# SDS192 Lab 2

Section 2 or 3 Answer Key

## Today's Dataset

Today we will be working with a data resource from the University of Wisconsin Population Health Institute. It aggregates data from a number of government sources to produce county health indicators for every county in the US. Something to keep in mind when reviewing this data:

> *While this is a powerful data source for visualizing health disparities, particularly in smaller counties, there can be large degrees of uncertainty in the reporting of certain health measures. We won't be working with this dataset's confidence intervals today, so we do want to consider that the visualizations that we see are not a perfect reflection of county health.

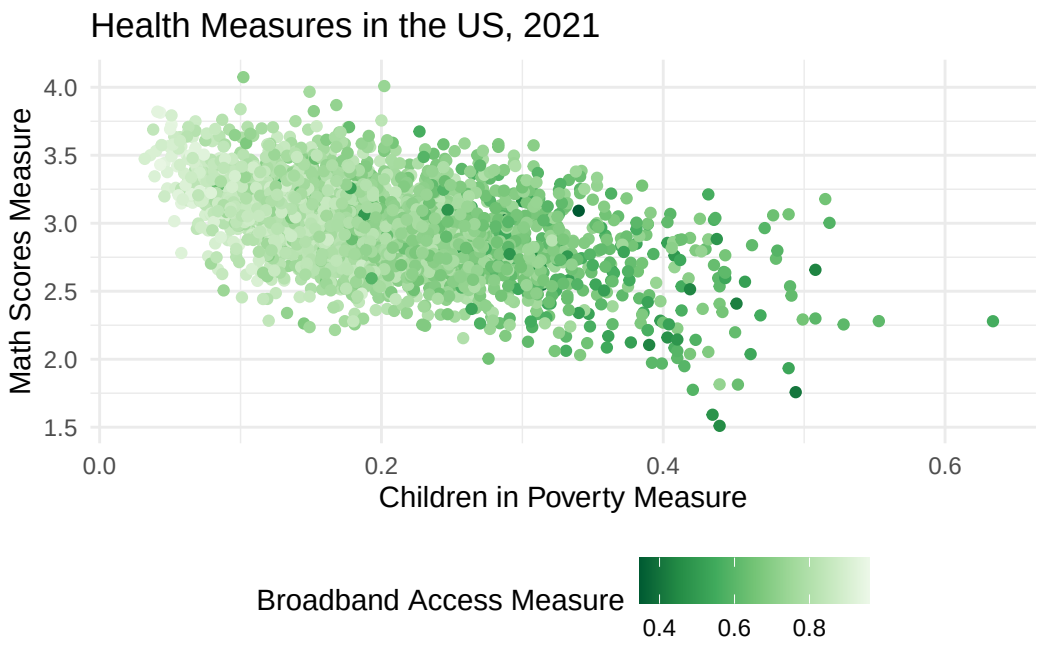
Run the code below to load the dataset we will be working with today.

## View the Data

```
# Write your code below
glimpse(county_health_2021)
```

## Overplotting

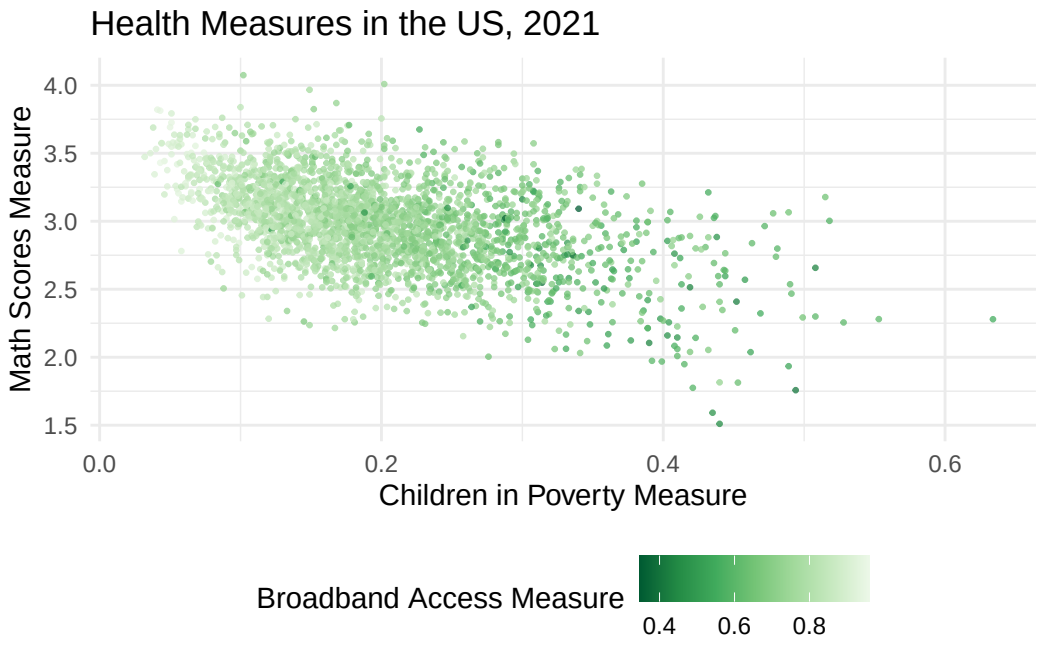
When we have so much overlapping data represented on a plot that it becomes difficult to draw any conclusions from it. Run the code below to see an example of this.




Fortunately, there are some adjustments we can make to deal with overplotting. First, we can adjust the transparency of the shapes on our plot by setting the *alpha* argument in our geom function to a number between 0 and 1. We can also reduce the size of the shapes on our plot by setting the *size* argument in our geom function to a number between 0 and 1.

### Exercise 1: Adjust the Alpha and Size of Points on Plot

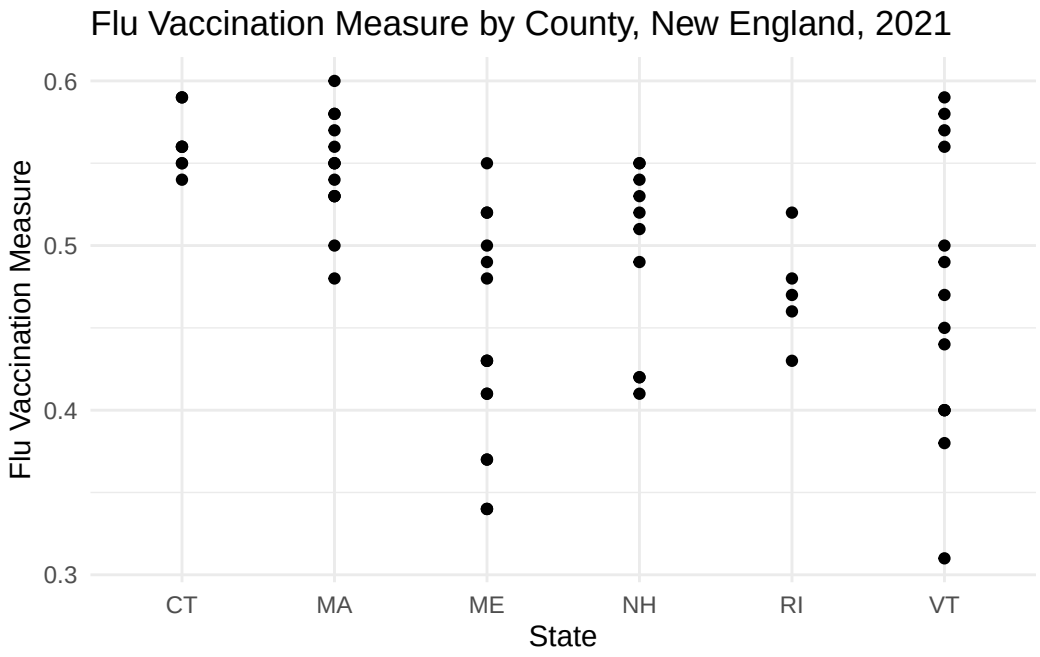
Copy and paste the code I created above into the chunk below. In the `geom_point()` function, set the *alpha* to 0.75 and the *size* to 0.5. Notice the adjustments to the plot as you update these attributes.



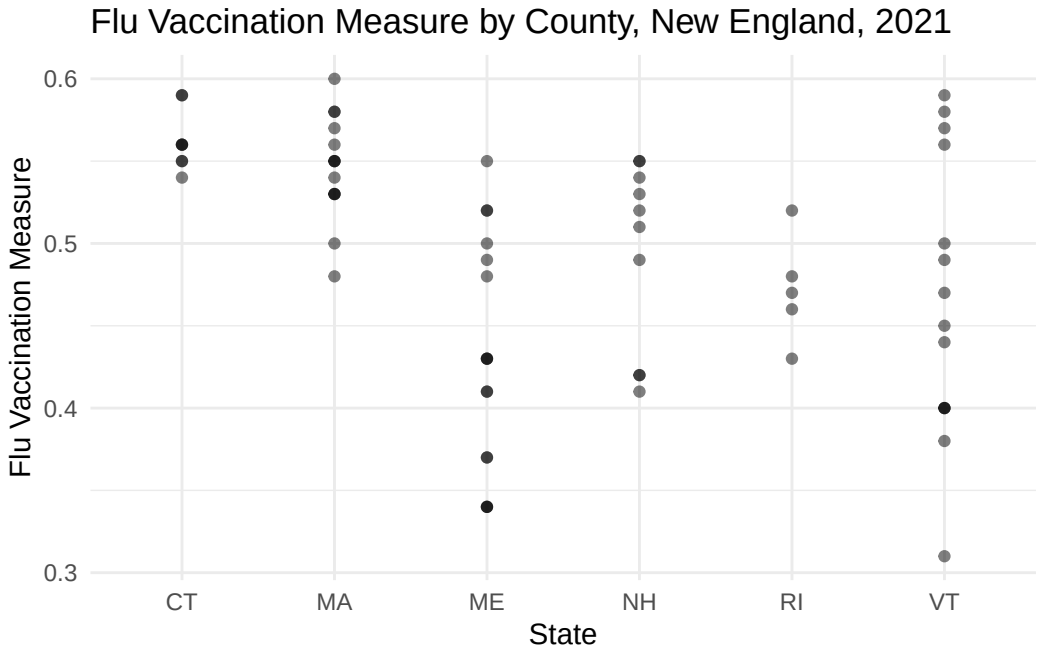

**Reflection**: What conclusions would you draw from this plot?

**Answer**: **As poverty increase math score decrease.**

Let's look at another example of overplotting. Run the code below and review the plot.



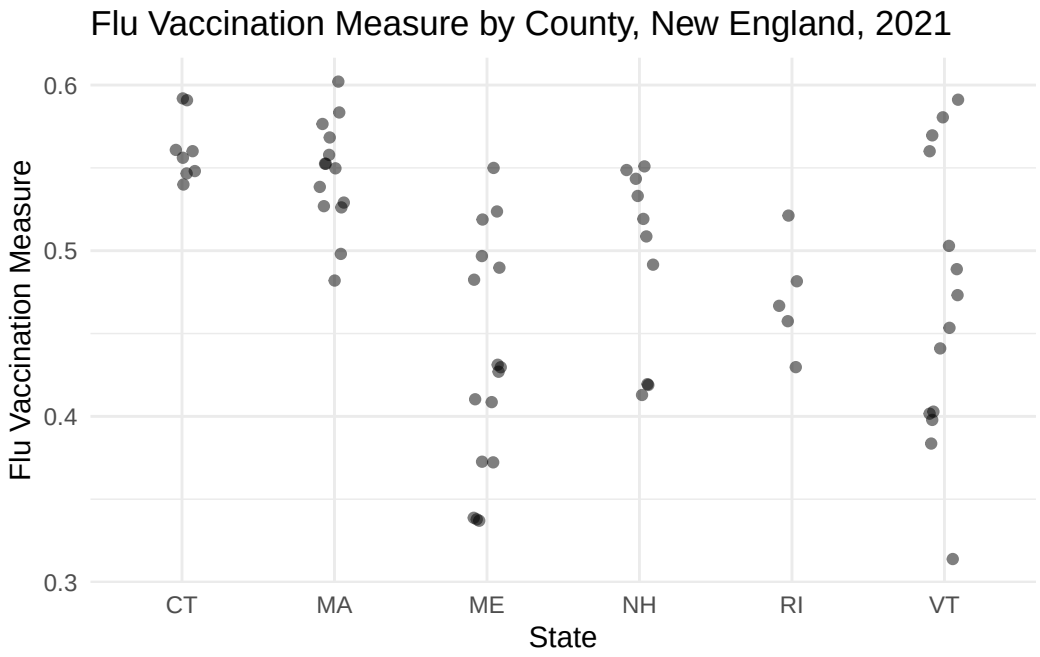
While it may look fine at first glance, it turns out that some counties have the exact same flu vaccination measure and are overlapping each other on that plot. We could add an *alpha* argument to see where there are overlaps.




However, in this case, to be sure that every county is represented on the plot, a better option, would be to add *jitter* to the plot. *jitter* offsets the points from their original position slightly so that we can make out points that overlap.

### Exercise 2: Add Jitter to the Plot

Copy the code that I created above into the chunk below. Refer to the ggplot2 cheatsheet to find the geom function you need to create this plot. Set the *width* and the *height* of the *jitter* to your choice. Notice the adjustments to the plot as you update these attributes.



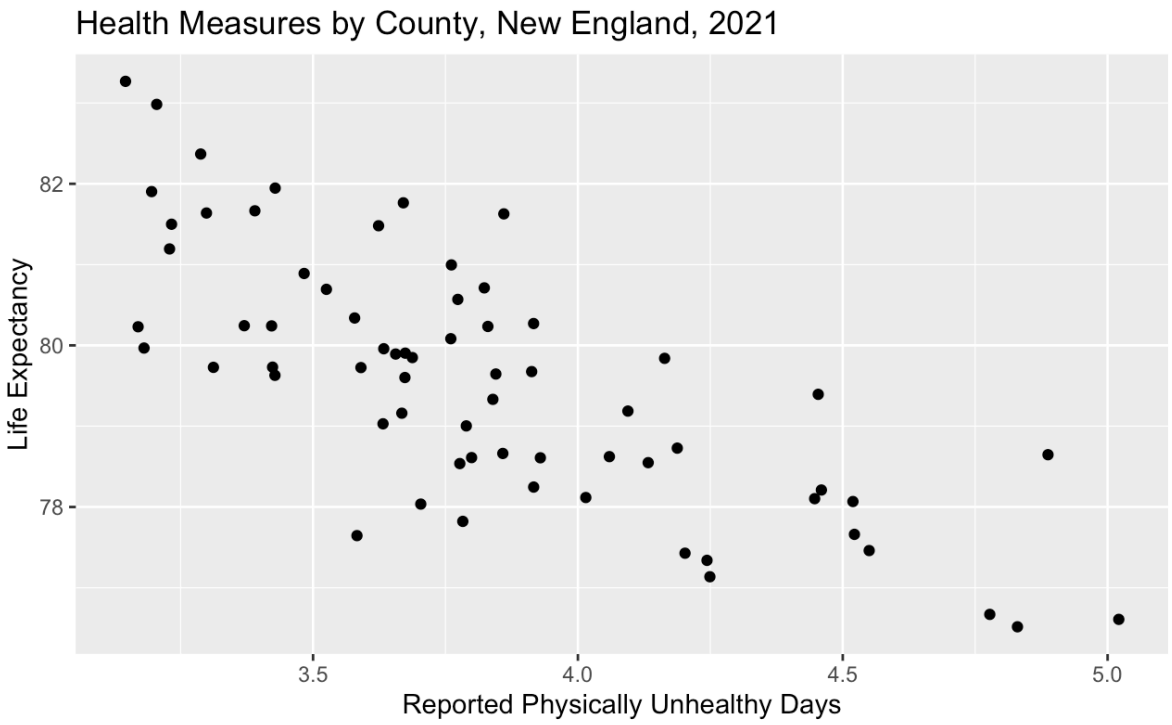

**Reflection**: Which New England states have counties with the highest flu vaccination rates? In which states are there greater disparities in vaccination rates across counties? I used geom_jitter(alpha= 0.5, width=0.1)

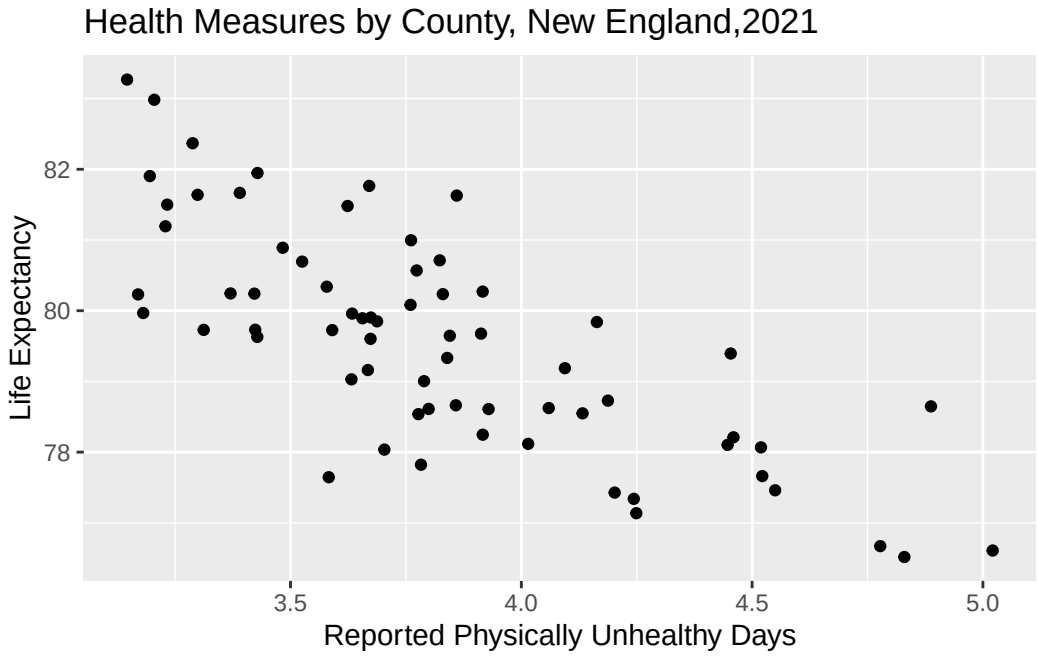
**Answer**: **CT, MA, and VT have counties with the highest vaccination rates. VT and ME have the largest disparity in rates**

## Graphics

### Exercise 3: Recreate This Image Using the `ggplot()` Function

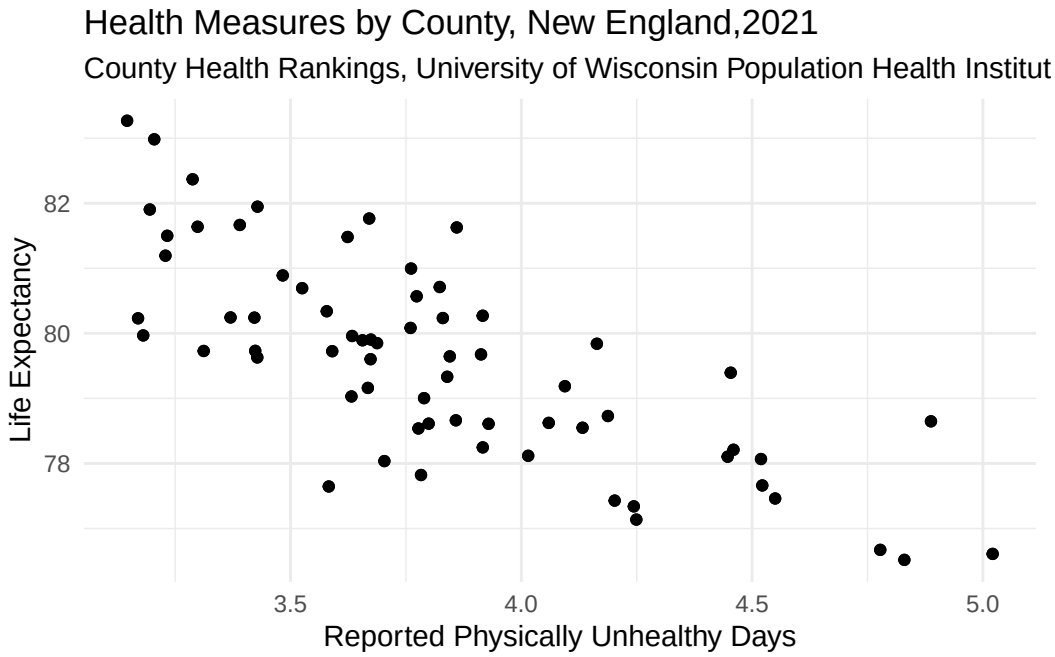
(Full size image in the images folder.)


Health Measures by County, New England, 2021
Life Expectancy
82
80
78
3.5
4.0
4.5
5.0
Reported Physically Unhealthy Days


Health Measures by County, New England,2021
Life Expectancy
82
80
78
3.5
4.0
4.5
5.0
Reported Physically Unhealthy Days

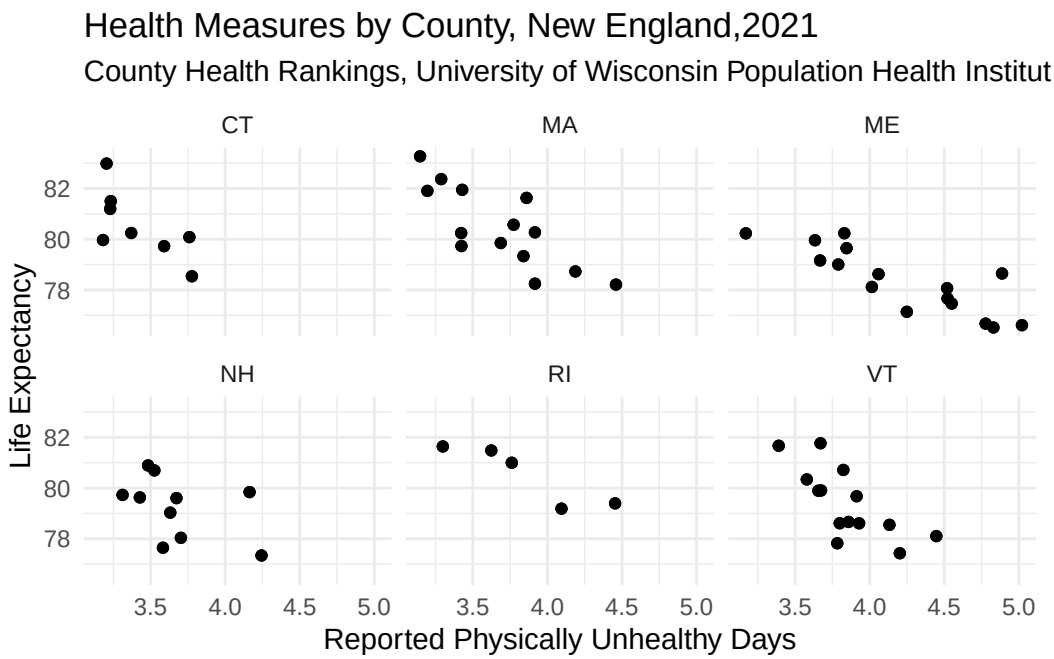
### Exercise 4: Add a Subtitle to This Plot and to Set the Theme to *Minimal* to Balance the Data-to-ink Ratio

Your subtitle should be the data's source: County Health Rankings, University of Wisconsin Population Health Institute




### Exercise 5: Divide This Plot by State

Sometimes, the narrative a plot conveys can be misleading until we starting layering in additional variables. For instance, in the above plot we look at the correlation between life expectancy and reported physically unhealthy days in New England counties, but different states in New England might have different relationships between the two variables, so it may make more sense to divide this plot out by state. We can do this by creating small multiples using the `facet_wrap()` function.




## Visulization Asthetics

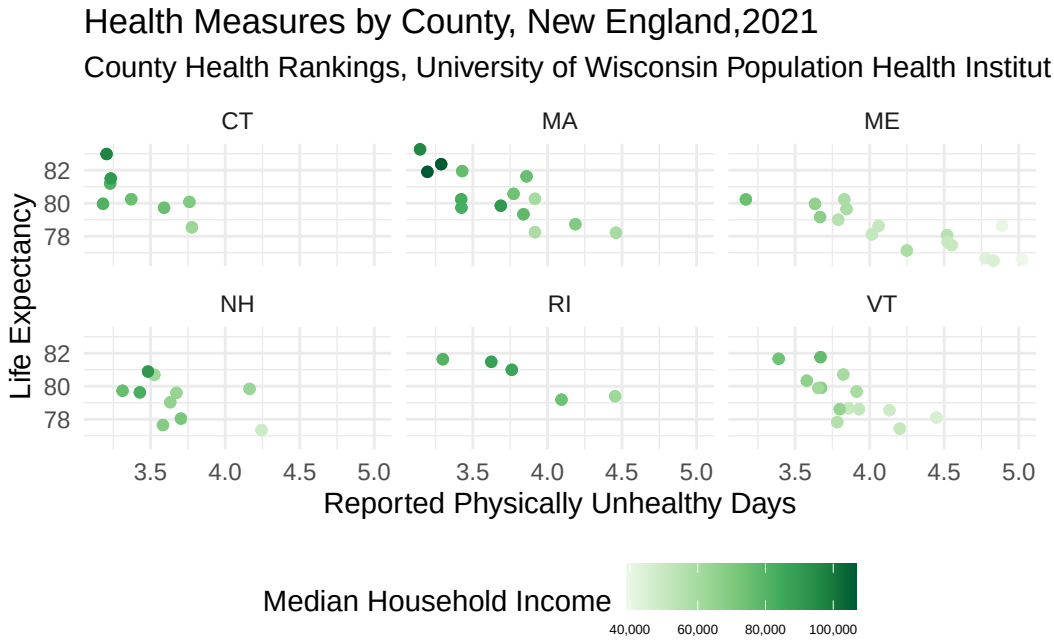
As we discussed during the colors day, there are three types of color palettes that can be added to a plot:

- Sequential: A uni-directional ordering of shades.
- Diverging: A bi-directional ordering of shades.
- Qualitative: A discrete set of colors.

We're going to use the `RBrewer` package to add appropriate palettes to these plots.

### Exercise 6: Add a Sequential Color Aesthetic to Plot

- Map the median household income raw value to the *color* aesthetic in your `aes()` function. Be sure to also change this argument in your `labs()` function to update your legend label. Then, append the following argument to the end of your plot: `scale_color_distiller(palette = "Greens", labels = scales::comma)`
- Set the legend's position to the bottom of the plot.
- The legend text may overlap. Use this argument `legend.text = element_text(size = 5)` in your `theme()` function to adjust the size of the legend text.




I used `scale_color_distiller(palette = "Greens", labels = scales::comma, direction = 1)`

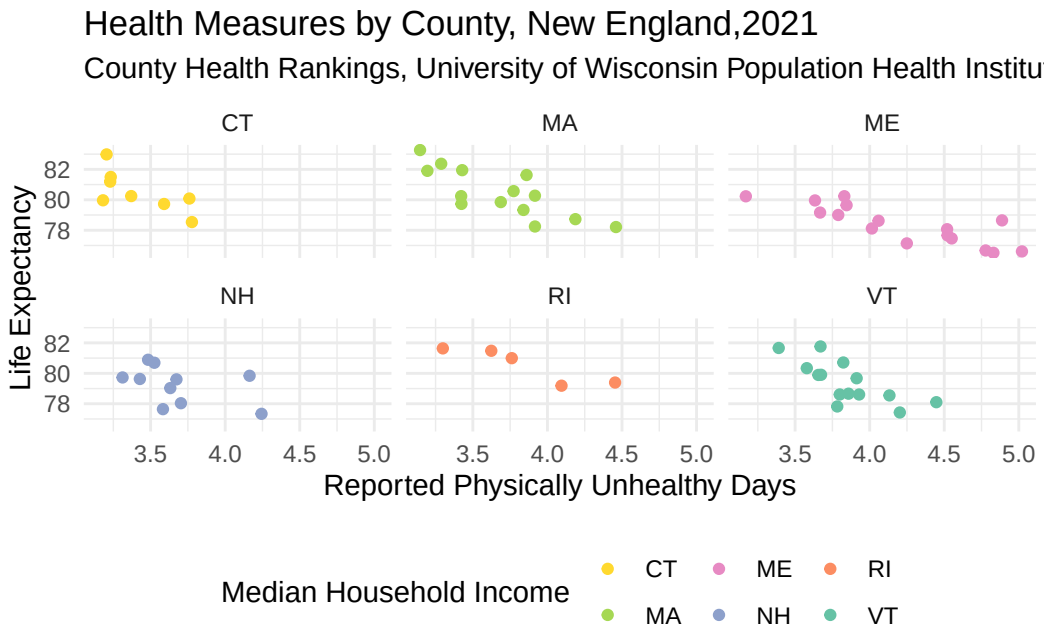
> Why do we use a sequential pattern here instead of a divergent pattern? Because Median Household Income is a continuous variable that starts at 0 and increases in one direction. There's not a middle or neutral value that we are comparing incomes around. We should only use diverging color palettes in cases when we are comparing distances in two directions from some middle point. For instance, maybe I would use a diverging palette to visualize temperatures above or below freezing.

### Exercise 7: Add a Qualitative Color Aesthetic to Plot

Copy and paste the plot we created above into the chunk below. Instead of coloring the points by median household income, color them by state. Be sure to update your legend title. Finally, convert the sequential palette into a categorical palette. To determine how to do this, direct your attention to the *Color and Fill Scales* section of the ggplot2 cheatsheet. How should the word "distiller" be edited in this function call in order to create a palette with a discrete set of colors? Once you determine this, update the function call and the palette to "Set2".

You need to delete `labels = scales::comma` in your `scale_color_*()` function. You can delete `legend.text = element_text(size = 5)` in your `theme()` function.

Hint: You don't need to change the word "color" to "fill" when adjusting the scale function name. We apply the color argument to points and lines, and the fill argument to bars and shapes. This is a point plot, so we will stick with color!



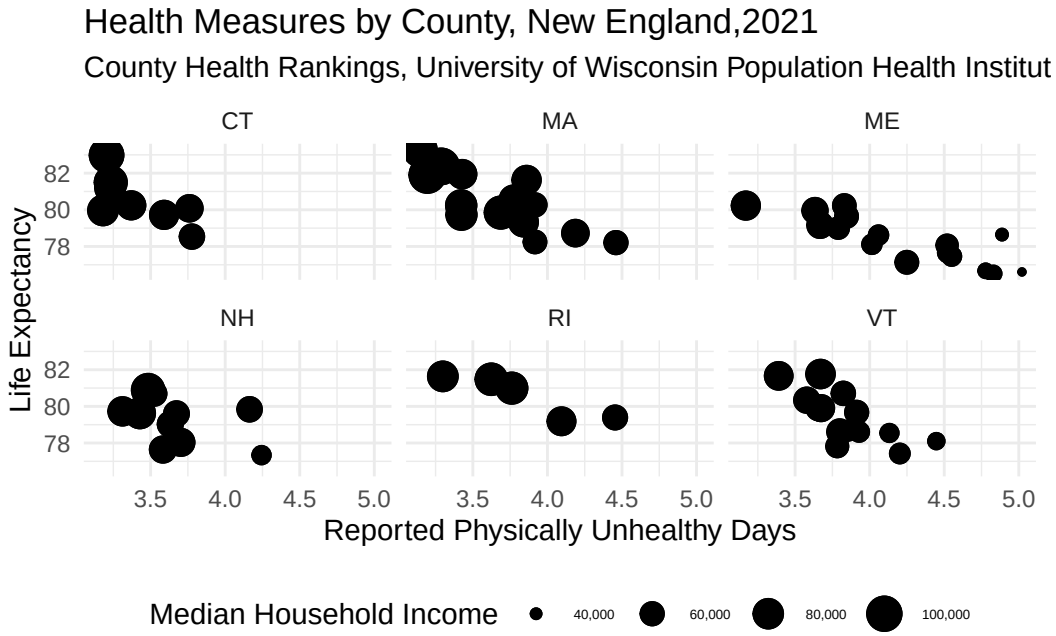

**Reflection**: Which states tend to have higher life expectancy and fewer reported physically unhealthy days? Which states tend to have lower life expectancy and more reported physically unhealthy days?

**Answer**: **CT has a higher life expectancy and fewer physcially unhealthy days. ME has a lower life expectancy and more physically unhealthy days.**

### Exercise 8: Add a Size Aesthetic to Plot

Reference the ggplot2 cheatsheet to adjust the following plot in these ways:

- Map the median household income raw value to the *size* aesthetic.
- Change the label for the *size* legend to Median Household Income (hint: you will supply the name of the aesthetic argument in the `labs()` function).
- Change `scale_color_distiller(palette = "Greens", labels = scales::comma)` to `scale_size_continuous(labels = scales::comma)`.



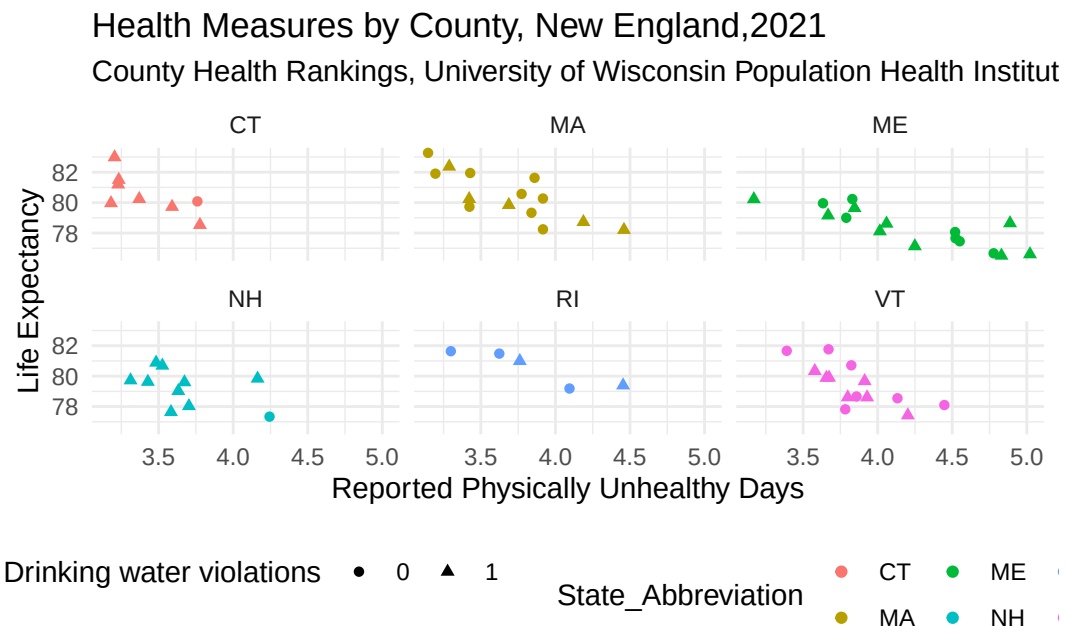

**Reflection**: What does this plot tell us that we couldn't see if we only considered the variables on the x and y axis?

**Answer**: **We can see that the counties with larger Median household income have higher life expectancy.**

## Exercise 9: Map a Shape Aesthetic to Plot

A different way to differentiate data on a plot is to map the shape aesthetic onto the plot. In this case, rather than all observations in the dataset appearing as points on a plot, observations will appear as different shapes based on their associated values in a *categorical* variable.

Copy the code that you wrote in Exercise 7 into the chunk below. Instead of the color aesthetic, assign the shape of the data on the plot by setting `shape =` to a *categorical* variable of your choice (hint: a categorical variable is a qualitative variable such as state abbreviation. We talked about types of variables in Lec 5). With the caveat that the shape palette can deal with a maximum of 6 discrete values because more than 6 becomes difficult to discriminate. Remember that, in this case, shape is an aesthetic, so this must be added inside of the `aes()` function. Add a label for the legend by setting `shape =` in the `labs()` function to a phrase that describes the shape variable. Remove the `scale_color_*()` function.




### Exercise 10: Adjust the Scale and Position of Plot

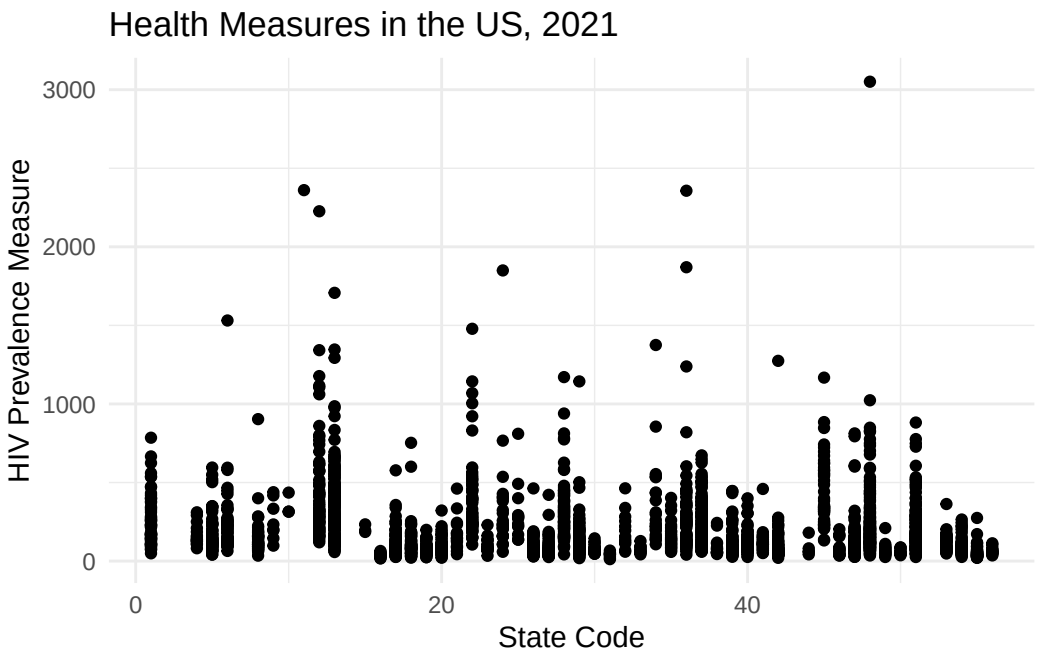
When we map a variable onto an aesthetic, we are only indicating that the variable should be mapped. We are not indicating how the variable should be mapped. In order to indicate how we want a variable mapped to an aesthetic, we can adjust its scales. Scales are adjusted by tacking the following onto a `ggplot()` object: `+ scale_*_<type>()`. For instance, let's say that I wanted to adjust the scale of my x-axis to a log scale. I would attach `+ scale_x_log10()` to my `ggplot()` object.

We can adjust the scale of our x and y-axes by adding `+ scale_x_<type>()` or `+ scale_y_<type>()` to our plots. Note what happens when we attempt to create a scatterplot that shows the relationship between all the counties in the US and the HIV prevalence measure. Run the code below to see an example of this.

```
#|echo: FALSE

ggplot(data = county_health_2021,
       aes(x = State_FIPS_Code,
           y = HIV_prevalence_raw_value)) +
  geom_point() +
  scale_y_continuous() +
  labs(title = "Health Measures in the US, 2021",
```

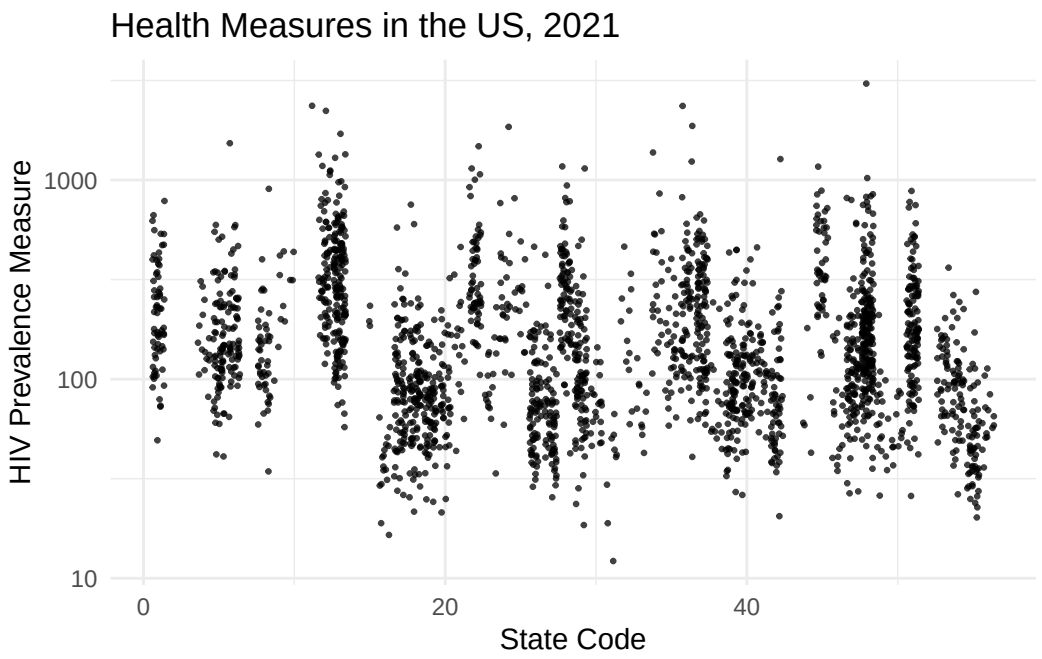
```
      x = "State Code",
      y = "HIV Prevalence Measure") +
theme_minimal()
```



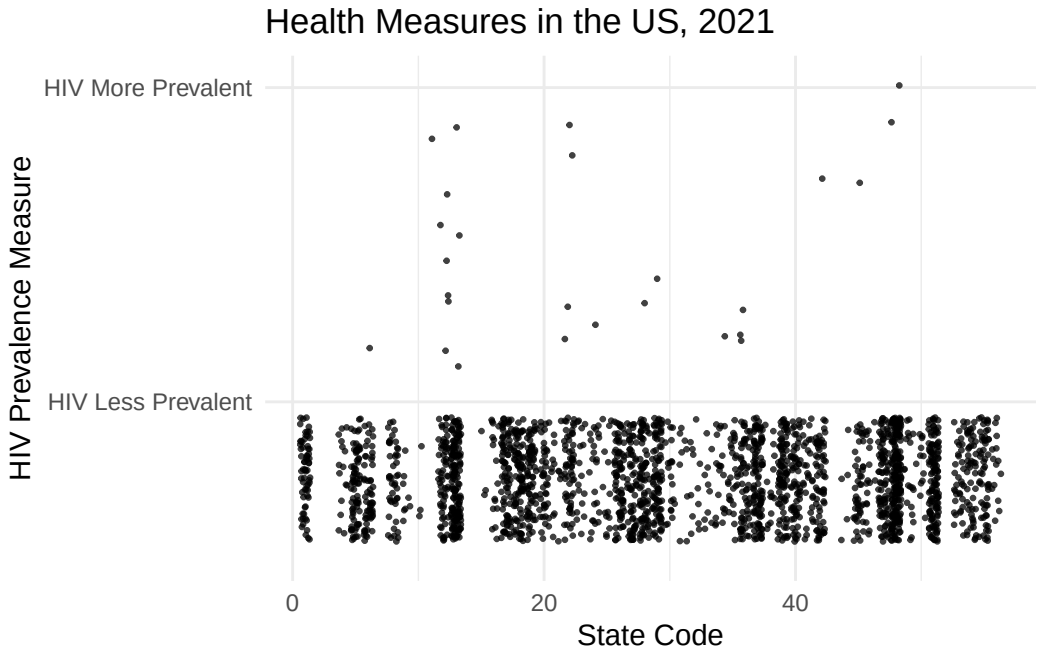

Even if we adjust the *alpha* and *size* of points on the plot as we did in Exercise 1, as well as use another method to add *jitter* to the plot, which is to set `position = "jitter"` in the `geom_point()` function (Reference the ggplot2 cheatsheet to learn more about Position Adjustments), the plot is still difficult to interpret. Run the code below to see an example of this.


Due to most states have fewer than 1,000 people newly diagnosed with HIV in 2021, the vast majority of the points on the plot appear at the bottom of the y-axis scale and are indiscernible from one another. This is a case when it makes sense to apply a log scale to the y-axis.

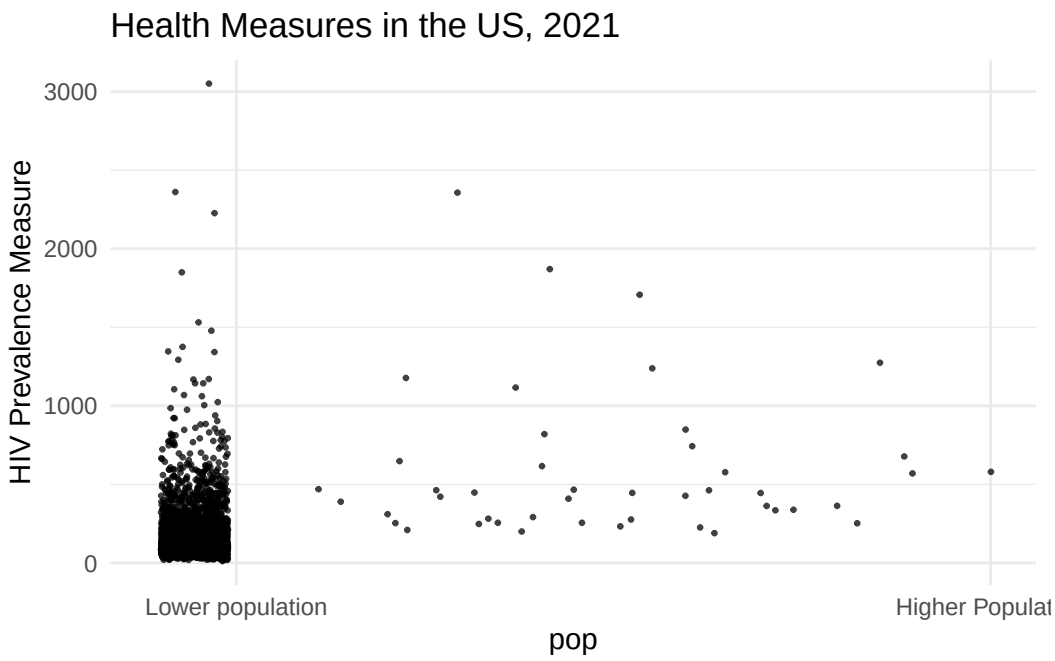
Copy the plot that I created above, and change the y-axis scale from continuous to log10. You might reference the formula specified above or reference the ggplot2 cheatsheet for help.




Sometimes I might wish to group certain numerical values into bins on a plot. For instance, let's say I just want to see how many states are more HIV prevalent in comparison to states that are less HIV prevalent. So I want to group the numerical values in HIV_prevalence_raw_value into two bins: 0-1000 and 1000-3000. To do this, I will set the scale to: `+ scale_y_binned()` and add an argument to establish the bin breaks: `breaks = c(1000, 3000)` as well as an argument to label the bin breaks: `labels = c("HIV Less Prevalent", "HIV More Prevalent")`. Check out what happens to the y-axis scale when I do this below.




This is just an exploratory data analysis. I was wondering if there is a relationship between population and HIV prevalence measure. I want to ask you to help explore this. You may need to adjust the x-axis scale of the plot. Specifically, you need to group the numerical values Population_raw_value into two bins of your choice. Just explore! Set appropriate labels for the two bins you choose.




## Exercise 11: Weekly Reflections

There are no weekly reflections