ITA0443 Statistics with R Programming

**1) Perform Matrix Addition &amp; Subtraction in R**

**r1=c(1,2,3,4)**

**r2=c(8,9,10,11)**

**mat1=matrix(r1,nrow=2,ncol=2,byrow = TRUE)**

**row.names(mat1)=c("A","B")**

**colnames(mat1)=c("C","D")**

**mat2=matrix(r2,nrow=2,ncol=2,byrow = TRUE)**

**row.names(mat2)=c("A","B")**

**colnames(mat2)=c("C","D")**

**print("First matrix :")**

**print(mat1)**

**print("Second matrix :")**

**print(mat2)**

**print("Matrix Addition :")**

**print(mat1+mat2)**

**print("Matrix subtraction :")**

**print(mat2-mat1)**

**output:-**

**> source('~/.active-rstudio-document')**

**[1] "First matrix :"**

**C D**

**A 1 2**

**B 3 4**

**[1] "Second matrix :"**

**C D**

**A 8 9**

**B 10 11**

**[1] "Matrix Addition :"**

**C D**

**A 9 11**

**B 13 15**

**[1] "Matrix subtraction :"**

**C D**

**A 7 7**

**B 7 7**

**2) Perform Scalar multiplication and matrix multiplication in R**

**r1=c(1,2,3,4)**

**r2=c(10,11,12,13)**

**mat1=matrix(r1,nrow=2,ncol=2,byrow = TRUE)**

**row.names(mat1)=c("A","B")**

**colnames(mat1)=c("C","D")**

**mat2=matrix(r2,nrow=2,ncol=2,byrow = TRUE)**

**row.names(mat2)=c("A","B")**

**colnames(mat2)=c("C","D")**

**print("Scalar multiplication :")**

**print(mat1\*2)**

**print("Matrix Multiplication :")**

**print(mat1\*mat2)**

**Output:**

**[1] "Scalar multiplication :"**

**C D**

**A 2 4**

**B 6 8**

**[1] "Matrix Multiplication :"**

**C D**

**A 10 22**

**B 36 52**

**>**

**3) Find Transpose of matrix in R.**

**r1=c(1,2,3,4)**

**r2=c(8,9,10,11)**

**mat1=matrix(r1,nrow=2,ncol=2,byrow = TRUE)**

**row.names(mat1)=c("A","B")**

**colnames(mat1)=c("C","D")**

**print("Matrix :")**

**print(mat1)**

**print("Transpose of Matrix :")**

**print(t(mat1))**

**Output: [1] "Matrix :"**

**C D**

**A 1 2**

**B 3 4**

**[1] "Transpose of Matrix :"**

**A B**

**C 1 3**

**D 2 4**

**>**

**4) Perform the operation of combining matrices in R using cbind() and rbind()**

**Function**

**r1=c(1,2,3,4)**

**r2=c(8,9,10,11)**

**mat1=matrix(r1,nrow=2,ncol=2,byrow = TRUE)**

**row.names(mat1)=c("A","B")**

**colnames(mat1)=c("C","D")**

**mat2=matrix(r2,nrow=2,ncol=2,byrow = TRUE)**

**row.names(mat2)=c("A","B")**

**colnames(mat2)=c("C","D")**

**print("Original matrices :")**

**print(mat1)**

**print(mat2)**

**print("Matrices after Cbind :")**

**print(cbind(mat1,mat2))**

**print("Matrices after Rbind :")**

**print(rbind(mat1,mat2))**

**output:-**

**[1] "Original matrices :"**

**C D**

**A 1 2**

**B 3 4**

**C D**

**A 8 9**

**B 10 11**

**[1] "Matrices after Cbind :"**

**C D C D**

**A 1 2 8 9**

**B 3 4 10 11**

**[1] "Matrices after Rbind :"**

**C D**

**A 1 2**

**B 3 4**

**A  8  9**

**B 10 11**

**5) Deconstruct a matrix in R**

**library(reshape)**

**r1=c(1,2,3,4)**

**r2=c(8,9,10,11)**

**mat1=matrix(r1,nrow=2)**

**r1=c(1,2,3,4)**

**r2=c(9,10,11,12)**

**print("Matrix :")**

**print(mat1)**

**print("Matrix after deconstruction :")**

**print(malt.matrix(mat1))**

**6) Perform array manipulation in R .**

**arr=array(data=c(1,2,3,4),dim=4,dimnames = row.names("a"))**

**print(arr)**

**print("Sum of Array :")**

**print(sum(arr))**

**print("adding an element into the array")**

**arr=array(data=c(arr[0:2],5,3:4))**

**print(arr)**

**output:-**

**[1] 1 2 3 4**

**[1] "Sum of Array :"**

**[1] 10**

**[1] "adding an element into the array"**

**[1] 1 2 5 3 4**

**>**

**7) Perform calculations across array elements in an array using the apply() function.**

**Output:-**

**print("Array :")**

**arr=array(data=c(1,2,3,4),dim=c(2,2),dimnames = row.names("a"))**

**print(arr)**

**print("Applying sum into the array :")**

**print(apply(arr,1,sum))**

**output:-**

**[1] "Array :"**

**[,1] [,2]**

**[1,] 1 3**

**[2,] 2 4**

**[1] "Applying sum into the array :"**

**[1] 4 6**

**8) Demonstrate Factor data structure in R.**

**fact=factor(c("apple", "banana","orange","apple","pineapple"))**

**print(fact)**

**output:-**

**[1] apple banana orange apple pineapple**

**Levels: apple banana orange pineapple**

**>**

**9) Create a data frame and print the structure of the data frame in R.**

**name=c("ab","cd","ef","gh")**

**age=c(20,30,40,50)**

**gender=c("M","F","M","F")**

**df=data.frame(name,age,gender)**

**print("Dataframe :")**

**print(df)**

**print("Stucture of Dataframe :")**

**print(str(df))**

**output:-**

**[1] "Dataframe :"**

**name age gender**

**1 ab 20 M**

**2 cd 30 F**

**3 ef 40 M**

**4 gh 50 F**

**[1] "Stucture of Dataframe :"**

**'data.frame': 4 obs. of 3 variables:**

**$ name : chr "ab" "cd" "ef" "gh"**

**$ age : num 20 30 40 50**

**$ gender: chr "M" "F" "M" "F"**

**NULL**

**10) Demonstrate the creation of S3 class in R.**

**x <- list(name ="Abc", reg\_no =19)**

**class(x)<-"student"**

**print(x)**

**output:-**

**$name**

**[1] "Abc"**

**$reg\_no**

**[1] 19**

**attr(,"class")**

**[1] "student"**

**11) Demonstrate the creation of S4 class in R.**

**setClass("Student",slots = list(name="character",reg\_no="numeric"))**

**student1=new("Student",name="abc",reg\_no=19)**

**print(student1)**

**output:-**

**An object of class "Student"**

**Slot "name":**

**[1] "abc"**

**Slot "reg\_no":**

**[1] 19**

**12) Demonstrate the creation of Reference class in R by defining a class called students**

**with fields – Name, Age , GPA. Also illustrate how the fields of the object can be**

**accessed using the $ operator. Modify the Name field by reassigning the name to Paul.**

**student=setRefClass("student",fields = list(name="character",age="numeric",GPA="numeric"))**

**student1=student(name="abc",age=19,GPA=10)**

**print("Class :")**

**print(student1)**

**print("Accessing with $ operator :")**

**print(student1$name)**

**print("Modifying name field :")**

**student1$name="venky"**

**print(student1)**

**output:-**

**[1] "Class :"**

**Reference class object of class "student"**

**Field "name":**

**[1] "abc"**

**Field "age":**

**[1] 19**

**Field "GPA":**

**[1] 10**

**[1] "Accessing with $ operator :"**

**[1] "abc"**

**[1] "Modifying name field :"**

**Reference class object of class "student"**

**Field "name":**

**[1] "venky"**

**Field "age":**

**[1] 19**

**Field "GPA":**

**[1] 10**

**Excercie-3**

**1.Write a program to check whether an integer (entered by the user) is a prime number or not using control statements.**

**n=as.integer(readline(prompt = "Enter a number :"))**

**start=2**

**flag=0**

**for(i in start:(n-1)){**

**if(n%%i==0){**

**flag=1**

**}**

**}**

**if(flag==0){**

**print("is a prime number")**

**}else if(flag==1){**

**print("is not a prime number")**

**}**

**Output:-**

**Enter a number :5**

**[1] "is a prime number"**

**> source('~/.active-rstudio-document')**

**Enter a number :4**

**[1] "is not a prime number"**

**2.Write a program to check whether a number entered by the user is positive number or a negative number or zero.**

**n=as.integer(readline(prompt = "Enter a number :"))**

**if(n==0){**

**print("Zero")**

**}else if(n>0){**

**print("Positive number")**

**}else if(n<0){**

**print("Negetive number")**

**}**

**Output:-**

**Enter a number :5**

**[1] "Positive number"**

**> source('~/.active-rstudio-document')**

**Enter a number :-6**

**[1] "Negetive number"**

**>**

**3.Write a program to check whether a number is an Armstrong number or not using a while loop.**

**n=as.integer(readline(prompt ="Enter a number :"))**

**s=0**

**c=n**

**while(n>0){**

**x=n%%10**

**s=s+x\*x\*x**

**n=n%/%10**

**}**

**if(s==c)**

**{**

**print("is Armstrong ")**

**}else{**

**print("is not Armstrong")**

**}**

**Output:-**

**> source('~/.active-rstudio-document')**

**Enter a number :25**

**[1] "is not Armstrong"**

**4.Write a program to demonstrate Repeat Loop in R**

**i <- 1**

**repeat {**

**print(i)**

**i <- i + 1**

**if(i >6) {**

**break**

**}**

**}**

**Output:-**

**[1] 1**

**[1] 2**

**[1] 3**

**[1] 4**

**[1] 5**

**[1] 6**

**>**

**5.Using functions develop a simple calculator in R.**

**summ<-function(x,y){**

**print(x+y)**

**}**

**sub<-function(x,y){**

**print(x-y)**

**}**

**mul<-function(x,y){**

**print(x\*y)**

**}**

**div<-function(x,y){**

**print(x/y)**

**}**

**power<-function(x,y){**

**print(x\*\*y)**

**}**

**choice<-function(){**

**cat("1=addition\n2=subraction\n3=multiplication\n4=division\n5=power")**

**ch=as.integer(readline(prompt = "Choose an operation :"))**

**return(ch)**

**}**

**x=choice()**

**if(x==1){**

**x=as.integer(readline(prompt = "enter left hand side operend :"))**

**y=as.integer(readline(prompt = "enter right hand side operend :"))**

**summ(x,y)**

**}else if(x==2){**

**x=as.integer(readline(prompt = "enter left hand side operend :"))**

**y=as.integer(readline(prompt = "enter right hand side operend :"))**

**sub(x,y)**

**}else if(x==3){**

**x=as.integer(readline(prompt = "enter left hand side operend :"))**

**y=as.integer(readline(prompt = "enter right hand side operend :"))**

**mul(x,y)**

**}else if(x==4){**

**x=as.integer(readline(prompt = "enter left hand side operend :"))**

**y=as.integer(readline(prompt = "enter right hand side operend :"))**

**div(x,y)**

**}else if(x==5){**

**x=as.integer(readline(prompt = "enter left hand side operend :"))**

**y=as.integer(readline(prompt = "enter right hand side operend :"))**

**power(x,y)**

**}else{**

**print("Invalid input... Try again")**

**choice()**

**}**

**Output:-**

**> source('~/.active-rstudio-document')**

**1=addition**

**2=subraction**

**3=multiplication**

**4=division**

**5=power**

**Choose an operation :4**

**enter left hand side operend :5**

**enter right hand side operend :1**

**[1] 5**

**>**

**6. Demonstrate the creation of a complex number in R.**

**a <- 1**

**b <- 1**

**x <- complex(real = a, imaginary = b)**

**print(x)**

**output:-**

**> source('~/.active-rstudio-document')**

**[1] 1+1i**

**>**

**7.Write a program to multiply two numbers using a function with a default value.**

**Assume default value as NULL.**

**mul<-function(x,y){**

**return(x\*y)**

**}**

**x=as.integer(readline(prompt = "enter a number :"))**

**y=as.integer(readline(prompt = "enter another number :"))**

**print(mul(x,y))**

**output:-**

**enter a number :5**

**enter another number :2**

**[1] 10**

**>**

**8.Find sum, mean and product of vector elements using built-in functions.**

**v=c(1:20)**

**print(paste("Sum of vector :",sum(v[1:20])))**

**print(paste("Mean :",mean(v)))**

**print(paste("Product :",prod(v)))**

**output:-**

**[1] "Sum of vector : 210"**

**[1] "Mean : 10.5"**

**[1] "Product : 2432902008176640000"**

**9.Sort a vector in R using sort() function. Also find the index of the sorted vector.**

**v=c(1,2,3,4,5,6)**

**v=sort(v)**

**print("Vector after sorting :")**

**print(v)**

**output:-**

**[1] "Vector after sorting :"**

**[1] 1 2 3 4 5 6**

**10.Find the L.C.M of two numbers entered by the user by creating a user-defined**

**function.**

**lcm <- function(x, y) {**

**v1=c()**

**v2=c()**

**for(i in 1:10){**

**v1=append(v1,i\*x)**

**}**

**for(i in 1:10){**

**v2=append(v2,i\*y)**

**}**

**for(i in v1){**

**for(j in v2){**

**if(i==j){**

**return(i)**

**}**

**}**

**}**

**}**

**num1 = as.integer(readline(prompt = "Enter first number: "))**

**num2 = as.integer(readline(prompt = "Enter second number: "))**

**print(paste("The LCM is", lcm(num1, num2)))**

**output:-**

**Enter first number: 5**

**Enter second number: 4**

**[1] "The LCM is 20"**

**>**