Homework 08

Peyton Hall

03/16/2024

```
rm(list=ls())
```

```
library(readxl)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(nycflights13)
```

```
COVID_19_Vaccine <- read_excel("~/Desktop/Data211/Week 7/COVID-19-Vaccine.xlsx")
COVID_19_Vaccine
```

```
## # A tibble: 199 x 4
##    `Developer / Researcher`  ProductCategory StageDevelopment ProductDescription
##    <chr>                     <chr>           <chr>            <chr>
##  1 "Genexine Consortium (Ge~ DNA-based       Phase I          DNA vaccine (GX-1~
##  2 "Inovio Pharmaceuticals/~ DNA-based       Phase I          DNA plasmid vacci~
##  3 "Zydus Cadila Healthcare~ DNA-based       Phase I          DNA plasmid (ZyCo~
##  4 "BioNet Asia"             DNA-based       Pre-clinical     DNA
##  5 "Chula Vaccine Research ~ DNA-based       Pre-clinical     DNA with electrop~
##  6 "Ege University"          DNA-based       Pre-clinical     DNA
##  7 "Entos Pharmaceuticals/ ~ DNA-based       Pre-clinical     DNA; Covigenix
##  8 "Immunomic Therapeutics ~ DNA-based       Pre-clinical     DNA plasmid, need~
##  9 "Mediphage Bioceuticals/~ DNA-based       Pre-clinical     msDNA vaccine
## 10 "National\r\n  Research ~ DNA-based       Pre-clinical     DNA plasmid vacci~
## # i 189 more rows
```

```r
# 1. The data COVID-19-Vaccine on D2L recorded the developer, product category
#    (type of vaccine), stage of development, and vaccine description by
#    September 2020. Do the following three parts in one pipeline.
# a. Keep non-missing values of product category
COVID_19_Vaccine <- COVID_19_Vaccine %>%
  filter(!is.na(ProductCategory)) %>%
  filter(!is.na(StageDevelopment)) %>%
  filter(!is.na(ProductDescription))
# b. Find the total number of vaccines of each product category using
#    group_by() and summarize()
vaccine_counts <- COVID_19_Vaccine %>%
  group_by(ProductCategory) %>%
  summarize(Count = n())
# c. Sort the total counts of each product category
#    in b) (i.e. "Inactivated virus")
sorted_vaccine_counts <- vaccine_counts %>%
  arrange(desc(Count))
```

```r
Recycling <- read_excel("~/Desktop/Data211/Week 7/Recycling.xlsx")
Recycling
```

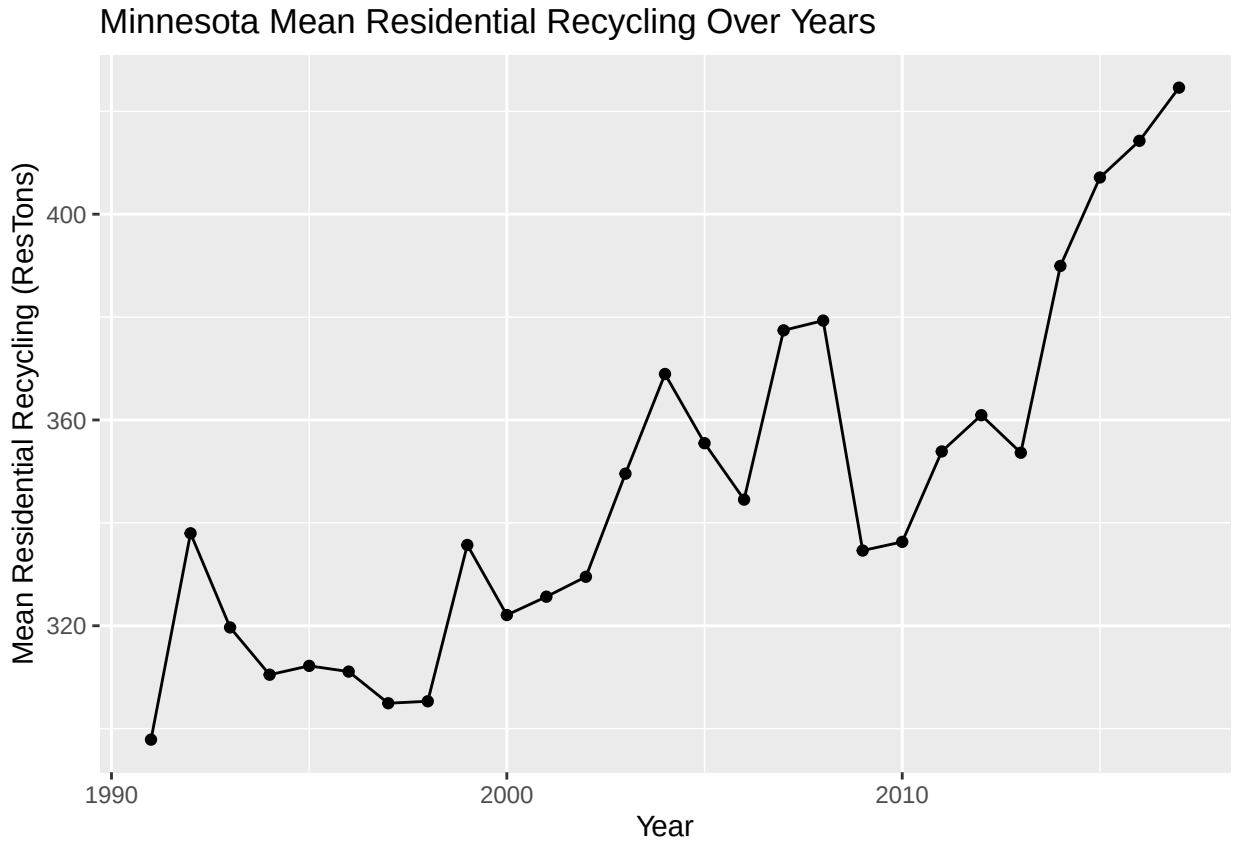
```
## # A tibble: 49,019 x 5
##     Year County Category  ResTons CIITons
##    <dbl> <chr>  <chr>       <dbl>   <dbl>
##  1  1991 Aitkin Paper          99       0
##  2  1991 Aitkin Paper           1       0
##  3  1991 Aitkin Paper          91       0
##  4  1991 Aitkin Paper           6       0
##  5  1991 Aitkin Metal          14       0
##  6  1991 Aitkin Metal         150       0
##  7  1991 Aitkin Metal          45       0
##  8  1991 Aitkin Glass          98       0
##  9  1991 Aitkin Plastic        17       0
## 10  1991 Aitkin Hazardous       6       0
## # i 49,009 more rows
```

```r
# 2. Use the data Recycling on d2l. This data is obtained from the Minnesota
#    Pollution Control Agency (MPCA) site. The data recorded the amount of
#    recycling in residential area (ResTons) and in commercial area
#    (CIITons) from 1991 to 2017 for the 87 Minnesota counties. Use tidyverse
#    functions to answer the following:
# a. Find the mean residential recycling (ResTons) by year
mean_residential_recycling <- Recycling %>%
  group_by(Year) %>%
  summarize(mean_residential_recycling = mean(ResTons))
mean_residential_recycling
```

```
## # A tibble: 27 x 2
##     Year mean_residential_recycling
##    <dbl>                      <dbl>
##  1  1991                       298.
##  2  1992                       338.
```

```
##  3  1993                       320.
##  4  1994                       310.
##  5  1995                       312.
##  6  1996                       311.
##  7  1997                       305.
##  8  1998                       305.
##  9  1999                       336.
## 10  2000                       322.
## # i 17 more rows
```

```r
# b. Graph the means in a) over years (note: a) and b) can be in one pipeline)
#    Graph the mean residential recycling over years
ggplot(mean_residential_recycling, aes(x = Year, y =
                                         mean_residential_recycling)) +
  geom_line() + geom_point() + labs(x = "Year",
                                    y = "Mean Residential Recycling (ResTons)",
       title = "Minnesota Mean Residential Recycling Over Years")
```




```r
# c. Find the mean residential recycling (ResTons) by county. What are the top
#    3 counties with the highest mean amount of residential recycling? Does that
#    make sense in MN? Use comments to answer in your R Markdown file.
mean_residential_recycling_by_county <- Recycling %>%
  group_by(County) %>%
  summarize(mean_residential_recycling = mean(ResTons))
# Sort the data to find the top 3 counties with the
```

```r
# highest mean residential recycling
top_3_counties <- mean_residential_recycling_by_county %>%
  top_n(3, mean_residential_recycling) %>%
  arrange(desc(mean_residential_recycling))
top_3_counties
```

```
## # A tibble: 3 x 2
##   County   mean_residential_recycling
##   <chr>                         <dbl>
## 1 Hennepin                      5021.
## 2 Ramsey                        2670.
## 3 Dakota                        1822.
```

```r
# Answer analysis:
# The top three counties with the highest mean amount of residential recycling
# are (#1) Hennepin, (#2) Ramsey, and (#3) Dakota. This seems to make sense to
# me, being a Minnesota resident my whole life, because these counties contain
# civilized cities including (but not limited to) Apple Valley, Maplewood,
# and Maple Grove.
# d. Find the mean residential recycling (ResTons) by category.
mean_residential_recycling_by_category <- Recycling %>%
  group_by(Category) %>% # Group the data by category
  summarize(mean_residential_recycling = mean(ResTons)) # calculate the mean
mean_residential_recycling_by_category
```

```
## # A tibble: 7 x 2
##   Category  mean_residential_recycling
##   <chr>                          <dbl>
## 1 Glass                          707.
## 2 Hazardous                      100.
## 3 Metal                          433.
## 4 Organic                        529.
## 5 Other                          221.
## 6 Paper                          680.
## 7 Plastic                         91.7
```

```r
Expenditure_and_Revenue <- read_excel("~/Desktop/Data211/Week 10/Expenditure and Revenue.xlsx")
Expenditure_and_Revenue
```

```
## # A tibble: 316 x 7
##     Year Administration Education Recycling `SCORE Revenue` `Local Revenue`
##    <dbl>          <dbl>     <dbl>     <dbl>           <dbl>           <dbl>
##  1  2014         161989      6073     68626           63803          163148
##  2  2014         295359     16133    320196          131222          246779
##  3  2014              0      3348    471161          125397          363168
##  4  2014         232003     22008         0          107633          178814
##  5  2014          73411      1525    108919           63803          120889
##  6  2014          82155     22310    168971          179551          122979
##  7  2014          56805      2125    351270           70283          369843
##  8  2014          79837      4716     70677           97711           55853
##  9  2014         181000         0    630676           78510          755611
## 10  2014          13557      2133    166254           63803          125353
```

```
## # i 306 more rows
## # i 1 more variable: `Other Revenue` <dbl>
```

```
# 3. The data Expenditure and Revenue on D2L recorded the expenditure and
#    revenue from Minnesota recycling from 2014 to 2017
# a. Create a new variable to show the total expenditure
#    (Administration+Education+Recycling)
Expenditure_and_Revenue <- Expenditure_and_Revenue %>%
  mutate(Total_Expenditure = Administration + Education + Recycling)
Expenditure_and_Revenue
```

```
## # A tibble: 316 x 8
##     Year Administration Education Recycling `SCORE Revenue` `Local Revenue`
##    <dbl>          <dbl>     <dbl>     <dbl>           <dbl>           <dbl>
##  1  2014         161989      6073     68626           63803          163148
##  2  2014         295359     16133    320196          131222          246779
##  3  2014              0      3348    471161          125397          363168
##  4  2014         232003     22008         0          107633          178814
##  5  2014          73411      1525    108919           63803          120889
##  6  2014          82155     22310    168971          179551          122979
##  7  2014          56805      2125    351270           70283          369843
##  8  2014          79837      4716     70677           97711           55853
##  9  2014         181000         0    630676           78510          755611
## 10  2014          13557      2133    166254           63803          125353
## # i 306 more rows
## # i 2 more variables: `Other Revenue` <dbl>, Total_Expenditure <dbl>
```

```
# b. Create a new variable to show the total revenue
#    (SCORE revenue+Local revenue+other revenue)
Expenditure_and_Revenue <- Expenditure_and_Revenue %>%
  mutate(Total_Revenue = `SCORE Revenue` + `Local Revenue` + `Other Revenue`)
Expenditure_and_Revenue
```

```
## # A tibble: 316 x 9
##     Year Administration Education Recycling `SCORE Revenue` `Local Revenue`
##    <dbl>          <dbl>     <dbl>     <dbl>           <dbl>           <dbl>
##  1  2014         161989      6073     68626           63803          163148
##  2  2014         295359     16133    320196          131222          246779
##  3  2014              0      3348    471161          125397          363168
##  4  2014         232003     22008         0          107633          178814
##  5  2014          73411      1525    108919           63803          120889
##  6  2014          82155     22310    168971          179551          122979
##  7  2014          56805      2125    351270           70283          369843
##  8  2014          79837      4716     70677           97711           55853
##  9  2014         181000         0    630676           78510          755611
## 10  2014          13557      2133    166254           63803          125353
## # i 306 more rows
## # i 3 more variables: `Other Revenue` <dbl>, Total_Expenditure <dbl>,
## #   Total_Revenue <dbl>
```

```
# c. Find the average expenditure and average revenue by year.
averages_by_year <- Expenditure_and_Revenue %>%
```

```
  group_by(Year) %>%
  summarise(Average_Expenditure = mean(Total_Expenditure, na.rm = TRUE),
            Average_Revenue = mean(Total_Revenue, na.rm = TRUE))
averages_by_year
```

```
## # A tibble: 4 x 3
##    Year Average_Expenditure Average_Revenue
##   <dbl>               <dbl>           <dbl>
## 1  2014             400251.         487970.
## 2  2015             428367.         517043.
## 3  2016             445394.         551990.
## 4  2017             451839.         584276.
```

```
# 4. Use the nycflights13 package planes data to answer the following questions:
# install.packages("nycflights13")
str(planes) # view the structure
```

```
## tibble [3,322 x 9] (S3: tbl_df/tbl/data.frame)
##  $ tailnum     : chr [1:3322] "N10156" "N102UW" "N103US" "N104UW" ...
##  $ year        : int [1:3322] 2004 1998 1999 1999 2002 1999 1999 1999 1999 1999 ...
##  $ type        : chr [1:3322] "Fixed wing multi engine" "Fixed wing multi engine" "Fixed wing multi e
##  $ manufacturer: chr [1:3322] "EMBRAER" "AIRBUS INDUSTRIE" "AIRBUS INDUSTRIE" "AIRBUS INDUSTRIE" ...
##  $ model       : chr [1:3322] "EMB-145XR" "A320-214" "A320-214" "A320-214" ...
##  $ engines     : int [1:3322] 2 2 2 2 2 2 2 2 2 2 ...
##  $ seats       : int [1:3322] 55 182 182 182 55 182 182 182 182 182 ...
##  $ speed       : int [1:3322] NA NA NA NA NA NA NA NA NA NA ...
##  $ engine      : chr [1:3322] "Turbo-fan" "Turbo-fan" "Turbo-fan" "Turbo-fan" ...
```

```
head(planes) # view the first few rows
```

```
## # A tibble: 6 x 9
##   tailnum  year type              manufacturer model engines seats speed engine
##   <chr>   <int> <chr>             <chr>        <chr>   <int> <int> <int> <chr>
## 1 N10156   2004 Fixed wing multi ~ EMBRAER      EMB-~       2    55    NA Turbo~
## 2 N102UW   1998 Fixed wing multi ~ AIRBUS INDU~ A320~       2   182    NA Turbo~
## 3 N103US   1999 Fixed wing multi ~ AIRBUS INDU~ A320~       2   182    NA Turbo~
## 4 N104UW   1999 Fixed wing multi ~ AIRBUS INDU~ A320~       2   182    NA Turbo~
## 5 N10575   2002 Fixed wing multi ~ EMBRAER      EMB-~       2    55    NA Turbo~
## 6 N105UW   1999 Fixed wing multi ~ AIRBUS INDU~ A320~       2   182    NA Turbo~
```

```
# a. Based on the planes data, group by manufacture and count the total using
#    one pipeline.
planes_summary <- planes %>%
  group_by(manufacturer) %>%
  summarise(Count = n())
planes_summary
```

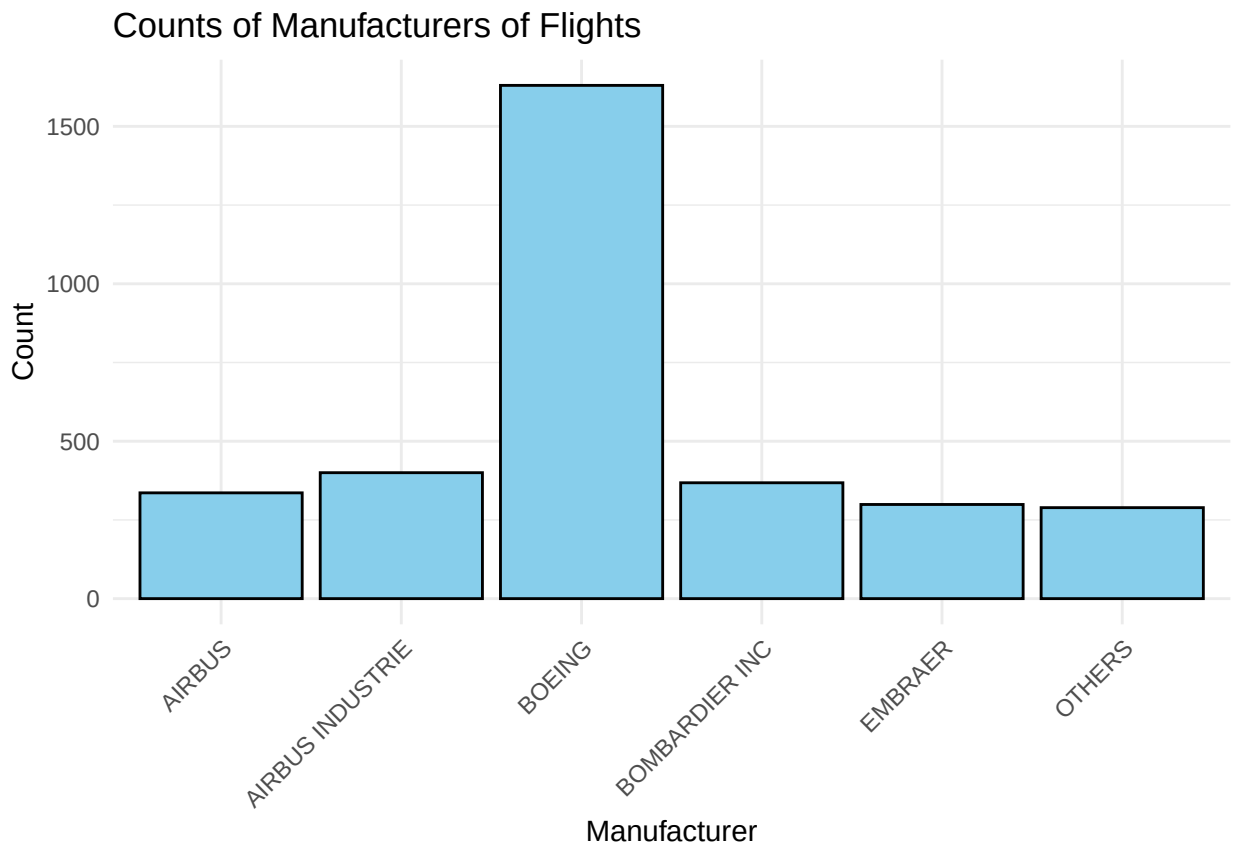
```
## # A tibble: 35 x 2
##    manufacturer           Count
##    <chr>                  <int>
##  1 AGUSTA SPA                 1
```

```
##  2 AIRBUS                   336
##  3 AIRBUS INDUSTRIE         400
##  4 AMERICAN AIRCRAFT INC      2
##  5 AVIAT AIRCRAFT INC         1
##  6 AVIONS MARCEL DASSAULT     1
##  7 BARKER JACK L              1
##  8 BEECH                      2
##  9 BELL                       2
## 10 BOEING                  1630
## # i 25 more rows
```

```r
# b. Start a new pipeline: generate a new column named "company" using mutate()
#    and the following is the rule:
#    If a manufacture is in the top five common manufactures, keep its
#    manufacture's name, if not, then name it "OTHERS".
#    (Hint: use ifelse( ... %in% c(....), manufacturer, OTHERS)
planes_modified <- planes %>%
  mutate(company = ifelse(manufacturer %in% names(head(sort(table(manufacturer),
         decreasing = TRUE), 5)), manufacturer, "OTHERS"))
planes_modified
```

```
## # A tibble: 3,322 x 10
##    tailnum  year type      manufacturer model engines seats speed engine company
##    <chr>   <int> <chr>     <chr>        <chr>   <int> <int> <int> <chr>  <chr>
##  1 N10156   2004 Fixed wi~ EMBRAER      EMB-~       2    55    NA Turbo~ EMBRAER
##  2 N102UW   1998 Fixed wi~ AIRBUS INDU~ A320~       2   182    NA Turbo~ AIRBUS~
##  3 N103US   1999 Fixed wi~ AIRBUS INDU~ A320~       2   182    NA Turbo~ AIRBUS~
##  4 N104UW   1999 Fixed wi~ AIRBUS INDU~ A320~       2   182    NA Turbo~ AIRBUS~
##  5 N10575   2002 Fixed wi~ EMBRAER      EMB-~       2    55    NA Turbo~ EMBRAER
##  6 N105UW   1999 Fixed wi~ AIRBUS INDU~ A320~       2   182    NA Turbo~ AIRBUS~
##  7 N107US   1999 Fixed wi~ AIRBUS INDU~ A320~       2   182    NA Turbo~ AIRBUS~
##  8 N108UW   1999 Fixed wi~ AIRBUS INDU~ A320~       2   182    NA Turbo~ AIRBUS~
##  9 N109UW   1999 Fixed wi~ AIRBUS INDU~ A320~       2   182    NA Turbo~ AIRBUS~
## 10 N110UW   1999 Fixed wi~ AIRBUS INDU~ A320~       2   182    NA Turbo~ AIRBUS~
## # i 3,312 more rows
```

```r
# c. Generate a bar graph to show the counts of manufactures of the flights
#    using the new variable generated      in b)
ggplot(planes_modified, aes(x = company)) +
  geom_bar(fill = "skyblue", color = "black") +
  labs(title = "Counts of Manufacturers of Flights",
       x = "Manufacturer", y = "Count") + theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```r
# 5. Create the following two tables first, and then merge the two tables:
#    StudentID, Gender, Age,
#    A,         Female, 21
#    B,         Male,   19
#    C,         Male,   20
#    D,         Female, 22
#    E,         Female, 20

#    StudentID, Midterm, Final
#    A,         78,      82
#    B,         97,      95
#    C,         81,      76
#    D,         93,      95
#    E,         82,      86

# Create the first table
table1 <- data.frame(StudentID = c("A", "B", "C", "D", "E"),
                     Gender = c("Female", "Male", "Male", "Female", "Female"),
                     Age = c(21, 19, 20, 22, 20))
table1
```

```
##   StudentID Gender Age
## 1         A Female  21
## 2         B   Male  19
## 3         C   Male  20
## 4         D Female  22
```

```
## 5         E Female  20
```

```r
# Create the second table
table2 <- data.frame(StudentID = c("A", "B", "C", "D", "E"),
                     Midterm = c(78, 97, 81, 93, 82),
                     Final = c(82, 95, 76, 95, 86))
table2
```

```
##   StudentID Midterm Final
## 1         A      78    82
## 2         B      97    95
## 3         C      81    76
## 4         D      93    95
## 5         E      82    86
```

```r
# Merge the two tables based on the StudentID column
merged_table <- merge(table1, table2, by = "StudentID")
merged_table
```

```
##   StudentID Gender Age Midterm Final
## 1         A Female  21      78    82
## 2         B   Male  19      97    95
## 3         C   Male  20      81    76
## 4         D Female  22      93    95
## 5         E Female  20      82    86
```