Lab 3

Math 241, Week 4

```r
# Put all necessary libraries here
library(tidyverse)
library(reprex)
library(dplyr)
library(infer)
library(moderndive)
library(lubridate)
library(ggthemes)
library(ggplot2)
```

Due: Thursday, February 25th at ~~8:30am~~ 6:00pm

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

```r
# Crash level dataset
crash <- read_csv("/home/courses/math241s21/Data/pdx_crash_2018_CRASH.csv")

# Participant level dataset
partic <- read_csv("/home/courses/math241s21/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.

```r
# CDC data
CDC <- read_csv("/home/courses/math241s21/Data/CDC2.csv")

# Regional data
USregions <- read_csv("/home/courses/math241s21/Data/USregions.csv")
```

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

```r
# Note I only want us to focus on a subset of the variables
polls <- read_csv("/home/courses/math241s21/Data/generic_topline.csv") %>%
  select(subgroup, modeldate, dem_estimate, rep_estimate)
```

- For Problem 6, we will use several datasets that came from `pdxTrees` but good messed up a bit:

```r
# Data on trees in a few parks in Portland
treez <- read_csv("/home/courses/math241s21/Data/treez.csv")
```

```r
treez_loc <- read_csv("/home/courses/math241s21/Data/treez_loc.csv")
treez_park <- read_csv("/home/courses/math241s21/Data/treez_park.csv")
```

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 labwork_username 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 data frame 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?

```r
#CRASH_TYP_CD
crash_freq <- crash %>%
  count(COLLIS_TYP_SHORT_DESC) %>%
  arrange(desc(n))
crash_freq
```

```
## # A tibble: 12 x 2
##    COLLIS_TYP_SHORT_DESC     n
##    <chr>                 <int>
##  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
```

Most common are rear, turn, and angle collisions. Least common are head, back, and parking collisions.

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.

```
crashweather <- crash %>%
  select(COLLIS_TYP_SHORT_DESC, WTHR_COND_SHORT_DESC) %>%
  filter(COLLIS_TYP_SHORT_DESC %in% c("REAR", "TURN", "ANGL")) %>%
  count(COLLIS_TYP_SHORT_DESC, WTHR_COND_SHORT_DESC)

propcrash <- crashweather %>%
  group_by(WTHR_COND_SHORT_DESC) %>%
  mutate(propcrfreqweather = n/sum(n)) %>%
  arrange(WTHR_COND_SHORT_DESC, desc(propcrfreqweather))

propcrash
```

```
## # A tibble: 19 x 4
## # Groups:   WTHR_COND_SHORT_DESC [8]
##    COLLIS_TYP_SHORT_DESC WTHR_COND_SHORT_DESC     n propcrfreqweather
##    <chr>                 <chr>                <int>             <dbl>
##  1 REAR                  CLD                     29             0.468
##  2 TURN                  CLD                     20             0.323
##  3 ANGL                  CLD                     13             0.210
##  4 REAR                  CLR                    549             0.535
##  5 TURN                  CLR                    290             0.282
##  6 ANGL                  CLR                    188             0.183
##  7 ANGL                  FOG                      2             0.667
##  8 TURN                  FOG                      1             0.333
##  9 REAR                  RAIN                    71             0.497
## 10 TURN                  RAIN                    44             0.308
## 11 ANGL                  RAIN                    28             0.196
## 12 ANGL                  SLT                      1             1
## 13 REAR                  SMOK                     1             1
## 14 ANGL                  SNOW                     3             0.5
## 15 TURN                  SNOW                     2             0.333
## 16 REAR                  SNOW                     1             0.167
## 17 REAR                  UNK                     20             0.588
## 18 TURN                  UNK                      8             0.235
## 19 ANGL                  UNK                      6             0.176
```

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.

```
weekendcrash <- crash %>%
  select(COLLIS_TYP_SHORT_DESC, CRASH_WK_DAY_CD) %>%
  mutate(date = case_when(
    CRASH_WK_DAY_CD %in% c(1,7) ~ "weekend",
     CRASH_WK_DAY_CD %in% c(2:6) ~ "weekday"))

weekend_dist <- count(weekendcrash, date, COLLIS_TYP_SHORT_DESC, sort = TRUE) %>%
  group_by(date) %>%
  mutate(prop = n/sum(n)) %>%
  arrange(COLLIS_TYP_SHORT_DESC, desc(n)) %>%
  as.data.frame()
```

```r
weekend_dist
```

```
##       date COLLIS_TYP_SHORT_DESC   n        prop
## 1  weekday                  ANGL 180 0.152027027
## 2  weekend                  ANGL  61 0.159268930
## 3  weekday                  BACK  10 0.008445946
## 4  weekend                  BACK   2 0.005221932
## 5  weekday                   FIX  31 0.026182432
## 6  weekend                   FIX  20 0.052219321
## 7  weekday                  HEAD  11 0.009290541
## 8  weekend                  HEAD   5 0.013054830
## 9  weekday                  NCOL   5 0.004222973
## 10 weekend                  NCOL   1 0.002610966
## 11 weekday                   OTH   3 0.002533784
## 12 weekday                  PARK   8 0.006756757
## 13 weekend                  PARK   2 0.005221932
## 14 weekday                   PED  67 0.056587838
## 15 weekend                   PED  19 0.049608355
## 16 weekday                  REAR 510 0.430743243
## 17 weekend                  REAR 161 0.420365535
## 18 weekday                  SS-M  11 0.009290541
## 19 weekend                  SS-M   6 0.015665796
## 20 weekday                  SS-O  67 0.056587838
## 21 weekend                  SS-O  22 0.057441253
## 22 weekday                  TURN 281 0.237331081
## 23 weekend                  TURN  84 0.219321149
```

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?

```r
ped <- crash %>%
  select(COLLIS_TYP_SHORT_DESC, CRASH_ID)

drive <- partic %>%
  select(DRVR_LIC_STAT_SHORT_DESC, CRASH_ID)

crash_ped <- full_join(ped, drive, by = c("CRASH_ID" = "CRASH_ID")) %>%
  distinct(CRASH_ID, .keep_all = TRUE) %>%
  mutate(pedest = case_when(COLLIS_TYP_SHORT_DESC == "PED" ~ "Involved",
                            COLLIS_TYP_SHORT_DESC != "PED" ~ "Not involved"))

crash_ped_overlap <- full_join(ped, drive, by = c("CRASH_ID" = "CRASH_ID")) %>%
  mutate(pedest = case_when(
    COLLIS_TYP_SHORT_DESC == "PED" ~ "Involved",
    COLLIS_TYP_SHORT_DESC != "PED" ~ "Not involved")
    )
crash_ped %>%
    group_by(pedest) %>%
    summarize(count = n()) %>%
    mutate(proportion = count/sum(count))
```

```
## # A tibble: 2 x 3
##   pedest       count proportion
```

```
## * <chr>        <int>      <dbl>
## 1 Involved        86     0.0549
## 2 Not involved  1481     0.945
```

```
crash_ped %>%
  group_by(DRVR_LIC_STAT_SHORT_DESC, pedest)%>%
  distinct(CRASH_ID) %>%
  summarize(count = n()) %>%
  mutate(prop = count/sum(count)) %>%
  arrange(pedest, desc(prop)) %>%
  as.data.frame
```

```
##    DRVR_LIC_STAT_SHORT_DESC       pedest count        prop
## 1                      SUSP     Involved     7 0.148936170
## 2                      OR-Y     Involved    71 0.062720848
## 3                     OTH-Y     Involved     6 0.036144578
## 4                       UNK     Involved     2 0.009852217
## 5                     N-VAL Not involved     3 1.000000000
## 6                      NONE Not involved    16 1.000000000
## 7                       UNK Not involved   201 0.990147783
## 8                     OTH-Y Not involved   160 0.963855422
## 9                      OR-Y Not involved  1061 0.937279152
## 10                     SUSP Not involved    40 0.851063830
```

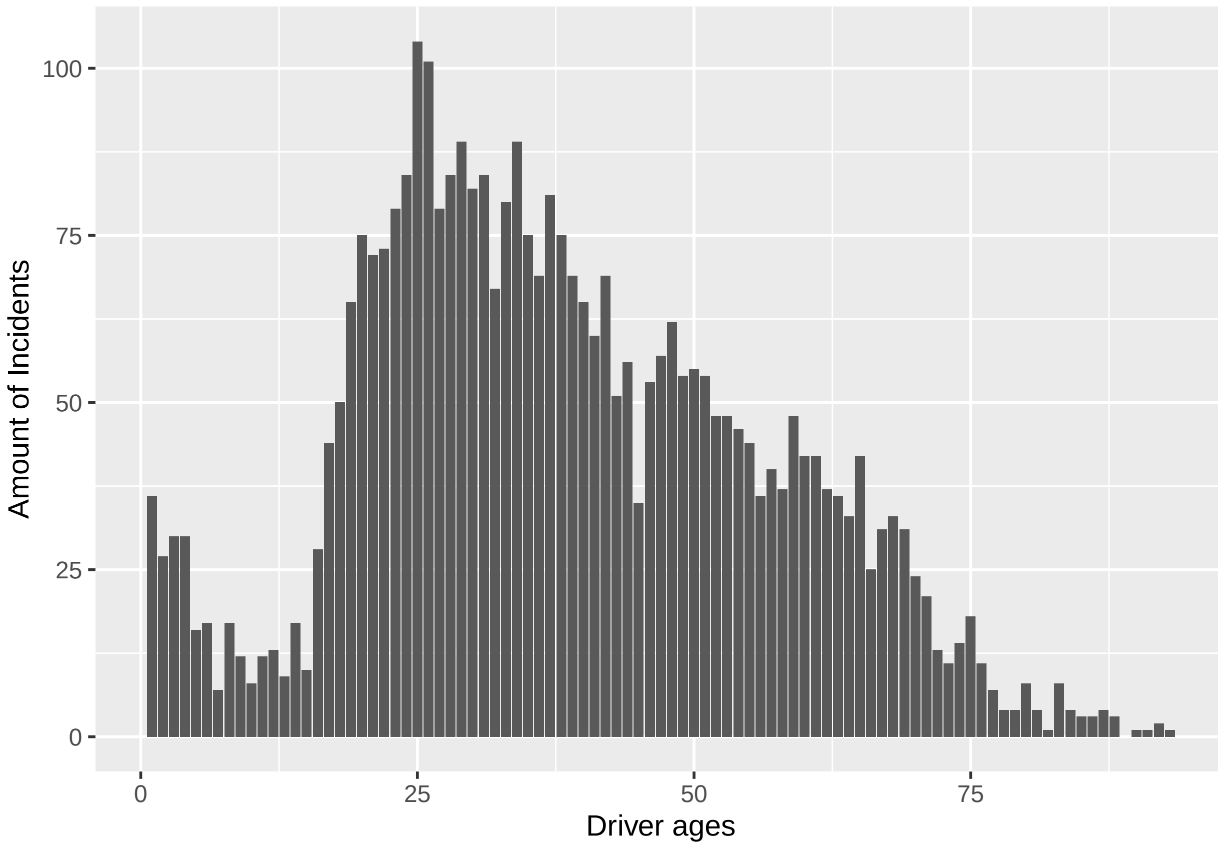
Drivers with a license type ORY have the highest rates of crashes involving pedestrians.

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.

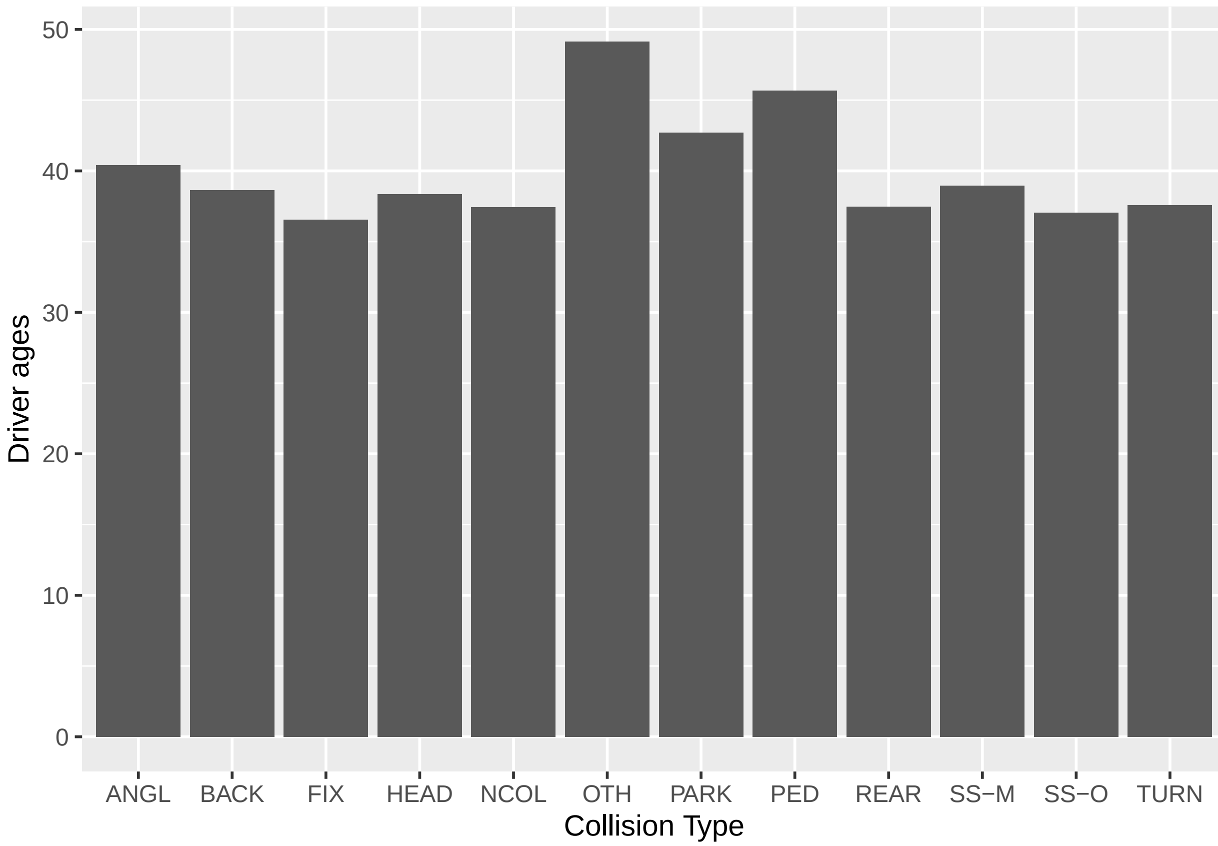
```
mergedcrash <- full_join(partic, crash, by = c("CRASH_ID" = "CRASH_ID"))

mergedcrash %>%
  mutate(AGE_VAL = as.numeric(AGE_VAL)) %>%
  filter(AGE_VAL != 0) %>%
  summarize(avgage = mean(AGE_VAL, na.rm = T),
            median = median(AGE_VAL, na.rm = T))
```

```
## # A tibble: 1 x 2
##   avgage median
##    <dbl>  <dbl>
## 1   38.2     36
```

```
mergedcrash %>%
  mutate(AGE_VAL = as.numeric(AGE_VAL)) %>%
  filter(AGE_VAL != 0) %>%
  ggplot(aes(x = AGE_VAL)) + geom_bar() +
  labs(x = "Driver ages",
       y = "Amount of Incidents")
```




```
mergedcrash %>%
  mutate(AGE_VAL = as.numeric(AGE_VAL)) %>%
  filter(AGE_VAL != 0) %>%
  group_by(COLLIS_TYP_SHORT_DESC) %>%
  summarize(avgage = mean(AGE_VAL, na.rm = T),
            median = median(AGE_VAL, na.rm = T)) %>%
  ggplot(aes(x = COLLIS_TYP_SHORT_DESC, y = avgage )) +
  geom_col() +
  labs(x = "Collision Type",
       y = "Driver ages")
```



Draw some conclusions.

Mean age is 38.1 median is 36 for all drivers. The first graph shows a cluster around 25 and shows that there is a wide range in age of drivers. The second graph shows the average age of people involved in specific types of collisions.

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?

```
unique(CDC$Topic)
```

```
##  [1] "Alcohol"
##  [2] "Arthritis"
##  [3] "Asthma"
##  [4] "Cancer"
##  [5] "Diabetes"
##  [6] "Chronic Obstructive Pulmonary Disease"
##  [7] "Mental Health"
##  [8] "Oral Health"
##  [9] "Cardiovascular Disease"
## [10] "Immunization"
## [11] "Chronic Kidney Disease"
## [12] "Nutrition, Physical Activity, and Weight Status"
## [13] "Older Adults"
## [14] "Tobacco"
## [15] "Overarching Conditions"
```

```
## [16] "Reproductive Health"
## [17] "Disability"
```

17 distinct topics were tracked

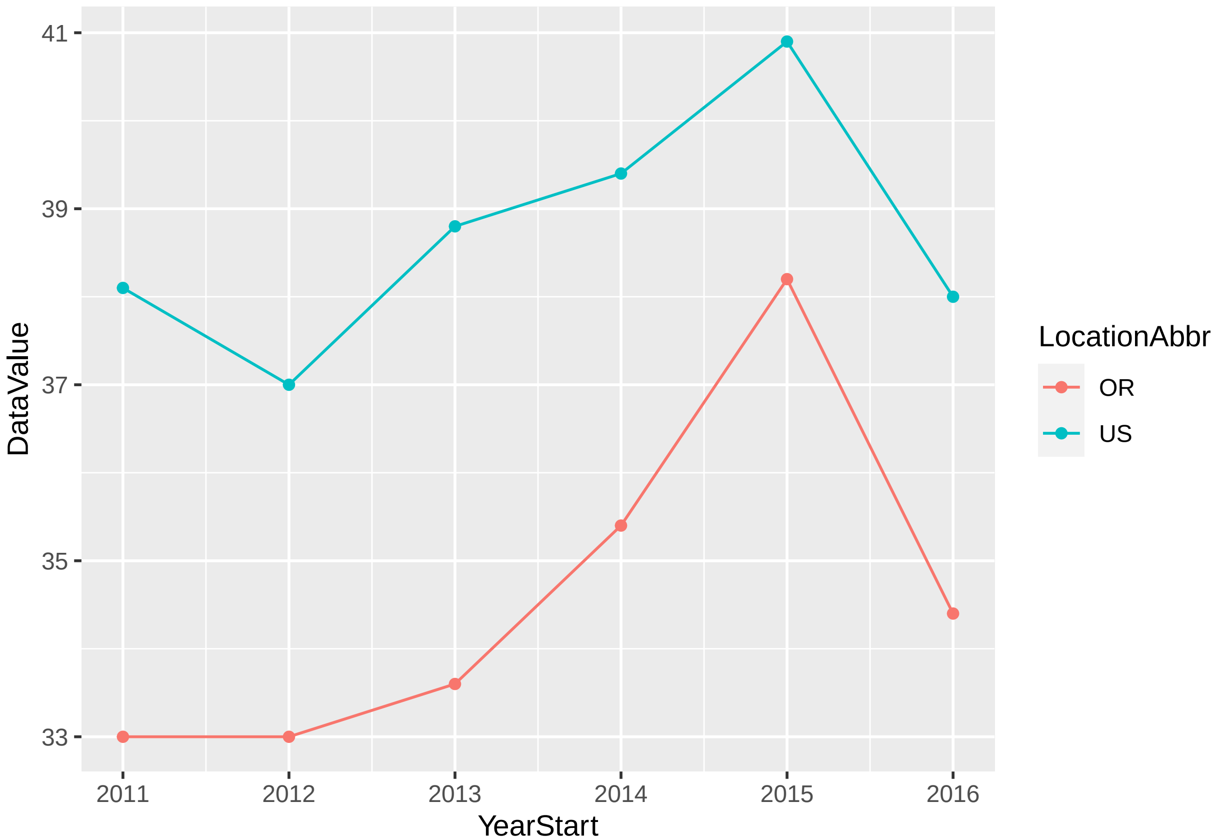
b. Let's study influenza vaccination patterns! Create a dataset that contains the age adjusted prevalence of the "Influenza vaccination among noninstitutionalized adults aged >= 18 years" for Oregon and the US from 2010 to 2016.

```
flupat <- CDC %>%
  select(Question, DataValueType, YearStart, LocationAbbr, Topic, DataValue) %>%
  filter(Question == "Influenza vaccination among noninstitutionalized adults aged >= 18 years",
         DataValueType == "Age-adjusted Prevalence",
         YearStart %in% c(2010: 2016),
         LocationAbbr %in% c("OR", "US"),
         Topic == "Immunization"
         )
flupat
```

```
## # A tibble: 12 x 6
##    Question               DataValueType   YearStart LocationAbbr Topic  DataValue
##    <chr>                  <chr>               <dbl> <chr>        <chr>      <dbl>
##  1 Influenza vaccinatio~ Age-adjusted P~      2016 US           Immun~      38
##  2 Influenza vaccinatio~ Age-adjusted P~      2016 OR           Immun~      34.4
##  3 Influenza vaccinatio~ Age-adjusted P~      2015 OR           Immun~      38.2
##  4 Influenza vaccinatio~ Age-adjusted P~      2015 US           Immun~      40.9
##  5 Influenza vaccinatio~ Age-adjusted P~      2012 OR           Immun~      33
##  6 Influenza vaccinatio~ Age-adjusted P~      2012 US           Immun~      37
##  7 Influenza vaccinatio~ Age-adjusted P~      2011 OR           Immun~      33
##  8 Influenza vaccinatio~ Age-adjusted P~      2011 US           Immun~      38.1
##  9 Influenza vaccinatio~ Age-adjusted P~      2014 OR           Immun~      35.4
## 10 Influenza vaccinatio~ Age-adjusted P~      2014 US           Immun~      39.4
## 11 Influenza vaccinatio~ Age-adjusted P~      2013 OR           Immun~      33.6
## 12 Influenza vaccinatio~ Age-adjusted P~      2013 US           Immun~      38.8
```

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

```
ggplot(flupat, aes(
  x = YearStart,
  y = DataValue,
  color = LocationAbbr)) +
  geom_point() +
  geom_line()
```



Both Oregon and the wider United States experienced an increase in vaccinations until 2016 where there was a decrease.

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.

```
totalimm <- left_join(CDC, USregions, by = c("LocationDesc" = "State"))

totalimm
```

```
## # A tibble: 74,811 x 35
##    YearStart YearEnd LocationAbbr LocationDesc   DataSource Topic Question
##        <dbl>   <dbl> <chr>        <chr>          <chr>      <chr> <chr>
##  1      2016    2016 US           United States  BRFSS      Alco~ Binge drinkin~
##  2      2016    2016 AL           Alabama        BRFSS      Alco~ Binge drinkin~
##  3      2016    2016 AK           Alaska         BRFSS      Alco~ Binge drinkin~
##  4      2016    2016 AZ           Arizona        BRFSS      Alco~ Binge drinkin~
##  5      2016    2016 AR           Arkansas       BRFSS      Alco~ Binge drinkin~
##  6      2016    2016 CA           California     BRFSS      Alco~ Binge drinkin~
##  7      2016    2016 CO           Colorado       BRFSS      Alco~ Binge drinkin~
##  8      2016    2016 CT           Connecticut    BRFSS      Alco~ Binge drinkin~
##  9      2016    2016 DE           Delaware       BRFSS      Alco~ Binge drinkin~
## 10      2016    2016 DC           District of C~ BRFSS      Alco~ Binge drinkin~
## # ... with 74,801 more rows, and 28 more variables: Response <lgl>,
## #   DataValueUnit <chr>, DataValueType <chr>, DataValue <dbl>,
## #   DataValueAlt <dbl>, DataValueFootnoteSymbol <chr>, DatavalueFootnote <chr>,
## #   LowConfidenceLimit <dbl>, HighConfidenceLimit <dbl>,
## #   StratificationCategory1 <chr>, Stratification1 <chr>,
## #   StratificationCategory2 <lgl>, Stratification2 <lgl>,
```

```
## #   StratificationCategory3 <lgl>, Stratification3 <lgl>, GeoLocation <chr>,
## #   ResponseID <lgl>, LocationID <chr>, TopicID <chr>, QuestionID <chr>,
## #   DataValueTypeID <chr>, StratificationCategoryID1 <chr>,
## #   StratificationID1 <chr>, StratificationCategoryID2 <lgl>,
## #   StratificationID2 <lgl>, StratificationCategoryID3 <lgl>,
## #   StratificationID3 <lgl>, Region <chr>
```

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

There are NAs because some places such as the entirety of US, district of columbia, or puerto rico are not technically regions of the United States.

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?

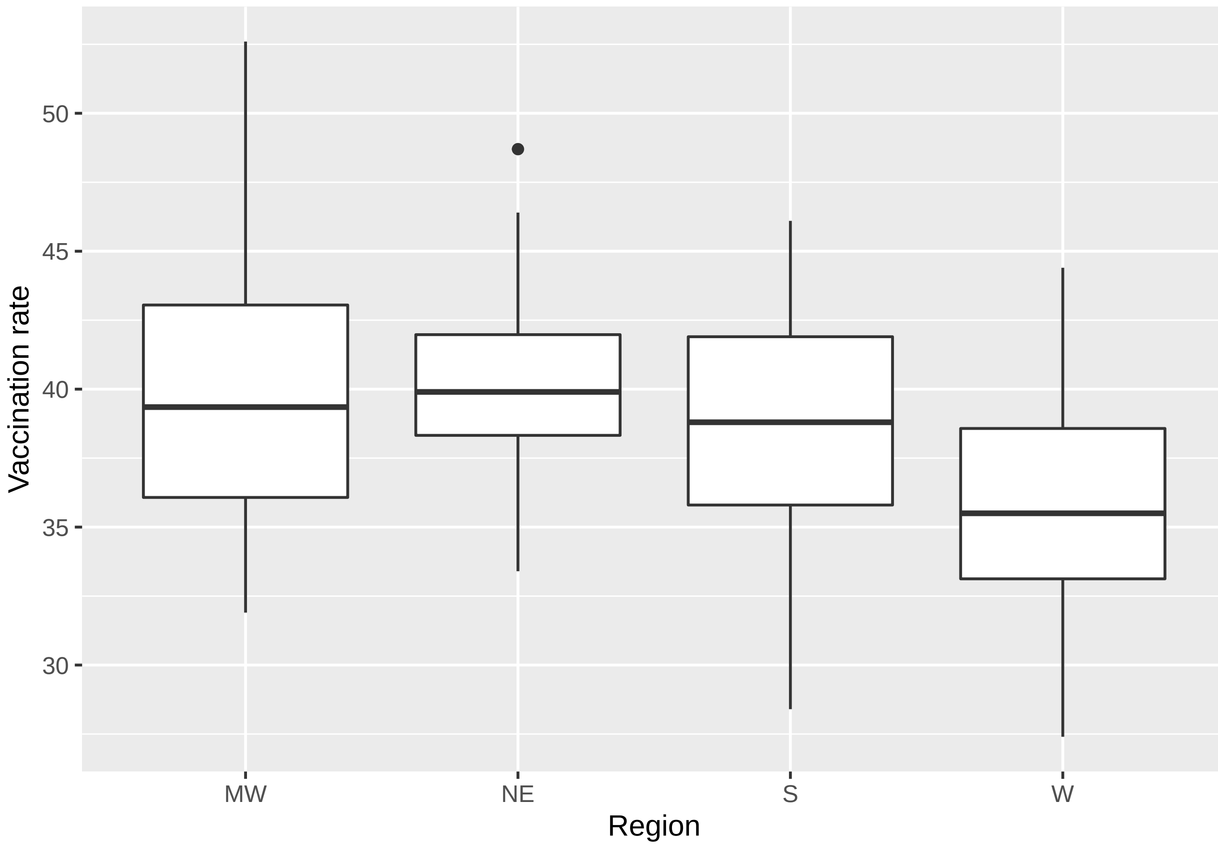
```
flurate2016 <- totalimm %>%
  select(DataValueType, YearStart, Topic, DataValue, LocationAbbr) %>%
  filter(DataValueType == "Age-adjusted Prevalence",
         YearStart == 2016 ,
         Topic == "Immunization"
         ) %>%
  arrange(desc(DataValue))

flurate2016
```

```
## # A tibble: 55 x 5
##    DataValueType           YearStart Topic        DataValue LocationAbbr
##    <chr>                       <dbl> <chr>            <dbl> <chr>
##  1 Age-adjusted Prevalence      2016 Immunization      47.2 SD
##  2 Age-adjusted Prevalence      2016 Immunization      45.1 RI
##  3 Age-adjusted Prevalence      2016 Immunization      45   IA
##  4 Age-adjusted Prevalence      2016 Immunization      43.4 NE
##  5 Age-adjusted Prevalence      2016 Immunization      43   NC
##  6 Age-adjusted Prevalence      2016 Immunization      42.9 MN
##  7 Age-adjusted Prevalence      2016 Immunization      42.3 CO
##  8 Age-adjusted Prevalence      2016 Immunization      42.1 MD
##  9 Age-adjusted Prevalence      2016 Immunization      41.9 VA
## 10 Age-adjusted Prevalence      2016 Immunization      41.4 CT
## # ... with 45 more rows
```

South Dakota has the highest immunization rate.

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.

```
totalimm %>%
  filter(DataValueType == "Age-adjusted Prevalence",
         Question == "Influenza vaccination among noninstitutionalized adults aged >= 18 years") %>%
  group_by(Region) %>%
  drop_na(Region) %>%
 ggplot(mapping = aes(x = Region, y = DataValue)) +
  geom_boxplot() +
  labs(x = "Region",
       y = "Vaccination rate")
```



My graphic uses boxplots to show the mean and median of immunization rates per region. It is able to show the variation in each regino and the audience can quickly compare means and medians.

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? (Consider the three rules of tidy data!)

```
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
```

For data to be considered tidy: each variable must have it's own column, each observation has its own row, and each value must have its own cell.

In the polls dataset dem_estimate and rep_estimate are made up of two variables. These two variables should have different columns for the dataset to be tidy.

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

```
all_polls <- polls %>%
  filter(subgroup == "All polls") %>%
  pivot_longer(cols = c(dem_estimate, rep_estimate),
               names_to = "Party",
               values_to = "Estimate")
```

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.

```
untidypolls <- polls %>%
  select(-rep_estimate) %>%
  pivot_wider(names_from = modeldate,
              values_from = dem_estimate)
```

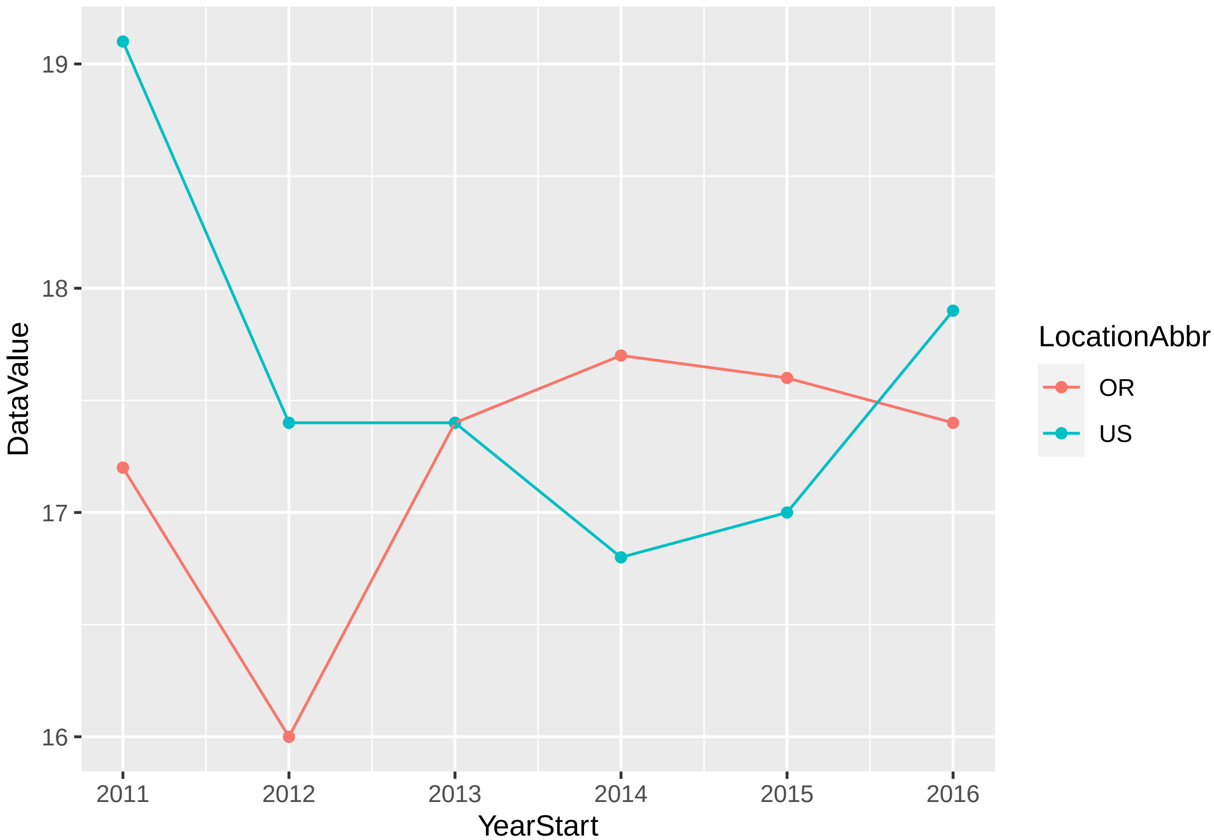
d. Why might someone want to transform the data like we did in part c?

Someone may use different software to use and analyze the data. This may require the data to be structured differently.

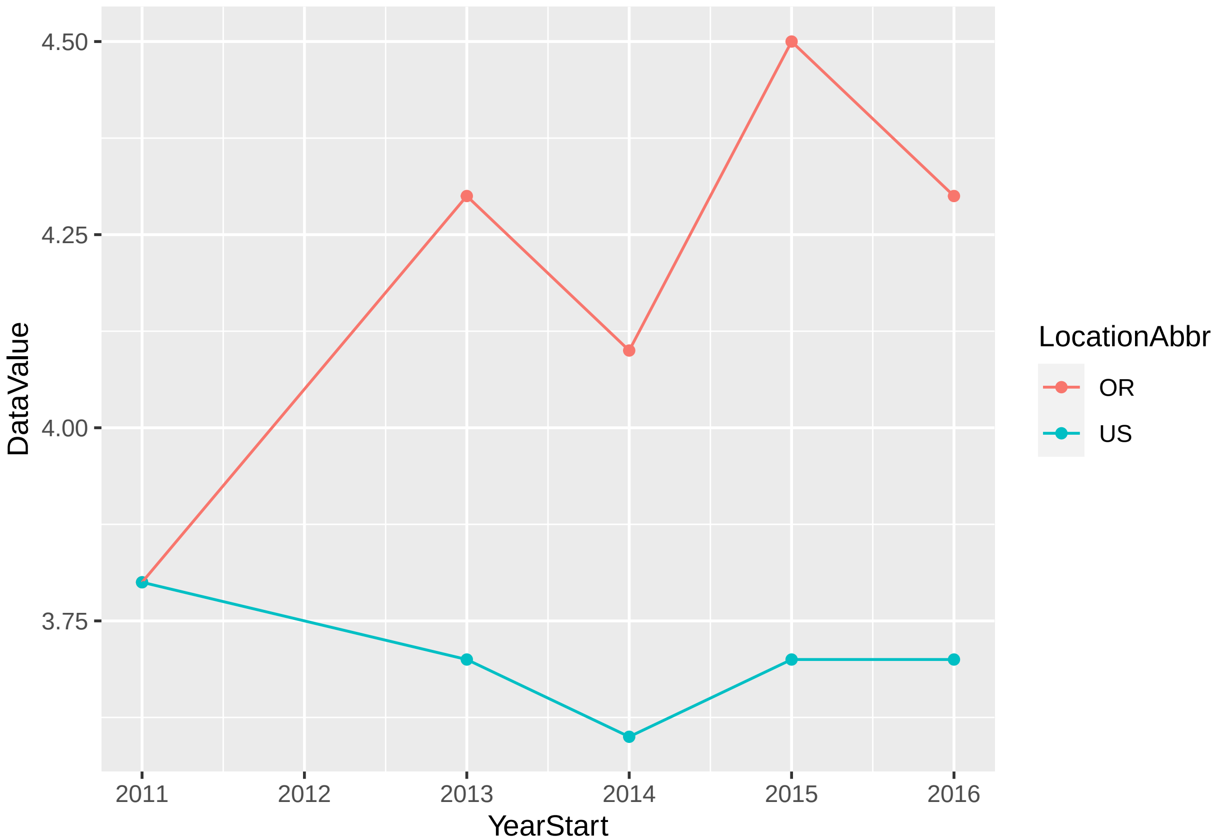
Problem 5: YOUR TURN!

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.

```
whole <- left_join(CDC, USregions, by = c("LocationDesc" = "State"))

whole <- whole %>%
  select(Question, DataValueType, YearStart, LocationAbbr, Topic, DataValue) %>%
  filter(Question %in% c("Binge drinking prevalence among adults aged >= 18 years",
                         "Recent mentally unhealthy days among adults aged >= 18 years"),
         DataValueType %in% c("Age-adjusted Prevalence", "Mean"),
         YearStart %in% c(2010: 2016),
         LocationAbbr %in% c("OR", "US"),
         Topic %in% c("Mental Health", "Alcohol")
         )

whole %>%
  filter(Topic == "Alcohol") %>%
  ggplot(aes(x = YearStart, y = DataValue, color = LocationAbbr)) +
  geom_point() +
  geom_line()
```




```
whole %>%
  filter(Topic == "Mental Health") %>%
  ggplot(aes(x = YearStart, y = DataValue, color = LocationAbbr)) +
  geom_point() +
  geom_line()
```



Question: what are the trends of mental health and alcohol consumption from the years 2010-2016? We can observe that from the years 2011-2016 there has been a decline in alcohol consumption in the United States but and increase in the state of Oregon. As for mental health there has been an increase in people experiencing more depressive symptoms while in the United States there has little to no change in people experiencing mental hardship.

Problem 6: Channeling your Inner Marie Kondo

In this problem, I am going to ask you to wrangle/clean up some data and then compare your "cleaned data" with a peer to see how your final versions differ.

a. Join `treez`, `treez_park`, and `treez_loc` to create one data frame where:

- Each row represents one tree (and there are no duplicates) from the following parks: Mt Tabor Park, Laurelhurst Park, Columbia Park
- All missing values (including suspicious values) are appropriately coded as `NA`.
- Each variable has a suitable `class`.
- Categories of categorical variables are appropriated encoded.
- And, any other cleaning is done.

It might take a little sleuthing to figure out which variables are your keys and what makes these datasets messy.

```
glimpse(treez_park)
```

```
## Rows: 25,534
## Columns: 2
## $ UserID <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, ~
## $ Park   <chr> "Gammans Park", "Gammans Park", "Gammans Park", "Gammans Park",~
```

```r
glimpse(treez)
```

```
## Rows: 4,057
## Columns: 9
## $ UserID            <dbl> 6855, 6856, 6857, 6858, 6859, 6860, 6861, 6862, 6863~
## $ DBH               <dbl> 14.0, 23.2, 25.8, 21.4, 22.9, 11.5, 30.6, 24.8, 25.1~
## $ Common_Name       <chr> "Magnolia", "Deodar Cedar", "European White Birch", ~
## $ Tree_Height       <dbl> 27, 66, 76, 45, 53, 31, 72, 69, 69, 19, 68, 33, 63, ~
## $ Crown_Width_NS    <dbl> 27, 45, 47, 45, 53, 19, 54, 53, 47, 30, 47, 34, 67, ~
## $ Crown_Width_EW    <dbl> 27, 37, 51, 47, 48, 17, 69, 66, 45, 21, 48, 27, 76, ~
## $ Crown_Base_Height <chr> "4", "4", "5", "8", "5", "6", "9", "12", "5", "9", "~
## $ Collected_By      <chr> "Staff", "Staff", "Staff", "Staff", "Staff", "Staff"~
## $ Edible            <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
```

```r
glimpse(treez_loc)
```

```
## Rows: 3,898
## Columns: 4
## $ IDUser    <dbl> 6855, 6856, 6857, 6858, 6859, 6860, 6861, 6862, 6863, 6864, ~
## $ Latitude  <dbl> 45.57692, 45.57633, 45.57593, 45.57540, 45.57542, 45.57611, ~
## $ Longitude <dbl> -122.7121, -122.7120, -122.7127, -122.7126, -122.7132, -122.~
## $ UserID    <chr> "bogus variable", "bogus variable", "bogus variable", "bogus~
```

```r
join_t1 <- treez_loc %>%
  select(IDUser, Latitude, Longitude) %>%
  rename(UserID = IDUser) %>%
  right_join(treez)

join_t <- treez_park %>%
  filter(Park %in% c("Mt Tabor Park", "Laurelhurst Park", "Columbia Park")) %>%
  right_join(join_t1) %>%
  distinct(UserID, .keep_all = TRUE)
```

b. Export your dataset to a csv file using `write_csv()`.

```r
# I recommend leaving in eval = FALSE
write_csv(join_t, file = "halp_tree.csv")
```

c. Find a classmate (maybe a project group member?) and share your cleaned datasets with each other. Save their data on RStudio and import it in the R chunk below. Also, state who you shared data with. (Feel free to share your data with multiple people but you only need to load one classmate's dataset.)

I shared my data with the slack and I used Lauren's dataset.

```r
# Import their dataset
larabey_trees <- read_csv("~/Math241/larabey_trees.csv")
glimpse(larabey_trees)
```

```
## Rows: 3,898
## Columns: 12
## $ UserID            <dbl> 6855, 6856, 6857, 6858, 6859, 6860, 6861, 6862, 6863~
## $ DBH               <dbl> 14.0, 23.2, 25.8, 21.4, 22.9, 11.5, 30.6, 24.8, 25.1~
## $ Common_Name       <chr> "Magnolia", "Deodar Cedar", "European White Birch", ~
## $ Tree_Height       <dbl> 27, 66, 76, 45, 53, 31, 72, 69, 69, 19, 68, 33, 63, ~
## $ Crown_Width_NS    <dbl> 27, 45, 47, 45, 53, 19, 54, 53, 47, 30, 47, 34, 67, ~
## $ Crown_Width_EW    <dbl> 27, 37, 51, 47, 48, 17, 69, 66, 45, 21, 48, 27, 76, ~
## $ Crown_Base_Height <dbl> 4, 4, 5, 8, 5, 6, 9, 12, 5, 9, 3, 7, 10, 9, 5, 5, 7,~
```

```
## $ Collected_By      <chr> "Staff", "Staff", "Staff", "Staff", "Staff", "Staff"~
## $ Edible            <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ Latitude          <dbl> 45.57692, 45.57633, 45.57593, 45.57540, 45.57542, 45~
## $ Longitude         <dbl> -122.7121, -122.7120, -122.7127, -122.7126, -122.713~
## $ Park              <chr> "Columbia Park", "Columbia Park", "Columbia Park", "~
```

```
nrow(larabey_trees)
```

```
## [1] 3898
```

```
nrow(join_t)
```

```
## [1] 3898
```

```
ncol(larabey_trees)
```

```
## [1] 12
```

```
ncol(join_t)
```

```
## [1] 12
```

```
setequal(larabey_trees, join_t)
```

```
## [1] FALSE
```

```
unique(larabey_trees$Park)
```

```
## [1] "Columbia Park"    "Laurelhurst Park" "Mt Tabor Park"    "Washington Park"
```

```
unique(join_t$Park)
```

```
## [1] "Columbia Park"    "Laurelhurst Park" "Mt Tabor Park"    NA
```

d. Compare your dataset and their dataset. In your comparison, answer the following questions:

- Do your datasets have the same number of rows? Same number of columns?

- Use `setequal()` to determine if they are exactly the same.
- How are they different?

Our data sets have the same number of rows (3898) and columns (12). We differ on the amount of parks we included.

e. A goal of this exercise with to experience both the **subjectivity** and **iterative nature** of data cleaning. Any time we clean data, we are making choices and often we don't catch all the bugs in our data the first (or second time around).

Based on your explorations of a classmate's cleaned dataset, do you think your dataset needs further wrangling? If not, justify. If so, do that now.

My dataset could not be further wrangled, we ended up with similar datasets with exception that I did not include one park.