Welcome to DATA 151

I'm so glad you're here!

DATA 151: CLASS 9B
INTRODUCTION TO DATA SCIENCE (WITH R)

TRENDS OVER TIME AND SPACE

NOTES PREPARED BY PROF. KITADA SMALLEY (FALL 2022)

ANNOUNCEMENTS

HOMEWORK REMINDER

Due next week:

- ***DUE 11/1*** *Project Milestone #5: EDA Step 3*
  - Numeric Distributions and Summary Statistics
- ***DUE 11/3*** *HW #9: DC Exploratory Data Analysis with Numerical Summaries*
  - *Just one chapter*
  - ***No submission on WISE necessary, do on DataCamp***

FRIENDLY REMINDER

Midterm #2 is Next Thursday

(content from weeks 5-9)

KNOWLEDGE CHECK

COMPREHENSION QUESTION: SPREAD

Which measure(s) of spread would be sensitive to the presence of outliers?

1. Variance
2. Standard deviation
3. IQR
4. Range

COMPREHENSION QUESTION: CENTER

Which measure(s) of center would be sensitive to the presence of outliers?

1. Mean
2. Median
3. Mode

COMPREHENSION QUESTION: STANDARD DEVIATION

A standard deviation can be negative.

- TRUE
- FALSE

COMPREHENSION QUESTION: STANDARD DEVIATION

A standard deviation can be negative.

- TRUE
- FALSE

FALSE, when calculating we square the deviations and the result will always be positive.

COMPREHENSION QUESTION: STANDARD DEVIATION

A standard deviation can be 0.

- TRUE
- FALSE

TRUE, when all values are exactly the same (5, 5, 5, 5) the data set will have zero spread

CONCLUSION

Choosing measures of center and spread:

- Skewed distortions or distributions with extreme outliers
  - Use median and quartiles

- Approximately symmetric distribution (with no outliers)
  - Use mean and standard deviation

DATA151: Trends Over Time and Space

Kitada Smalley

Learning Objectives

In this lesson students will learn how to create

- Time series plots
- Choropleths (colored map plots)

TRENDS OVER TIME

EXAMPLE 1: SALEM AQI

Time Series Plots

Time series plots show how a variable (on the y-axis) changes over time (on the x-axis).

Example 1: Salem, Oregon AQI

Step 0: Library Tidyverse

```
library(tidyverse)
```

Step 1: Load the Data

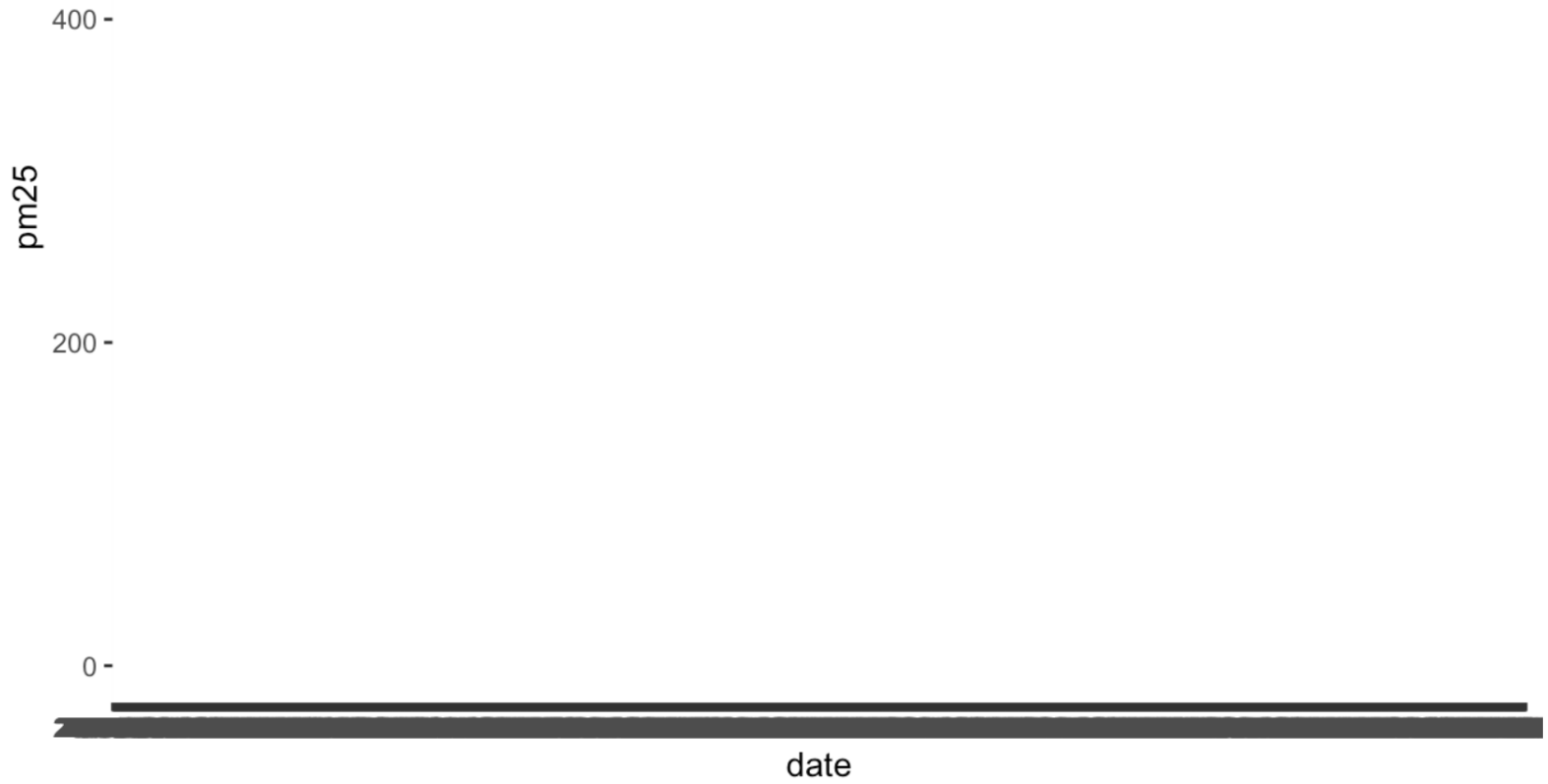
```
salem<- read.csv("https://raw.githubusercontent.com/kitadasmalley/DATA151/main/Data/salemOR_AQI.csv",
                 header=TRUE)

str(salem)
```

```
## 'data.frame':    2799 obs. of  4 variables:
##  $ date: Factor w/ 2799 levels "2014/1/1","2014/1/10",..: 2545 2551 2552 2553 2554 2555 2556 2557 2535 2536 ...
##  $ pm25: int  41 42 26 35 57 72 68 72 91 63 ...
##  $ pm10: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ o3  : int  33 27 12 NA NA NA NA NA NA NA ...
```

Step 2: `geom_line()`

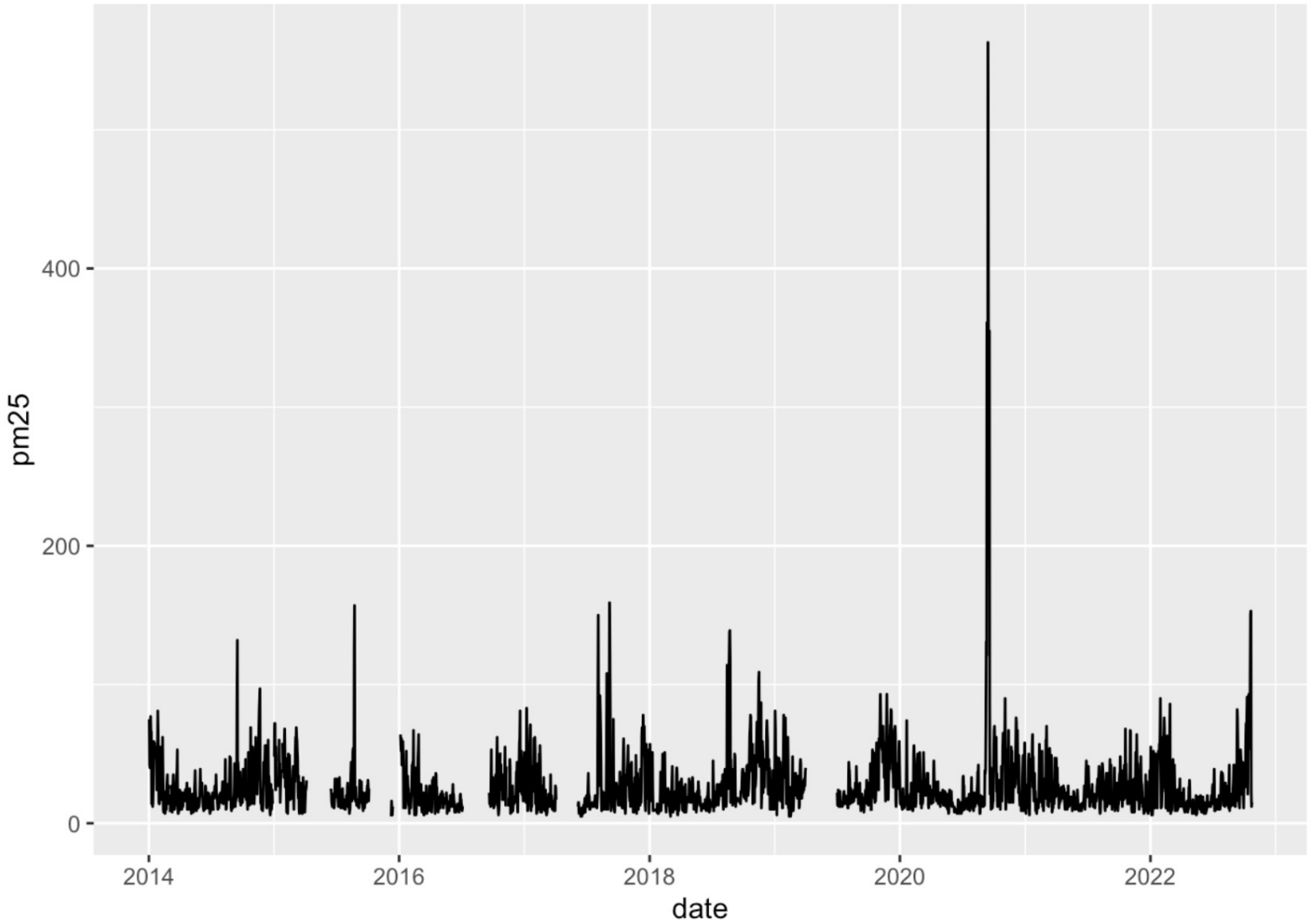
Let's just try using `geom_line()`:

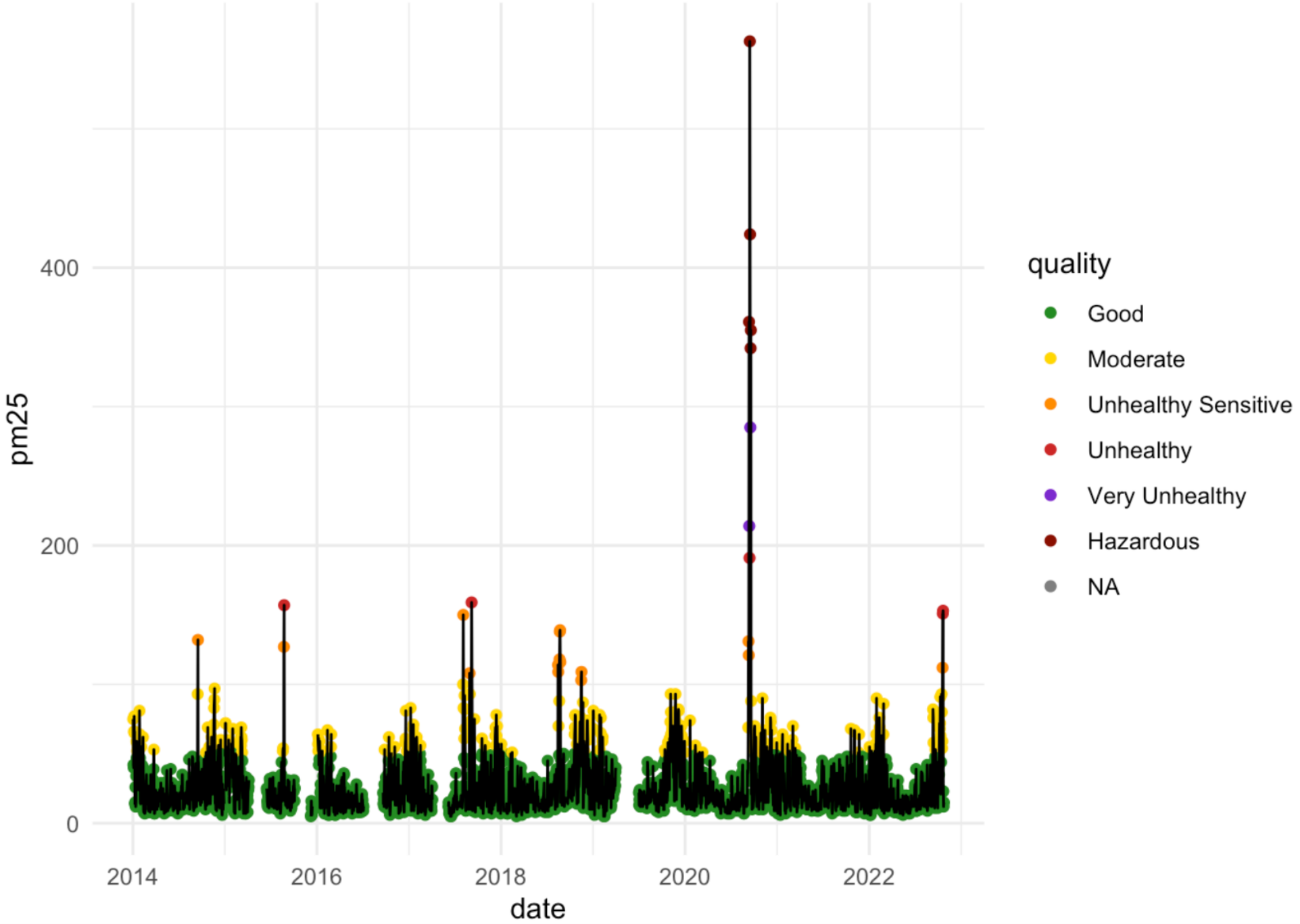
```
ggplot(salem, aes(date, pm25))+
  geom_line()
```



What's wrong with this?

```
salem$date<-as.Date(salem$date)

ggplot(salem, aes(date, pm25))+
  geom_line()
```




400
200
0
pm25
2014
2016
2018
2020
2022
date
quality
Good
Moderate
Unhealthy Sensitive
Unhealthy
Very Unhealthy
Hazardous
NA

EXAMPLE 2: CRYPTOCURRENCY

Step 1: Load the Data

These data are in three separate files:

```
coin_Bitcoin <- read_csv("https://raw.githubusercontent.com/kitadasmalley/DATA151/main/Data/coin_Bitcoin.csv")
coin_Dogecoin <- read_csv("https://raw.githubusercontent.com/kitadasmalley/DATA151/main/Data/coin_Dogecoin.csv")
coin_Ethereum <- read_csv("https://raw.githubusercontent.com/kitadasmalley/DATA151/main/Data/coin_Ethereum.csv")
```

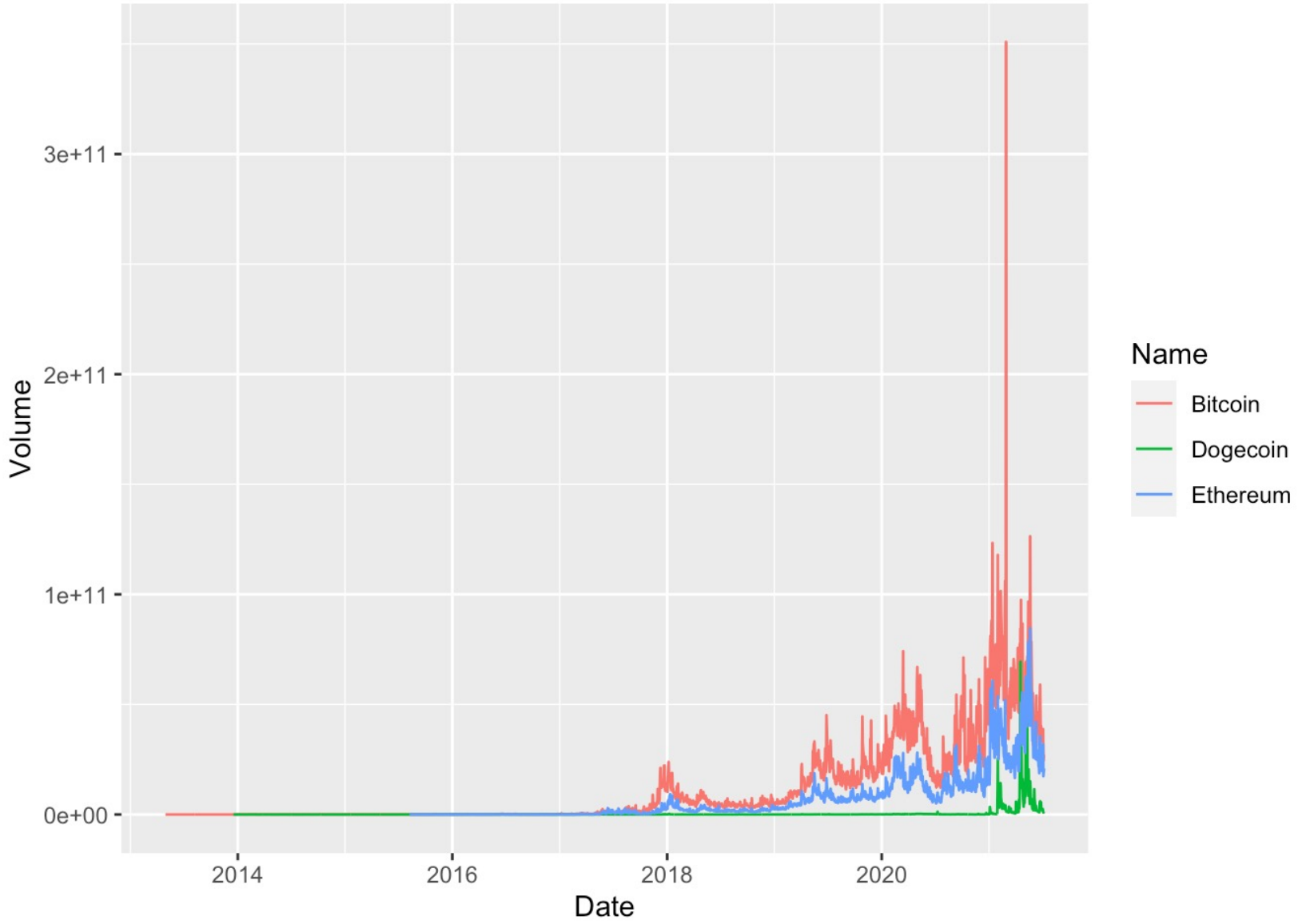
Step 2: Combine the data

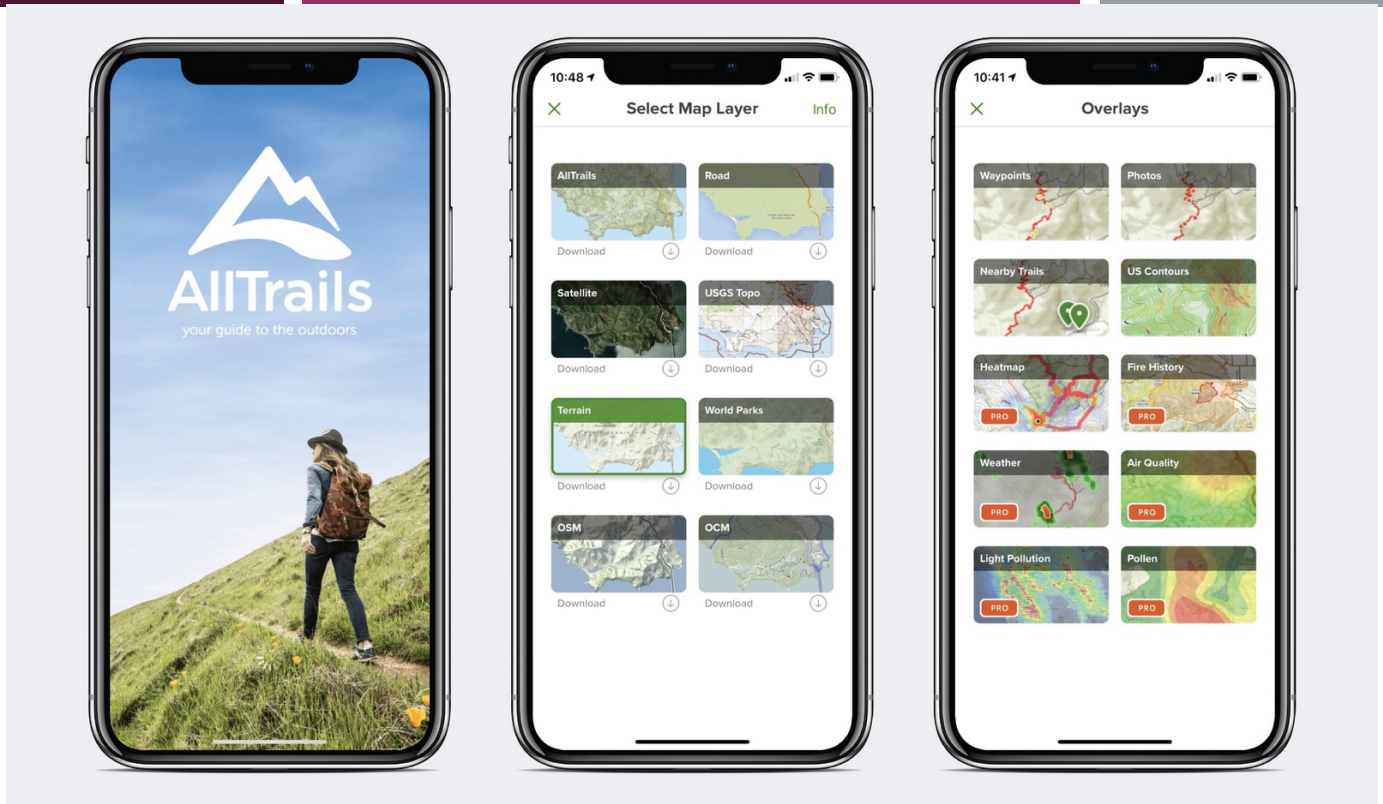
```
coinBind<-coin_Bitcoin %>%
  rbind(coin_Dogecoin)%>%
  rbind(coin_Ethereum)
```

Step 3: Time Series Plot

Since `Date` is already a date type variable we can go ahead and plot it. Here `color=Name` works as a grouping variable.

```
#str(coinBind)

ggplot(coinBind, aes(x=Date, y=Volume, color=Name))+
  geom_line()
```





CHOROPLETHS (MAP PLOTS)

Example 3: All Trails

Step 1: Load the Data

```
npark <- read_csv("https://raw.githubusercontent.com/kitadasmalley/DATA151/main/Data/AllTrails%20data%20-%20nationalpark.csv")
```

Step 2: State Level Data

Group by state to create summaries for metrics within a state.

```
stateNP<-npark%>%
  group_by(state_name)%>%
  summarise(stateTrails=n(),
            avgPop=mean(popularity, na.rm=TRUE),
            avgElev=mean(elevation_gain, na.rm=TRUE))
```

Step 3: `usmap` Package

```
#install.packages("usmap")
library(usmap)
```

```
## Warning: package 'usmap' was built under R version 3.6.2
```

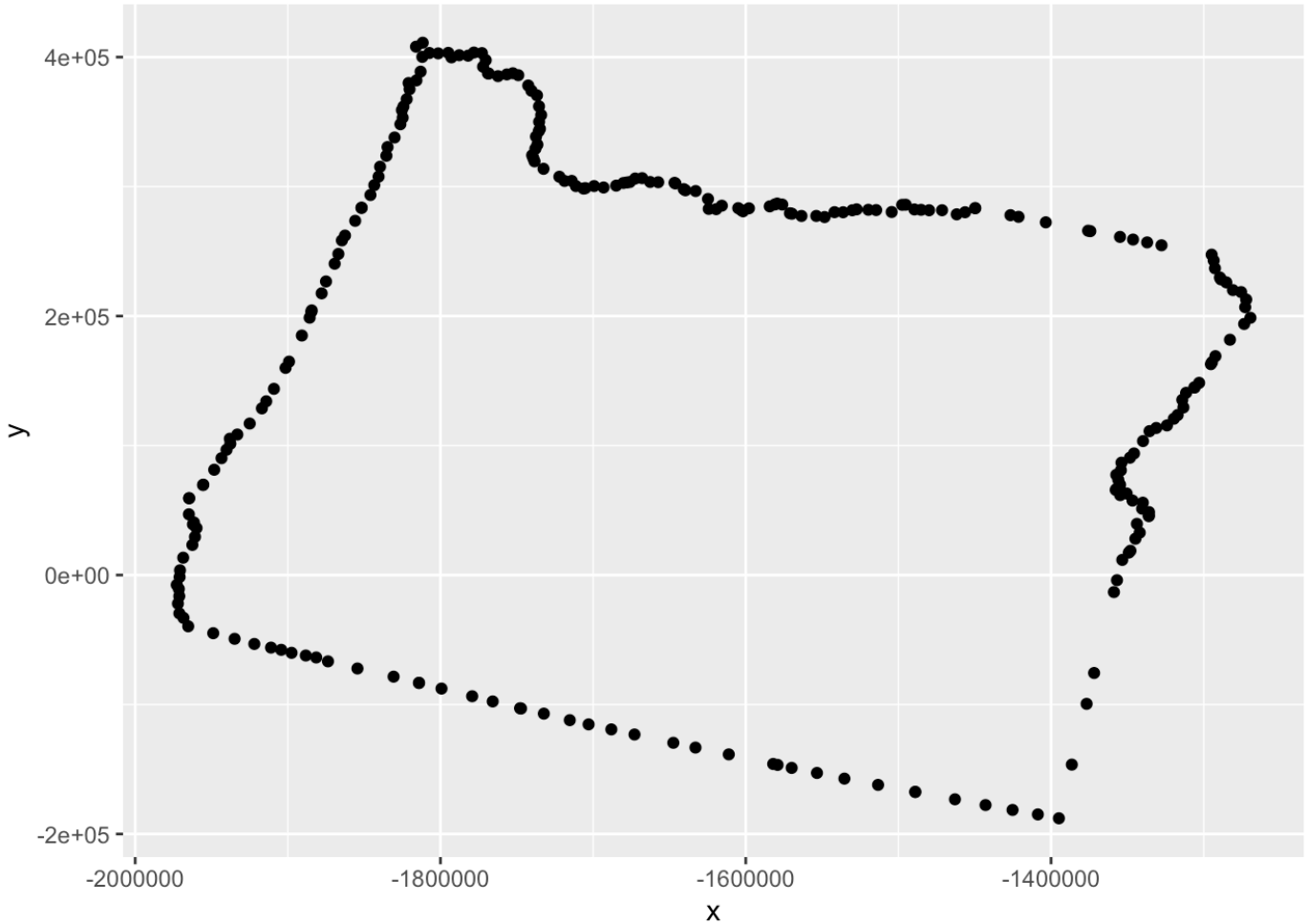
```
states <- usmap::us_map()

head(states)
```

```
##         x        y order  hole piece group fips abbr    full
## 1 1091779 -1380695     1 FALSE     1  01.1   01   AL Alabama
## 2 1091268 -1376372     2 FALSE     1  01.1   01   AL Alabama
## 3 1091140 -1362998     3 FALSE     1  01.1   01   AL Alabama
## 4 1090940 -1343517     4 FALSE     1  01.1   01   AL Alabama
## 5 1090913 -1341006     5 FALSE     1  01.1   01   AL Alabama
## 6 1090796 -1334480     6 FALSE     1  01.1   01   AL Alabama
```

Let's investigate the data for Oregon.

Points

```
oregon<-states%>%
  filter(full=="Oregon")

ggplot(oregon, aes(x, y))+
  geom_point()
```

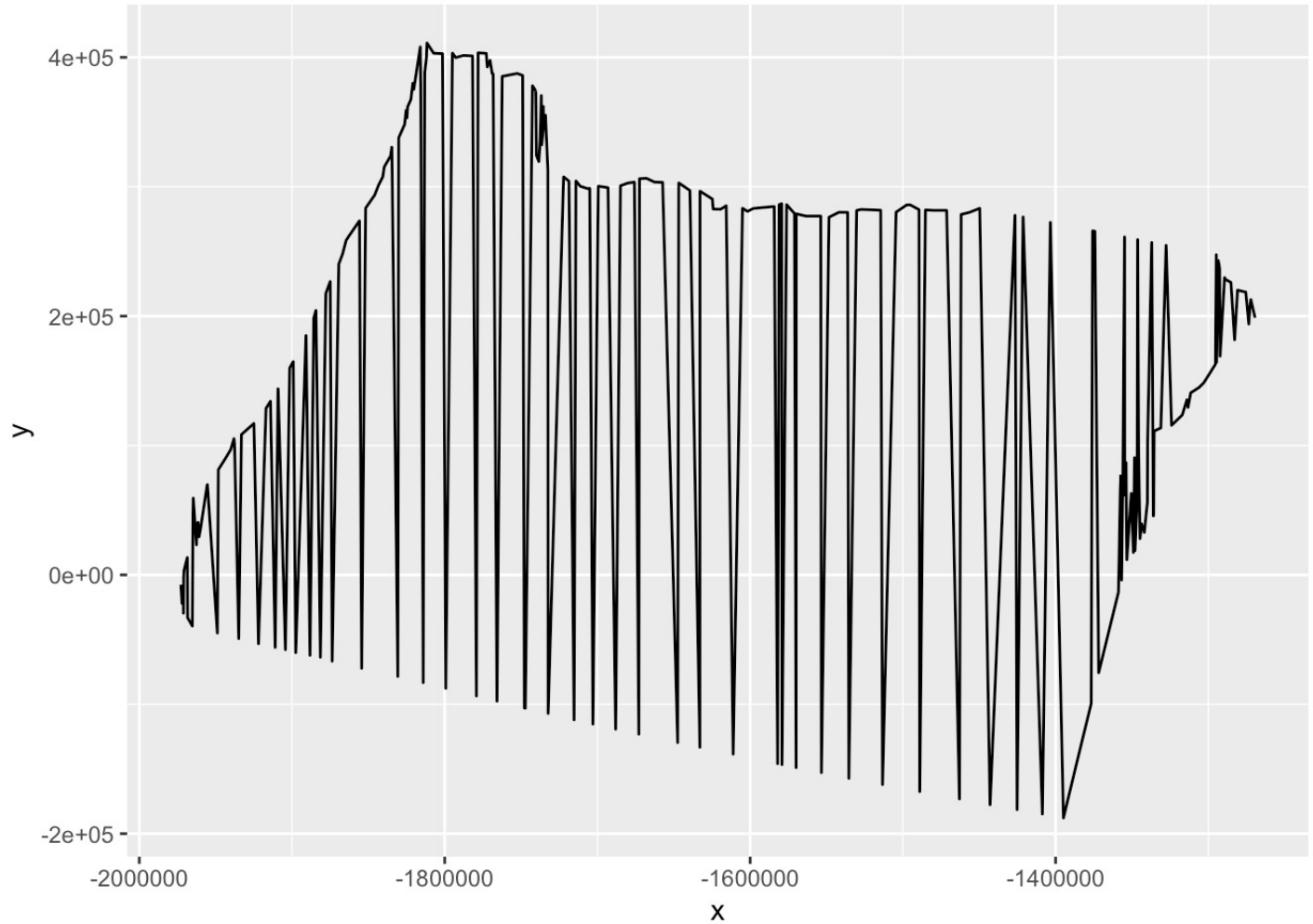


These data allow us to play “connect the dots” to draw the shape of the state of Oregon.

Connect the dots

Oh no, what happened?

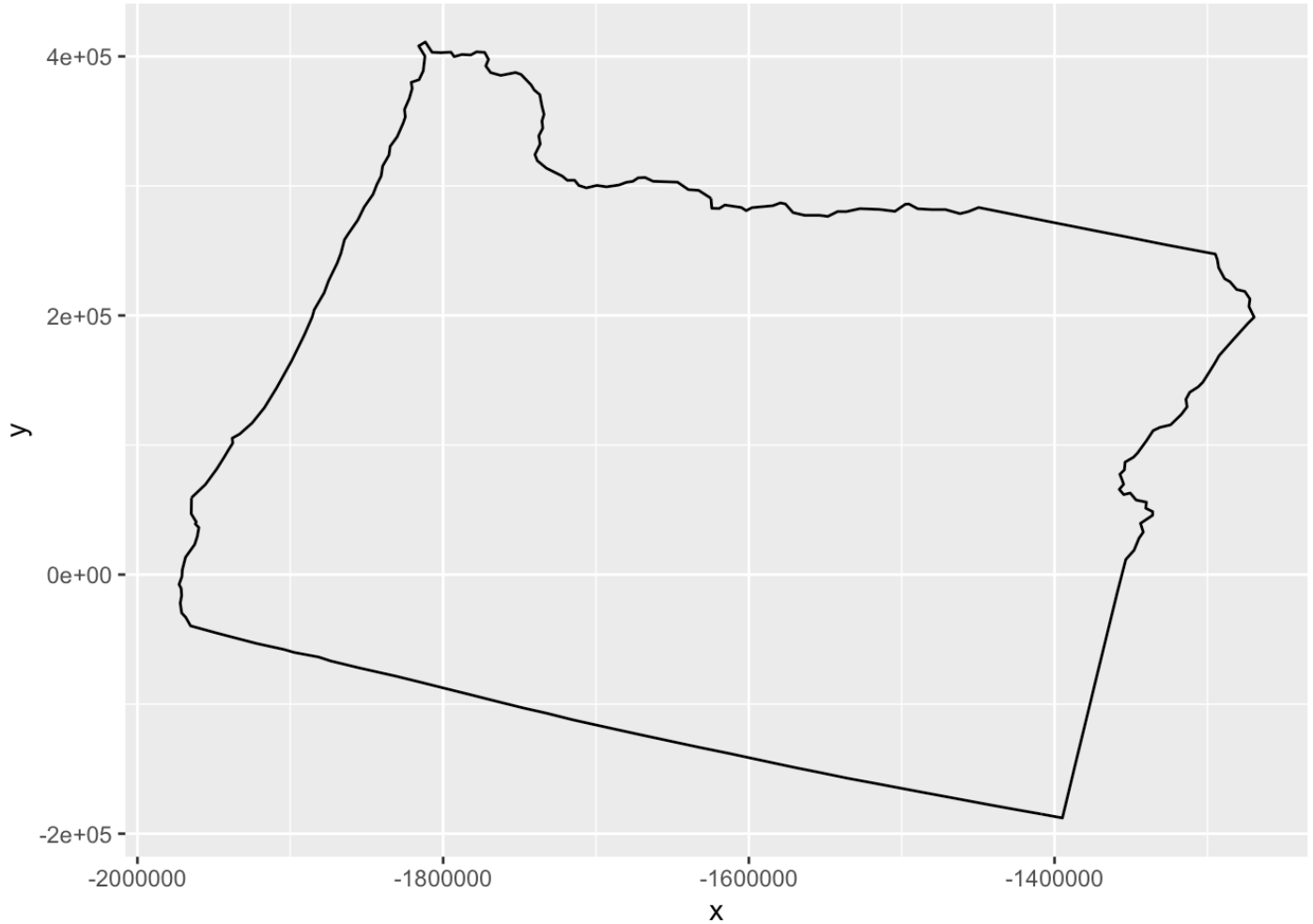
```
ggplot(oregon, aes(x, y))+
  geom_line()
```



We need to tell R what order to connect the dots.

- `geom_path()` connects the observations in the order in which they appear in the data.
- `geom_line()` connects them in order of the variable on the x axis.

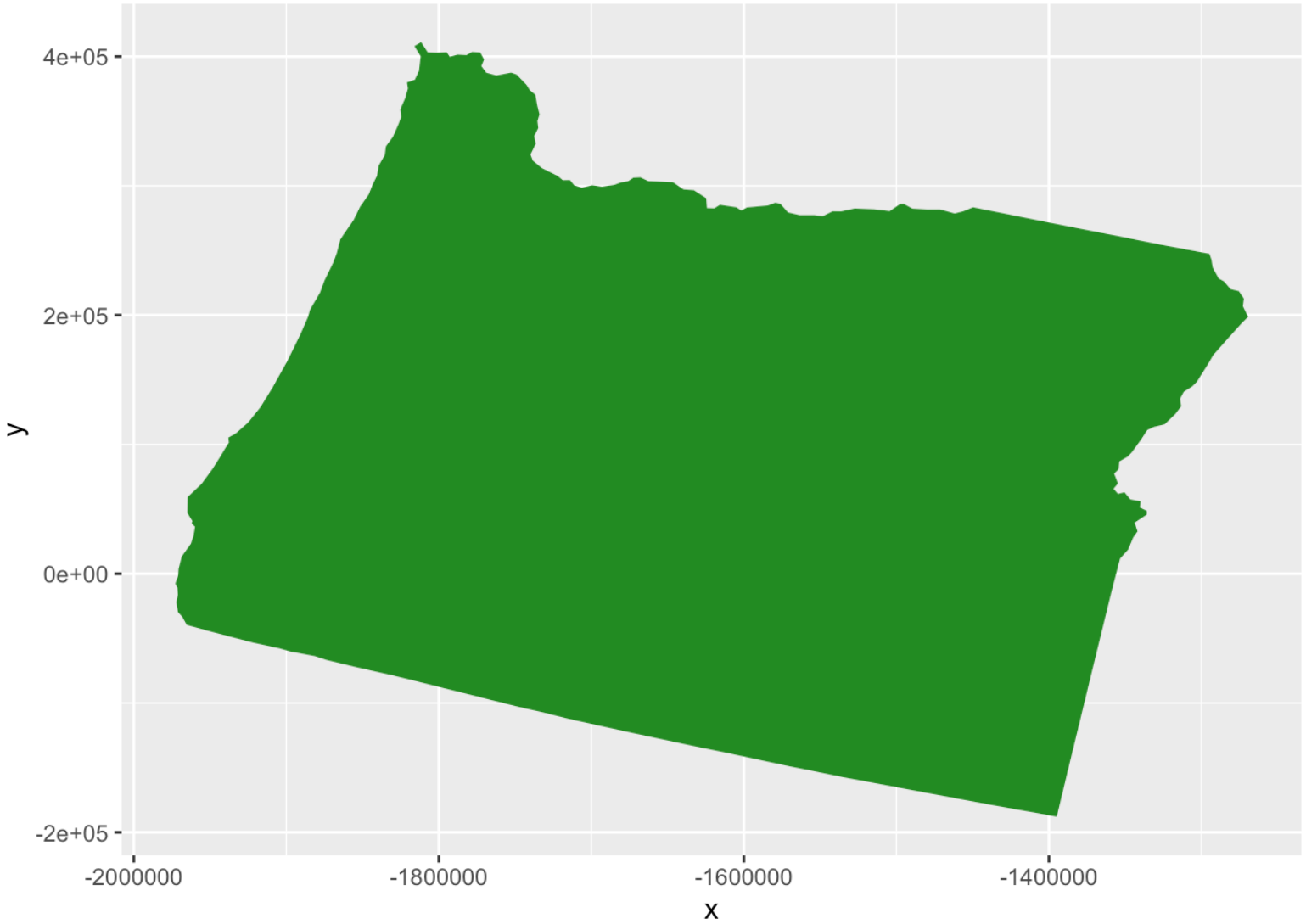
```
ggplot(oregon, aes(x, y, group=group))+
  geom_path()
```



Filling in the space

We can actually think of geographies as generalized polygons!

```
ggplot(oregon, aes(x, y, group=group))+
  geom_polygon(fill="forestgreen")
```



Step 4: Join the Map and Data

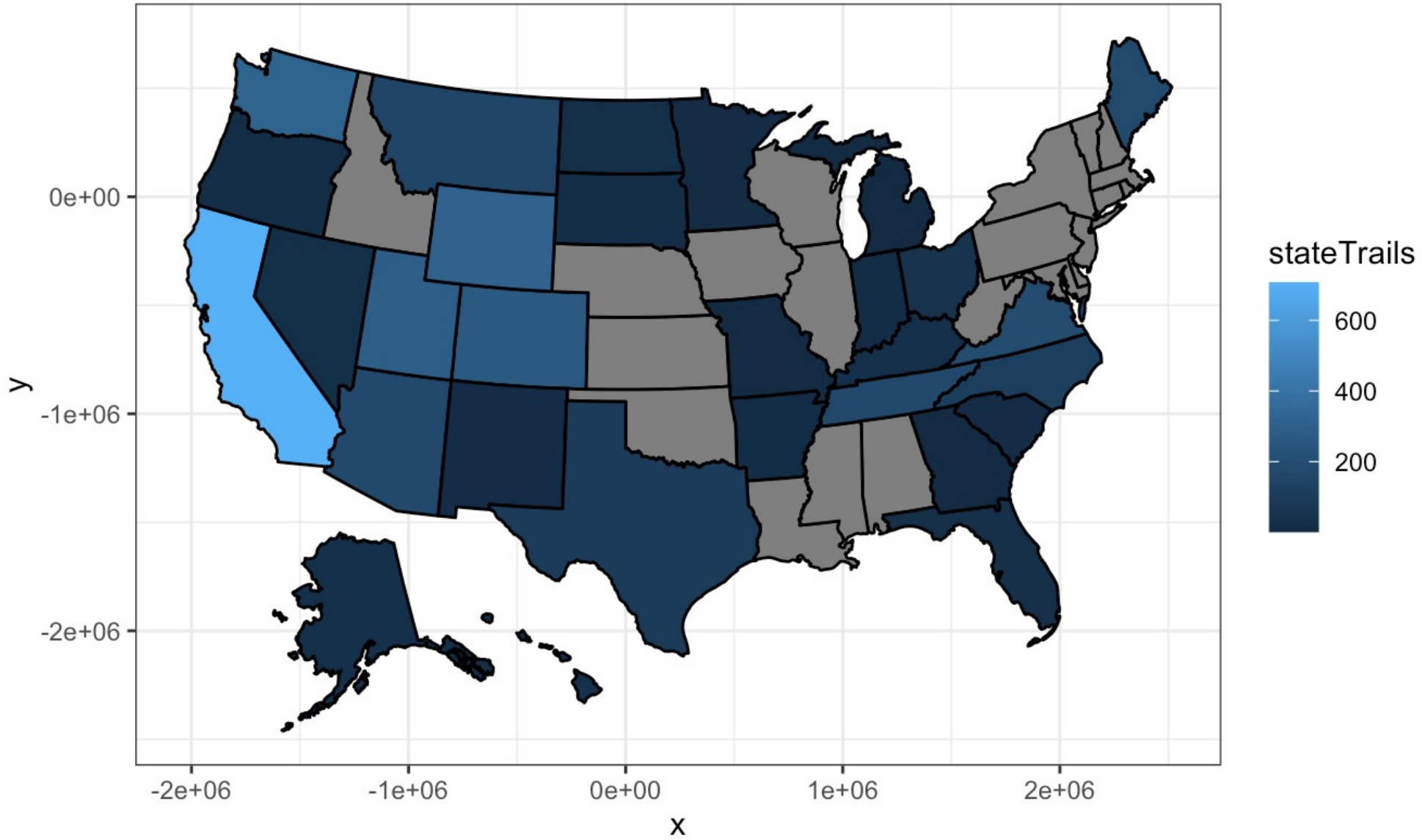
When joining the data to the map we need to have the same variable name in both. Let's create a new column named `state_name`.

```
stateNP_Map<-states%>%
  mutate(state_name=full)%>%
  left_join(stateNP)
```

```
## Joining, by = "state_name"
```

Step 5: Make a Map

```
stateNP_Map%>%
  ggplot(aes(x, y, group = group)) +
  geom_polygon(aes(fill = stateTrails),color="black")+
  theme_bw()+
  coord_equal()
```

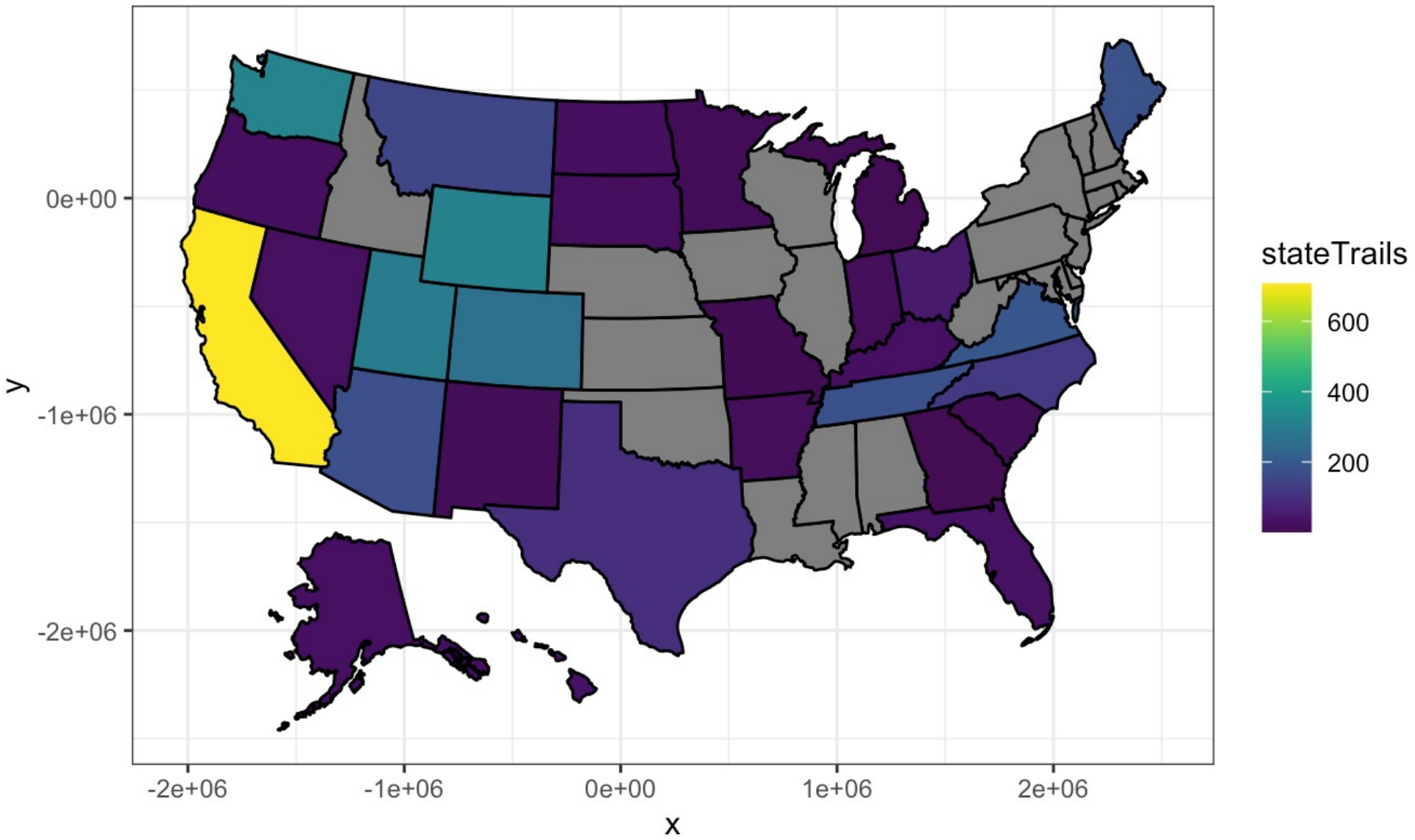

0e+00
-1e+06
-2e+06
y
-2e+06
-1e+06
0e+00
1e+06
2e+06
x
stateTrails
600
400
200

STEP 6: Changing Color Palette

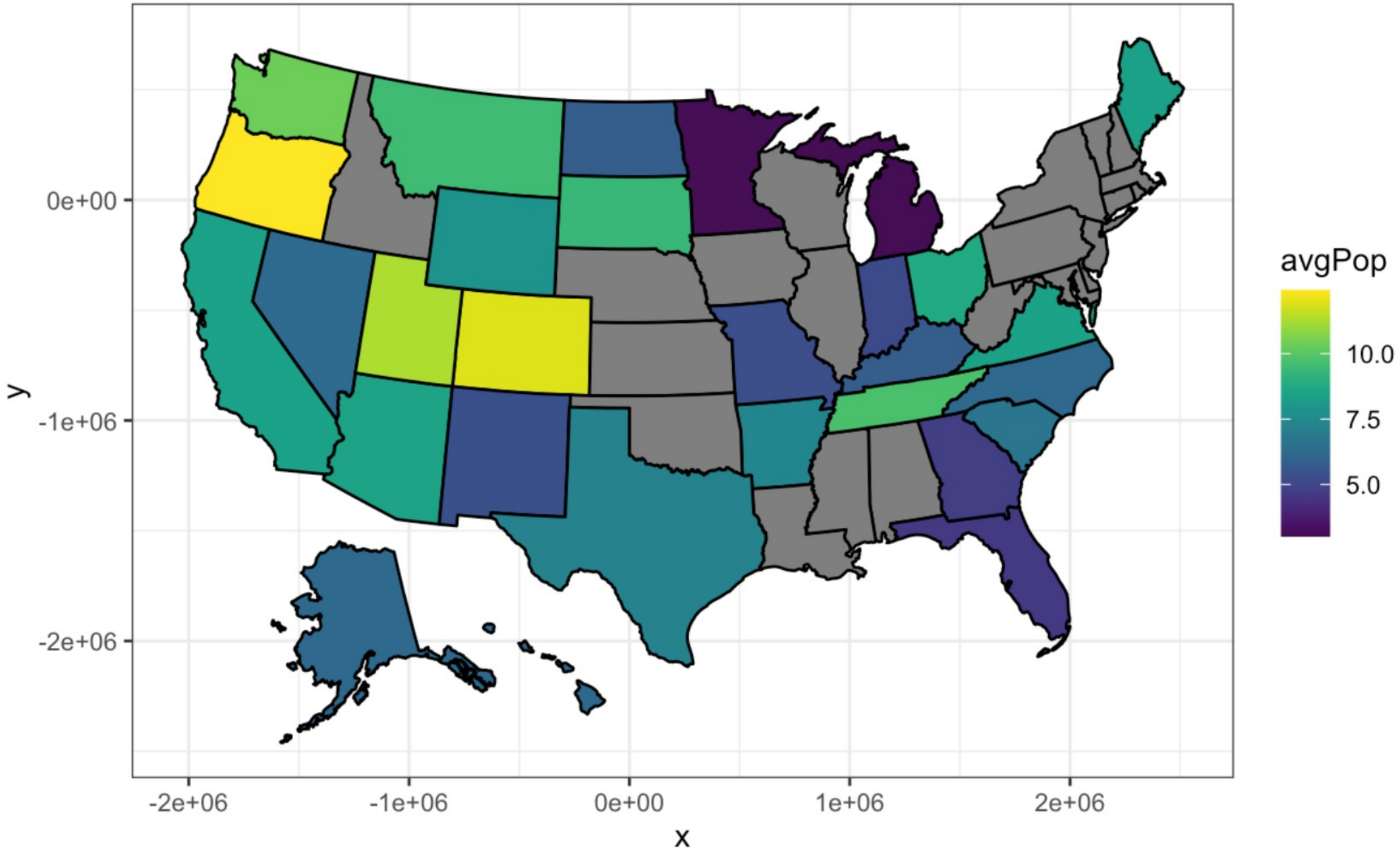
Viridis is a colorblind friendly color palette that can be used to create accessible heatmaps.

```
#install.packages("viridis")
library(viridis)

stateNP_Map%>%
  ggplot(aes(x, y, group = group)) +
  geom_polygon(aes(fill = stateTrails),color="black")+
  theme_bw()+
  coord_equal()+
  ggtitle("California has the MOST trails, but...")+
  scale_fill_viridis(option="viridis", direction = 1)
```

California has the MOST trails, but...



..Oregon trails are the MOST popular



Your turn!

Create maps to show the distribution of…

- Average elevation by state
- Average trail length by state

TIME FOR GROUP WORK

MILESTONE #5

DATA 151: Project Milestone #5

Due 11 - Milestone #5: Exploratory Data Analysis Step #3
Distributions, Summary Statistics, and Comparing Subgroups

Goal: Work to answer at least one of your questions of interest for numeric variables of interest.

- Describe the shape of the distribution for a numeric variable of your choice. Convey the appropriate summary statistics
- Explore possible subgroups

Please submit using Rmarkdown