Review of Data Science 1

You can download this .qmd file from here. Just hit the Download Raw File button.

Determinants of COVID vaccination rates

First, a little detour to describe several alternatives for reading in data:

If you navigate to my Github account, and find the `264_spring_2025` repo, there is a Data folder inside. You can then click on `vacc_Mar21.csv` to see the data we want to download. This link should also get you there, but it's good to be able to navigate there yourself.

```
# Approach 1
vaccine_data <- read_csv("~/Documents/SDS-264/Data/vaccinations_2021.csv")  ①

# Approach 2
vaccine_data <- read_csv("~/264_spring_2025/Data/vaccinations_2021.csv")    ②

# Approach 3
vaccine_data <- read_csv("https://joeroith.github.io/264_spring_2025/Data/vaccinations_2021.c

# Approach 4
vaccine_data <- read_csv("https://raw.githubusercontent.com/joeroith/264_spring_2025/refs/hea
```

① Approach 1: create a Data folder in the same location where this .qmd file resides, and then store vaccinations_2021.csv in that Data folder
② Approach 2: give R the complete path to the location of vaccinations_2021.csv, starting with Home (~)
③ Approach 3: link to our course webpage, and then know we have a Data folder containing all our csvs
④ Approach 4: navigate to the data in GitHub, hit the Raw button, and copy that link

A recent Stat 272 project examined determinants of covid vaccination rates at the county level. Our data set contains 3053 rows (1 for each county in the US) and 14 columns; here is a quick description of the variables we'll be using:

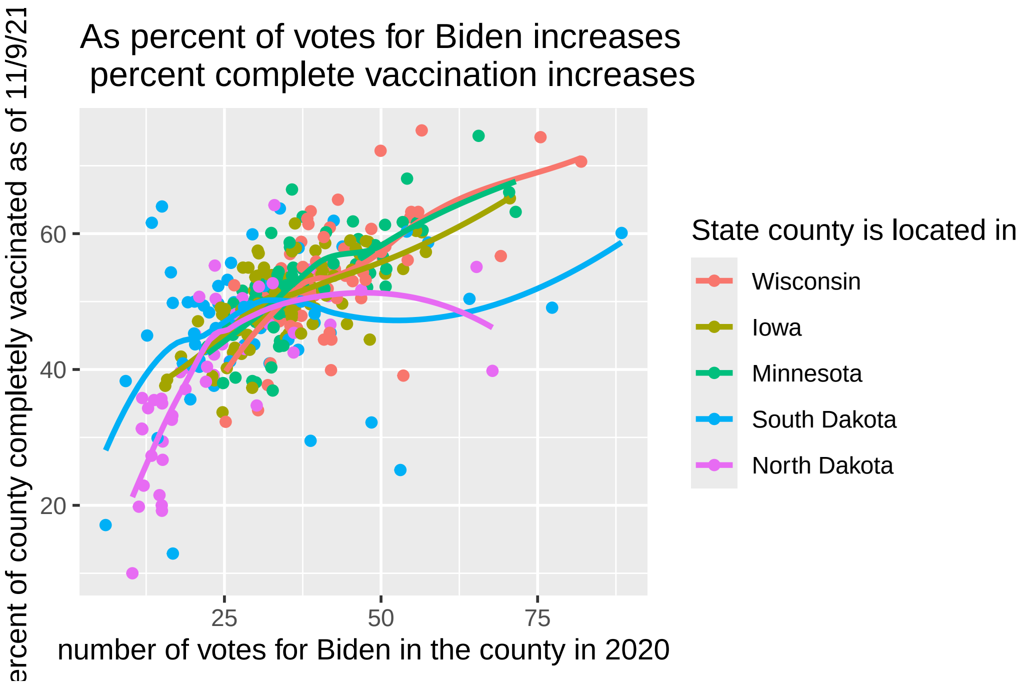
- `state` = state the county is located in
- `county` = name of the county
- `region` = region the state is located in
- `metro_status` = Is the county considered “Metro” or “Non-metro”?
- `rural_urban_code` = from 1 (most urban) to 9 (most rural)
- `perc_complete_vac` = percent of county completely vaccinated as of 11/9/21
- `tot_pop` = total population in the county
- `votes_Trump` = number of votes for Trump in the county in 2020
- `votes_Biden` = number of votes for Biden in the county in 2020
- `perc_Biden` = percent of votes for Biden in the county in 2020
- `ed_somecol_perc` = percent with some education beyond high school (but not a Bachelor’s degree)
- `ed_bachormore_perc` = percent with a Bachelor’s degree or more
- `unemployment_rate_2020` = county unemployment rate in 2020
- `median_HHincome_2019` = county’s median household income in 2019

1. Consider only Minnesota and its surrounding states (Iowa, Wisconsin, North Dakota, and South Dakota). We want to examine the relationship between the percentage who voted for Biden and the percentage of complete vaccinations by state. Generate two plots to examine this relationship:

a) A scatterplot with points and smoothers colored by state. Make sure the legend is ordered in a meaningful way, and include good labels on your axes and your legend. Also leave off the error bars from your smoothers.

```
vaccine_data |>
  filter(state %in% c("Minnesota", "Iowa", "Wisconsin", "North Dakota", "South Dakota")) |>
  ggplot(aes(x = perc_Biden, y = perc_complete_vac, color = fct_reorder2(state, perc_Biden, p
  geom_point() +
  geom_smooth(se = FALSE) +
  labs(title = "As percent of votes for Biden increases \n percent complete vaccination incre
       color = "State county is located in",
       y = "percent of county completely vaccinated as of 11/9/21",
       x = "number of votes for Biden in the county in 2020")
```

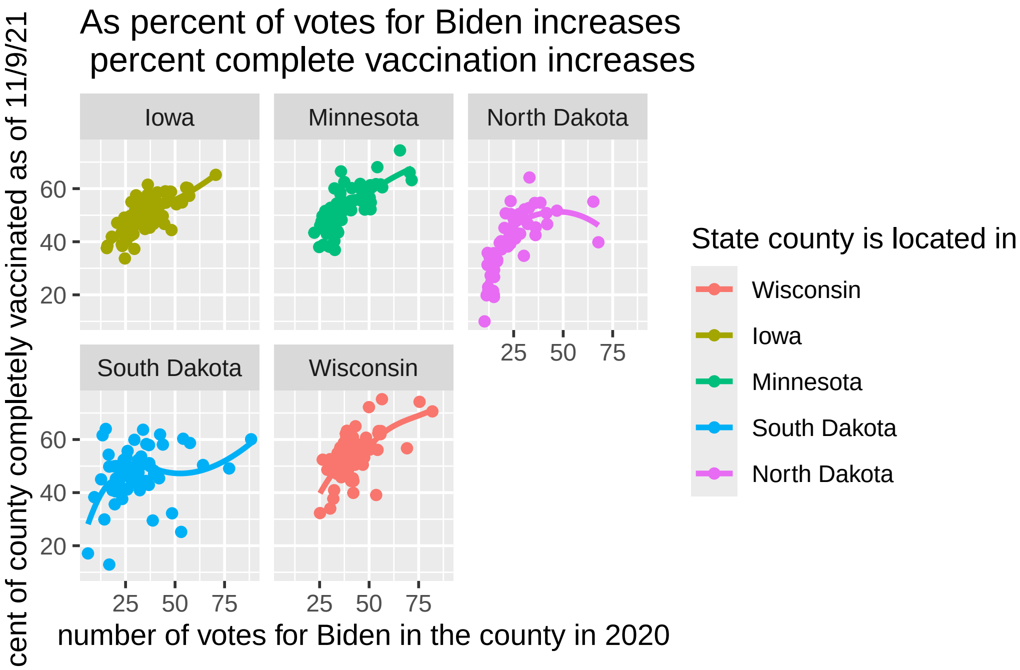
```
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



b) One plot per state containing a scatterplot and a smoother.

```
vaccine_data |>
  filter(state %in% c("Minnesota", "Iowa", "Wisconsin", "North Dakota", "South Dakota")) |>
  ggplot(aes(x = perc_Biden, y = perc_complete_vac, color = fct_reorder2(state, perc_Biden, p
  geom_point() +
  geom_smooth(se = FALSE) +
  facet_wrap(~state) +
  labs(title = "As percent of votes for Biden increases \n percent complete vaccination incre
       color = "State county is located in",
       y = "percent of county completely vaccinated as of 11/9/21",
       x = "number of votes for Biden in the county in 2020")
```

```
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Describe which plot you prefer and why. What can you learn from your preferred plot? I prefer the individual plots per state because it allows for a clearer comparison of the smoother lines. This helps me see the general trend of the data without it being cluttered by the other states.

2. We wish to compare the proportions of counties in each region with median household income above the national median ($69,560).

a) Fill in the blanks below to produce a segmented bar plot with regions ordered from highest proportion above the median to lowest.

b) Create a table of proportions by region to illustrate that your bar plot in (a) is in the correct order (you should find two regions that are *really* close when you just try to eyeball differences).

c) Explain why we can replace `fct_relevel(region, FILL IN CODE)` with

```
mutate(region_sort = fct_reorder(region, median_HHincome_2019 < 69560, .fun =
mean))
```

but not

```
mutate(region_sort = fct_reorder(region, median_HHincome_2019 < 69560))
```

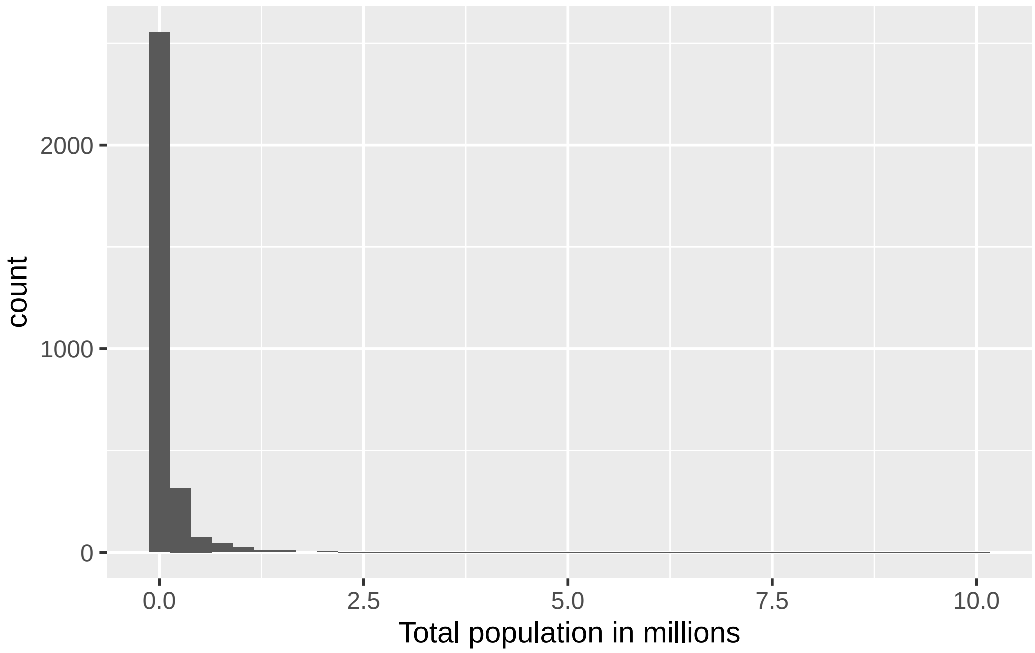
```
#part a
vaccine_data |>
  mutate(HHincome_vs_national = ifelse(median_HHincome_2019 < 69560, "Under Median", "Above
  mutate(region_sort = fct_relevel(region, "Northeast", "West", "Midwest", "South")) |>
  ggplot(mapping = aes(x = region_sort, fill = HHincome_vs_national)) +
    geom_bar(position = "fill")

#part b to find prop
vaccine_data |>
  mutate(HHincome_vs_national = ifelse(median_HHincome_2019 < 69560, "Under Median", "Above
  group_by(region) |>
  summarize(prop_above = mean(HHincome_vs_national == "Above Median"))
```

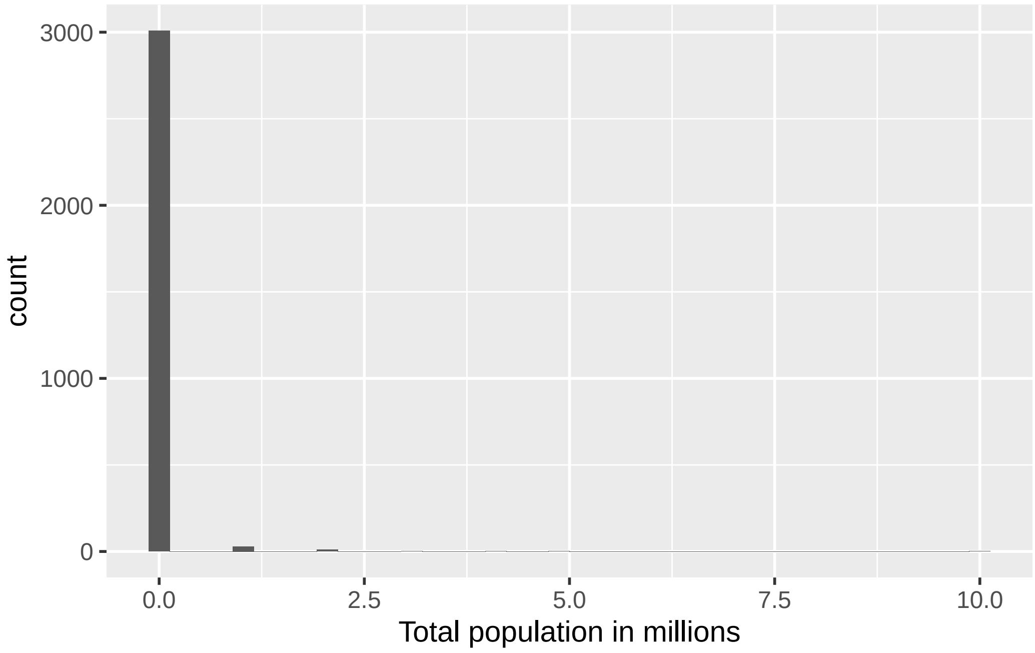
3. We want to examine the distribution of total county populations and then see how it's related to vaccination rates.

a) *Carefully and thoroughly* explain why the two histograms below provide different plots.

```
vaccine_data |>
  mutate(tot_pop_millions = tot_pop / 1000000) |>
  ggplot(mapping = aes(x = tot_pop_millions)) +
    geom_histogram(bins = 40) +
    labs(x = "Total population in millions")
```



```
vaccine_data |>
  mutate(tot_pop_millions = tot_pop %/% 1000000) |>
  ggplot(mapping = aes(x = tot_pop_millions)) +
    geom_histogram(bins = 40) +
    labs(x = "Total population in millions")
```



b) Find the top 5 counties in terms of total population.

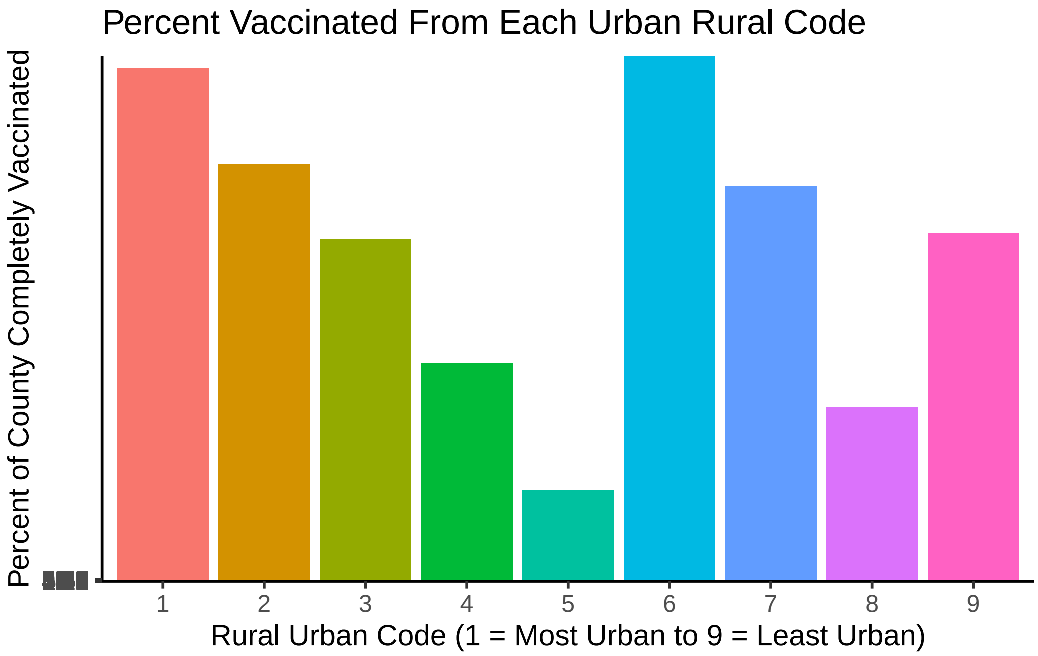
c) Plot a histogram of logged population and describe this distribution.

d) Plot the relationship between log population and percent vaccinated using separate colors for Metro and Non-metro counties (be sure there's no 3rd color used for NAs). Reduce the size and transparency of each point to make the plot more readable. Describe what you can learn from this plot.

4. Produce 3 different plots for illustrating the relationship between the rural_urban_code and percent vaccinated. Hint: you can sometimes turn numeric variables into categorical variables for plotting purposes (e.g. `as.factor()`, `ifelse()`).

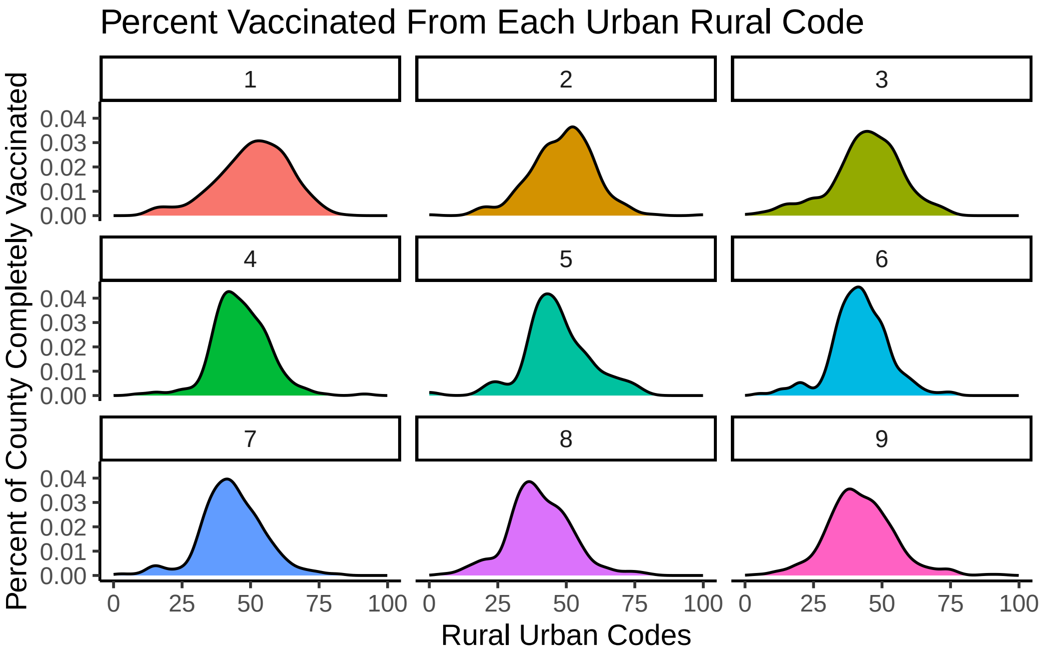
#Graph 1

```
vaccine_data |>
  mutate(rural_urban_code = as.factor(rural_urban_code),
         perc_complete_vac = as.factor(perc_complete_vac),
         vax_buckets = fct_collapse(perc_complete_vac, )) |>
  ggplot() +
  geom_bar(aes(x = rural_urban_code, y = perc_complete_vac, fill = rural_urban_code),
           stat = "identity", show.legend = FALSE) +
  theme_classic() +
  labs(title = "Percent Vaccinated From Each Urban Rural Code",
       x = "Rural Urban Code (1 = Most Urban to 9 = Least Urban)",
       y = "Percent of County Completely Vaccinated")
```



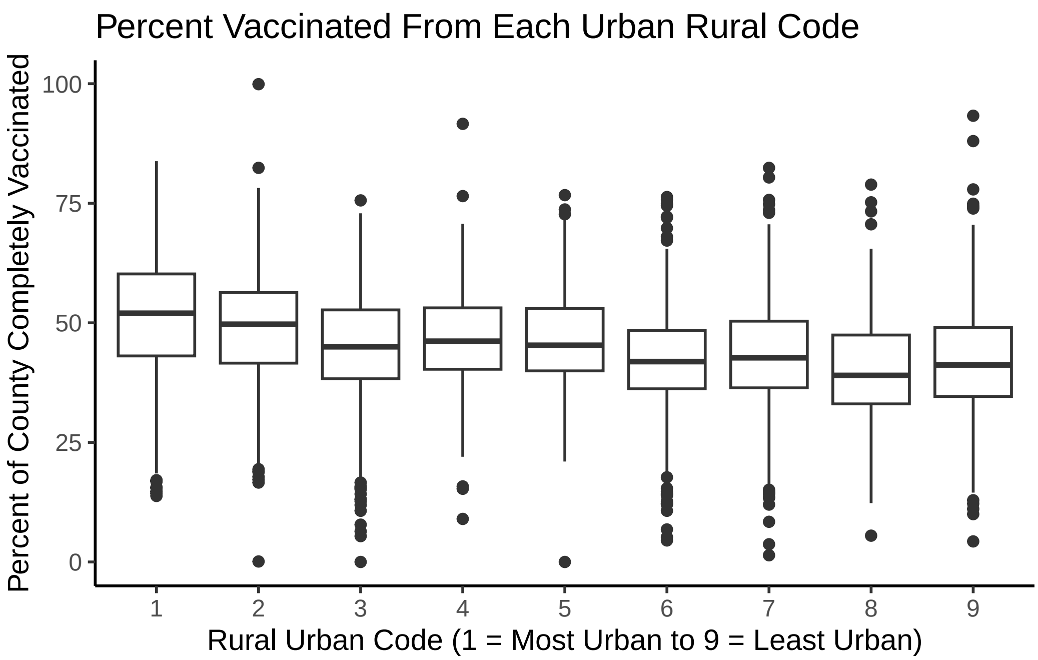
#Graph 2

```
vaccine_data |>
  mutate(rural_urban_code = as.factor(rural_urban_code))|>
  ggplot() +
  geom_density(aes(x = perc_complete_vac, fill = rural_urban_code)) +
  theme_classic() +
  theme(legend.position = "none") +
  facet_wrap(~rural_urban_code) +
  labs(title = "Percent Vaccinated From Each Urban Rural Code",
       y = "Percent of County Completely Vaccinated",
       x = "Rural Urban Codes",
       fill = "Rural Urban Code (1 = Most Urban to 9 = Least Urban)")
```



#Graph #3

```
vaccine_data |>
  mutate(rural_urban_code = as.factor(rural_urban_code)) |>
  ggplot() +
  geom_boxplot(aes(x = rural_urban_code, y = perc_complete_vac)) +
  theme_classic() +
  labs(title = "Percent Vaccinated From Each Urban Rural Code",
       x = "Rural Urban Code (1 = Most Urban to 9 = Least Urban)",
       y = "Percent of County Completely Vaccinated")
```

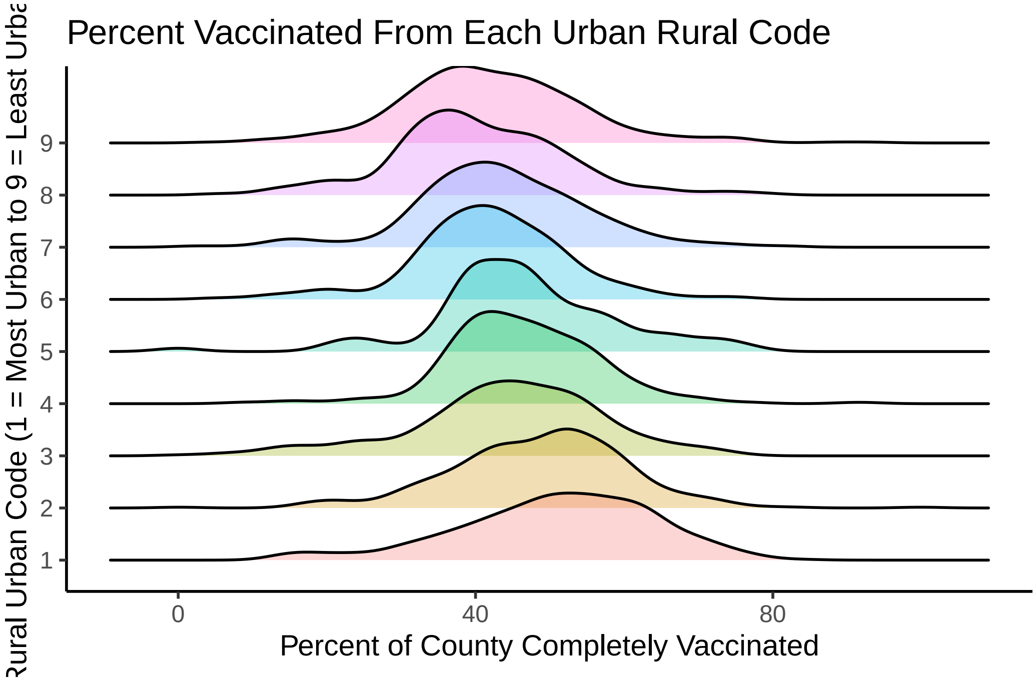


#Graph 4

```
library(ggridges)

vaccine_data |>
  mutate(rural_urban_code = as.factor(rural_urban_code)) |>
  ggplot(aes(x = perc_complete_vac, y = rural_urban_code, fill = rural_urban_code)) +
  geom_density_ridges(alpha = 0.3) +
  theme_classic() +
  theme(legend.position = "none") +
    labs(title = "Percent Vaccinated From Each Urban Rural Code",
       y = "Rural Urban Code (1 = Most Urban to 9 = Least Urban)",
       x = "Percent of County Completely Vaccinated")
```

```
Picking joint bandwidth of 3.04
```



Alt Text for Graph 4 This is a ridge line graph that displays the relationship between the percent of counties completely vaccinated and their coordinating rural urban codes. The x axis shows the percent of counties completely vaccinated and ranges from 0 - 100. The y axis shows the rural urban codes, ranging from 1 (most urban) to 9 (least urban). Generally, we can see that as urban codes increase so does the percent of counties completely vaccinated.

State your favorite plot, why you like it better than the other two, and what you can learn from your favorite plot. Create an alt text description of your favorite plot, using the Four Ingredient Model. See this link for reminders and references about alt text.

- What Kind of Graph or Chart Is It?
- What Variables Are on the Axes?
- What Are the Ranges of the Variables?
- What Does the Appearance Tell You About the Relationships Between the Variables?

I enjoyed graph #1 the most, it felt the most practical and easiest to comprehend. I can learn the importance of clear graphs and not over complicating them.

Alt Text for Graph 1 This is a bar chart graph that shows the relationship between rural urban codes and the percent of counties completely vaccinated. The x axis's variable is a leveled categorical variable that demonstrates urban codes, the "minimum" being 1 is the most urban rural code and the "maximum" being 9 is the least urban rural code. The y axis's variable is numeric with a minimum of 0 percent of a county being completely vaccinated and maximum of 25,000 being the percent of county completely vaccinated. Based on the appearance of the graph, we can see most counties completely vaccinated are in rural urban code #6, which is less urban compared to the second most completely vaccinated counties in rural urban code #1 and the most urban. Urban code #5 which isnt most or least urban, has the least percent of counties completely vaccinated.

5. BEFORE running the code below, sketch the plot that will be produced by R. AFTER running the code, describe what conclusion(s) can we draw from this plot?

We can conclude, after viewing this plot that shows the IQR of the percentage of votes for Biden in the county in 2020, we know larger values mean more variability and a generally larger spread of data. States that we predefined as "big states", such as Tennessee have a generally small IQR or perc_biden, which means not a lot of variability, as a state, their votes on him were very similar. Meanwhile, in states like Virginia, They have a very high IQR of 25, indicating high variability of votes, with a widely spread range of data. Here, the counties votes for Biden had a lot of variety.

```r
vaccine_data |>
  filter(!is.na(perc_Biden)) |>
  mutate(big_states = fct_lump(state, n = 10)) |>
  group_by(big_states) |>
  summarize(IQR_Biden = IQR(perc_Biden)) |>
  mutate(big_states = fct_reorder(big_states, IQR_Biden)) |>
  ggplot() +
    geom_point(aes(x = IQR_Biden, y = big_states))
```

6. In this question we will focus only on the 12 states in the Midwest (i.e. where region == "Midwest").

a) Create a tibble with the following information for each state. Order states from least to greatest state population.

- number of different `rural_urban_code`s represented among the state's counties (there are 9 possible)
- total state population - tot_pop
- proportion of Metro counties - metro_status
- median unemployment rate - unemployment_rate

```r
vax <- vaccine_data |>
  filter(region == "Midwest") |>
  group_by(state) |>
  summarize(tot_pop = sum(tot_pop),
            dist_urban_codes = n_distinct(rural_urban_code),
            prop_metro_counties = mean(metro_status == "Metro"),
            median_unemply_rate = median(unemployment_rate_2020)) |>
  arrange(tot_pop)

vax
```

```
# A tibble: 12 x 5
   state        tot_pop dist_urban_codes prop_metro_counties median_unemply_rate
   <chr>          <dbl>            <int>               <dbl>               <dbl>
 1 North Dakota  7.62e5                6               0.113                4.4 
 2 South Dakota  8.85e5                6               0.121                4.35
 3 Nebraska      1.26e6                6               0.292                3.3 
 4 Kansas        2.91e6                9               0.181                4.1 
 5 Iowa          3.16e6                8               0.212                4.6 
 6 Minnesota     5.64e6                9               0.310                5.6 
 7 Wisconsin     5.82e6                8               0.361                6.3 
 8 Missouri      6.14e6                9               0.296                5.6 
 9 Indiana       6.73e6                8               0.478                6.5 
10 Michigan      9.99e6                9               0.313                9.1 
11 Ohio          1.17e7                7               0.432                8.1 
12 Illinois      1.27e7                9               0.392                7.75
```

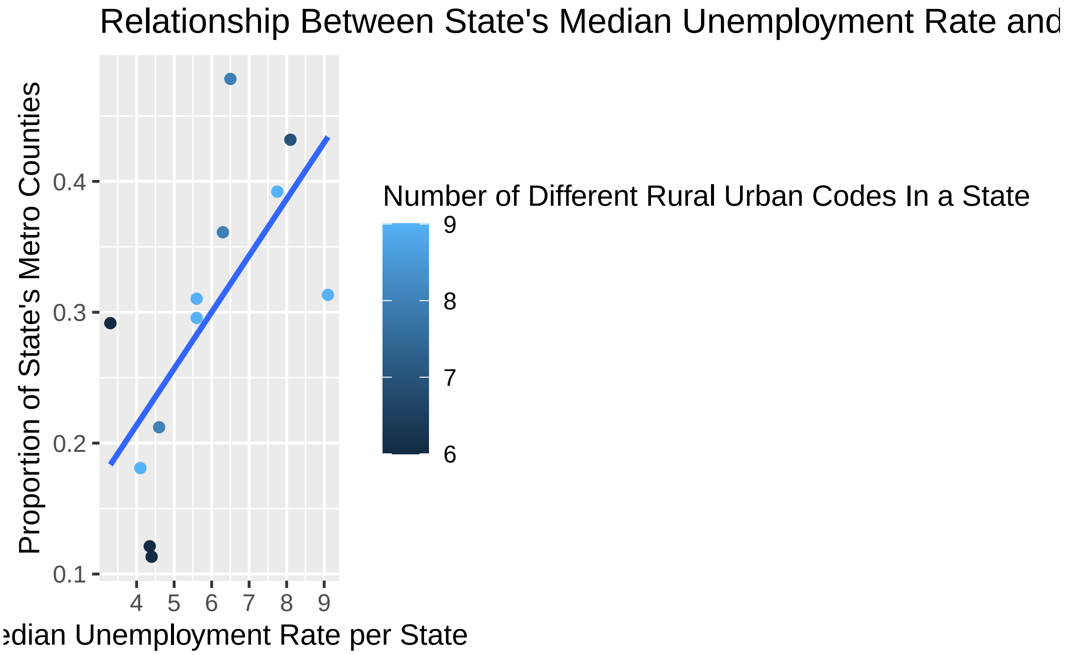
b) Use your tibble in (a) to produce a plot of the relationship between **proportion of Metro counties and median unemployment rate**. Points should be colored by the number of different `rural_urban_codes` in a state, but a single linear trend should be fit to all points. What can you conclude from the plot?

```
vax |>
  ggplot(aes(x = median_unemply_rate, y = prop_metro_counties, color = dist_urban_codes)) +
  geom_point() +
  geom_smooth(se = FALSE, method = "lm") +
  labs(
    title = "Relationship Between State's Median Unemployment Rate and Their Proportion of Me
    x = "Median Unemployment Rate per State",
    y = "Proportion of State's Metro Counties",
    color = "Number of Different Rural Urban Codes In a State"
  )
```

```
`geom_smooth()` using formula = 'y ~ x'
```

```
Warning: The following aesthetics were dropped during statistical transformation:
colour.
i This can happen when ggplot fails to infer the correct grouping structure in
```

```
  the data.
i Did you forget to specify a `group` aesthetic or to convert a numerical
  variable into a factor?
```



7. Generate an appropriate plot to compare vaccination rates between two subregions of the US: New England (which contains the states Maine, Vermont, New Hampshire, Massachusetts, Connecticut, Rhode Island) and the Upper Midwest (which, according to the USGS, contains the states Minnesota, Wisconsin, Michigan, Illinois, Indiana, and Iowa). What can you conclude from your plot?

In this next section, we consider a few variables that could have been included in our data set, but were NOT. Thus, you won't be able to write and test code, but you nevertheless should be able to use your knowledge of the tidyverse to answer these questions.

Here are the hypothetical variables:

- HR_party = party of that county's US Representative (Republican, Democrat, Independent, Green, or Libertarian)
- people_per_MD = number of residents per doctor (higher values = fewer doctors)
- perc_over_65 = percent of residents over 65 years old
- perc_white = percent of residents who identify as white

8. Hypothetical R chunk #1:

```r
# Hypothetical R chunk 1
temp <- vaccine_data |>
  mutate(new_perc_vac = ifelse(perc_complete_vac > 95, NA, perc_complete_vac),
         MD_group = cut_number(people_per_MD, 3)) |>
  group_by(MD_group) |>
  summarise(n = n(),
            mean_perc_vac = mean(new_perc_vac, na.rm = TRUE),
            mean_white = mean(perc_white, na.rm = TRUE))

vaccine_data |>
  mutate(new_perc_vac = ifelse(perc_complete_vac > 95, perc_complete_vac, NA)) |>
  summarise(n = n(),
            mean_perc_vac = mean(new_perc_vac))
```

a) Describe the tibble `temp` created above. What would be the dimensions? What do rows and columns represent? **Three rows for our MD_groups, then columns for mean_perc_vac and mean_white**

b) What would happen if we replaced `new_perc_vac = ifelse(perc_complete_vac > 95, NA, perc_complete_vac)` with `new_perc_vac = ifelse(perc_complete_vac > 95, perc_complete_vac, NA)`? **In the first statement, if perc_complete_vac is greater than 95, we replace it with NA, if it's false, we just use the value from perc_complete_vac and put it in the new_perc_vac column. In the second statement, if perc_complete_vac is greater than 95, the value from that column will get used in the new_perc_vac column, if its false, then its replaced with NA.**

c) What would happen if we replaced `mean_white = mean(perc_white, na.rm = TRUE)` with `mean_white = mean(perc_white)`? **It would become an NA value**

d) What would happen if we removed `group_by(MD_group)`? **Our data wouldnt be grouped by the three groups we created from people_per_MD, we might have fewer rows now**

9. Hypothetical R chunk #2:

```r
# Hypothetical R chunk 2
ggplot(data = vaccine_data) +
  geom_point(mapping = aes(x = perc_over_65, y = perc_complete_vac,
                           color = HR_party)) +
  geom_smooth()

temp <- vaccine_data |>
  group_by(HR_party) |>
  summarise(var1 = n()) |>
  arrange(desc(var1)) |>
  slice_head(n = 3)

vaccine_data |>
  mutate(rural_urban_code = as.factor(rural_urban_code)) |>
  ggplot(mapping = aes(x = rural_urban_code, y = perc_complete_vac)) +
    geom_boxplot()


ggplot(data = vaccine_data) +
  geom_point(mapping = aes(x = perc_Biden, y = perc_complete_vac,
                           color = ed_somecol_perc)) +
  geom_smooth()

temp <- vaccine_data |>
  group_by(ed_somecol_perc) |>
  summarise(var1 = n()) |>
  arrange(desc(var1)) |>
  slice_head(n = 3)
```

a) Why would the first plot produce an error? **Our issue would occur within the geom_smooth because we declared our aes in the geom_point, it doesn't drop down to geom_smooth. To fix this problem I would declare our aes in the ggplot.**

b) Describe the tibble `temp` created above. What would be the dimensions? What do rows and columns represent?

Three rows with HR_party and var1 as columns, var1 in descending order from greatest to least values.

c) What would happen if we replaced `fct_reorder(HR_party, perc_over_65, .fun = median)` with `HR_party`? **The x axis would no longer be ordered by median of HR party and perc_over65. instead it would only be what is in the HR_party variable. Assuming the HR+party variable is a factor, t he code should run and create box plots.**

10. Hypothetical R chunk #3:

```
# Hypothetical R chunk 3
vaccine_data |>
  filter(!is.na(people_per_MD)) |>
  mutate(state_lump = fct_lump(state, n = 4)) |>
  group_by(state_lump, rural_urban_code) |>
  summarise(mean_people_per_MD = mean(people_per_MD)) |>
  ggplot(mapping = aes(x = rural_urban_code, y = mean_people_per_MD,
      colour = fct_reorder2(state_lump, rural_urban_code, mean_people_per_MD))) +
    geom_line()

vaccine_data |>
  filter(!is.na(perc_complete_vac)) |>
  mutate(state_lump = fct_lump(state, n = 4)) |>
  group_by(state_lump, rural_urban_code) |>
  summarise(mean_perc_complete_vac = mean(perc_complete_vac)) |>
  ggplot(mapping = aes(x = rural_urban_code, y = mean_perc_complete_vac,
      colour = fct_reorder2(state_lump, rural_urban_code, mean_perc_complete_vac))) +
    geom_line()
```

a) Describe the tibble piped into the ggplot above. What would be the dimensions? What do rows and columns represent? **The tibble has removed any NA's from people_per_md, created a new variable called state_lump, which lumps the four states with largest people_per_md values, the others will be lumped into an other category. Then, the tibble groups by state_lump and rural urban codes. Finally it summarises to find mean people_per_md. This tibble will have dimensions for 9 "states" including the top four and the "other" category, then urban codes 1-9 for each state. The columns would include the state lump, rural urban code, and the mean people_per_md. 45x3**

b) Carefully describe the plot created above.

mean people per md is the y axis, the rural urban codes are the x axis. Each state and the "other" catgeory will have a line (color coded) descending from greatest to least based off the fct_reorder2 using the statelump, rural urban codes, and mean people per md.

c) What would happen if we removed `filter(!is.na(people_per_MD))`? **The data for that column would still have the NAs and might be incorrectly portrayed. There's also the risk of the column being replaced with all NAs in the summarise later on in the code which could prevent it from running.**

d) What would happen if we replaced `fct_reorder2(state_lump, rural_urban_code, mean_people_per_MD)` with `state_lump`? **The lines in graph would not be ordered anymore**