QUIZ #4

MSAN 593

August 9, 2018

NAME:

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.
5. **For all questions you may assume that the `tidyverse`, `scales` and `magrittr` packages are loaded into memory.**

n.b. The question will be scored in part based on the concision and cleanliness of your code, as well as your use of piping and `dplyr`.

Question 1 ($5 \times 1 = 5$ pts)

The following function is defined:

```
myFunc <- function(maxIter = 10, reps = 10, k) maxIter * reps + k
```

What does `myFunc()` return?

```
myFunc()
```

```
## Error in myFunc(): argument "k" is missing, with no default
```

What does `myFunc(2)` return?

```
myFunc(2)
```

```
## Error in myFunc(2): argument "k" is missing, with no default
```

What does `myFunc(k = 2)` return?

```
myFunc(k = 2)
```

```
## [1] 102
```

What does `myFunc(, , 2)` return?

```
myFunc(, , 2)
```

```
## [1] 102
```

What does `myFunc(4, m = 2, 3)` return?

```
myFunc(4, m = 2, 3)
```

```
## [1] 11
```

Question 2 (6 pts)

Write a function called myFunc which accepts three arguments with no default values. The function will only be called with numeric arguments. If all three values are provided to the function, the three arguments should be summed and returned. If one or more values are missing from the argument, the function should return an error message which says: "*You are missing x arguments*", where x is the number of arguments missing from the function call.

```r
myFunc <- function(x, y, z) {
    if (nargs() == 3) {
        sum(x, y, z)
    } else {
        print(paste(3 - nargs(), " variable(s) are missing"))
    }
}
```

Question 3 (4 pts)

Using an lapply

1. Write code that will take a data frame with columns that are exclusively numeric named myNum_DF (of any size) as input
2. Add 3 to all values in the data frame using an *anonymous function.*
3. Return an updated *data frame*

```r
(myNum_DF <- data.frame(a = 1:3, b = 10:12))

data.frame(lapply(myNum_DF, function(x) x + 3))
```

Question 4 (6 + 7 + 3 = 16 pts)

Data on all participants in all Olympic games up to an including the 2016 Olympics is imported into the data frame `olympics`. The fields and data are structured as follows:

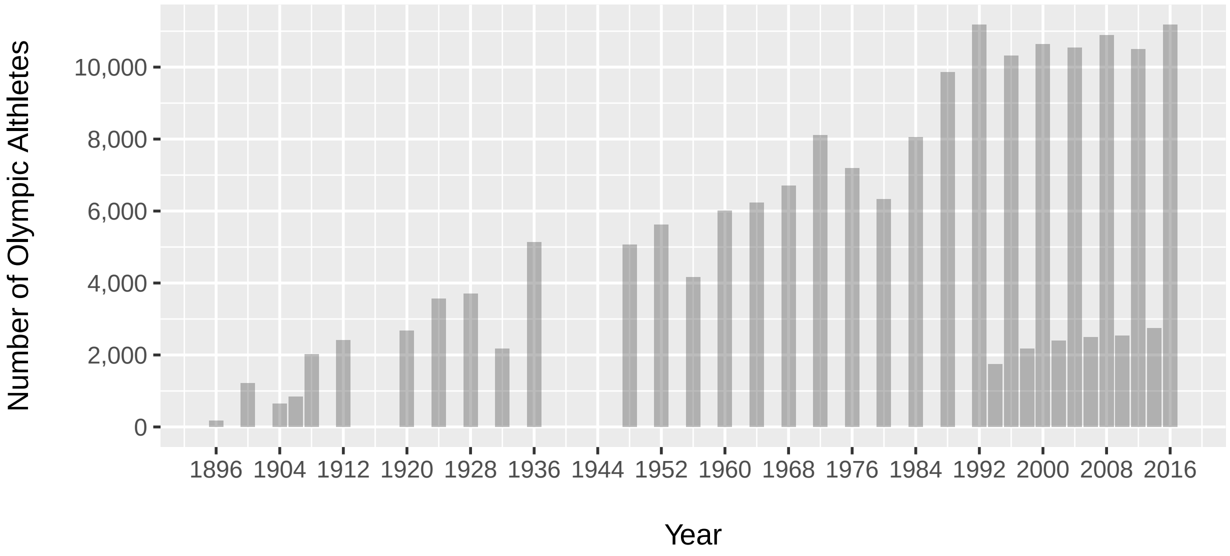
```
glimpse(olympics)
```

```
## Observations: 271,116
## Variables: 15
## $ ID     <int> 1, 2, 3, 4, 5, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6, 7...
## $ Name   <chr> "A Dijiang", "A Lamusi", "Gunnar Nielsen Aaby", "Edgar ...
## $ Sex    <chr> "M", "M", "M", "M", "F", "F", "F", "F", "F", "F", "M", ...
## $ Age    <int> 24, 23, 24, 34, 21, 21, 25, 25, 27, 27, 31, 31, 31, 31,...
## $ Height <int> 180, 170, NA, NA, 185, 185, 185, 185, 185, 185, 188, 18...
## $ Weight <dbl> 80, 60, NA, NA, 82, 82, 82, 82, 82, 82, 75, 75, 75, 75,...
## $ Team   <chr> "China", "China", "Denmark", "Denmark/Sweden", "Netherl...
## $ NOC    <chr> "CHN", "CHN", "DEN", "DEN", "NED", "NED", "NED", "NED",...
## $ Games  <chr> "1992 Summer", "2012 Summer", "1920 Summer", "1900 Summ...
## $ Year   <int> 1992, 2012, 1920, 1900, 1988, 1988, 1992, 1992, 1994, 1...
## $ Season <chr> "Summer", "Summer", "Summer", "Summer", "Winter", "Wint...
## $ City   <chr> "Barcelona", "London", "Antwerpen", "Paris", "Calgary",...
## $ Sport  <chr> "Basketball", "Judo", "Football", "Tug-Of-War", "Speed ...
## $ Event  <chr> "Basketball Men's Basketball", "Judo Men's Extra-Lightw...
## $ Medal  <chr> NA, NA, NA, "Gold", NA, NA, NA, NA, NA, NA, NA, NA, NA,...
```

(a) Write the code that will recreate the following graph of the number of Olympians participating in each Olympics. Pay attention to details, including

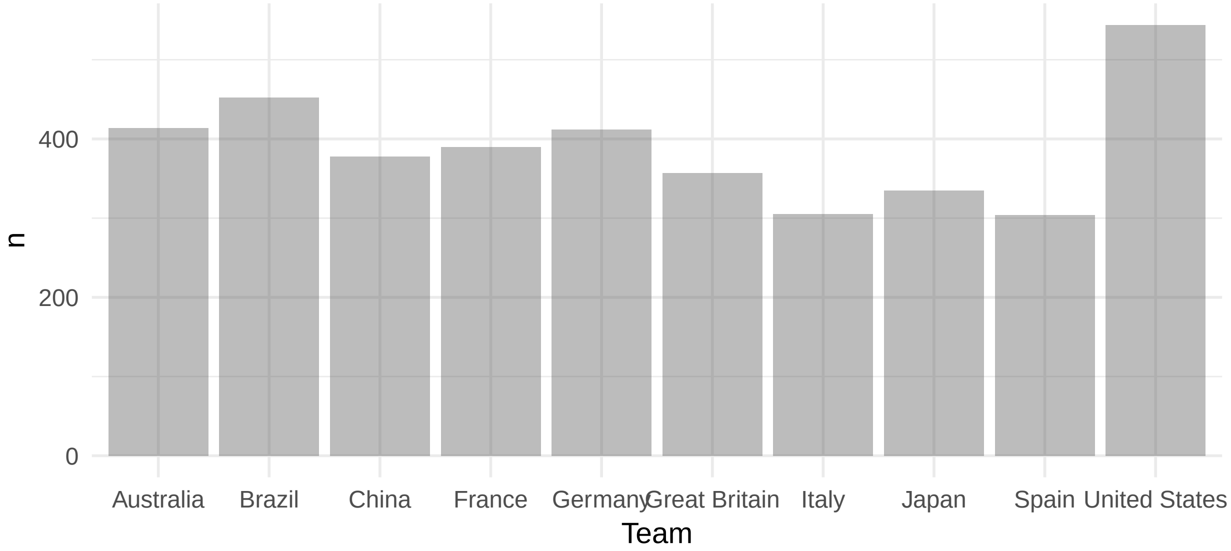
- `alpha = 0.4`
- y-axis commas
- x- and y-axis title
- x- and y-axis tick marks

```
olympics %>% count(Name, Year) %>% ggplot() + geom_bar(aes(x = Year), alpha = 0.4) +
    xlab("\nYear") + ylab("Number of Olympic Althletes\n") + scale_y_continuous(labels = comma,
    breaks = seq(0, 16000, by = 2000)) + scale_x_continuous(breaks = seq(1896, 2016,
    by = 8))
```


Number of Olympic Althletes
10,000
8,000
6,000
4,000
2,000
0
1896
1904
1912
1920
1928
1936
1944
1952
1960
1968
1976
1984
1992
2000
2008
2016
Year

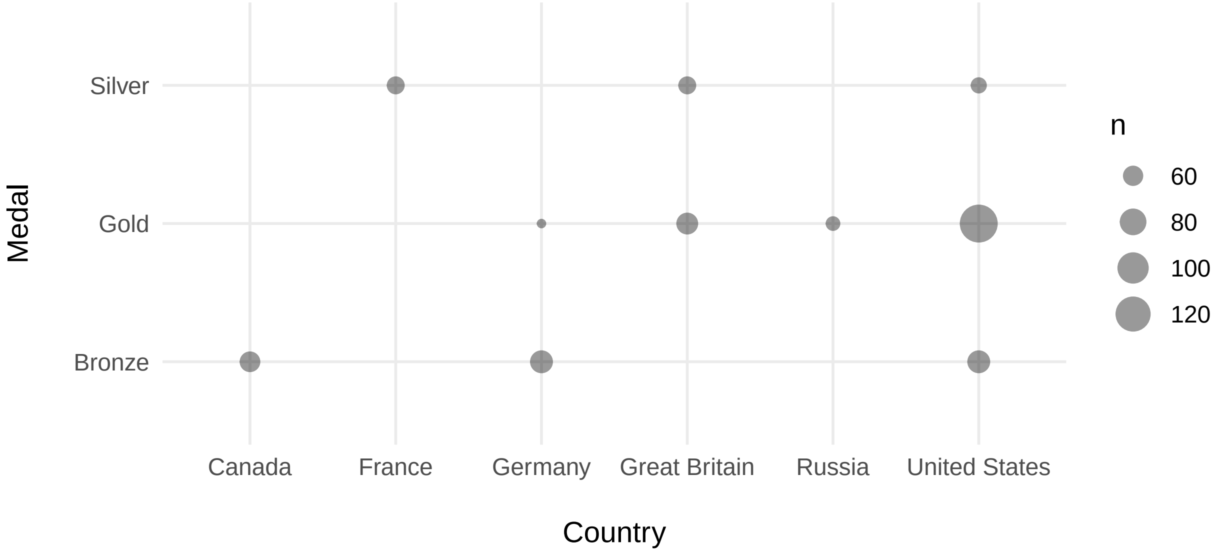
(b) Write the code that will recreate the following graph, selecting the top 10 countries with the most Olympians in the 2016 Olympics. *No need* to pay attention to axis details or themes.

```r
olympics %>% filter(Year == 2016) %>% distinct(Name, .keep_all = T) %>% count(Team) %>%
    top_n(10, n) %>% ggplot() + geom_bar(aes(x = Team, y = n), stat = "identity",
    alpha = 0.4) + theme_minimal()
```



(c) Fill in the missing line of code, represented by a . . . , that will generate the following graph:

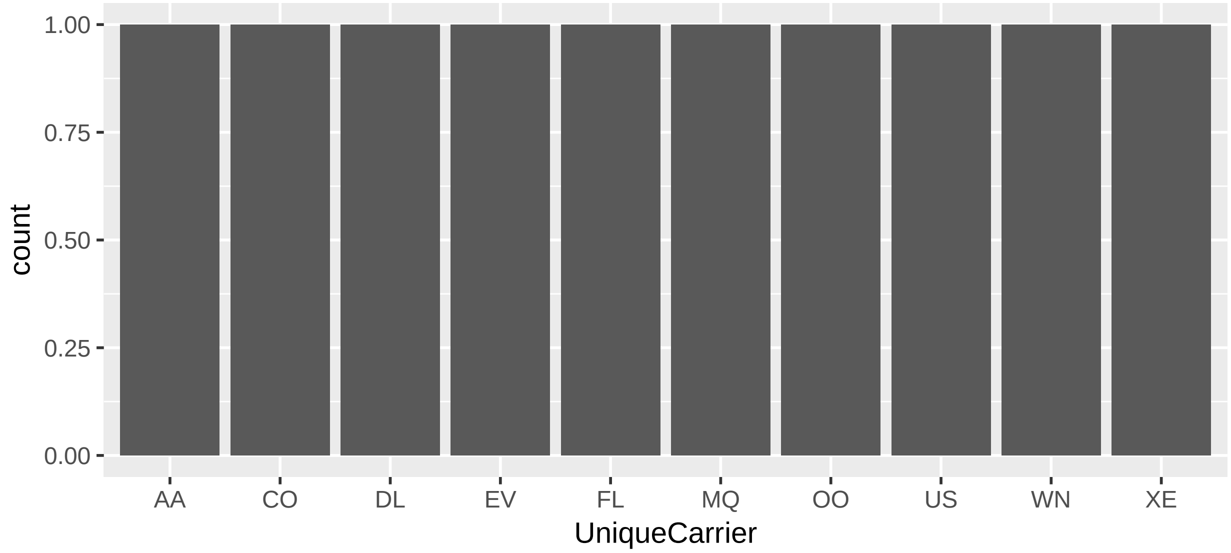
```
olympics %>% filter(Year == 2016, !is.na(Medal)) %>% count(Team, Medal) %>% top_n(10,
    n) %>% ggplot() + geom_point(aes(x = Team, y = Medal, size = n), alpha = 0.4) +
    xlab("\nCountry") + ylab("Medal\n") + theme_minimal()
```



Question 5 (4 pts)

Using `hflights` data, peer who is attempting to generate a bar graph of of the number of total flights per carrier in `ggplot` is frustrated, ending up with the following output:

```
hflights::hflights %>% count(UniqueCarrier) %>% top_n(10, n) %>% ggplot() + geom_bar(aes(x = UniqueCarr
```



Clearly this is not the output your colleague is attempting to generate. **Succintly** explain what might be happening and why, and suggest a correction. Be clear but **concise**.

Your colleague is trying to create a bar plot, but is not passing raw data to `geom_ bar()`. Rather, your colleague is passing a frequency table, i.e., a two-by-two table with categories and frequencies already determined. To pass this type of data to `geom_bar()`, you are reuired to include a `y =` inside the `aes()`, as well as include the option `stat = "identity" outside the aesthetic but inside the`geom_bar()‘.

Question 6 (4 pts)

You are provided with a vector `myVec` of length n, which contains exclusively integer values. Write the most succinct (shortest) possible code that evaluates whether or not each element in `myVec` is positive or negative, printing the result to the console window. **Hint: Do not overthink, answer is very short.**

```
ifelse(myVec > 0, "positive", "negative")
```

USE THE FOLLOWING DESCRIPTION OF DATA FOR BOTH QUESTIONS 7 & 8

You are provided with two data frames. The first data frame is a `customer` data frame with 92 rows and contains three variables: `custID`, `fname` and `lname`. The second data frame is a `transactions` data frame that contains information on purchase transactions at an online retailer. The `transactions` data frame contains 432 observations and 3 variables: `custID`, `date` and `purchAmt`.

Question 7 (4 pts)

The following code is used to `join` both tables:

```
newDF <- left_join(transactions, customer, by = "custID")
```

How many rows will `newDF` have? Choose the **most correct** answer.

(a) Exactly 92
(b) Exactly 432
(c) Less than or equal to 432
(d) Less than or equal to 92
(e) Greater than or equal to 92
(f) Greater than or equal to 432

ANSWER: (b)

Question 8 (4 pts)

The following code is used to `join` both tables:

```
newDF <- full_join(transactions, customer, by = "custID")
```

How many rows will `newDF` have? Choose the **most correct** answer.

(a) Exactly 92
(b) Exactly 432
(c) Less than or equal to 432
(d) Less than or equal to 92
(e) Greater than or equal to 92
(f) Greater than or equal to 432

ANSWER: (f)