Homework 5

Brady Lamson

3/12/2022

PDF PROBLEMS

Problem 1

```
setwd("/home/brady/repos/mth_3270_data_science/module_5/hw_5")
houses <- read.csv("houses-for-sale.txt", header = TRUE, sep = "\t")
translations <- read.csv("house_codes.txt", header = TRUE, sep = "\t")

houses_small <- select(houses, fuel, heat, sewer, construction)

codes <- translations %>%
    tidyr::pivot_wider(
        names_from = system_type,
        values_from = meaning,
        values_fill = "invalid"
    )
```

```
# A
# Join in codes df based on each type of code
# Then select only those code columns to remove the integer columns
houses_small_coded <- houses_small %>%

    dplyr::left_join(
        codes %>%
            dplyr::select(code, fuel_type),
            by = c(fuel = "code")
    ) %>%
    dplyr::left_join(
        codes %>%
            dplyr::select(code, heat_type),
            by = c(heat = "code")
    ) %>%
    dplyr::left_join(
        codes %>%
            dplyr::select(code, sewer_type),
            by = c(sewer = "code")
    ) %>%
    dplyr::left_join(
        codes %>%
```

```
            dplyr::select(code, new_const),
            by = c(construction = "code")
    ) %>%
    dplyr::select(fuel_type, heat_type, sewer_type, new_const)

houses_small_coded %>% head()
```

```
##   fuel_type heat_type sewer_type new_const
## 1  electric  electric    private        no
## 2       gas hot water    private        no
## 3       gas hot water     public        no
## 4       gas   hot air    private        no
## 5       gas   hot air     public       yes
## 6       gas   hot air    private        no
```

```
arrange(summarize(group_by(select(filter(houses_small_coded, new_const == "no"),
fuel_type, heat_type), fuel_type), count = n()), desc(count))
```

```
## # A tibble: 3 x 2
##   fuel_type count
##   <chr>     <int>
## 1 gas        1117
## 2 electric    314
## 3 oil         216
```

This command does the following:

First, it **filters** out only the rows with **NO** new construction. Then, we **select** the fuel_type and heat_type columns, ignoring all the others. After that, we **group by** the type of fuel. Then we **summarize** this data frame by the **count** of each **type** of fuel and we **order** those counts in **descending** order.

```
houses_small_coded %>%
    dplyr::filter(
        new_const == "no"
    ) %>%
    dplyr::select(fuel_type, heat_type) %>%
    dplyr::group_by(fuel_type) %>%
    dplyr::summarise(count = n()) %>%
    dplyr::arrange(dplyr::desc(count))
```

```
## # A tibble: 3 x 2
##   fuel_type count
##   <chr>     <int>
## 1 gas        1117
## 2 electric    314
## 3 oil         216
```

Problem 2

```
flights <- nycflights13::flights
```

```
# Group by destination and get total and average minutes of delay
flights %>%
    dplyr::group_by(dest) %>%
    dplyr::summarise(
        total_delay = sum(dep_delay, arr_delay, na.rm = TRUE),
        average_delay = c(dep_delay, arr_delay) %>%
            mean(na.rm = TRUE) %>%
            round(digits = 3)
    ) %>%
    dplyr::arrange(
        dplyr::desc(average_delay)
    )
```

```
## # A tibble: 105 x 3
##    dest  total_delay average_delay
##    <chr>       <dbl>         <dbl>
##  1 CAE          8233          38.7
##  2 TUL         20333          34.3
##  3 OKC         19641          30.6
##  4 JAC          1174          27.3
##  5 TYS         30410          26.3
##  6 BHM         12617          23.3
##  7 DSM         23791          22.6
##  8 MSN         24481          21.9
##  9 RIC        102711          21.9
## 10 CAK         34138          20.3
## # ... with 95 more rows
```

Problem 3

```
planes <- nycflights13::planes

# Using some hacky tricks we can figure out which column names match automatically
names(flights)[which(names(flights) %in% names(planes))]
```

```
## [1] "year"    "tailnum"
```

From this we can see that 'year' and 'tailnum' are our two candidates. **Year** is, based purely on intuition, probably not a good option. Year is tied to the plane in the planes data set, but not the flights data set. The year represents totally different things in each. Thankfully **tailnum** is tied to the tail number in both data sets so we can utilize that. I feel using **inner_join** should work out just fine as that will remove rows without a proper tail number and, by extension, those that lack the manufacturer information we need.

```
flights %>%
    dplyr::inner_join(
        planes,
        by = 'tailnum'
    ) %>%
    dplyr::group_by(manufacturer) %>%
    dplyr::summarise(count = n()) %>%
    dplyr::arrange(dplyr::desc(count))
```

```
## # A tibble: 35 x 2
##    manufacturer                count
##    <chr>                       <int>
##  1 BOEING                      82912
##  2 EMBRAER                     66068
##  3 AIRBUS                      47302
##  4 AIRBUS INDUSTRIE            40891
##  5 BOMBARDIER INC              28272
##  6 MCDONNELL DOUGLAS AIRCRAFT CO  8932
##  7 MCDONNELL DOUGLAS            3998
##  8 CANADAIR                     1594
##  9 MCDONNELL DOUGLAS CORPORATION  1259
## 10 CESSNA                        658
## # ... with 25 more rows
```

What we can see from this is that **Boeing** made the most flights with a count of **82912** flights to its name.

Textbook Problems

Chapter 5 Problem 3

- How many planes have a missing date of manufacture?

```
planes %>%
    dplyr::filter(is.na(year)) %>%
    dplyr::summarise(count = n()) %>%
    paste()
```

```
## [1] "70"
```

From this we can say that 70 of the planes in the planes data set are missing a data of manufacture.

- What are the five most common manufactures?

```
# We group by the manufacturer, count up the number for each
# Sort from most common to least and then
# extract the first 5 rows
planes %>%
    dplyr::group_by(manufacturer) %>%
    dplyr::summarise(count = n()) %>%
    dplyr::arrange(
        dplyr::desc(count)
    ) %>%
    # Extract only the top 5 rows
    dplyr::top_n(5)
```

```
## Selecting by count
```

```
## # A tibble: 5 x 2
##   manufacturer     count
##   <chr>            <int>
## 1 BOEING            1630
## 2 AIRBUS INDUSTRIE   400
## 3 BOMBARDIER INC     368
## 4 AIRBUS             336
## 5 EMBRAER            299
```

The 5 most common manufacturers are Boeing, airbus industrie, bombardier, airbus and embraer.

Chapter 5 Problem 4

- What is the oldest plane that flew from NYC airports in 2013?

For this we want to combine the planes and flights data sets again. We can combine a smaller version though as we are only concerned with flights done in 2013.

```
flights %>%
    dplyr::filter(year == 2013) %>%
    # Rename year to flight year so we can keep the planes year column
    dplyr::rename(flight_year = year) %>%
    dplyr::left_join(planes, by = 'tailnum') %>%
    dplyr::select(tailnum, year) %>%
    dplyr::filter(year == min(year, na.rm = TRUE))
```

```
## # A tibble: 22 x 2
##    tailnum  year
##    <chr>   <int>
##  1 N381AA   1956
##  2 N381AA   1956
##  3 N381AA   1956
##  4 N381AA   1956
##  5 N381AA   1956
##  6 N381AA   1956
##  7 N381AA   1956
##  8 N381AA   1956
##  9 N381AA   1956
## 10 N381AA   1956
## # ... with 12 more rows
```