Name:- Mithun

USN:- 19BTRCR006

LAB PROGRAM 3

1. Create some complex data structure variables such as list and data frames using list() and data.frame commands.

In [1]:

```
# creating a dataframe
df <- data.frame(int_col= c(1:5), double_col=c(5,5.1,5.2,5.3,5.4))
df
```

A data.frame: 5 × 2

int_col <int>	double_col <dbl>
1	5.0
2	5.1
3	5.2
4	5.3
5	5.4

In [2]:

```
# creating a list
list_var <- list(1:10,11:20)
print(list_var)
```

```
[[1]]
 [1]  1  2  3  4  5  6  7  8  9 10

[[2]]
 [1] 11 12 13 14 15 16 17 18 19 20
```

2. Create data using data.frames, lists, and tables.

In [3]:

```
# Creating a dataframe using data.frame
let <- letters[1:5]     # generating english letters, for capital letters use LETTERS
d <- data.frame(x = 1, y = 1:10, letters=let)     # factors means categorical variables
d
```

A data.frame: 10 × 3

x	y	letters
<dbl>	<int>	<fct>
1	1	a
1	2	b
1	3	c
1	4	d
1	5	e
1	6	a
1	7	b
1	8	c
1	9	d
1	10	e

In [4]:

```
# Creating a list
let <- letters[1:5]
l <- list(1:5, let)
print(l)
```

```
[[1]]
[1] 1 2 3 4 5

[[2]]
[1] "a" "b" "c" "d" "e"
```

In [5]:

```
# importing iris dataset
iris <- datasets::iris
head(iris)
```

A data.frame: 6 × 5

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
	<dbl>	<dbl>	<dbl>	<dbl>	<fct>
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa

In [6]:

```
# Creating a table and counting the values from the iris dataset
t <- table(iris$Species, iris$Petal.Width)
t
```

```
            0.1 0.2 0.3 0.4 0.5 0.6  1 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9
2
  setosa      5  29   7   7   1   1  0   0   0   0   0   0   0   0   0   0
0
  versicolor  0   0   0   0   0   0  7   3   5  13   7  10   3   1   1   0
0
  virginica   0   0   0   0   0   0  0   0   0   0   1   2   1   1  11   5
6

            2.1 2.2 2.3 2.4 2.5
  setosa      0   0   0   0   0
  versicolor  0   0   0   0   0
  virginica   6   3   8   3   3
```

3. Implement basic R operations (data input, missing values, Importing data into R using different formats : xlsx, CSV, Text files).

- use read.text for reading a dataset of format .txt
- use read.excel for reading a dataset of format .xlsx

In [9]:

```
# importing a csv file to a dataframe
data <- read.csv('titanictrain.csv')
head(data)
```

A data.frame: 6 × 12

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
	<int>	<int>	<int>	<fct>	<fct>	<dbl>	<int>	<int>	<fct>	<dbl>
1	1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.2500
2	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833
3	3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.9250
4	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1000
5	5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.0500
6	6	0	3	Moran, Mr. James	male	NA	0	0	330877	8.4583

In [10]:

```
sum(is.na(data))
```

177

In [11]:

```
sum(is.na(data$Age))   # age feature is having all the 177 missing values
```

177

In [12]:

```
# removing the NA values
data_new <- na.omit(data)
sum(is.na(data_new))
```

0

In [13]:

```
cat("Rows before removing the NA values:", nrow(data), "\n")  # shape before removing the N
cat("Rows after removing the NA values: ", nrow(data_new) )  #   (891-177)
```

```
Rows before removing the NA values: 891
Rows after removing the NA values:  714
```

In [14]:

```
# filling the missing values with mean
data[is.na(data)]= mean(data_new$Age)
sum(is.na(data))
```

0

4. Explore data type conversions from one data structure to another with commands such as as.data.frame(), as.vector(), is.data.frame(), is.vector; and find the data type with class() command.

In [15]:

```
head(data)    # a dataframe
```

A data.frame: 6 × 12

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
	<int>	<int>	<int>	<fct>	<fct>	<dbl>	<int>	<int>	<fct>	<db
1	1	0	3	Braund, Mr. Owen Harris	male	22.00000	1	0	A/5 21171	7.25(
2	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38.00000	1	0	PC 17599	71.28:
3	3	1	3	Heikkinen, Miss. Laina	female	26.00000	0	0	STON/O2. 3101282	7.92:
4	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.00000	1	0	113803	53.10(
5	5	0	3	Allen, Mr. William Henry	male	35.00000	0	0	373450	8.05(
6	6	0	3	Moran, Mr. James	male	29.69912	0	0	330877	8.45:

In [16]:

```
# is.data.frame()
is.data.frame(data)
```

TRUE

In [24]:

```
# converting a list into vector
list_vec <- list(1:10)
data_vec= as.vector(list_vec)
is.vector(data_vec)
```

TRUE

In [27]:

```
# converting a vector into dataframe
data_new <- as.data.frame(data_vec)
is.data.frame(data_new)
```

TRUE

5. Explore function programming in R.

In [37]:

```
# Printing the squares
square_func <- function(a)
{
  for(i in 1:a)
  {
    b <- i^2
    print(b)
  }
}
square_func(6)
```

```
[1] 1
[1] 4
[1] 9
[1] 16
[1] 25
[1] 36
```

6. Explore loops in R programming such as if-else-ifelse, for, while, repeat-break, etc

In [38]:

```
# for loop
cubic_func <- function(a)
{
  for(i in 1:a)
  {
    b <- i*3
    print(b)
  }
}
cubic_func(6)
```

```
[1] 3
[1] 6
[1] 9
[1] 12
[1] 15
[1] 18
```

In [39]:

```
# while loop
power<-2
i<-1
while(i<=5){
  print(power**i)
  i=i+1
}
```

```
[1] 2
[1] 4
[1] 8
[1] 16
[1] 32
```

In [48]:

```
# if-else conditional statements
num1 <- 3
num2 <- 6
num3 <- 9
if(num1>num2 && num1>num3){
    max = num1
} else if(num2>num1 && num2>num3){
    max = num2
} else{
    max =num3
}

print(max)
```

```
[1] 9
```

In [51]:

```
# repeat-break
x <- 1
repeat {
    print(x)
    x = x+1
    if (x == 6){
        break
}
}
```

```
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```