# In-Class Exercise 6

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

This set seems trickier than the last, right from the get go. The question asks us to determine how to list US-headquartered firms from highest assets to lowest, and then subsort again on firm name alphabetically as a tiebreaker. Answer B will filter for missing `fyear` values rather than filtering them out, so that can't be it. Answer C uses a Boolean OR for location, so that's not what we want, and answer D arranges the assets in ascending order (the default sort order) rather than descending so that's also not what we want. Answer A does what we want.

## Question 2

This one's pretty straightforward:

```
q2 = companies %>%
  filter(naicsh > 99999, naicsh < 1000000)

nrow(q2)
```

And we have this many rows:

```
224129
```

## Question 3

Straightforward-ish:

```
q3 = q2 %>%
  group_by(gvkey, naicsh) %>%
  summarize(count = n()) %>%
  ungroup() %>%
  group_by(gvkey) %>%
  summarize(count2 = n()) %>%
  filter(count2 > 1)

nrow(q3)
```

Essentially the first `summarize()` gets you the number of fyears in which a given company was listed under a given NAICS code. The trick is to run a second `summarize()` which gives you the number of NAICS codes under which a company has existed, and thus filtering for that second count to be greater than one gets us the answer:

```
5645
```

## Question 4

There are multiple ways to solve this (obviously). Angus and I originally both solved it during class using an inner join, but then later we realized you can do it without a second table:

```
q4 = companies %>%
  filter(sale >= 100, at >= 100, naicsh > 99999, naicsh < 1000000) %>%
  filter(!is.na(emp), !is.na(ni), loc == "USA") %>%
  group_by(naicsh, fyear) %>%
  mutate(count = n_distinct(gvkey)) %>%
  filter(count > 2)

nrow(q4)
```

What this does is first do the required screening and then, for every combination of fiscal year and NAICS code, figure out how many different firms (`gvkeys`) are in that group, and as directed in the question we drop groups with fewer than 3 firms. You could also do it using a second table/data frame with a `summarize()` and then joining them together. I ran it both ways and got the same result which is:

```
63796
```

## Question 5

Again, a few different ways you could solve this but I felt it was cleaner to do without merges.

```
q5 = q4 %>%
  mutate(roa = ni / at) %>%
  group_by(naicsh, fyear) %>%
  mutate(medroa = median(roa)) %>%
  ungroup() %>%
  mutate(fsp = roa - medroa) %>%
  filter(fsp > 0)

nrow(q5)
```

Basically rather than doing the work in a second table I opted to just compute the median using group and storing in a column on the original table. Then we computed the Firm Specific Profits and filtered for `fsp` strictly greater than zero, which gives:

```
29995
```

## Question 6

I actually struggled with this one a bit because I kept getting a number much too low to be plausible, and it's because I was doing the wrong comparisons when filtering. It helps to build up the dataframe one pipe operation at a time, and then explore the result to make sure it matches what you were expecting.

```
q6 = q4 %>%
  mutate(roa = ni / at) %>%
  group_by(naicsh, fyear) %>%
  mutate(medroa = median(roa)) %>%
  ungroup() %>%
  mutate(fsp = roa - medroa) %>%
  group_by(gvkey) %>%
  filter(fyear >= 2012, fyear <= 2018, fsp > 0) %>%
  summarize(num = n()) %>%
  filter(num == 7)

nrow(q6)
```

I'm certain there's a more elegant way to get to the answer, and I know a couple people got here using a second table and joins, but this looks uglier than it is because it re-creates the `q5` datataset on the way. The trick here was to drop all the rows outside the date range that the question specifies, which makes it simple to count `gvkey` group sizes with `n()` and then filter for groups equal to 7 in size, and that gives us our answer:

```
274
```