Assignment 3

Daniel Rios

## Question 1

echo = TRUE  
library(tidyverse)

── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
✔ dplyr 1.1.1 ✔ readr 2.1.4  
✔ forcats 1.0.0 ✔ stringr 1.5.0  
✔ ggplot2 3.4.2 ✔ tibble 3.2.1  
✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
✔ purrr 1.0.1   
── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
✖ dplyr::filter() masks stats::filter()  
✖ dplyr::lag() masks stats::lag()  
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(rio)  
  
USMA <- import('USMA\_Progression.xlsx')

## Question 2

USMA <- USMA %>%  
 group\_by(year, company\_n) %>%   
 summarize(femalepeers2 = sum(female), .groups = "drop") %>%  
 left\_join(USMA, by = c('year', 'company\_n')) %>%  
 mutate(femalepeers2 = case\_when(  
 female == 1 ~ femalepeers2 - 1,  
 TRUE ~ femalepeers2))  
   
USMA

# A tibble: 17,223 × 9  
 year company\_n femalepeers2 class female femalespeers malespeers  
 <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
 1 78 A-1 0 1 0 0 34  
 2 78 A-1 0 1 0 0 34  
 3 78 A-1 0 1 0 0 34  
 4 78 A-1 0 1 0 0 34  
 5 78 A-1 0 1 0 0 34  
 6 78 A-1 0 1 0 0 34  
 7 78 A-1 0 1 0 0 34  
 8 78 A-1 0 1 0 0 34  
 9 78 A-1 0 1 0 0 34  
10 78 A-1 0 1 0 0 34  
# ℹ 17,213 more rows  
# ℹ 2 more variables: continue\_or\_grad <dbl>, totpeople <dbl>

USMA <- USMA %>%  
 group\_by(year, company\_n) %>%   
 summarize(malepeers2 = sum(case\_when(female == 0 ~ 1, TRUE ~ 0)),  
 .groups = "drop") %>%  
 left\_join(USMA, by = c('year', 'company\_n')) %>%  
 mutate(malepeers2 = case\_when(  
 female == 0 ~ malepeers2 - 1,  
 TRUE ~ malepeers2))  
  
USMA

# A tibble: 17,223 × 10  
 year company\_n malepeers2 femalepeers2 class female femalespeers malespeers  
 <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
 1 78 A-1 33 0 1 0 0 34  
 2 78 A-1 33 0 1 0 0 34  
 3 78 A-1 33 0 1 0 0 34  
 4 78 A-1 33 0 1 0 0 34  
 5 78 A-1 33 0 1 0 0 34  
 6 78 A-1 33 0 1 0 0 34  
 7 78 A-1 33 0 1 0 0 34  
 8 78 A-1 33 0 1 0 0 34  
 9 78 A-1 33 0 1 0 0 34  
10 78 A-1 33 0 1 0 0 34  
# ℹ 17,213 more rows  
# ℹ 2 more variables: continue\_or\_grad <dbl>, totpeople <dbl>

USMA <- USMA %>%   
 group\_by(year, company\_n) %>%  
 summarize(totpeople2 = n(), .groups = "drop") %>%  
 left\_join(USMA, by = c('year', 'company\_n'))  
  
USMA

# A tibble: 17,223 × 11  
 year company\_n totpeople2 malepeers2 femalepeers2 class female femalespeers  
 <dbl> <chr> <int> <dbl> <dbl> <dbl> <dbl> <dbl>  
 1 78 A-1 34 33 0 1 0 0  
 2 78 A-1 34 33 0 1 0 0  
 3 78 A-1 34 33 0 1 0 0  
 4 78 A-1 34 33 0 1 0 0  
 5 78 A-1 34 33 0 1 0 0  
 6 78 A-1 34 33 0 1 0 0  
 7 78 A-1 34 33 0 1 0 0  
 8 78 A-1 34 33 0 1 0 0  
 9 78 A-1 34 33 0 1 0 0  
10 78 A-1 34 33 0 1 0 0  
# ℹ 17,213 more rows  
# ℹ 3 more variables: malespeers <dbl>, continue\_or\_grad <dbl>, totpeople <dbl>

## Question 3

For the malespeers and femalespeers columns the data not lining up seems to be occurring because they tend to display the **total** number of males or females rather than the peers (not including themselves). I would trust my recreation more because the calculation in itself should be accurate as it accounts for themself using the female column directly, in addition to that I have gone through a handful of classes and can see the problem being fixed.

The totpeople column typically has 1 more value than the recreation and it’s not entirely clear why. The variable is described as being femalespeers + malespeers, but that’s not accurate, this could be found just by looking at the first few classes. Perhaps it’s assumed that the peers factor is taken into account, so it’s adjusted (+1 person) each time to account for the total. I decided to look into the instances where the recreation did match the original, and in these instances the peers data is accurate. So for the majority of the classes, with peer data being inaccurate, it could also be causing total people to be. I trust my recreation more because this counts the number of records in each class which seems (as we saw) to be a more concrete and accurate way of counting the total number of people, as well as checking over a handful of classes.

i <- USMA %>%  
 filter(totpeople == totpeople2) %>%  
 select(company\_n,`totpeople`,`totpeople2`)  
i

# A tibble: 300 × 3  
 company\_n totpeople totpeople2  
 <chr> <dbl> <int>  
 1 A-2 39 39  
 2 A-2 39 39  
 3 A-2 39 39  
 4 A-2 39 39  
 5 A-2 39 39  
 6 A-2 39 39  
 7 A-2 39 39  
 8 A-2 39 39  
 9 A-2 39 39  
10 A-2 39 39  
# ℹ 290 more rows

## Question 4

USMA <- USMA %>%  
 mutate(company = str\_sub(company\_n, 1, 1)) %>%  
 mutate(division = str\_sub(company\_n, 3))   
  
USMA

# A tibble: 17,223 × 13  
 year company\_n totpeople2 malepeers2 femalepeers2 class female femalespeers  
 <dbl> <chr> <int> <dbl> <dbl> <dbl> <dbl> <dbl>  
 1 78 A-1 34 33 0 1 0 0  
 2 78 A-1 34 33 0 1 0 0  
 3 78 A-1 34 33 0 1 0 0  
 4 78 A-1 34 33 0 1 0 0  
 5 78 A-1 34 33 0 1 0 0  
 6 78 A-1 34 33 0 1 0 0  
 7 78 A-1 34 33 0 1 0 0  
 8 78 A-1 34 33 0 1 0 0  
 9 78 A-1 34 33 0 1 0 0  
10 78 A-1 34 33 0 1 0 0  
# ℹ 17,213 more rows  
# ℹ 5 more variables: malespeers <dbl>, continue\_or\_grad <dbl>,  
# totpeople <dbl>, company <chr>, division <chr>

## Question 5

USMA <- USMA %>%  
 group\_by(year) %>%  
 summarize(totclasses = n\_distinct(class)) %>%  
 filter(totclasses >= 4) %>%  
 left\_join(USMA, by = 'year')  
  
USMA

# A tibble: 7,559 × 14  
 year totclasses company\_n totpeople2 malepeers2 femalepeers2 class female  
 <dbl> <int> <chr> <int> <dbl> <dbl> <dbl> <dbl>  
 1 80 4 A-1 89 87 1 1 0  
 2 80 4 A-1 89 87 1 1 0  
 3 80 4 A-1 89 87 1 1 0  
 4 80 4 A-1 89 87 1 1 0  
 5 80 4 A-1 89 87 1 1 0  
 6 80 4 A-1 89 87 1 1 0  
 7 80 4 A-1 89 87 1 1 0  
 8 80 4 A-1 89 87 1 1 0  
 9 80 4 A-1 89 87 1 1 0  
10 80 4 A-1 89 87 1 1 0  
# ℹ 7,549 more rows  
# ℹ 6 more variables: femalespeers <dbl>, malespeers <dbl>,  
# continue\_or\_grad <dbl>, totpeople <dbl>, company <chr>, division <chr>

## Question 6

### a.

top\_grad\_rates <- USMA %>%  
 group\_by(company) %>%  
 summarize(rate = sum(continue\_or\_grad == 1) / n()) %>%  
 arrange(desc(rate)) %>%  
 slice(1:4)  
  
top\_grad\_rates

# A tibble: 4 × 2  
 company rate  
 <chr> <dbl>  
1 D 0.918  
2 H 0.914  
3 A 0.905  
4 E 0.901

### b.

class\_grad\_rates <- USMA %>%  
 group\_by(class) %>%  
 summarize(c\_rate = sum(continue\_or\_grad == 1) / n())  
  
class\_grad\_rates

# A tibble: 4 × 2  
 class c\_rate  
 <dbl> <dbl>  
1 1 0.849  
2 2 0.886  
3 3 0.952  
4 4 0.953

### c.

female\_grad\_rates <- USMA %>%  
 group\_by(class) %>%  
 filter(female == 1) %>%  
 summarize(f\_rate = sum(continue\_or\_grad == 1) / n())  
  
female\_grad\_rates

# A tibble: 4 × 2  
 class f\_rate  
 <dbl> <dbl>  
1 1 0.822  
2 2 0.749  
3 3 0.931  
4 4 0.984