DSCI 1411: Fundamentals of Data Science

Second Exam, Fall 2022 (Answers)

Table of contents

1 Q1 - Choose the correct answer

1.1 What would the following code print?

```r
x<-c("T", "0", 1)
as.logical(x)
```

```
[1] TRUE   NA   NA
```

Answer: (d) None of the above

1.2 What would be the output of the following code?

```r
x<-1:3
names(x)
```

```
NULL
```

Answer: (b) `NULL`

Here is a named version of `x`:

```r
x = c(a = 1, b = 2, c = 3)
x
```

```
a b c
1 2 3
```

```r
names(x)
```

```
[1] "a" "b" "c"
```

1.3 Which of the following statement changes column name of a matrix `m` to `h` and `f` ?

```r
m = matrix (1:6, ncol = 2)
m
```

```
     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6
```

```r
colnames(m)<-c("h","f")
m
```

```
     h f
[1,] 1 4
[2,] 2 5
[3,] 3 6
```

Answer: (a) `colnames(m)<-c("h","f")`

Similarly, rows can be named using the `rownames()` function:

```r
rownames(m) = c("x", "y", "z")
m
```

```
  h f
x 1 4
y 2 5
z 3 6
```

1.4 The object `x` is a scalar defined in the work space. Which of the following code snippet will create a vector with NAs in it?

```r
x<-c(1,2,NA,10,3)
x
```

```
[1]  1  2 NA 10  3
```

Answer: (c) `x<-c(1,2,NA,10,3)`

1.5 What is the class of a in the following R code? `a<-c(1,"a",FALSE)`

```
a<-c(1,"a",FALSE)
a
```

```
[1] "1"     "a"     "FALSE"
```

```
class(a)
```

```
[1] "character"
```

Answer: (b) `character`

1.6 What is the class of b in the following R code? `b<-c(TRUE,TRUE,FALSE)`

```
b<-c(TRUE,TRUE,FALSE)
b
```

```
[1]  TRUE  TRUE FALSE
```

```
class(b)
```

```
[1] "logical"
```

Answer: (d) `logical`

1.7 If an object q is not defined in an R workspace, then the output of the command `class(q)` is:

```
q
```

```
function (save = "default", status = 0, runLast = TRUE)
.Internal(quit(save, status, runLast))
<bytecode: 0x115eaf0d0>
<environment: namespace:base>
```

Answer: (b) `function`

1.8 For the following array `y<-c(1,2,3,4,5)`, the R command to find the median is

```r
y<-c(1,2,3,4,5)
y
```

```
[1] 1 2 3 4 5
```

```r
median(y)
```

```
[1] 3
```

Answer: (a) `median(y)`

What is the output of the code below?

```r
x=c("apples","pears","kiwis")
y=c(10,6,12)
z1=paste(x,y,sep=":")
z2=paste0(x,y)
z1;z2
```

```
[1] "apples:10" "pears:6"   "kiwis:12"

[1] "apples10" "pears6"   "kiwis12"
```

```r
x=c("apples","pears","kiwis")
x
```

```
[1] "apples" "pears"  "kiwis"
```

```
y=c(10,6,12)
y
```

```
[1] 10  6 12
```

```
z1=paste(x,y,sep=":")
z1
```

```
[1] "apples:10" "pears:6"   "kiwis:12"
```

```
z2=paste0(x,y)
z2
```

```
[1] "apples10" "pears6"   "kiwis12"
```

```
z1;z2 # the semicolon is just to exectue more than function in the same line
```

```
[1] "apples:10" "pears:6"   "kiwis:12"

[1] "apples10" "pears6"   "kiwis12"
```

Answer: (c)

2 Q2 - Write an R code to implement the following

2.1 Create the following matrix:

$$\begin{pmatrix} 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \end{pmatrix}$$

```
matrix (1:8, nrow = 2)
```

```
     [,1] [,2] [,3] [,4]
[1,]    1    3    5    7
[2,]    2    4    6    8
```

2.2 Compute the sum of last 5 numbers in an object x containing 51 numeric values.

```r
set.seed (123)
size = 51
x = sample (1:100, size)
x
```

```
 [1] 31 79 51 14 67 42 50 43 97 25 90 69 57  9 72 26  7 95 87 36 78 93 76 15 32
[26] 84 82 41 23 27 60 53 75 89 71 38 91 34 29  5  8 12 13 18 33 66 64 65 21 77
[51] 73
```

```r
length(x)
```

```
[1] 51
```

```r
n = 5
```

```r
x[(length(x) - n + 1) : length(x)] # last 5 numbers
```

```
[1] 64 65 21 77 73
```

```r
sum(x[(length(x) - n + 1) : length(x)]) # sum of last 5 numbers
```

```
[1] 300
```

2.3 Compute the sum of smallest 4 numbers in an object x containing 21 numeric values.

```r
set.seed(123)
size = 21
x = sample (1:100, size)
x
```

```
 [1] 31 79 51 14 67 42 50 43 97 25 90 69 57  9 72 26  7 95 87 36 78
```

```r
length(x)
```

```
[1] 21
```

```r
sorted_x = sort(x)
sorted_x
```

```
 [1]  7  9 14 25 26 31 36 42 43 50 51 57 67 69 72 78 79 87 90 95 97
```

```r
smallest_four_numbers = sorted_x[1:4]
smallest_four_numbers
```

```
[1]  7  9 14 25
```

```r
sum(smallest_four_numbers)
```

```
[1] 55
```

Or directly:

```r
sum(sort(x)[1:4])
```

```
[1] 55
```

2.4 Find the maximum value of the second column in a given data frame `df`.

```r
set.seed(123)
df = data.frame (X = sample(1:10, 5), Y = sample(1:10, 5))
df
```

```
   X Y
1  3 5
2 10 4
3  2 6
4  8 8
5  6 1
```

```r
max(df[, 2])
```

```
[1] 8
```

2.5 Find the odd values in the second column in a given data frame `df`.

```r
set.seed(456)
df = data.frame (X = sample(1:10, 5), Y = sample(1:10, 5))
df
```

```
   X  Y
1  5  9
2  3 10
3  6  7
4 10  2
5  4  3
```

```r
df[, 2] %% 2 != 0
```

```
[1]  TRUE FALSE  TRUE FALSE  TRUE
```

```r
which(df[, 2] %% 2 != 0)
```

```
[1] 1 3 5
```

```r
df[which(df[, 2] %% 2 != 0), 2]
```

```
[1] 9 7 3
```

3 Q3

```r
b = c(78, 72, 78, 79, 105) # before
b
```

```
[1]  78  72  78  79 105
```

```r
a = c(67, 65, 79, 70, 93) # after
a
```

```
[1] 67 65 79 70 93
```

3.1 Compute the amount of weight loss/gain for each participant.

```r
c = b - a # loss (difference)
c
```

```
[1] 11  7 -1  9 12
```

3.2 Compute the average amount of weight loss.

```r
mean(c)
```

```
[1] 7.6
```

3.3 Arrange the object b in ascending order.

```r
sort(b)
```

```
[1]  72  78  78  79 105
```

3.4 Rank the object b in ascending order.

```
rank(b)
```

```
[1] 2.5 1.0 2.5 4.0 5.0
```

Note: the `rank()` function averages the rank of entries of the same value. This is different from the `order()` function:

```
order(b)
```

```
[1] 2 1 3 4 5
```

3.5 Create a matrix m that contains b and a as rows.

```
mat = matrix(c(b, a), nrow = 2, byrow = TRUE) # by default byrow is FALSE
mat
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]   78   72   78   79  105
[2,]   67   65   79   70   93
```

3.6 Compute the Z-score for both the rows of the above matrix.

- **Version 1: normalization by row**

```
row_means = apply(mat, 1, mean)
row_means
```

```
[1] 82.4 74.8
```

```
row_sd = apply(mat, 1, sd)
row_sd
```

```
[1] 12.93445 11.49783
```

```r
zscores_by_row = sweep(sweep(mat, 1, row_means, "-"), 1, row_sd, "/")
zscores_by_row
```

```
           [,1]       [,2]       [,3]       [,4]     [,5]
[1,] -0.3401768 -0.8040543 -0.3401768 -0.2628639 1.747272
[2,] -0.6783891 -0.8523351  0.3652865 -0.4174702 1.582908
```

- **Version 2: global normalization**

```r
zscores_overall = (mat - mean(mat))/sd(mat)
zscores_overall
```

```
            [,1]      [,2]        [,3]        [,4]     [,5]
[1,] -0.04912828 -0.540411 -0.04912828  0.03275218 2.161644
[2,] -0.94981336 -1.113574  0.03275218 -0.70417197 1.179079
```