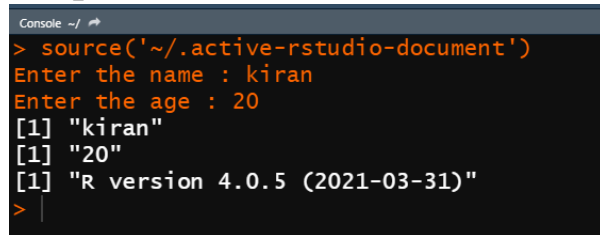


## Basics Programs

1. Write an R program to take input from the user (name and age) and display the values and also print the version of R installation Program:

```
name <- readline("Enter the name : ") age
<- readline("Enter the age : ")
print(name) print(age)
print(R.version.string)
```

Output:



```
Console ~/ / ➤
> source('~/.active-rstudio-document')
Enter the name : kiran
Enter the age : 20
[1] "kiran"
[1] "20"
[1] "R version 4.0.5 (2021-03-31)"
> |
```

2. Write an R program to get all prime numbers up to a given number Program:

```
n <- readline("Enter the number n : ")
n <- as.integer(n) for(j in 2:n){ f = 1
i = 2  n = j
  while(i<=n/2){
    if(n%%i == 0){
      f = 0
      break
    }
    i = i + 1
  }
  if(f == 1)
  {
    print(n)
  }
```

```
}
```

Output:

```
[1] 2
[1] 3
[1] 5
[1] 7
[1] 11
[1] 13
[1] 17
[1] 19
> |
```

3. Write an R program to get the first 10 Fibonacci numbers.

**Program:**

```
n1 = 0 n2 = 1
```

```
print(n1)
```

```
print(n2)
```

```
for(i in 2:10){
```

```
  n3 = n2 + n1
```

```
  print(n3)  n1
```

```
  = n2  n2 = n3
```

```
} Output:
```


```
[1] 0
[1] 1
[1] 1
[1] 2
[1] 3
[1] 5
[1] 8
[1] 13
[1] 21
[1] 34
[1] 55
> |
```

4. Write an R program to create a sequence of numbers from 20 to 50 and find the mean of numbers from 20 to 60 and sum of numbers from 51 to 91.

Program:

```
s <- seq(20:50) print(s)
print(mean(20:60))
print(sum(51:91))
```

Output:

A screenshot of an R console window. The title bar shows 'Console', 'Terminal', and 'Jobs'. The R version is 'R 4.1.1' and the path is '~/R Lab/'. The code executed is '> source("C:/Pentagon/R/Basic.r")'. The output shows three lines: '[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31', '[1] 40', and '[1] 2911'.

```
R 4.1.1 · ~/R Lab/
> source("C:/Pentagon/R/Basic.r")
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
[1] 40
[1] 2911
> |
```

5. Write an R program to convert other types of objects into complex type objects.

```
var1 <- 20
var2 <- as.complex(var1)
print(var2)
```

Output:

A screenshot of an R console window showing the output of the code for question 5. The output is '[1] 20+0i'.

```
[1] 20+0i
> |
```

6. Write an R program to print the numbers from 1 to 100 and print "AAA" for multiples of 3, print "BBBBB" for multiples of 5, and print "AAABBBBB" for multiples of both.

```
print(seq(1:100))
```

```

n <- readline("Enter the number n : ") n
= as.integer(n)
if(n%%15 == 0){
print("AAABBBB") }
else if(n%%3 == 0){
print("AAA") } else
if(n%%5 == 0){
  print("BBBBB")
}

```

Output:

```

[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
[27] 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
[53] 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78
[79] 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
Enter the number n : 45
[1] "AAABBBB"
> |

```

7. Write an R program to extract first 10 English letters in lowercase and last 10 letters in upper case and extract letters between 22nd to 24th letters in uppercase.

```

print(head(letters,10)) print(tail(LETTERS,10))
print(head(letters[22:24]))

```

Output:

```

[1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j"
[1] "Q" "R" "S" "T" "U" "V" "W" "X" "Y" "Z"
[1] "v" "w" "x"
> |

```

8. Write an R program to find a list of even numbers from 1 to n.

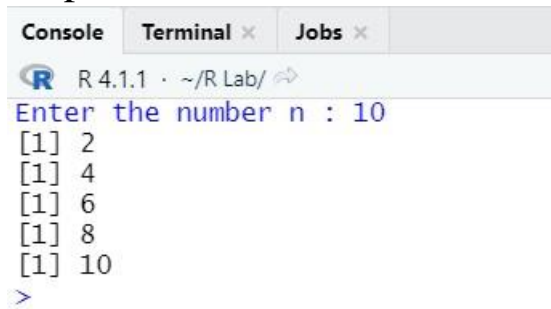
```

n <- readline("Enter the number n : ")
n <- as.integer(n) for(i in 1:n){
  if(i%%2 == 0){

```

```
    print(i)
  }
}
```

Output:

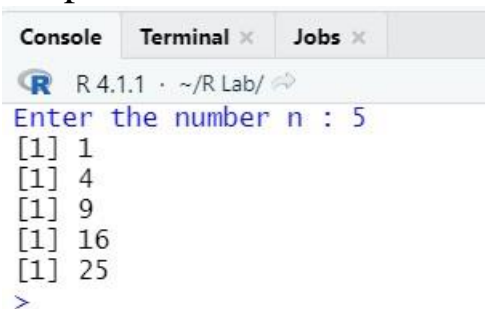


```
Console Terminal x Jobs x
R 4.1.1 · ~/R Lab/ ↗
Enter the number n : 10
[1] 2
[1] 4
[1] 6
[1] 8
[1] 10
>
```

9. Write an R program to create a function to print squares of a number in sequence.

```
square <- function(n) {
  for(i in 1:n) {
    print(i^2)
  } }
n <- readline("Enter the number n : ") n
<- as.integer(n)
square(n)
```

Output:



```
Console Terminal x Jobs x
R 4.1.1 · ~/R Lab/ ↗
Enter the number n : 5
[1] 1
[1] 4
[1] 9
[1] 16
[1] 25
>
```

## Data Types:

1. Write an R program to create a vector which contains 10 random integer values between -50 and +50.

```
n = sample(-50:50, 10, replace=TRUE)
```

print(n) Output:

```
[1] -41 -21 38 11 18 7 -34 48 43 46  
> |
```

2. Write an R program to create a  $5 \times 4$  matrix,  $3 \times 3$  matrix with labels and fill the matrix by rows and  $2 \times 2$  matrix with labels and fill the matrix by columns.

```
m1 = matrix(1:20, nrow=5, ncol=4)
```

```
print(m1) n =
```

```
c(11,2,10,8,7,9,1,15,14) rn =
```

```
c("R1", "R2", "R3") cn = c("C1",  
"C2", "C3")
```

```
m2 = matrix(n, nrow=3, ncol=3, byrow=TRUE, dimnames=list(rn, cn))
```

```
print(m2) n = c(20,12,23,45) rn = c("R1", "R2") cn = c("C1", "C2")
```

```
m3 = matrix(n, nrow=2, ncol=2, byrow=FALSE, dimnames=list(rn, cn))  
print(m3)
```

Output:

```

      [,1] [,2] [,3] [,4]
[1,]    1    6   11   16
[2,]    2    7   12   17
[3,]    3    8   13   18
[4,]    4    9   14   19
[5,]    5   10   15   20

      C1 C2 C3
R1 11  2 10
R2  8  7  9
R3  1 15 14

      C1 C2
R1 20 23
R2 12 45
> |

```

3. Write an R program to sort a Vector in ascending and descending order.

```

n = c(10, 20, 30, 25, 9, 26)
print(n) print(sort(n))
print(sort(n, decreasing=TRUE))

```

Output:

```

[1] 10 20 30 25  9 26
[1]  9 10 20 25 26 30
[1] 30 26 25 20 10  9
> |

```

4. Write an R program to find the maximum and the minimum value of a given vector

```

n = c(10, 20, 30, 25, 9, 26) print(n)
print(paste("Max",max(n)))
print(paste("Min : ",min(n)))

```

Output:

```
[1] 10 20 30 25 9 26
[1] "Max 30"
[1] "Min : 9"
> |
```

5. Write an R program to create a list of heterogeneous data, which include character, numeric and logical vectors. Print the lists.

```
li = list("Hello",23,23L,TRUE) print(li)
```

Output:

```
[[1]]
[1] "Hello"

[[2]]
[1] 23

[[3]]
[1] 23

[[4]]
[1] TRUE

> |
```

6. Write an R program to create a Data frame which contain details of 5 employees and display the details

```
Id = c(21,22,23,24,25)
Name = c("John","Jane","Lucy","Stuart","George")
Age = c(34,32,28,28,40)
Gender = c("M","F","F","M","M") Emp
= data.frame(Id,Name,Age,Gender)
print(Emp)
```

Output:



```

  Id   Name Age Gender
1 21   John  34      M
2 22   Jane  32      F
3 23   Lucy  28      F
4 24 Stuart  28      M
5 25 George  40      M
> |

```

7. Write an R program to find the maximum and the minimum value of a given vector.

```

n <- c(20,12,4,45,32,65,2,7,38) print(max(n))
print(min(n))

```

Output:

```

[1] 65
[1] 2
> |

```

8. Write an R program to combine three arrays so that the first row of the first array is followed by the first row of the second array and then the first row of the third array.

```

num1 = rbind(rep("A",3), rep("B",3), rep("C",3)) print(num1)
num2 = rbind(rep("P",3), rep("Q",3), rep("R",3)) print(num2)
num3 = rbind(rep("X",3), rep("Y",3), rep("Z",3)) print(num3)
a = matrix(t(cbind(num1,num2,num3)),ncol=3, byrow=TRUE)
print(a)

```

Output:

```

      [,1] [,2] [,3]
[1,] "A"  "A"  "A"
[2,] "B"  "B"  "B"
[3,] "C"  "C"  "C"
      [,1] [,2] [,3]
[1,] "P"  "P"  "P"
[2,] "Q"  "Q"  "Q"
[3,] "R"  "R"  "R"
      [,1] [,2] [,3]
[1,] "X"  "X"  "X"
[2,] "Y"  "Y"  "Y"
[3,] "Z"  "Z"  "Z"
      [,1] [,2] [,3]
[1,] "A"  "A"  "A"
[2,] "P"  "P"  "P"
[3,] "X"  "X"  "X"
[4,] "B"  "B"  "B"
[5,] "Q"  "Q"  "Q"
[6,] "Y"  "Y"  "Y"
[7,] "C"  "C"  "C"
[8,] "R"  "R"  "R"
[9,] "Z"  "Z"  "Z"
> |

```

9. Write an R program to create an array of two 3x3 matrices each with 3 rows and 3 columns from two given two vectors. Print the second row of the second matrix of the array and the element in the 3rd row and 3rd column of the 1st matrix.

```

v1 = c(21,23,24,25) v2
= c(0,1,2,3,4,5)
re = array(c(v1,v2),dim = c(3,3,2))
print(re) print(re[2,,2])
print(re[3,3,1])

```

Output:

```

, , 1
      [,1] [,2] [,3]
[1,]    21    25    2
[2,]    23     0    3
[3,]    24     1    4

, , 2
      [,1] [,2] [,3]
[1,]     5    24    1
[2,]    21    25    2
[3,]    23     0    3

[1] 21 25 2
[1] 4
> |

```

10. Write an R program to create a Data frame which contain details of 5 employees and display summary of the data

```

ID <- c(101,102,103,104,105)
Job <- c('Manager','Analyst','Clerk','Analyst','Intern')
Salary <- c(75000,50000,20000,51000,15000)
data <- data.frame(ID,Job,Salary)
print(data)

```

Output:

```

  ID   Job Salary
1 101 Manager 75000
2 102 Analyst 50000
3 103  Clerk 20000
4 104 Analyst 51000
5 105 Intern 15000
> |

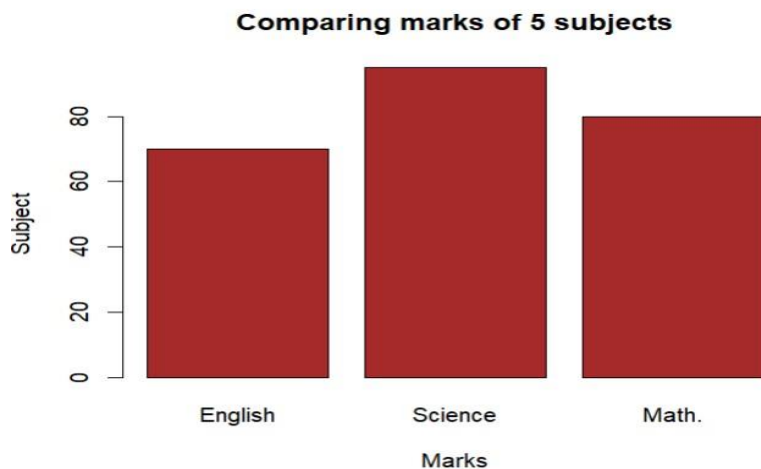
```

## Statistics

- Write an R program to create a simple bar plot of three subject's marks, change the border color to brown and make inside bar lines as 90 degrees.

```
marks = c(70, 95, 20)
barplot(marks,
        main = "Comparing marks of 5
subjects",      xlab = "Marks",      ylab =
"Subject",
        names.arg = c("Psychology", "DWDM", "Maths"),
        col = "brown",
        horiz = FALSE)
```

Output:



- Write a program to read a csv file and analyze the data in the file in R.

```
data <- read.csv("C:/Users/kiran/Desktop/data.csv")
print(data)
```

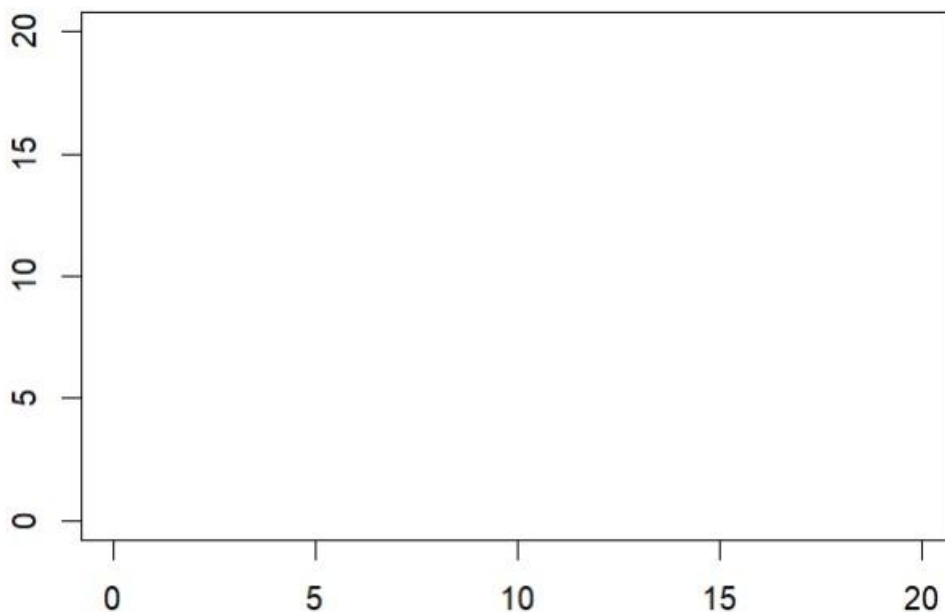
Output:

```
Console Terminal x Jobs x
R 4.1.1 · ~/R Lab/
> source("C:/Pentagon/R/Stat.r")
  Subject Marks Highest
1  Maths    75      87
2 Physics    78      88
3 Biology    76      90
4  Social    70      93
> |
```

- Write an R program to draw an empty pie chart and empty plots specify the axes limits of the graph

```
plot.new()
plot(1, type="n", xlab="", ylab="", xlim=c(0, 20), ylim=c(0, 20))
```

Output:

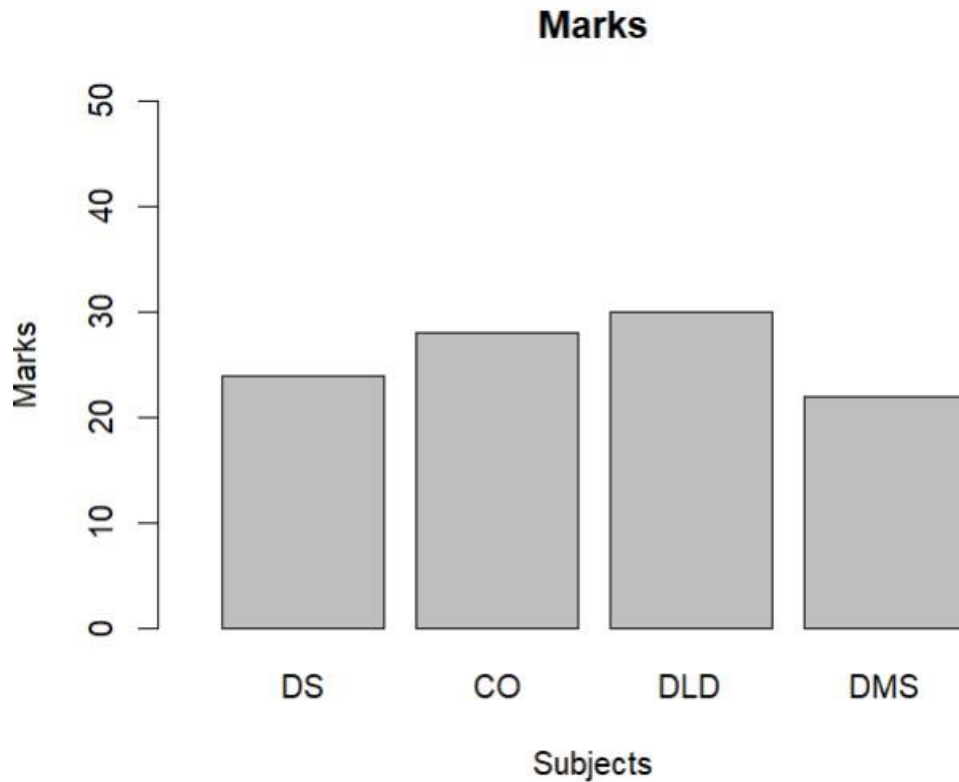


- Write an R program to create a simple bar plot of four subject's registered, assign the colors to each bar and assign the limit to x-axis as c(0,5) and y-axis as c(0,50).

```
m = c(24,28,30,22)
barplot(m,      main
="Marks",      xlab =
"Subjects",    ylab
="Marks",      xlim
=c(0,5),       ylim =
c(0,50),
```

```
names.arg = c('DS','CO','DLD','DMS'))
```

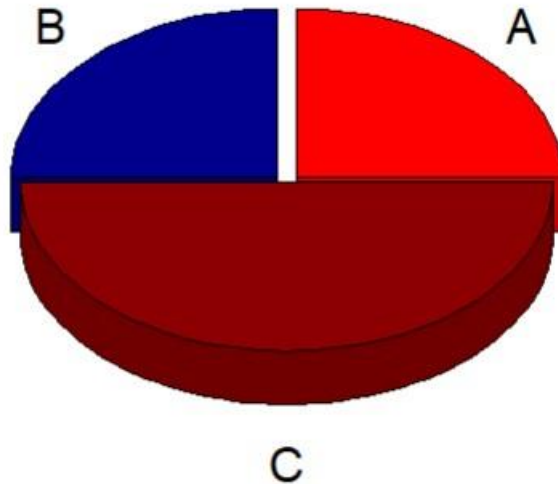
Output:



- Write an R program to create a simple 3D pie chart, assign color and labels to each part. `library(plotrix) pie3D(x=c(25,25,50), explode = 0.05, theta = 1, col = c('red','darkblue','darkred'), labels = c('A','B','C')) title(main = "Sample pie")`

Output:

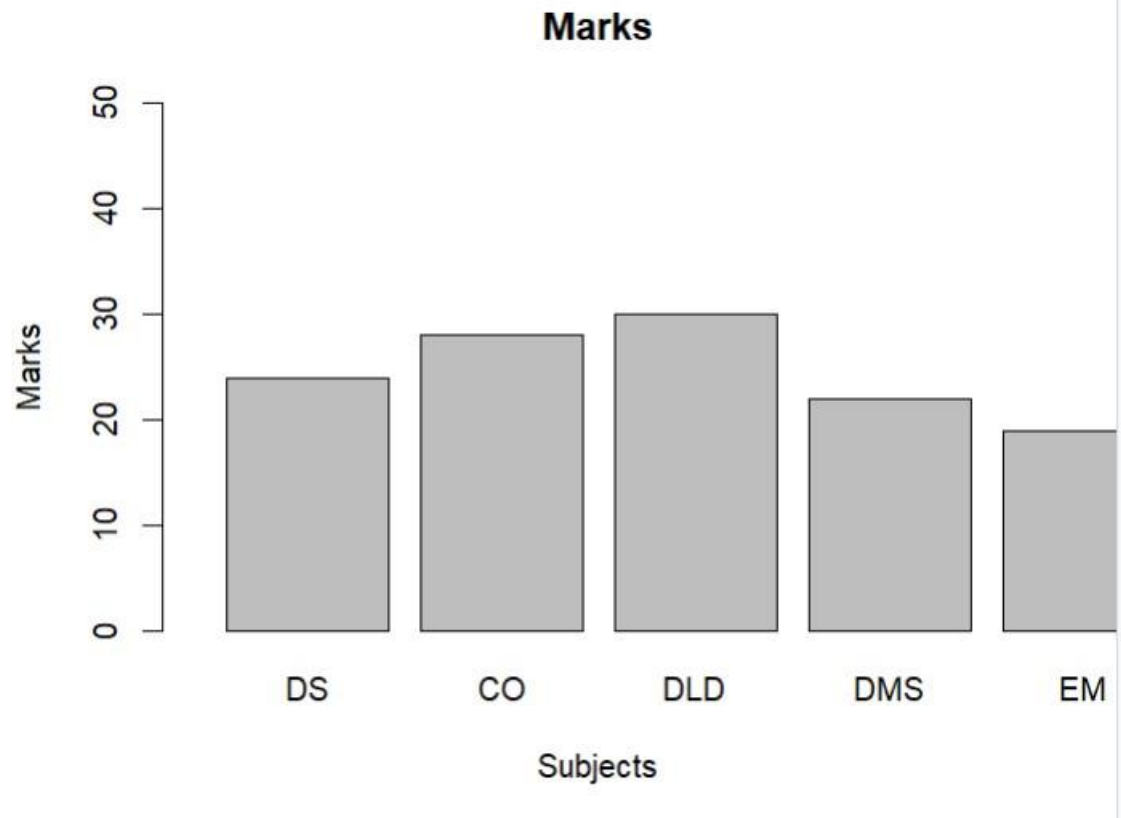
Sample pie



- Write an R program to create a simple bar plot of five subject's marks.

```
m = c(24,28,30,22,19) barplot(m,      main  
="Marks",      xlab = "Subjects",      ylab =  
"Marks",      xlim = c(0,5),      ylim = c(0,50),  
names.arg = c('DS','CO','DLD','DMS','EM'))
```

Output:

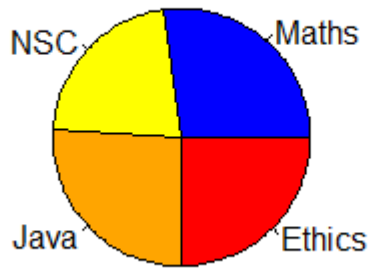


- Write an R program to create a simple pie chart of four subjects registered, assign the colors to each block and display in anticlockwise direction.

```
m <- c(25,20,24,23)
sub <- c('Maths','NSC','Java','Ethics')
pie(m,labels = sub,col = c('blue','yellow','orange','red'))
```

Output:



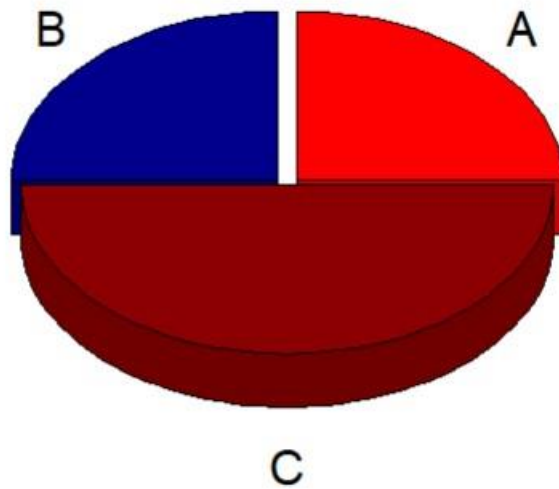


- Write an R program to create a simple 3D pie chart, assign title to the chart and also split each part.

```
library(plotrix)
pie3D(x=c(25,10,50),
explode = 0.05,    theta
= 1,
      col = c('red','darkblue','darkred'),
labels = c('A','B','C'))
title(main = "Sample pie")
```

Output:

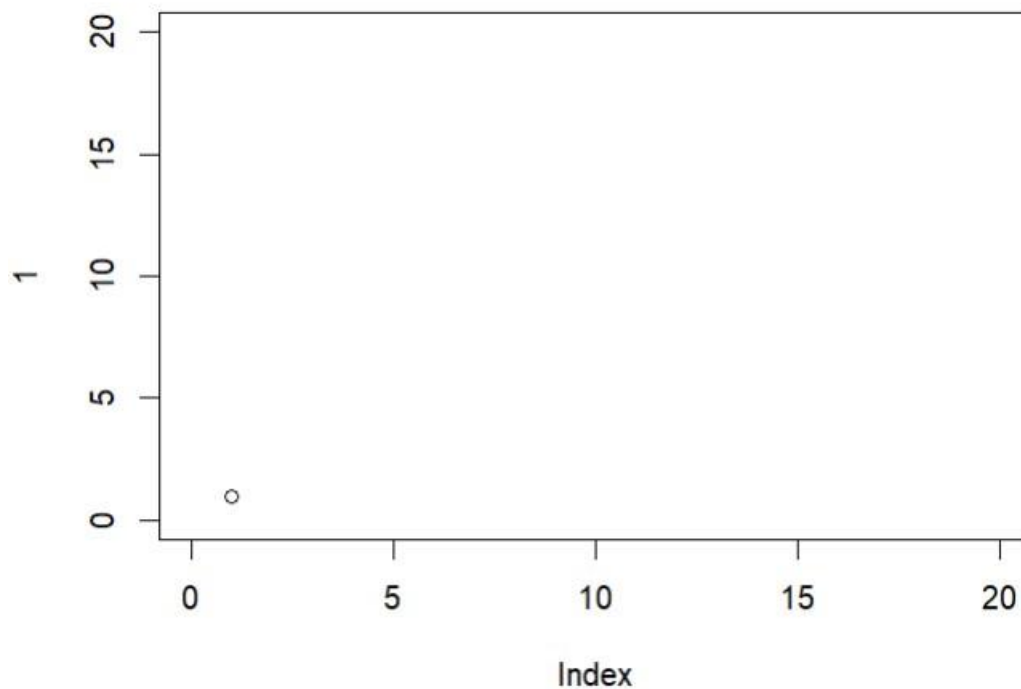
**Sample pie**



- Write an R program to draw an empty bar plot and empty plots specify the axes limits of the graph.

```
plot.new()  
plot(1,xlim=c(0,20),ylim=c(0,20))
```

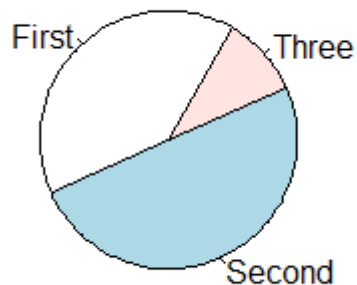
Output:



- Write an R program to create a simple pie chart of three subject's marks, change the border color to pink and make inside bar lines as 60 degrees.

```
pie(x = c(40,50,10),  
    init.angle = 60,  
    labels = c('First','Second','Three'))
```

Output:



**Strings**

- Write an R program to check if the given string is palindrome or not.

Program:

```
s <- readline() rev = "" for(i in
  nchar(s):1){
  rev = paste(rev,substr(s,i,i),sep="")
} if(s == rev){
print(TRUE)
}else{
print(FALSE)
}
```

Output:

```
Hello
[1] FALSE
abcba
[1] TRUE
> |
```

- Write an R program to convert the string into lowercase and uppercase letters.

Program:

```
s <- readline("Enter any string : ")
print(toupper(s)) print(tolower(s))
```

Output:

```
> source('~/.active-rstudio-document')
Enter any string : abcdef
[1] "ABCDEF"
[1] "abcdef"
> |
```

- Write an R program to extract characters from 5th to 7th position from a given string.

Program:

```
s <- readline("Enter any string ")
```

```
print(substr(s,5,7))
```

Output:

```
Enter any string RProgrammingLAB  
[1] "gra"  
> |
```

- Write an R program to combine three arrays so that the first row of the first array is followed by the first row of the second array and then first row of the third array.

**Program:**

```
num1 = rbind(rep("A",3), rep("B",3), rep("C",3)) print(num1)  
num2 = rbind(rep("P",3), rep("Q",3), rep("R",3)) print(num2)  
num3 = rbind(rep("X",3), rep("Y",3), rep("Z",3)) print(num3)  
a = matrix(t(cbind(num1,num2,num3)),ncol=3, byrow=TRUE)  
print(a)
```

Output:

```

      [,1] [,2] [,3]
[1,] "A"  "A"  "A"
[2,] "B"  "B"  "B"
[3,] "C"  "C"  "C"

      [,1] [,2] [,3]
[1,] "P"  "P"  "P"
[2,] "Q"  "Q"  "Q"
[3,] "R"  "R"  "R"

      [,1] [,2] [,3]
[1,] "X"  "X"  "X"
[2,] "Y"  "Y"  "Y"
[3,] "Z"  "Z"  "Z"

      [,1] [,2] [,3]
[1,] "A"  "A"  "A"
[2,] "P"  "P"  "P"
[3,] "X"  "X"  "X"
[4,] "B"  "B"  "B"
[5,] "Q"  "Q"  "Q"
[6,] "Y"  "Y"  "Y"
[7,] "C"  "C"  "C"
[8,] "R"  "R"  "R"
[9,] "Z"  "Z"  "Z"
> |

```

- Write an R program to find duplicate characters from a given string.

Program:

```

s <- "Hello there!" d = c() for(i in
  1:nchar(s)){ d =
  append(substr(s,i,i),d,i)
}

```

```

k = c() j =
1 for(i in
d){
  if(i %in% k){
print(i) }else{
k = append(i,k,j)
j = j+1
}
}

```

Output:

```
Console Terminal x Jobs x
R 4.1.1 · ~/R Lab/ ↗
> source("C:/Pentagon/R/string.r")
[1] "e"
[1] "]"
[1] "e"
> |
```

- Write an R program to extract substring from a given main string.

Program: s <- "R  
Programming Lab "  
print(substr(s,5,8))

Output:

```
[1] "ogra"
> |
```

- Write an R program to describe any five string handling functions.

Program:  
s1 = "Data"  
s2 = "Analytics"

```
print(length(s1))
print(nchar(s2))
print(toupper(s1))
print(tolower(s2))
print(paste(s1,s2))
```

Output:

```
[1] 1
[1] 9
[1] "DATA"
[1] "analytics"
[1] "Data Analytics"
> |
```

- Write an R program to describe any five string handling functions.

Program: `st <-`

`cat("Hello,","How","are","you?",sep=" ")`

`print(st)`

`print(casefold("GoOd mOrnInG"))`

`print(chartr("o","O","Good Morning"))`

`print(strsplit("See you soon!"," "))`

`s <- "Good Afternoon"`

`print(substr(s,11,14))`

Output:

```

Hello, How are you?NULL
[1] "good morning"
[1] "GOOd MORning"
[[1]]
[1] "See" "you" "soon!"

[1] "noon"
> |

```

- Write an R program to check whether the substring is present in main string or not Program:

`s <- "To check for a substring in a  
string"`

`t <- grep("substring",s)`

`if(t == 1){`

`print("present")`

`}else{ print("Not  
Present")`



```
}
```

Output:

```
[1] "present"  
> |
```