ITA04 STATISTICS WITH R PROGRAMMING

Rithik vasan.k

191921033

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

(i) the first 12 letters;

(ii) the odd ‘numbered’ letters;

(iii) the (English) consonants.

**SOLUTION :**

1. the first 12 letters

first\_12 <- LETTERS[1:12]

**OUTPUT**: [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L”

1. the odd ‘numbered’ letters;

odd\_letters <- LETTERS[c(TRUE, FALSE)]

**OUTPUT**: [1] "A" "C" "E" "G" "I" "K" "M" "O" "Q" "S" "U" "W" "Y"

1. the (English) consonants.

consonants <- setdiff(LETTERS, c("A", "E", "I", "O", "U"))

**OUTPUT**:[1] "B" "C" "D" "F" "G" "H" "J" "K" "L" "M" "N" "P" "Q" "R" "S" "T" "V" "W" "X" "Y" "Z"

2. The function rnorm() generates normal random variables. For instance, rnorm(10) gives a vector

of 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;

**SOLUTION:**

set.seed(123)

x <- rnorm(20)

less\_than\_1 <- x[x < 1]

print(less\_than\_1).

**OUTPUT:**

[1] -0.56047565 -0.23017749 0.07050839 0.12928774 0.46091621 -1.26506123 -0.686852[8] -0.44566197 0.35981383 0.40077145 0.11068272 -0.55584113 0.49785048 -1.966617[15] 0.70135590 -0.47279141

1. the entries between – 0.5 and 1;

SOLUTION:

set.seed(123)

x <- rnorm(20)

between\_neg05\_pos1 <- x[x > -0.5 & x < 1]

print(between\_neg05\_pos1)

OUTPUT:

[1] -0.23017749 0.07050839 0.12928774 0.46091621 -0.445661970.35981383

[7] 0.40077145 0.11068272 0.49785048 0.70135590 -0.47279141

1. the entries whose absolute value is larger than 1.5.

**SOLUTION:**

set.seed(123)

x <- rnorm(20)

abs\_larger\_than\_15 <- x[abs(x) > 1.5]

print(abs\_larger\_than\_15)

**OUTPUT:**

[1] 1.558708 1.715065 1.786913 -1.966617

3. 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

**SOLUTION**

A <- 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, byrow = TRUE)

b <- c(-5, 2, 5, 10, 11)

x <- solve(A, b)

x

**OUTPUT:**

[1] 3.5066667 0.1066667 -0.6933333 -0.2933333 -1.0933333

4. Create a factor object for an apple color such as &#39;green&#39;, &#39;green&#39;, &#39;yellow&#39;, &#39;red&#39;, &#39;red&#39;, &#39;red&#39;,&#39;

green&#39;. Print the factor and applying the nlevels function to know the number of distinct

values

**SOLUTION**:

apple\_colors <- factor(c('green', 'green', 'yellow', 'red', 'red', 'red', 'green','light green','light red','light red'))

print(apple\_colors)

nlevels(apple\_color)

**OUTPUT:**

[1] green green yellow red red red

[7] green light green light red light red

Levels: green light green light red red yellow

[1] 5

5. 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

**SOLUTION:**

fruit <- function(name, quantity, cost) {

lst <- list(name = name, quantity = quantity, cost = cost)

class(lst) <- "fruit"

return(lst)

}

apple <- fruit(name = "Apple", quantity = 25, cost = 20.50)

print(apple$name)

print(apple$quantity)

print(apple$cost)

**OUTPUT:**

[1] "Apple"

[1] 25

[1] 20.5