

# QUIZ #2 SOLUTIONS

MSAN 593

July 26, 2018

## Instructions

1. No computer, no notes or electronic devices permitted in this quiz.
2. You may only use a pencil and eraser or pen.
3. Write your name at the top of the first page of this quiz.
4. You have 45 minutes to complete the quiz.

## Question 1 (2 pts)

What does the last line of code return?

```
x <- 0:3
x <- as.logical(x)
x[4]
```

```
## [1] TRUE
```

## Question 2 (4 = 2 + 2 pts)

I have created a data frame using the following code:

```
myDF <- tibble::data_frame(1:4, c("Paul", "Jen", "David", "Susan"), runif(4, 10, 100))
```

- (a) Write code that returns observations 2 and 3 of myDF using base R.

```
myDF[2:3, ]
```

```
## # A tibble: 2 x 3
##   `1:4` `c("Paul", "Jen", "David", "Susan")` `runif(4, 10, 100)`
##   <int> <chr>                                <dbl>
## 1     2 Jen                                56.8
## 2     3 David                               42.0
```

- (b) When calling `str(myDF)`, what is the type of column 2?

```
str(myDF)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    4 obs. of  3 variables:
## $ 1:4                                : int  1 2 3 4
## $ c("Paul", "Jen", "David", "Susan"): chr  "Paul" "Jen" "David" "Susan"
## $ runif(4, 10, 100)                  : num  19.9 56.8 42 79.4
```

### Question 3 (3 pts)

List the homogeneous data structures in R.

- Atomic Vectors
- Matrix
- Array

### Question 4 (2 pts)

Write code that creates an `integer` vector, stored in `myVec`, with 5 sequential integer values.

```
myVec <- 1:5
```

OR

```
myVec <- as.integer(c(1, 2, 3, 4, 5))
```

OR

```
myVec <- c(1L, 2L, 3L, 4L, 5L)
```

### Question 5 (2 pts)

What happens when the preceding code is executed?

```
myNum <- 3 + "3"
```

Error in 3 + "3" : non-numeric argument to binary operator

### Question 6 (2 pts)

```
myAtomicVector_01 <- c(99.1, 98.2, 97.3, 96.4, NA)
```

Write code which stores the mean of `myAtomicVector_01` in the variable `myMean`.

```
(myMean <- mean(myAtomicVector_01, na.rm = T))
```

```
## [1] 97.75
```

### Question 7 (2 pts)

What does the preceding code return? Explain. *hint: read the statement carefully*

```
(myVec <- logical(T, F, 4, 1, -1))
```

Error in logical(T, F, 4, 1, -1) : unused arguments (F, 4, 1, -1)

### Question 8 (4 pts)

List the four common types of vectors.

- Logical
- Integer
- Double
- Character

### Question 9 (2 pts)

You are provided with the following code. What does the last line of code return?

```
x <- 1:4
y <- 5:6
(z <- x + y)
```

```
## [1] 6 8 8 10
```

### Question 10 (2 pts)

You are provided with the following code. What does the last line of code return?

```
x <- 1:5
y <- 5:6
(z <- x + y)
```

```
## Warning in x + y: longer object length is not a multiple of shorter object
## length
## [1] 6 8 8 10 10
```

### Question 11 (4 = 2 + 2 pts)

You are provided with the following code.

(a) What does the last line of code return?

```
x <- c(1:5, NULL, NA)
length(x)
```

```
## [1] 6
```

(b) What does the last line of code return?

```
x <- c(1:5, NULL, NA)
x <- NULL
length(x)
```

```
## [1] 0
```

### Question 12 (2 pts)

```
myDF <- tibble::data_frame(col1 = 1:3, col2 = LETTERS[1:3], col3 = letters[24:26])
```

Using base R, write code that accesses the second column of `myDF` by referencing the column's name.

```
myDF["col2"]
```

```
## # A tibble: 3 x 1
##   col2
##   <chr>
## 1 A
## 2 B
## 3 C
```

### Question 13 (8 = 4 + 4 pts)

```
# the following code generates a random vector of 100,000 capital letters
set.seed(99)
myVec <- sample(LETTERS, 100000, replace = T)
```

- (a) Write code that will return all entries of `myVec` that are between "A" and "C" inclusive.

```
# the first line of code is unnecessary but not wrong
myVec <- factor(myVec, levels = LETTERS[1:26], ordered = T)
myVec[myVec <= "C" & myVec >= "A"]
```

- (b) Write the code that computes how many entries are between "H" and "K" in `myVec`, inclusive?

```
length(myVec[myVec <= "K" & myVec >= "H"])
```

```
## [1] 15611
```

### Question 14 (4 pts)

In a single statement, write code that creates a data frame with two columns, `myCol1` and `myCol2`, where `myCol1` has all upper case English letters stored as characters, and where `myCol2` has a sequence of integers from -101 to -126. Store this data frame in a variable called `myDataFrame`.

```
myDataFrame <- data.frame(myCol1 = LETTERS, myCol2 = -101:-126)
```

### Question 15 (2 pts)

What is returned when the following lines of code are run?

```
myDF <- tibble::data_frame(x = 1:5, y = letters[1:5])
length(myDF)
```

```
## [1] 2
```

### Question 16 (1 pt)

What distinct advantage does one get by creating a `tibble` (`data_frame`) instead of a `data.frame`? (*hint*: this is particularly noticeable when importing data)

`stringsAsFactors = F` as default

### Question 17 (3 pts)

Indicate whether the following `dplyr` functions operate on rows (observations) or columns (variables). Write your answer directly after the function

- `select` operates on columns
- `filter` operates on rows
- `slice` operates on rows

### Question 18 (6 = 3 + 3 pts)

A subset of columns from the `starwars` data set, stored in a `data frame` named `myStarWars`. The first 10 rows of `myStarWars` follows:

name	height	mass	birth_year	gender	homeworld	species
Luke Skywalker	172	77	19.0	male	Tatooine	Human
C-3PO	167	75	112.0	NA	Tatooine	Droid
R2-D2	96	32	33.0	NA	Naboo	Droid
Darth Vader	202	136	41.9	male	Tatooine	Human
Leia Organa	150	49	19.0	female	Alderaan	Human
Owen Lars	178	120	52.0	male	Tatooine	Human
Beru Whitesun lars	165	75	47.0	female	Tatooine	Human
R5-D4	97	32	NA	NA	Tatooine	Droid
Biggs Darklighter	183	84	24.0	male	Tatooine	Human
Obi-Wan Kenobi	182	77	57.0	male	Stewjon	Human

- Answer the following questions **exclusively** using as many `dplyr` functions as possible. You may use certain base R functions, but these questions are designed to showcase your knowledge of `dplyr`. Using base R where an equivalent or better `dplyr` function is available may result in losing marks.
- You may assume that `dplyr` is already loaded.
- There is no need to store the results of any of the following questions in a variable. I am simply interested in the code that generates the output.

- (a) Write code that will return a subset of the that includes only `name`, `homeworld` and `birth_year`. The abridged result should looks like this (don't worry about formatting):

```
knitr::kable(head(select(myStarWars, name, homeworld, birth_year)))
```

name	homeworld	birth_year
Luke Skywalker	Tatooine	19.0
C-3PO	Tatooine	112.0
R2-D2	Naboo	33.0
Darth Vader	Tatooine	41.9
Leia Organa	Alderaan	19.0
Owen Lars	Tatooine	52.0

- (b) Write code that will return all **male** Star Wars characters whose home world is **Tatooine** from `myStarWars` (don't worry about NAs). The abridged result should looks like this (don't worry about formatting):

```
knitr::kable(head(filter(myStarWars, gender == 'male', homeworld == "Tatooine")))
```

name	height	mass	birth_year	gender	homeworld	species
Luke Skywalker	172	77	19.0	male	Tatooine	Human
Darth Vader	202	136	41.9	male	Tatooine	Human
Owen Lars	178	120	52.0	male	Tatooine	Human
Biggs Darklighter	183	84	24.0	male	Tatooine	Human
Anakin Skywalker	188	84	41.9	male	Tatooine	Human
Cliegg Lars	183	NA	82.0	male	Tatooine	Human