In-Class Exercise 2

Nathan Van Ymeren

Preface:

First let's load the tidyverse:

```
library(tidyverse)
```

Question 1, 1.42 pts

Copy the following code into R (note: don't worry if you don't know why this works..we didn't cover this code in class and the set.seed() or the runif() functions are not on the exam..this is just for fun..)

```
set.seed(888)
rand_vec <- runif(1000000, min = 0, max = 100)
```

You should have a vector of 1,000,000 randomly generated numbers starting with 2.55, 35.67, 6.14...

What is the standard deviation of `rand_vec`?

The answer is closest to:

1. 0
2. 22.390
3. 25.663
4. 27.483
5. *28.869*

The standard deviation function is just `sd()` which I determined by the advanced technique of "Searching Google which leads you to the R Documentation website":

```
sd(rand_vec)
```

```
28.8693771657007
```

Thus it's the fifth answer.

Question 2, 1.43 pts

The mtcars dataset is a built-in dataset in R. Some information about it can be found here.

To see mtcars in your Environment page, you can create a new data frame by entering the following comand in the Console:

```
my_cars <- mtcars
```

Question: The whole dataset has ____ rows (i.e., observations), and ____ columns (i.e., variables)

```
nrow(my_cars)
```

```
32
```

```
ncol(my_cars)
```

```
11
```

Thus the answers are 32 and 11.

Question 3, 1.43 pts

What is the minimum value of mpg and the maximum value of hp, respectively? (rounded to one decimal place)

1. 10.4; 52
2. 15.0; 335
3. 10.4; 205
4. 10.4; 335
5. None of the above.

```
min(my_cars$mpg)
```

```
10.4
```

```
max(my_cars$hp)
```

```
335
```

Thus the answer is the fourth choice.

Question 4, 1.43 pts

What is the average horsepower for all cars in the `my_cars` dataset? Fill in below, round to one decimal place.

```
round( mean( my_cars$hp ), digits = 1 )
```

```
146.7
```

Question 5, 1.43 pts

Suppose you have created the following vectors:

```
v1 <- c(5, 10, 15)
v2 <- c("Red", "Yellow", "Blue")
v3 <- c("a", "b")
```

Now, you want to create a data frame called df1 that contains these three vectors.

True or False: The following code will successfully create that data frame:

```
df1 <- data.frame(v1, v2, v3)
```

It's false because `v3` has length 2, whereas `v1` and `v2` have length 3. Dataframes need to be rectangular, and we can confirm this by running the code and seeing R complain about mismatched lengths, thus the answer is False.

Question 6, 1.43 pts

Suppose you run the following code to create a data frame:

```
mydf <- data_frame(
  a = c(1,2,3,NA,5),
  b = c(1,4,9,NA,25)
)
```

View the dataframe before completing the following questions, using the View(mydf) command.

I won't use `View()` when creating a pdf, but the df looks like this:

a	b
1	1
2	4
3	9
NA	NA
5	25

How would you calculate the variance of the a column in mydf based on non-missing values? Select all possible options that would do this. If none of them work, then select none of the above.

1. `var(a, na.rm = TRUE)`
2. `var(a, na.rm = FALSE)`
3. `var(mydf$a, na.rm = TRUE)`
4. `var(mydf$a, na.rm = FALSE)`
5. None of the above (if you choose this option do NOT choose any of the other options)

You can run them all and see that they all work but what's important to note is that `a` is declared only inside the parentheses of the `data_frame()` function, which means there isn't a global variable `a`, and `a` only exists as a column within `mydf` so we need to reference it with the `$` operator. We want to exclude missing values so you need to pick the one with `na.rm = TRUE` as well as the one that uses the `$` operator, and that's the fourth choice.

Question 7, 1.43 pts

You would now like to count the number of NA values in column a from the dataframe mydf. Select all possible lines of code that can achieve this. If none of them work, then select none of the above.

1. `sum(is.na(mydf$a))`
2. `sum(is.na(a))`
3. `sum(mydf$a, na.rm = TRUE)`
4. `sum(a, na.rm = TRUE)`
5. `None of the above`

Similar to Q6. To calculate the number of NA values in column `a` we need to test each value with `is.na()` which produces a vector:

```
is.na(mydf$a)
```

```
FALSE
FALSE
FALSE
TRUE
FALSE
```

Remembering the principle of implicit coercion we can just sum this whole vector we just produced:

```
sum( is.na( mydf$a ) )
```

```
1
```

And that result of 1 matches what we can plainly see from the declaration of `mydf` in question 6, so the answer to this question is the first choice.