# Exam BST02

23 March 2018

1. a

Examine the following code in R.

g <- function(a, b){  
 res1 <- min(a)  
 res2 <- max(b)  
 res2  
}

What does the above function g() do?

1. The function g() takes the input vector a and return the minimum and maximum.
2. The function g() takes the input vectors a and b and return the minimum of a and the maximum of b.
3. The function g() takes the input vectors a and b and return the maximum of b.
4. The function g() takes the input vectors a and b and return the minimum of a and the minimum of b.
5. The function g() takes the input vectors a and b and return the maximum of a and the maximum of b.

Examine the following code in R:

for (i in (seq(2,10,2) + 1)){  
 print(i)  
}

What does the above code do?

1. The above code prints the values: 1, 3, 5, 7 and 9.
2. The above code prints the values: 3, 5, 7, 9 and 11
3. The above code prints the values: 3, 4, 5, 6, 7, 8, 9 and 11.
4. The above code prints the values: 2, 4, 6, 8 and 11.
5. The above code prints the values: 3, 4, 5, 6, 7, 8, 9, 10, 11, 2, 4, 5, 6, 7, 8, 9, 10 and 11.

Examine the following code in R:

WEIGHT <= 90

What does the above code compute?

1. A vector with value 0 if WEIGHT is lower than or equal to 90 and 1 if WEIGHT larger than 90.
2. A vector with value 0 if WEIGHT larger than 90 and 1 if WEIGHT is lower than or equal to 90.
3. A vector with TRUE if WEIGHT larger than 90 and FALSE if WEIGHT is lower than or equal to 90.
4. A vector with value 0 if WEIGHT = 90 and 1 otherwise.
5. A vector with value TRUE if WEIGHT is lower than or equal to 90 and FALSE if WEIGHT larger than 90.

Examine the following code in R.

dat[dat$sex == "male" | (dat$weight > 68 & dat$height > 1.80), ]

What does the above code return?

1. The dataset dat with only males that have weight higher than 68 or height higher than 1.80.
2. The dataset dat with only males that have weight lower than 68 or height higher than 1.80.
3. The dataset dat with only males that have weight higher than 68 or height lower than 1.80.
4. The dataset dat with males and with patients that have weight higher than 68 or height higher than 1.80.
5. The dataset dat with males and with patients that have weight higher than 68 and height higher than 1.8.

Which of the following codes will investigate the association of the explanatory variable BMI (as a continuous variable) with education level (as response variable, higher versus lower education) accounting for gender and age from a dataset called dat?

1. lm(education ~ ns(BMI, df = 3) + age + gender, data = dat)
2. glm(education ~ BMI + age + gender, data = dat, family = binomial)
3. glm(BMI ~ education, data = dat, family = binomial)
4. glm(BMI ~ education + age + gender + height, data = dat)
5. lm(BMI ~ education + age + gender, data = dat)

Examine the following code in R.

h <- function(k){  
 if (k <= 20){  
 3 \* k  
 } else {

2 \* k

}  
}

What, if anything, will the function call h(x) return if x is a vector?

1. A vector where each element is triple the value of corresponding element of the input vector x.
2. A vector where each element is double the value of corresponding element of the input vector x.
3. A vector where each element is triple the value of corresponding element of the input vector x if the element is greater or equal to 20 and double the corresponding element otherwise.
4. A vector where each element is triple the value of corresponding element of the input vector x if the element is less than or equal to 20 and double the corresponding element otherwise.k
5. None of the above answers is always correct.

7

Consider the following code:

alphabeta <- factor(rep(letters[1:6],20))

bp <- rnorm(120, 120, 10)

df <- data.frame(alphabeta, BP=bp)

Which of the following statements is CORRECT?

1. The code will not run because the first argument of data.frame does not have a name.
2. The code will run.
3. The code will not run because alphabet and bp have different types.
4. The code will not run because alphabet is not numeric.
5. The code will not run because alphabet and bp have different numbers of rows.

8

We want to estimate a linear model in which we model y with the main effects and the interaction of treatment arm (treat) and time. All variables are included in the data.frame df

Which of the following codes does this correctly?

1. lm(y ~ time - arm, data=df)
2. lm(y ~ time == arm data=df)
3. lm(y ~ time + arm data=df)
4. lm(y ~ time \* arm data=df)
5. lm(y ~ time : arm, data=df)

9

Examine the following R code:

the\_list <- list( str=c('R', 'S'), vec=1:3, bool=TRUE )

answer <- length(the\_list[2])

What will be the value of answer?

A 0

B 1

C 3

D NA

E None of the above answers is correct.

10

Examine the following R code.

M <- matrix(seq(1,6.5,by=0.5), nrow = 3)[1,]  What is the output of class(M)?

A "vector"

B "matrix"

C "list"

D "numeric"

E None of the above answers is correct.

11

Wich of the following commands can be used to add a title to an existing plot?

A title('the title')

B title\_add('the title')

C main('the title')

D main\_title('the title')

E text('the title')

12

The documentation for the function append tells you:

?append

Vector Merging

**Description**

Add elements to a vector.

**Usage**

append(x, values, after = length(x))

**Arguments**

|  |  |
| --- | --- |
| x | the vector the values are to be appended to. |
| values | to be included in the modified vector. |
| after | a subscript, after which the values are to be appended. |

**Value**

A vector containing the values in x with the elements of values appended after the specified element of x.

**Put the following lines of code of the function append in the correct order:**

A: append<- function (x, values, after = length(x))

B: c(values, x)

C: if (!after)

D: {lengx <- length(x)

E: else c(x[1L:after], values, x[(after + 1L):lengx])

F: else if (after >= lengx)

G: c(x, values)

H: }

13

The function diff(x, lag = 1, differences = 1, ...) computes lagged and iterated differences. According to the documentation:

If x is a vector of length n and differences = 1, then the computed result is equal to the successive differences x[(1+lag):n] - x[1:(n-lag)].

If difference is larger than one this algorithm is applied recursively to x. Note that the returned value is a vector which is shorter than x.

If x is a matrix then the difference operations are carried out on each column separately.

Put the following code fragments of the function in the correct order.

A

function (x, lag = 1L, differences = 1L, ...) {

ismat <- is.matrix(x)

B:

if (ismat)

for (i in seq\_len(differences)) r <- r[i1, , drop = FALSE] -

r[-nrow(r):-(nrow(r) - lag + 1L), , drop = FALSE]

else for (i in seq\_len(differences)) r <- r[i1] - r[-length(r):-(length(r) -

lag + 1L)]

C

xlen <- if (ismat)

dim(x)[1L]

else length(x)

if (length(lag) != 1L || length(differences) > 1L || lag <

1L || differences < 1L)

D:

stop("'lag' and 'differences' must be integers >= 1")

if (lag \* differences >= xlen)

return(x[0L])

r <- unclass(x)

i1 <- -seq\_len(lag)

E:

class(r) <- oldClass(x)

r

}

14 Are the following statements **true or false**?

1. logical is one of the data types in R
2. matrix and lists are both examples of classes in R
3. rep(2, 4) has length 4
4. seq(2, 4)has length 4
5. The function debug can be used to check the speed of your code
6. The function call lapply(list(1,2,3),function(x){x\*x})returns a vector of length 3
7. The command plot(x=x, y=y, xlim='X') will generate a plot where the x-axis will be labelled X
8. The command library can be used to load an R package.
9. The function cbind can be used to add columns to a data.frame
10. If we use the function plot(x, y) we create a scatter plot with x on the x-axis and y on the y-axis