ITA04 STATISTICS WITH R PROGRAMMING 192011223

Assignment-I

1. The built-in vector LETTERS contains the uppercase letters of the alphabet. Produce a vector of
   1. the first 12 letters;

A) LETTERS[1:12]

* 1. the odd ‘numbered’ letters;

for(i in 1:26){

if(i %% 2 != 0){

print(LETTERS[i])

}

}

(iii) the (English) consonants.

for( i in LETTERS){

if(i=='A' || i=='E' || i=='I' || i=='O' || i=='U'){

}

else{

print(i)

}

}

1. The function rnorm() generates normal random variables. For instance, rnorm(10) gives a vectorof 10 i.i.d. standard normals. Generate 20 standard normals, and store them as x. Then obtain subvectors of
   1. the entries in x which are less than 1;
   2. the entries between – 0.5 and 1;
   3. the entries whose absolute value is larger than 1.5.

data<- rnorm(n = 10)

print(data)

summary(data)

1. Solve the following system of simultaneous equations using matrix methods.

a + 2b + 3c + 4d + 5e = −5 2a

+3b + 4c + 5d + e = 2

3a + 4b + 5c + d + 2e = 5

4a + 5b + c + 2d + 3e = 10

5a + b + 2c + 3d + 4e = 11

lm <- matrix(c(1,2,3,4,5,2,3,4,5,1,3,4,5,1,2,4,5,1,2,3,5,1,2,3,4),nrow=5)

rm <- matrix(c(-5,2,5,10,11),nrow =5)

solve(lm,rm)

1. Create a factor object for an apple color such as 'green', 'green', 'yellow', 'red', 'red', 'red',' green'. Print the factor and applying the nlevels function to know the number of distinct values

x <- c('green', 'green', 'yellow', 'red', 'red', 'red','green')

fac <- factor(x)

print(fac)

print(nlevels(fac))

1. Create an S3 object of class fruit contains a list with following required components such as name, quantity, cost and also Define and create s4 objects.Define a reference class of fruit

setClass("ABOUT\_FRUITS", fruits <- list(name="character", Quantity="numeric",cost="numeric"))

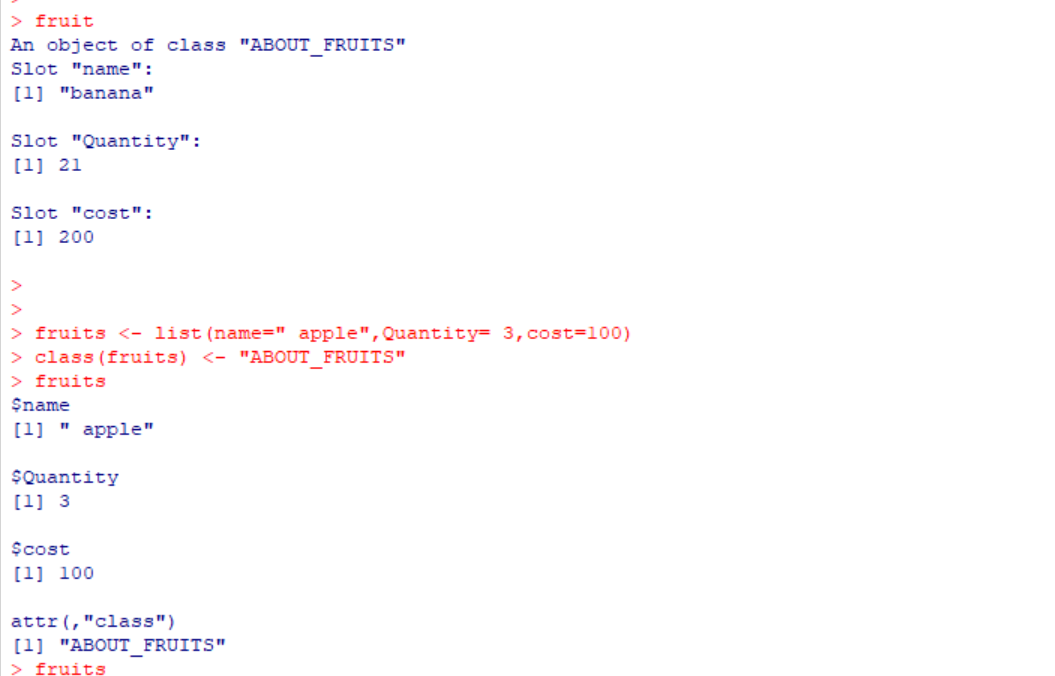
fruit <- new("ABOUT\_FRUITS", name = "banana", Quantity = 21, cost = 200)

fruit

fruits <- list(name=" apple",Quantity= 3,cost=100)

class(fruits) <- "ABOUT\_FRUITS"

fruits



ASSESSMENT-2 192011223

1. Write a R program to take input from the user (name and age) and display the

values. Also print the version of R installation.

name = readline(prompt="Input your name: ")

age = readline(prompt="Input your age: ")

print(paste("My name is",name, "and I am",age ,"years old."))

print(R.version.string)

Output:

Input your name: mahesh

Input your age:19

[1] "My name is mahesh and I am 19 years old"

[1] "R version 4.2.2 (2022-10-31 ucrt)"

1. Write a R program to get the details of the objects in memory.

name = "java";

n1 = 10;

n2 = 0.5

nums = c(10, 20, 30, 40, 50, 60)

print(ls())

print("Details of the objects in memory:")

print(ls.str())

Output:

1. “n1” “n2” “name” “nums”
2. “Details of the objects in memory:”

n1:num 10

n2:num 0.5

name: chr “java”

nums: num[1:6] 10 20 30 40 50 60

3. Write a 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.

print(seq(20,50))

print("Mean of numbers from 20 to 60:")

print(mean(20:60))

print("Sum of numbers from 51 to 91:")

print(sum(51:91))

Output:

[1] 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44

[26] 45 46 47 48 49 50

[1] "Mean of numbers from 20 to 60:"

[1] 40

[1] "Sum of numbers from 51 to 91:"

[1] 2911

4. Write a R program to create a vector which contains 10 random integer values

between -50 and +50.

v = sample(-50:50, 10, replace=TRUE)

print("Content of the vector:")

print("10 random integer values between -50 and +50:")

print(v)

Output:

[1] "Content of the vector:"

[1] "10 random integer values between -50 and +50:"

[1] 36 32 17 45 -12 28 27 -32 -13 -29

5. Write a R program to get all prime numbers up to a given number (based on

the sieve of Eratosthenes).

prime\_numbers <- function(n) {

if (n >= 2) {

x = seq(2, n)

prime\_nums = c()

for (i in seq(2, n)) {

if (any(x == i)) {

prime\_nums = c(prime\_nums, i)

x = c(x[(x %% i) != 0], i)

}

}

return(prime\_nums)

}

else

{

stop("Input number should be at least 2.")

}

}

prime\_numbers(12)

Output:

1. 2 3 5 7 11

6. Write a R program to extract first 10 english letter in lower case and last 10

letters in upper case and extract letters between 22 nd  to 24 th  letters in upper case.

t = head(letters, 10)

print(t)

t = tail(LETTERS, 10)

print(t)

e = tail(LETTERS[22:24])

print(e)

Output:

[1] "Last 10 letters in upper case:"

[1] "Q" "R" "S" "T" "U" "V" "W" "X" "Y" "Z"

[1] "V" "W" "X"

8. Write a R program to get the unique elements of a given string and unique

numbers of vector.

str1 = "The quick brown fox jumps over the lazy dog."

print("Original vector(string)")

print(str1)

print("Unique elements of the said vector:")

print(unique(tolower(str1)))

nums = c(1, 2, 2, 3, 4, 4, 5, 6)

print("Original vector(number)")

print(nums)

print("Unique elements of the said vector:")

print(unique(nums))

Output:

[1] "Original vector(string)"

[1] "The quick brown fox jumps over the lazy dog."

[1] "Unique elements of the said vector:"

[1] "the quick brown fox jumps over the lazy dog."

[1] "Original vector(number)"

[1] 1 2 2 3 4 4 5 6

[1] "Unique elements of the said vector:"

[1] 1 2 3 4 5 6

9. Write a R program to create three vectors a,b,c with 3 integers. Combine the

three vectors to become a 3×3 matrix where each column represents a vector.

Print the content of the matrix.

a<-c(1,2,3)

b<-c(4,5,6)

c<-c(7,8,9)

m<-cbind(a,b,c)

print("Content of the said matrix:")

print(m)

Output:

[1] "Content of the said matrix:"

a b c

[1,] 1 4 7

[2,] 2 5 8

[3,] 3 6 9

10. Write a R program to create a list of random numbers in normal distribution

and count occurrences of each value.

n = floor(rnorm(1000, 50, 100))

print('List of random numbers in normal distribution:')

print(n)

t = table(n)

print("Count occurrences of each value:")

print(t)

11. Write a R program to create three vectors numeric data, character data and

logical data. Display the content of the vectors and their type.

a = c(1, 2, 5, 3, 4, 0, -1, -3)

b = c("Red", "Green", "White")

c = c(TRUE, TRUE, TRUE, FALSE, TRUE, FALSE)

print(a)

print(typeof(a))

print(b)

print(typeof(b))

Print(c)

print(typeof(c))

Output:

[1] 1 2 5 3 4 0 -1 -3

[1] "double"

[1] "character"

[1] TRUE TRUE TRUE FALSE TRUE FALSE

[1] "logical"

12. Write a R program to create a 5 x 4 matrix , 3 x 3 matrix with labels and fill

the matrix by rows and 2 × 2 matrix with labels and fill the matrix by columns.

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

print("5 × 4 matrix:")

print(m1)

cells = c(1,3,5,7,8,9,11,12,14)

rnames = c("Row1", "Row2", "Row3")

cnames = c("Col1", "Col2", "Col3")

m2 = matrix(cells, nrow=3, ncol=3, byrow=TRUE, dimnames=list(rnames, cnames)) print("3 × 3 matrix with labels, filled by rows: ")

print(m2)

print("3 × 3 matrix with labels, filled by columns: ")

m3 = matrix(cells, nrow=3, ncol=3, byrow=FALSE, dimnames=list(rnames, cnames)) print(m3)

Output:

[1] "5 × 4 matrix:"

[,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

[1] "3 × 3 matrix with labels, filled by rows: "

Col1 Col2 Col3

Row1 1 3 5

Row2 7 8 9

Row3 11 12 14

[1] "3 × 3 matrix with labels, filled by columns: "

Col1 Col2 Col3

Row1 1 7 11

Row2 3 8 12

Row3 5 9 14

13. Write a R program to create an array, passing in a vector of values and a

vector of dimensions. Also provide names for each dimension.

a = array( 6:30, dim = c(4, 3, 2), dimnames = list( c("Col1", "Col2", "Col3", "Col4"), c("Row1", "Row2", "Row3"), c("Part1", "Part2")

)

)

print(a)

Output:

, , Part1

Row1 Row2 Row3

Col1 6 10 14

Col2 7 11 15

Col3 8 12 16

Col4 9 13 17

, , Part2

Row1 Row2 Row3

Col1 18 22 26

Col2 19 23 27

Col3 20 24 28

Col4 21 25 29

14. Write a R program to create an array with three columns, three rows, and two

&quot;tables&quot;, taking two  vectors as input to the array.  Print the array

v1 = c(1, 3, 5, 7)

v2 = c(2, 4, 6, 8, 10)

arr = array(c(v1, v2),dim = c(3,3,2))

print(arr)

Output:

, , 1

[,1] [,2] [,3]

[1,] 1 7 6

[2,] 3 2 8

[3,] 5 4 10

, , 2

[,1] [,2] [,3]

[1,] 1 7 6

[2,] 3 2 8

[3,] 5 4 10

15. Write a R program to create a list of elements using vectors, matrices and a

functions. Print the content of the list.

l = list( c(1, 2, 2, 5, 7, 12), month.abb, matrix(c(3, -8, 1, -3), nrow = 2), asin )

print("Content of the list:")

print(l)

Output:

[1] "Content of the list:"

[[1]]

[1] 1 2 2 5 7 12

[[2]]

[1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct" "Nov" "Dec"

[[3]]

[,1] [,2]

[1,] 3 1

[2,] -8 -3

[[4]]

function (x) .Primitive("asin")