In-Class Exercise 5

Nathan Van Ymeren

Preface:

In previous documents I reproduced the question text entirely. This time to make it clearer I'm going to omit the question text and just have some explanatory prose in each section.

First let's load the tidyverse and our dataset:

```
library(tidyverse)
setwd("~/Developer/r/")
companies = readRDS("~/Developer/r/North American Stock Market 1994-2018.rds")
```

Question 1

So this question doesn't technically require us to run the code, if we know what the dataset looks like, but I ran it anyway because we need it for the following questions.

```
q1 = companies %>%
  filter(fyear >= 2016, fyear <= 2017, at >= 10, sale >= 10) %>%
  select(gvkey, fyear, conm, at, sale)
```

If you look at the first few rows of the data you can see that you can't uniquely address rows by `gvkey` alone, but you could by the combination of `gvkey` and `fyear`, so we'll go with answer B.

Question 2

Not too different from Question 1 but this time they want us to filter differently and then merge the result with the dataset from question 1, so let's just go ahead and do that:

```
    q2 = companies %>%
      filter(fyear >= 2017, fyear <= 2018) %>%
      filter(at >= 100, sale >= 100) %>%
      select(gvkey, fyear, tic, ni)

  merged1 = inner_join(q1, q2, by = c("gvkey" = "gvkey", "fyear" = "fyear"))
```

```
nrow(merged1)
```

And calling `nrow()` gives us our answer, which is:

```
4303
```

Question 3

Another one where you *technically* don't need to run the code but you need the dataframe for question 4 so why not.

```
q3 = companies %>%
  filter(at >= 50, sale >= 50, !is.na(emp), !is.na(ni)) %>%
  mutate(ROA = ni / at) %>%
  group_by(gvkey) %>%
  summarise(ROA_avg = mean(ROA))
```

This time because we've condensed (reduced) the dataset using what R calls summary functions, we have only two columns which are `gvkey` and `ROA_avg` meaning we can uniquely address any row by its `gvkey`, unlike in question 1. So the answer is A.

Question 4

Counting the number of observations (rows) is trivial:

```
nrow(q3)
```

And our answer is:

```
13835
```

Question 5

So *technicallyyyyyy* there's a single row that has an `NA` value for fiscal year, but non-missing employment data. Whether or not that row should be included is not clear. That single row can be found like so:

```
companies %>%
  filter(is.na(fyear), !is.na(emp)) %>%
  select(gvkey, fyear, emp)
```

And voila:

gvkey	fyear	emp
20115	NA	1.033

Anyways, it'll change the answer by 1 depending on whether or not you include this row. But assuming we keep this row:

```
 q5 = companies %>%
  filter(!is.na(emp)) %>%
  group_by(gvkey) %>%
  summarise(emp_avg = mean(emp))

merge5 = inner_join(q3, q5, by = "gvkey")

nrow(merge5)
```

Which produces:

```
13835
```

Question 6

This question and the next few are meant to get you to reason about the code but you can always make a mockup:

```
student1 = tibble(ubcid = c(1, 2, 3, 4, 5, 6, 7))
student2 = tibble(ubcid = c(5, 6, 7, 8, 9, 10, 11, 12, 13, 14))
merge6 <- left_join(student1, student2)
nrow(merge6)
```

That gives:

```
7
```

So there are exactly seven (thousand) rows in what the question refers to as `merged1`.

Question 7

This time it's a left-join:

```
merge7 <- left_join(student2, student1)
nrow(merge7)
```

This time our answer is:

```
10
```

So there are exactly ten (thousand) rows in what the question refers to as `merged2`.

Question 8

Inner joins (per the question) on these sets ought to produce 3000 rows based on what they gave us. Let's see:

```r
merge8 <- inner_join(student1, student2)
nrow(merge8)
```

Gives:

```
3
```

Yup.

Question 9

This one you don't even really need to think about. We merged on more than one key variable in Question 2, so clearly this is False.