Lab 3

Christy Lei

Math 241, Week 4

```
# Put all necessary libraries here
library(tidyverse)
library(ggplot2)
library(ggthemes)
library(RColorBrewer)
library("ggpubr")
```

Due: Thursday, February 27th at 8:30am

Goals of this lab

1. Practice using GitHub.
2. Practice wrangling data.

Data Notes:

- For Problem 2, we will continue to dig into the SE Portland crash data but will use two datasets:
  - `CRASH`: crash level data
  - `PARTIC`: participant level data

```
# Crash level dataset
crash <- read_csv("/home/courses/math241s20/Data/pdx_crash_2018_CRASH.csv")

# Participant level dataset
partic <- read_csv("/home/courses/math241s20/Data/pdx_crash_2018_PARTIC.csv")
```

- For Problem 3, we will look at chronic illness data from the CDC along with the regional mapping for each state.

```
# CDC data
CDC <- read_csv("/home/courses/math241s20/Data/CDC2.csv")

# Regional data
USregions <- read_csv("/home/courses/math241s20/Data/USregions.csv")
```

- For Problem 4, we will use polling data from FiveThirtyEight.com.

```
# Note I only want us to focus on a subset of the variables
polls <- read_csv("/home/courses/math241s20/Data/generic_topline.csv") %>%
  select(subgroup, modeldate, dem_estimate, rep_estimate)
```

Problems

Problem 1: Git Control

In this problem, we will practice interacting with GitHub on the site directly and from the RStudio Server. Do this practice on **your repo**, not your group's Project 1 repo, so that the graders can check your progress with Git.

a. Let's practice creating and closing **Issues**. In a nutshell, **Issues** let us keep track of our work. Within your repo on GitHub.com, create an Issue entitled "Complete Lab 3". Once Lab 3 is done, close the **Issue**. (If you want to learn more about the functionalities of Issues, check out this page.)

b. Edit the ReadMe of your repo to include your name and a quick summary of the purpose of the repo. You can edit from within GitHub directly or on the server. If you edit on the server, make sure to push your changes to GitHub.

c. Upload both your Lab 3 .Rmd and .pdf to your repo on GitHub.

Problem 2: `dplyr` madness

Each part of this problem will require you to wrangle the data and then do one or both of the following:

- Display the wrangled data frame. To ensure it displays the whole data frame, you can pipe `as.data.frame()` at the end of the wrangling.
- Answer a question(s).

Some parts will require you to do a data join but won't tell you that.

a. Produce a table that provides the frequency of the different collision types, ordered from most to least common. What type is most common? What type is least common?

```
# Create a frequency table of the different collision types
crash %>%
  count(COLLIS_TYP_SHORT_DESC) %>%
  arrange(desc(n)) %>%
  as.data.frame()
```

```
##    COLLIS_TYP_SHORT_DESC   n
## 1                   REAR 671
## 2                   TURN 365
## 3                   ANGL 241
## 4                   SS-O  89
## 5                    PED  86
## 6                    FIX  51
## 7                   SS-M  17
## 8                   HEAD  16
## 9                   BACK  12
## 10                  PARK  10
## 11                  NCOL   6
## 12                   OTH   3
```

Rear-End (REAR) is the most common collision type (n=671) and Miscellaneous (OTHER) is the least common one (n=3).

b. For the three most common collision types, create a table that contains:
   - The frequencies of each collision type and weather condition combination.
   - The proportion of each collision type by weather condition.

Arrange the table by weather and within type, most to least common collision type.

```
# create a frequency and prop table for the three most collision types
# arranged by weather and frequency of types

crash %>%
  filter(COLLIS_TYP_SHORT_DESC %in% c("REAR","TURN","ANGL")) %>%
  group_by(WTHR_COND_SHORT_DESC) %>%
  count(COLLIS_TYP_SHORT_DESC, WTHR_COND_SHORT_DESC) %>%
  mutate(prop = prop.table(n)) %>%
  arrange(desc(n), .by_group = TRUE) %>%
  as.data.frame()
```

```
##    WTHR_COND_SHORT_DESC COLLIS_TYP_SHORT_DESC   n      prop
## 1                   CLD                  REAR  29 0.4677419
## 2                   CLD                  TURN  20 0.3225806
## 3                   CLD                  ANGL  13 0.2096774
## 4                   CLR                  REAR 549 0.5345667
## 5                   CLR                  TURN 290 0.2823759
## 6                   CLR                  ANGL 188 0.1830574
## 7                   FOG                  ANGL   2 0.6666667
## 8                   FOG                  TURN   1 0.3333333
## 9                  RAIN                  REAR  71 0.4965035
## 10                 RAIN                  TURN  44 0.3076923
## 11                 RAIN                  ANGL  28 0.1958042
## 12                  SLT                  ANGL   1 1.0000000
## 13                 SMOK                  REAR   1 1.0000000
## 14                 SNOW                  ANGL   3 0.5000000
## 15                 SNOW                  TURN   2 0.3333333
## 16                 SNOW                  REAR   1 0.1666667
## 17                  UNK                  REAR  20 0.5882353
## 18                  UNK                  TURN   8 0.2352941
## 19                  UNK                  ANGL   6 0.1764706
```

c. Create a column for whether or not a crash happened on a weekday or on the weekend and then create a data frame that explores if the distribution of collision types varies by whether or not the crash happened during the week or the weekend.

```
# create a "weekday_true" column that shows whether a crash happened on a weekday
weekday = c(1,2,3,4,5)
weekend = c(6,7)
crash$weekday_true =
  ifelse(crash$CRASH_WK_DAY_CD %in% weekday, "Yes",
  ifelse(crash$CRASH_WK_DAY_CD %in% weekend, "No", NA))

# create a data frame that contains collision types and the weekday/weekend info
# arranged by weekday info and frequency
crash_weekday <- crash %>%
  count(COLLIS_TYP_SHORT_DESC, weekday_true) %>%
  group_by(weekday_true) %>%
  mutate(prop = prop.table(n)) %>%
  arrange(desc(n), .by_group = TRUE) %>%
  as.data.frame()

crash_weekday
```

```
##    COLLIS_TYP_SHORT_DESC weekday_true   n        prop
## 1                   REAR           No 210 0.449678801
## 2                   TURN           No 109 0.233404711
## 3                   ANGL           No  69 0.147751606
## 4                   SS-O           No  29 0.062098501
## 5                    PED           No  25 0.053533191
## 6                    FIX           No  11 0.023554604
## 7                   BACK           No   4 0.008565310
## 8                   SS-M           No   4 0.008565310
## 9                   PARK           No   3 0.006423983
## 10                  HEAD           No   2 0.004282655
## 11                  NCOL           No   1 0.002141328
## 12                  REAR          Yes 461 0.419090909
## 13                  TURN          Yes 256 0.232727273
## 14                  ANGL          Yes 172 0.156363636
## 15                   PED          Yes  61 0.055454545
## 16                  SS-O          Yes  60 0.054545455
## 17                   FIX          Yes  40 0.036363636
## 18                  HEAD          Yes  14 0.012727273
## 19                  SS-M          Yes  13 0.011818182
## 20                  BACK          Yes   8 0.007272727
## 21                  PARK          Yes   7 0.006363636
## 22                  NCOL          Yes   5 0.004545455
## 23                   OTH          Yes   3 0.002727273
```

It seems that the most common collision types (REAR, TURN, ANGL) happened more often on weekdays.

d. First determine what proportion of crashes involve pedestrians. Then, for each driver license status, determine what proportion of crashes involve pedestrians. What driver license status has the highest rate of crashes that involve pedestrians?

```
#determine proportion of crashes that involves pedestrians
crash %>%
  count(pedestrians_involved = (CRASH_TYP_SHORT_DESC == "PED")) %>%
  mutate(prop = prop.table(n)) %>%
  as.data.frame()
```

```
##   pedestrians_involved    n       prop
## 1                FALSE 1481 0.94511806
## 2                 TRUE   86 0.05488194
```

```
#join the crash and partic datasets by "CRASH_ID"
crash_and_partic <- left_join (crash, partic,
                               by = c("CRASH_ID" = "CRASH_ID"))

#proportion of crashes that involves pedestrians for each driver license status
crash_and_partic %>%
  group_by(DRVR_LIC_STAT_SHORT_DESC) %>%
  count(pedestrians_involved = (CRASH_TYP_SHORT_DESC == "PED")) %>%
  mutate(prop = prop.table(n)) %>%
  filter(pedestrians_involved == TRUE) %>%
  arrange(desc(n)) %>%
  as.data.frame()
```

```
##   DRVR_LIC_STAT_SHORT_DESC pedestrians_involved  n        prop
## 1                    OR-Y                 TRUE 72 0.029008864
## 2                    SUSP                 TRUE  7 0.120689655
```

```
## 3                    <NA>                TRUE  7 0.008728180
## 4                  OTH-Y                 TRUE  6 0.019292605
## 5                    UNK                 TRUE  2 0.007692308
```

5.49% of all the crashs involves pedestrians. Valid Oregon license or permit (OR-Y/Type 1 license) has the highest rate of crashs (2.9%) that involves pedestrians.

e. Create a data frame that contains the age of drivers and collision type. (Don't print it.) Complete the following:
   - Find the average and median age of drivers.
   - Find the average and median age of drivers by collision type.
   - Create a graph of driver ages.
   - Create a graph of driver ages by collision type.

Draw some conclusions.

```
# create a data drame that contains the age of drivers and collision type
collision_type_age <- crash_and_partic %>%
  filter(PARTIC_TYP_SHORT_DESC == "DRVR") %>%
  select(COLLIS_TYP_SHORT_DESC, AGE_VAL) %>%
  as.data.frame()

# find the average and median age of drivers excluding unknown ages (00)
collision_type_age %>%
  filter(AGE_VAL != "00") %>%
  summarize(mean_age = mean(as.numeric(AGE_VAL)),
            median_age = median(as.numeric(AGE_VAL)))
```

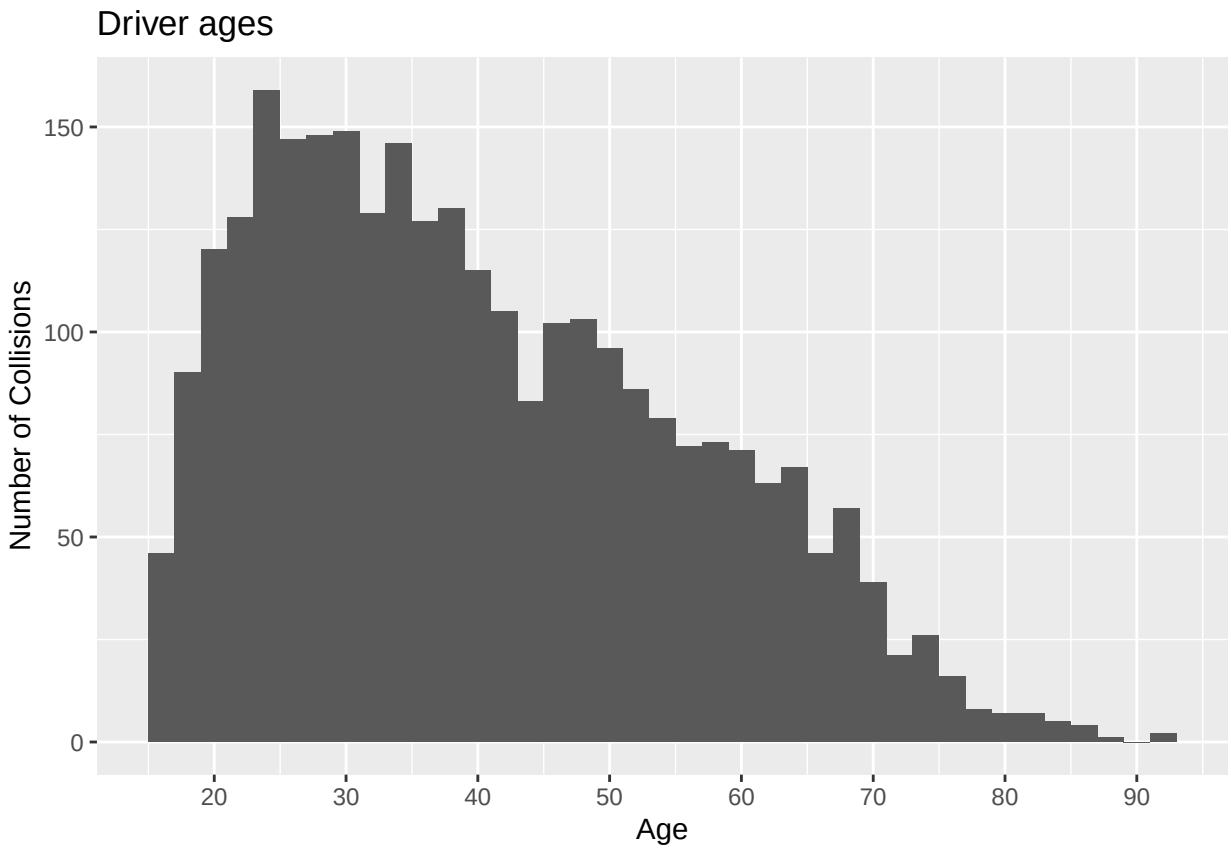
```
##   mean_age median_age
## 1 40.90184         38
```

```
# find the mean & median age of drivers by collision type without unknown ages
collision_type_age %>%
  filter(AGE_VAL != "00") %>%
  group_by(COLLIS_TYP_SHORT_DESC) %>%
  summarize(mean_age = mean(as.numeric(AGE_VAL)),
            median_age = median(as.numeric(AGE_VAL))) %>%
  arrange(desc(median_age))
```

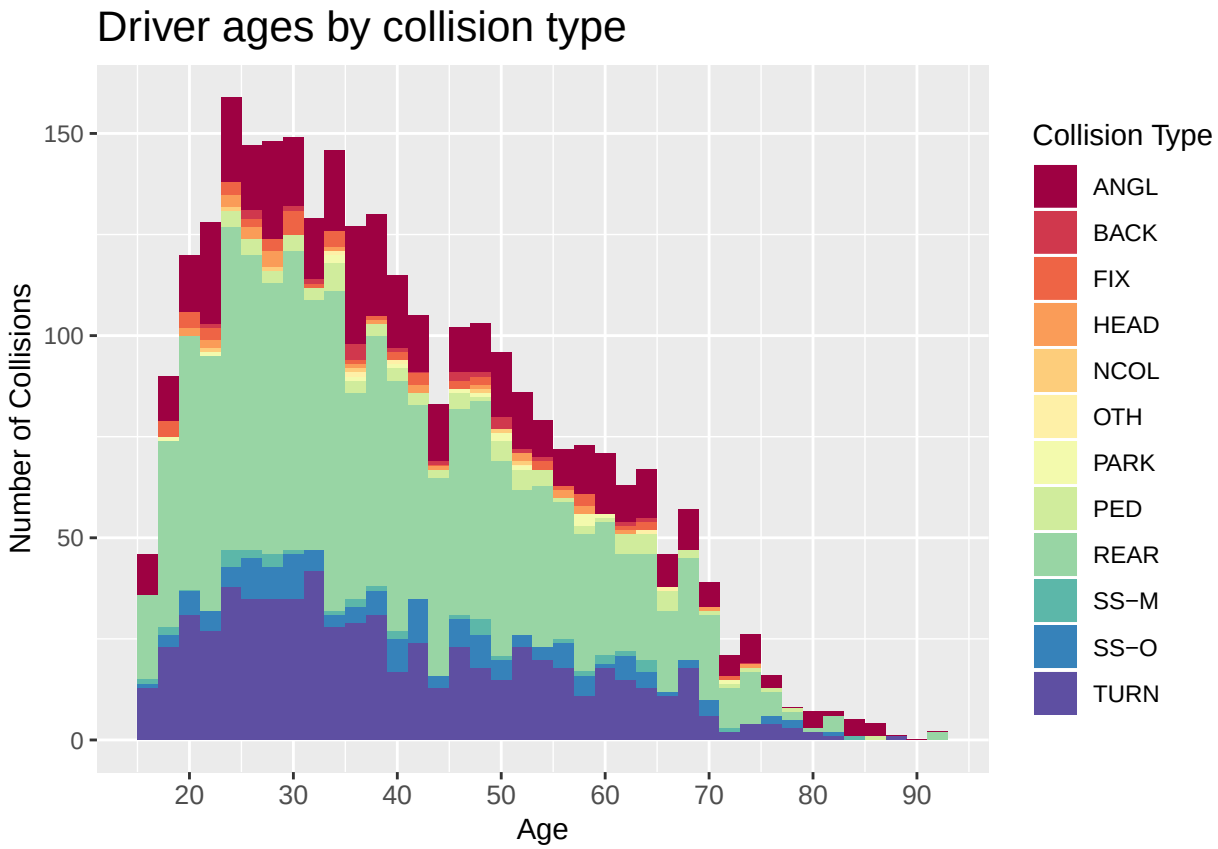
```
## # A tibble: 12 x 3
##    COLLIS_TYP_SHORT_DESC mean_age median_age
##    <chr>                    <dbl>      <dbl>
##  1 PARK                      46.4         50
##  2 PED                       48.0         47
##  3 BACK                      42.8         44
##  4 SS-O                      42.4         41
##  5 OTH                       48.6         40
##  6 SS-M                      42.5         40
##  7 ANGL                      42.5         39
##  8 REAR                      40.2         38
##  9 TURN                      40.0         37
## 10 HEAD                      39.4         36
## 11 NCOL                      37.4         36
## 12 FIX                       36.7         33
```

```
# create a graph of driver ages without unknown ages
collision_type_age %>%
  filter(AGE_VAL != "00") %>%
```

```
  ggplot(aes(x = as.numeric(AGE_VAL))) +
  geom_bar(binwidth = 2) +
  scale_x_continuous(breaks = c(0,10,20,30,40,50,60,70,80,90)) +
  labs(title = "Driver ages",
       x = "Age", y = "Number of Collisions")
```




```
# get a palette for the graph
colourCount = length(unique(collision_type_age$COLLIS_TYP_SHORT_DESC))
getPalette = colorRampPalette(brewer.pal(12, "Spectral"))

# create a graph of driver ages by collision type without unknown ages
collision_type_age %>%
  filter(AGE_VAL != "00") %>%
  ggplot(aes(x = as.numeric(AGE_VAL), fill = COLLIS_TYP_SHORT_DESC)) +
  geom_histogram(binwidth = 2) +
  scale_fill_manual(values = getPalette(colourCount)) +
  scale_x_continuous(breaks = c(0,10,20,30,40,50,60,70,80,90)) +
  theme(plot.title = element_text(size = 16)) +
  theme(axis.text.x = element_text(size = 10)) +
  labs(title = "Driver ages by collision type",
       x = "Age", y = "Number of Collisions", fill = "Collision Type")
```



- After excluding the unknown ages (00), the mean age of drivers is 41 and the median age is 38.
- After excluding the unknown ages (00), the average age of drivers seems to be the highest for collisions that involve parking maneuver (PARK) and pedestrians (PED), and the average age of drivers seems to be the lowest for types that involve non-collision (NCOL) and fixed-object (FIX).
- From the histogram that displays the driver ages, the distribution is right-skewed, which means that the drivers are relatively young (in their 20s, 30s, and 40s) in most incidents.
- From the histogram that displays the driver ages by collision type, we can see that for all age groups, the most common types of collisions involve rear-end (REAR), angle (ANGL), and turning movement (TURN).

Problem 3: Chronically Messy Data

a. Turning to the CDC data, let's get a handle of what is represented there. For 2016 (use `YearStart`), how many distinct topics were tracked?

```
CDC %>%
  filter(YearStart == 2016) %>%
  count(Topic) %>%
  nrow()
```

```
## [1] 16
```

16 distinct topics were tracked.

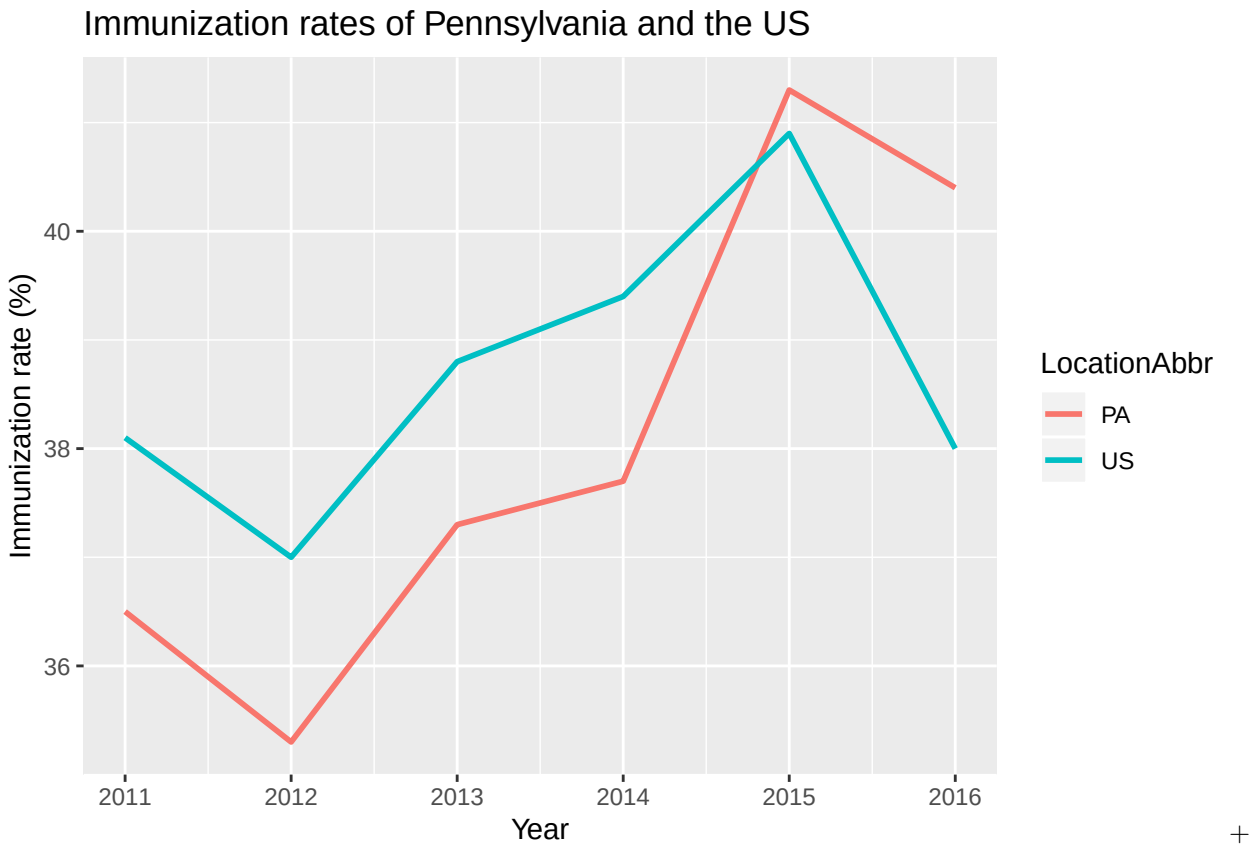
b. Let's study influenza vaccination patterns! Create a dataset that contains the age adjusted prevalence of the "Influenza vaccination among noninstitutionalized adults aged >= 18 years" for Oregon and the US from 2010 to 2016.

```r
# create a data frame that contains the age adjusted prevalence
# of the specified question from 2010 to 2016
influenza_vacc_prevalance <- CDC %>%
  filter(YearStart %in% c(2010, 2011, 2012, 2013, 2014, 2015, 2016),
         Question == "Influenza vaccination among noninstitutionalized adults aged >= 18 years",
         DataValueType == "Age-adjusted Prevalence") %>%
  select(YearStart, LocationAbbr, DataValue, DataValueType) %>%
  arrange(YearStart) %>%
  pivot_wider(names_from = DataValueType,
              values_from = DataValue) %>%
  as.data.frame()

# create a data frame that contains the age adjusted prevalence
# of the specified question from 2010 to 2016 for Oregon and the US
influenza_vacc_oregon_us <- influenza_vacc_prevalance %>%
  filter(LocationAbbr%in% c("OR", "US"))

influenza_vacc_oregon_us
```

```
##    YearStart LocationAbbr Age-adjusted Prevalence
## 1       2011           OR                    33.0
## 2       2011           US                    38.1
## 3       2012           OR                    33.0
## 4       2012           US                    37.0
## 5       2013           OR                    33.6
## 6       2013           US                    38.8
## 7       2014           OR                    35.4
## 8       2014           US                    39.4
## 9       2015           OR                    38.2
## 10      2015           US                    40.9
## 11      2016           US                    38.0
## 12      2016           OR                    34.4
```

c. Create a graph comparing the immunization rates of Pennsylvania and the US. Comment on the observed trends in your graph

```r
# create a graph of immunization rates of PA and the US from 2010 to 2016
influenza_vacc_prevalance %>%
  filter(LocationAbbr %in% c("PA", "US")) %>%
  ggplot(aes(x = YearStart, y = `Age-adjusted Prevalence`,
             color = LocationAbbr)) +
  geom_line(size = 1) +
  labs(x = "Year", y = "Immunization rate (%)",
       title = "Immunization rates of Pennsylvania and the US")
```



For both Pennsylvania and the US, there is a steady increase in immunization rates from 2011 to 2015 (with the exception of a slight drop in 2012), followed by a decrease in immunization rates in 2016. In general, the immunization rate in Pennslyvania is slightly lower than that in the US with the exception of year 2015 and 2016.

d. Let's see how immunization rates vary by region of the country. Join the regional dataset to our CDC dataset so that we have a column signifying the region of the country.

```
# join the regional dataset to CDC dataset
CDC <- left_join (CDC, USregions, by = c("LocationDesc" = "State"))
```

e. Why are there NAs in the region column of the new dataset?

- A left join in question (d) returns all rows from CDC and all columns from CDC and USregions. However, some location names used in the CDC dataset (55 locations under `LocationDesc`) are not present in the USRegions dataset (50 locations under `State`), so this creates NAs for the variable `Region` in the new dataset.

f. Create a dataset that contains the age adjusted influenza immunization rates in 2016 for each state in the country and sort it by highest immunization to lowest. Which state has the highest immunization?

```
# create a data frame that contains the age adjusted immunization rates
# in 2016 for each state

immunization_by_state <- CDC %>%
  filter(YearStart == 2016,
         Question == "Influenza vaccination among noninstitutionalized adults aged >= 18 years",
         DataValueType == "Age-adjusted Prevalence") %>%
  select(YearStart, LocationAbbr, DataValue, DataValueType, Region) %>%
  pivot_wider(names_from = DataValueType,
```

```
              values_from = DataValue) %>%
  arrange(desc(`Age-adjusted Prevalence`)) %>%
  as.data.frame()

immunization_by_state
```

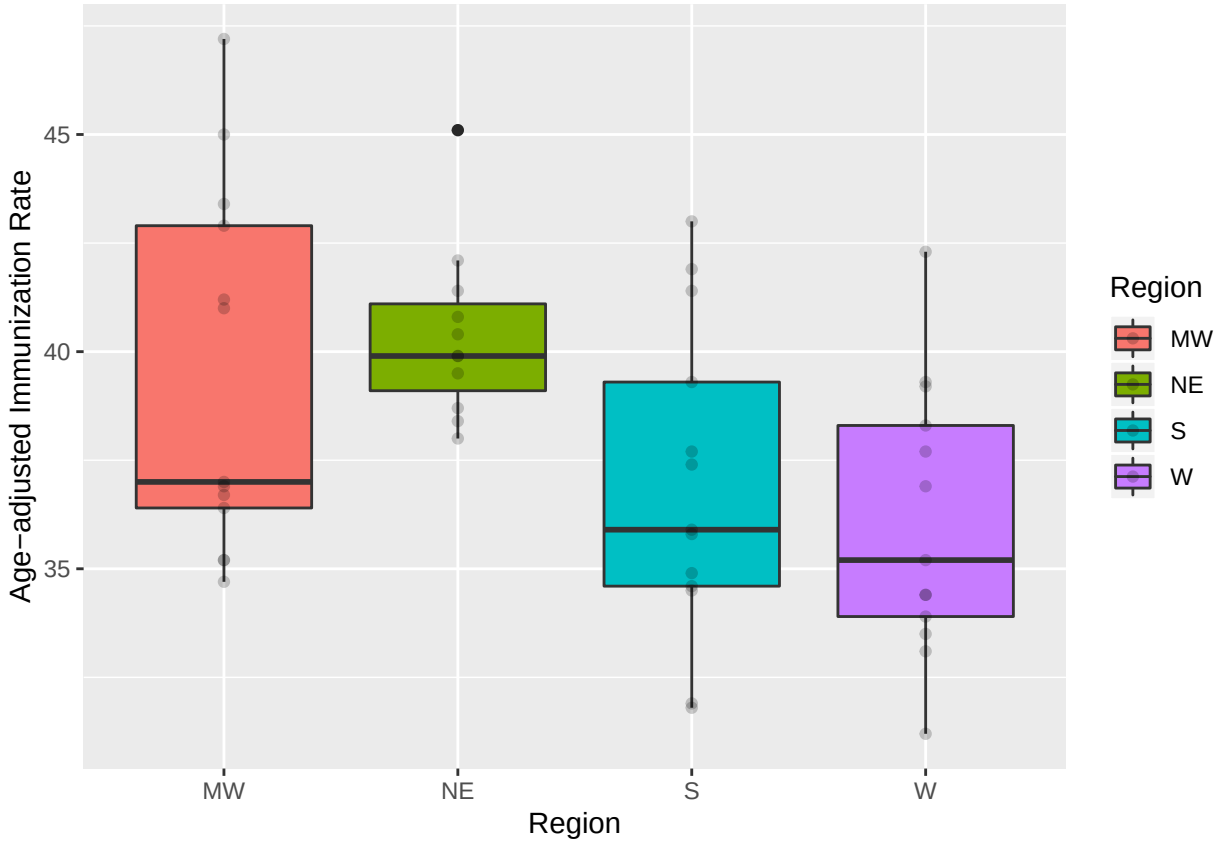
```
##    YearStart LocationAbbr Region Age-adjusted Prevalence
## 1       2016           SD     MW                    47.2
## 2       2016           RI     NE                    45.1
## 3       2016           IA     MW                    45.0
## 4       2016           NE     MW                    43.4
## 5       2016           NC      S                    43.0
## 6       2016           MN     MW                    42.9
## 7       2016           CO      W                    42.3
## 8       2016           MD     NE                    42.1
## 9       2016           VA      S                    41.9
## 10      2016           CT     NE                    41.4
## 11      2016           WV      S                    41.4
## 12      2016           ND     MW                    41.2
## 13      2016           MO     MW                    41.0
## 14      2016           MA     NE                    40.8
## 15      2016           PA     NE                    40.4
## 16      2016           NH     NE                    39.9
## 17      2016           VT     NE                    39.9
## 18      2016           DE     NE                    39.5
## 19      2016           OK      S                    39.3
## 20      2016           WA      W                    39.3
## 21      2016           NM      W                    39.2
## 22      2016           NY     NE                    38.7
## 23      2016           ME     NE                    38.4
## 24      2016           HI      W                    38.3
## 25      2016           US   <NA>                    38.0
## 26      2016           DC   <NA>                    38.0
## 27      2016           NJ     NE                    38.0
## 28      2016           AR      S                    37.7
## 29      2016           MT      W                    37.7
## 30      2016           TX      S                    37.4
## 31      2016           OH     MW                    37.0
## 32      2016           KY     MW                    36.9
## 33      2016           UT      W                    36.9
## 34      2016           KS     MW                    36.7
## 35      2016           IN     MW                    36.4
## 36      2016           MS      S                    35.9
## 37      2016           SC      S                    35.8
## 38      2016           CA      W                    35.2
## 39      2016           IL     MW                    35.2
## 40      2016           WI     MW                    35.2
## 41      2016           AL      S                    34.9
## 42      2016           MI     MW                    34.7
## 43      2016           GA      S                    34.6
## 44      2016           TN      S                    34.5
## 45      2016           AK      W                    34.4
## 46      2016           OR      W                    34.4
## 47      2016           AZ      W                    33.9
```

```
## 48       2016           WY      W                    33.5
## 49       2016           ID      W                    33.1
## 50       2016           LA      S                    31.9
## 51       2016           FL      S                    31.8
## 52       2016           NV      W                    31.2
## 53       2016           GU   <NA>                    30.9
## 54       2016           PR   <NA>                    23.9
## 55       2016           VI   <NA>                    14.9
```

South Dakota has the highest immunization rate in 2016.

g. Construct a graphic of the 2016 influenza immunization rates by region of the country. Don't include locations without a region. Comment on your graphic.

```
# create a graph of the immunization rates in 2016 by region
immunization_by_state %>%
  drop_na() %>%
  ggplot(aes(x = Region, y = `Age-adjusted Prevalence`, fill = Region)) +
  geom_boxplot() + geom_point(width = 1, alpha = 0.2) +
  labs(y = "Age-adjusted Immunization Rate")
```



In 2016, the Northeast (NE) has the highest average of the influenza immunization rates compared to the other regions, and the Midwest (MW) has the largest IQR which means the immunization rates for the states in MW are more variable. The South (S) and the West (W) have a similar pattern of influenza immunization prevalence in terms of the average (i.e. median) and variability (i.e. IQR).

Problem 4: Tidying Data Like a Boss

I was amazed by the fact that many of the FiveThirtyEight datasets are actually not in a perfectly *tidy* format. Let's tidy up this dataset related to polling.

a. Why is this data not currently in a tidy format?

- The rows contain mutiple observations (e.g. for each date, "All polls" also include data from "Voters" and "Adults").

```
polls
```

```
## # A tibble: 1,529 x 4
##    subgroup  modeldate dem_estimate rep_estimate
##    <chr>     <chr>            <dbl>        <dbl>
##  1 All polls 9/18/2018         48.8         39.8
##  2 All polls 9/17/2018         49.0         39.9
##  3 All polls 9/16/2018         49.0         39.9
##  4 All polls 9/15/2018         49.0         39.9
##  5 All polls 9/14/2018         48.9         39.8
##  6 All polls 9/13/2018         48.8         39.7
##  7 All polls 9/12/2018         48.8         39.6
##  8 All polls 9/11/2018         48.5         39.9
##  9 All polls 9/10/2018         48.4         39.9
## 10 All polls 9/9/2018          48.4         39.9
## # ... with 1,519 more rows
```

b. Create a tidy dataset of the `All polls` subgroup.

```
tidy_poll <- polls %>%
  pivot_wider(names_from = subgroup,
              values_from = c(dem_estimate, rep_estimate)) %>%
  as.data.frame()

# display the first few rows for the new dataset
head(tidy_poll,3)
```

```
##   modeldate dem_estimate_All polls dem_estimate_Voters dem_estimate_Adults
## 1 9/18/2018               48.83349                  NA                  NA
## 2 9/17/2018               48.99541                  NA                  NA
## 3 9/16/2018               49.00679                  NA                  NA
##   rep_estimate_All polls rep_estimate_Voters rep_estimate_Adults
## 1               39.82040                  NA                  NA
## 2               39.89773                  NA                  NA
## 3               39.90728                  NA                  NA
```

c. Now let's create a new untidy version of `polls`. Focusing just on the estimates for democrats, create a data frame where each row represents a subgroup (given in column 1) and the rest of the columns are the estimates for democrats by date.

```
untidy_poll <- polls %>%
  select(!rep_estimate) %>%
  pivot_wider(names_from = modeldate,
              values_from = dem_estimate) %>%
  as.data.frame()

# display the first few columns for the new dataset
untidy_poll[0:7]
```

```
##    subgroup 9/18/2018 9/17/2018 9/16/2018 9/15/2018 9/14/2018 9/13/2018
## 1 All polls  48.83349  48.99541  49.00679   48.9885   48.9038  48.81726
## 2    Voters        NA        NA        NA        NA        NA        NA
## 3    Adults        NA        NA        NA        NA        NA        NA
```

d. Why might someone want to transform the data like we did in part c?

- They want to scroll horizontally to see if there's a general trend of increasing or decreasing number of estimates over time for a particular subgroup
- They are interested in comparing the estimates of votes from the three subgroups on a particular day (e.g. on July 4th)
- They can quickly see which subgroup has a "NA" value for each date

Problem 5: YOUR TURN!

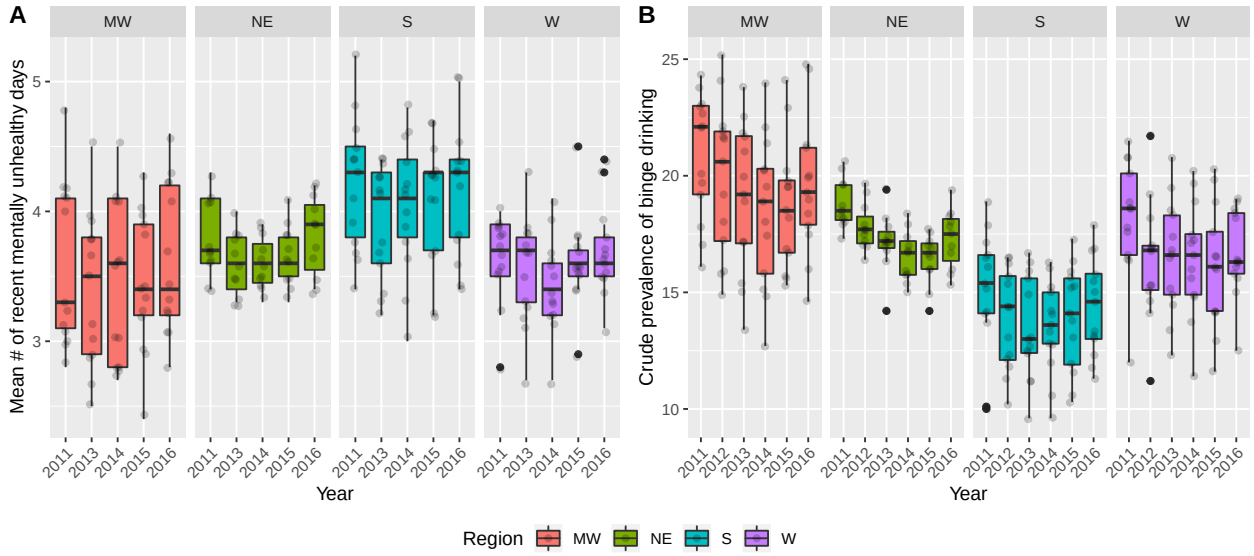
Now it is your turn. Pick one (or multiple) of the datasets used on this lab. Ask a question of the data. Do some data wrangling to produce statistics (use at least two wrangling verbs) and a graphic to answer the question. Then comment on any conclusions you can draw about your question.

- For the CDC dataset: I want to explore more about the data under the topic "Mental Health" and "Alcohol", especially the relationship between the question "Recent mentally unhealthy days among adults aged >= 18 years" and "Binge drinking prevalence among adults aged >= 18 years".

  I'm curious to know if the trend of the average number of mentally unhealthy days is similar to the crude prevalance of binge drinking over the years, and how that differs in each region.

```
# create a data frame that has the mean of the recently mentally unhealthy days
mental_health_region <- CDC %>%
  filter(Question == "Recent mentally unhealthy days among adults aged >= 18 years",
         DataValueType == "Mean") %>%
  select(YearStart, DataValue, DataValueType, Region, LocationAbbr) %>%
  pivot_wider(names_from = DataValueType,
              values_from = DataValue) %>%
  as.data.frame()

# create a data frame that has the crude prevalence of binge drinking prevalence
binge_drinking_region <- CDC %>%
  filter(YearStart %in% c(2010,2011,2012,2013,2014,2015,2016),
         Question %in% c("Binge drinking prevalence among adults aged >= 18 years"),
         DataValueType == "Crude Prevalence") %>%
  pivot_wider(names_from = DataValueType,
              values_from = DataValue) %>%
  select(YearStart, Topic, `Crude Prevalence`, Region, LocationAbbr) %>%
  drop_na() %>%
  as.data.frame()


# create a graph that displays the trend over the years by region for
# mentally unhealthy days
mental_health_region <- mental_health_region %>%
  drop_na() %>%
  ggplot(aes(x = factor(YearStart), y = Mean, fill = Region)) +
  geom_boxplot() +
  geom_jitter(width = 0.1, alpha = 0.2) +
  xlab("Year") +
  ylab("Mean # of recent mentally unhealthy days") +
```

```
  facet_wrap(~Region, ncol = 4) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# create a graph that displays the trend over the years by region for binge drinking
binge_drinking_region <- binge_drinking_region %>%
  drop_na() %>%
  ggplot(aes(x = factor(YearStart), y = `Crude Prevalence`, fill = Region)) +
  geom_boxplot() +
  geom_jitter(width = 0.1, alpha = 0.2) +
  xlab("Year") +
  ylab("Crude prevalence of binge drinking") +
  facet_wrap(~Region, ncol = 4) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# combine the two graphs together for comparison
ggarrange(mental_health_region, binge_drinking_region,
          labels = c("A", "B"), common.legend = TRUE, legend = "bottom")
```



- **Mentally unhealthy days**: adults who live in the South (S) consistanly experience more mentally unhealthy days compared to those in the other regions. For all regions, it seems that the mean number of mentally unhealthy days stayed relatively stable throughout the years.

  It's also worthy to note that the IQRs are the largest for the states in the Midwest (MW), which means that there's a higher variability in the mean number of mentally unhealthy days among adults who are from different states in the MW.

- **Binge drinking prevalence**: the binge drinking prevalence rate for adults who live in the South (S) is the lowest among the four regions, whereas binge drinking is the most prevalent in the Midwest (MW). For all regions, it seems that the binge drinking prevalence is decreasing throughout the years.

  It's also worthy to note that the IQRs are the smallest for the states in the Northeast (NE), which means that there's a low variability in the binge drinking prevalence among adults who are from different states in the NE.

- **Compare the two questions**: it's interesting that the South (S) has the lowest binge drinking prevalence but the highest mean number of mentally unhealthy days, while the Midwest (MW) has the highest binge drinking prevalence but the lowest mean number of mentally unhealthy days. This is contrary to my expectation that more binge drinking is positively related to more mental health

problems.

Also, I noticed that in both graphs the IQRs are the largest for the MW. This might indicate that the between-states difference is the largest in the Midwest compared to that in other regions for other measures as well.