rm=TRUE)
q3=quantile(x,0.75,na.rm=TRUE)
IQR_value=q3-q1

Lower_Bound=q1-1.5*IQR_value
Upper_Bound=q3+1.5*IQR_value
print(Lower_Bound)
print(Upper_Bound)
#1.Replace outliers with mean
mean_value=mean(x,na.rm=TRUE)
x[x<Lower_Bound|x>Upper_Bound]=mean_value # | or
boxplot(x,main="ozone after outlier treatment",col="pink")

#method 2 apply capping (winsorization)
x[x<Lower_Bound]=Lower_Bound
x[x>Upper_Bound]=Upper_Bound
boxplot(x,main="ozone after capping",col="lightblue")
```