Mapping Essential Concepts

School of Information Studies
Syracuse University

Geographic Coordinate Systems

Latitude lines are parallel, horizontal circles

- 0 is the equator; +90 degrees is the North Pole; −90 degrees is the South Pole (latitude lines are full circles)
- Think of ‘x’ values

Longitude lines are vertical arcs; not parallel

- 0 is the prime meridian in Greenwich, UK; +90 degrees is east and bisects Asia; −90 degrees is west and passes through middle of Canada and US; +/−180 degrees is mid-Pacific (longitude lines are half circles)
- Think of ‘y’ values

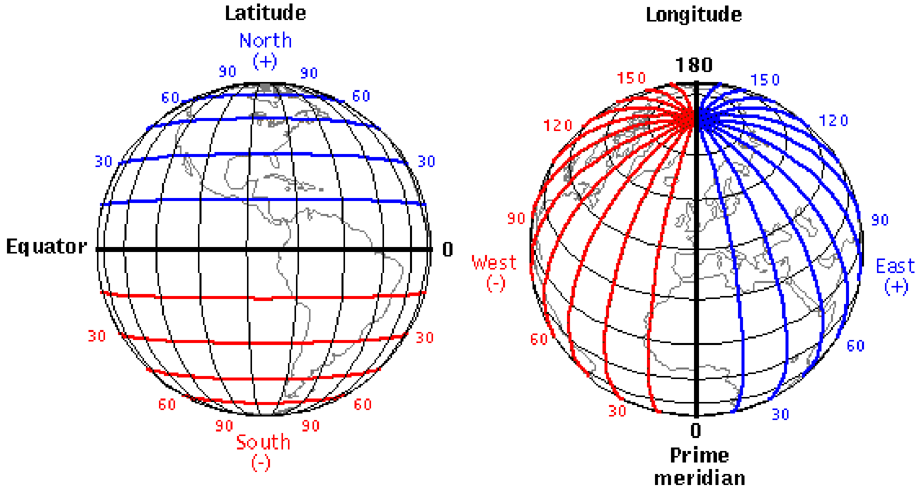


Image credit: 60n95w.com

Map Projections

- At small scales, longitude and latitude work like a regular Cartesian grid.
- At large scales, the spherical shape of Earth interferes with plotting on a flat surface.
- A map projection renders a spherical area onto a flat surface.
- All map projections create distortions.

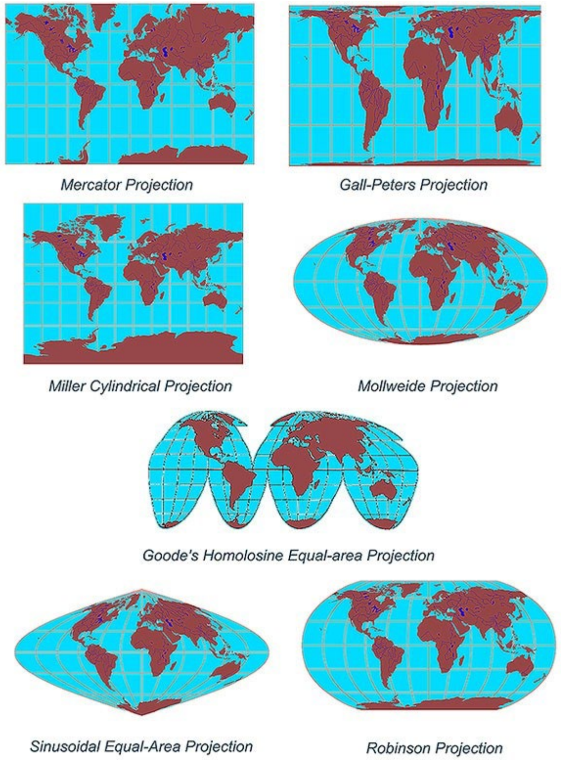

Mercator Projection
Gall-Peters Projection
Miller Cylindrical Projection
Mollweide Projection
Goode's Homolosine Equal-area Projection
Sinusoidal Equal-Area Projection
Robinson Projection

Image credit: geoawesomeness.com

All Projections Create Distortions

Map distortion in context: Alaska vs. Lower 48

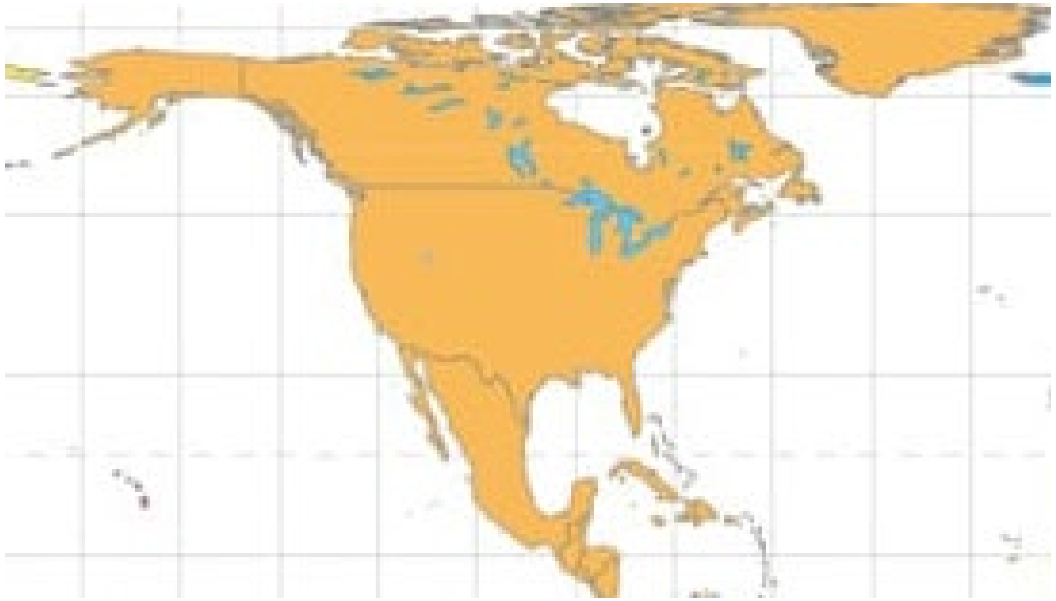
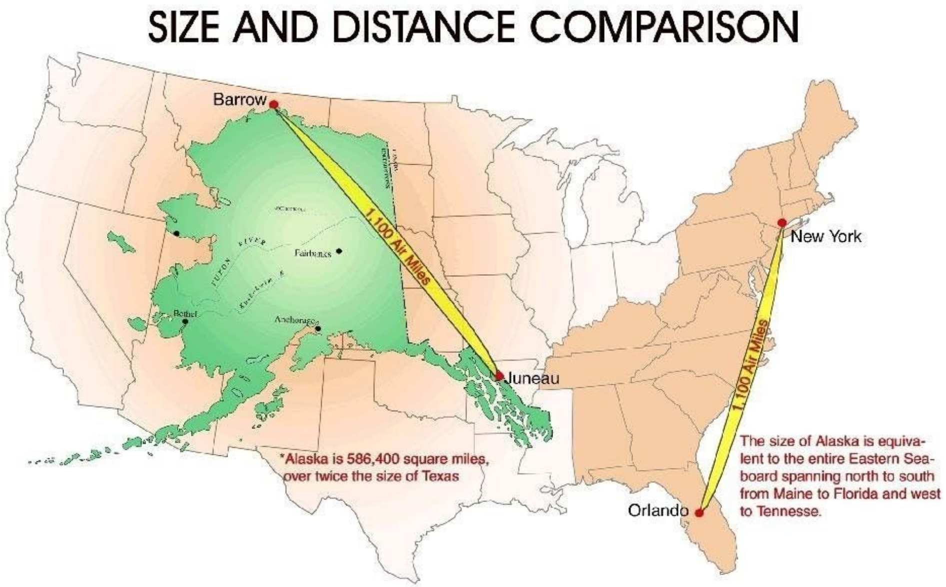


Image credit: The Guardian



SIZE AND DISTANCE COMPARISON

*Alaska is 586,400 square miles, over twice the size of Texas

The size of Alaska is equivalent to the entire Eastern Seaboard spanning north to south from Maine to Florida and west to Tennessee.

Image credit: Matador Network

Four Major Classes of Spatial Data

1. **Point—**individual point, usually (x,y), or collection of points
2. **Line—**an ordered collection of points, assumed to be connected
3. **Polygon—**an area enclosed by lines
4. **Raster—**a collection of locations, usually organized in a rectangular lattice (e.g., an 'image')

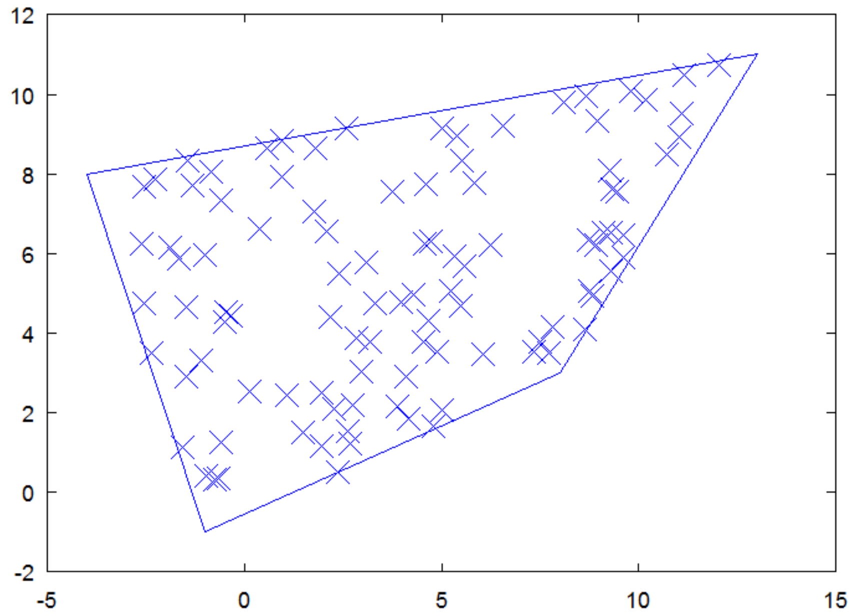


Image credit: particleincell.com

Note: A filled polygon is called a choropleth.

Vector vs. Raster

- Vector graphics work by drawing lines between points.
- Raster graphics treat every image as a grid of pixels.
- Both types of maps exists: “shapefile” maps contain vector graphics info; “tile maps” contain raster graphics.
- We will use ggplot2 to make vector-based maps (and ggmap for raster graphics)

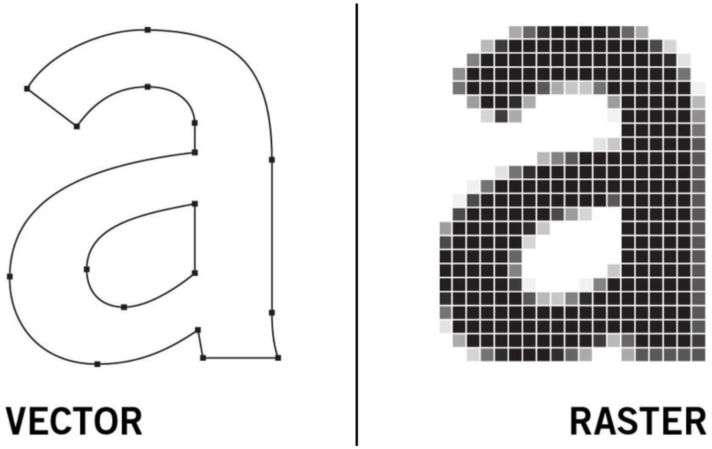
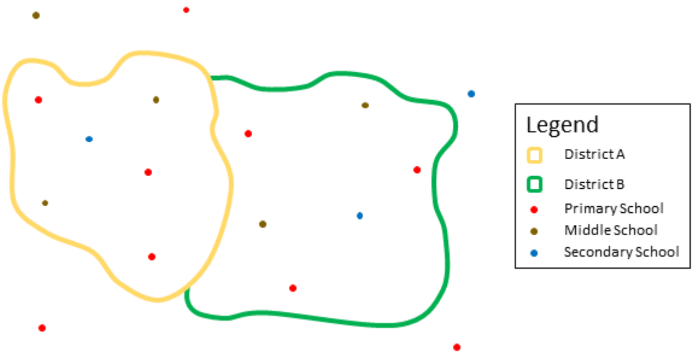


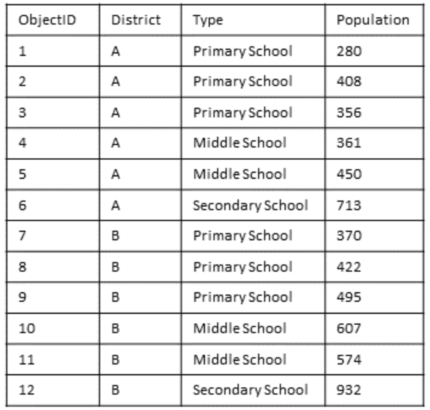
Image credit: Seeka Creative

Spatial Data Attributes

In addition to x and y (or other coordinate) position, any point, line, polygon, or grid can have attributes. For example:

- Points on Earth's surface can have an altitude.
- Spatial objects can have names.
- Spatial objects can have attached data, such as the population contained within a region.





ObjectID	District	Type	Population
1	A	Primary School	280
2	A	Primary School	408
3	A	Primary School	356
4	A	Middle School	361
5	A	Middle School	450
6	A	Secondary School	713
7	B	Primary School	370
8	B	Primary School	422
9	B	Primary School	495
10	B	Middle School	607
11	B	Middle School	574
12	B	Secondary School	932

Image credit: arcgis.com

Question

A minimum of three data fields are needed to create any data-based map. What are they?

Mapping Essential Concepts (cont.)

School of Information Studies
Syracuse University

Answer

X location (longitude)

Y location (latitude)

An attribute (e.g., income that can be mapped to a visual)

Note that the point itself could represent information (e.g., location of an ATM machine). In that case, only two attributes are needed.

Visualization With Maps Using R

School of Information Studies
Syracuse University

Polygon Data Using map_data()

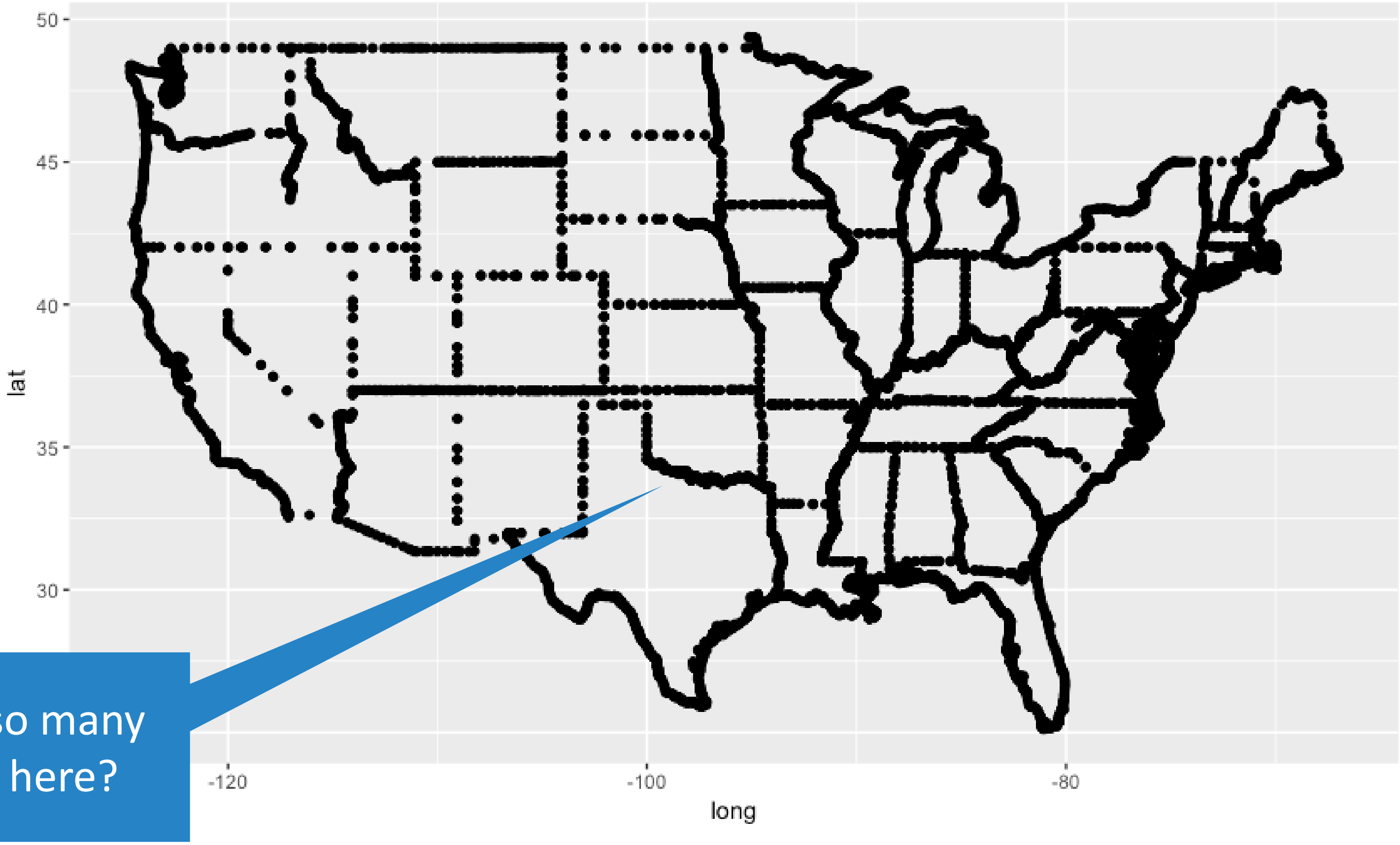
```
#Shapes of states
state_geomDF <- map_data("state")

head(state_geomDF )
```

Group is a bunch of points that belong together.

```
Longitude    Latitude   Group    Order    Region    Subregion
1 -87.46201  30.38968   1        1        alabama   <NA>
2 -87.48493  30.37249   1        2        alabama   <NA>
3 -87.52503  30.37249   1        3        alabama   <NA>
4 -87.53076  30.33239   1        4        alabama   <NA>
5 -87.57087  30.32665   1        5        alabama   <NA>
6 -87.58806  30.32665   1        6        alabama   <NA>
```

Order shows the order in which to plot the points of the polygon.

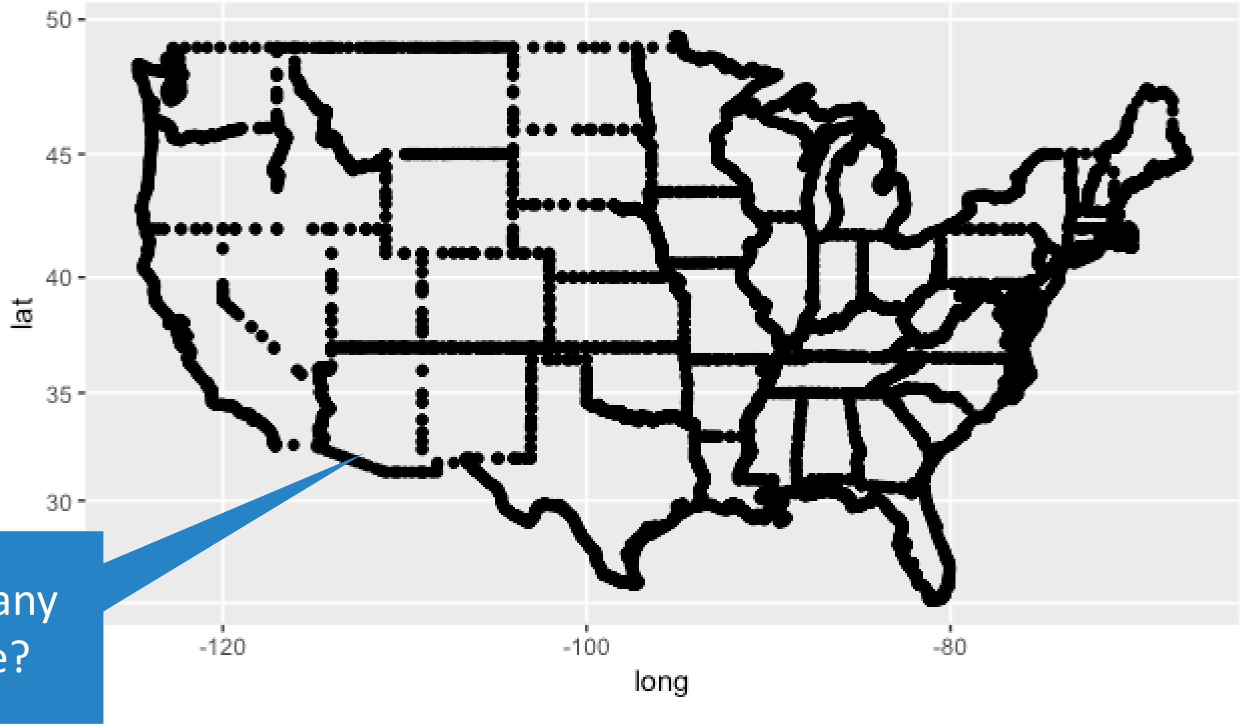
ggplot(state_geomDF) + geom_point(aes(x=long, y=lat))



Why so many dots here?

Correct Aspect Ratio

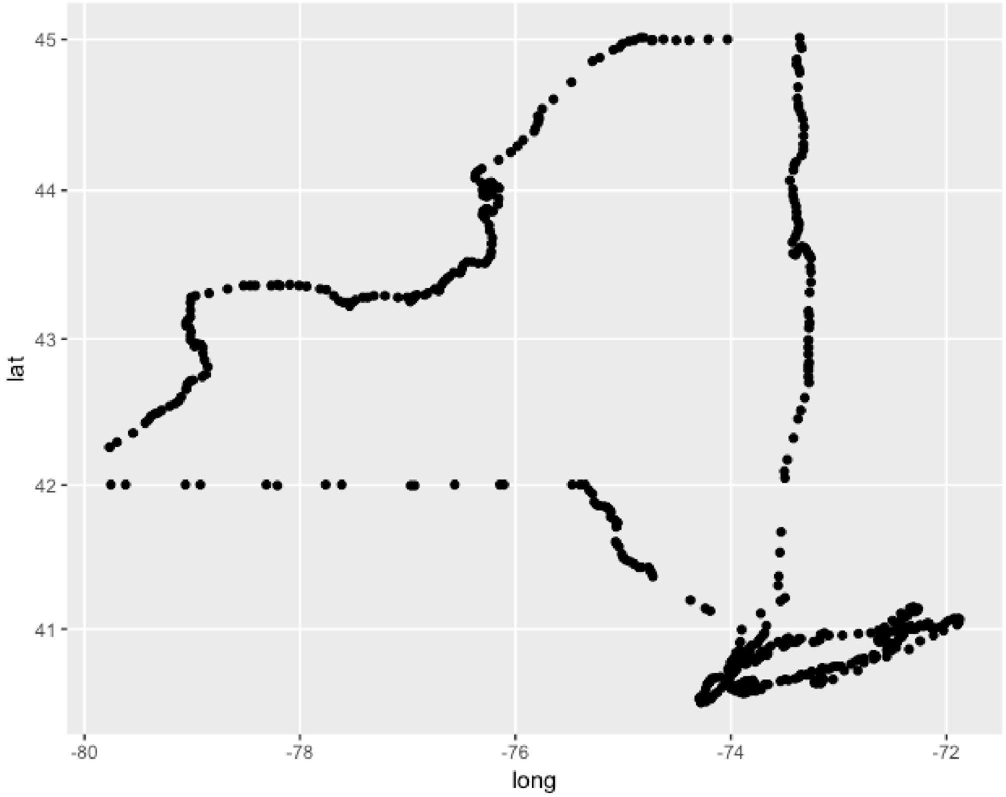
```
ggplot(state_geomDF ) + geom_point(aes(x=long,y=lat)) + coord_map()
```



Why so many dots here?

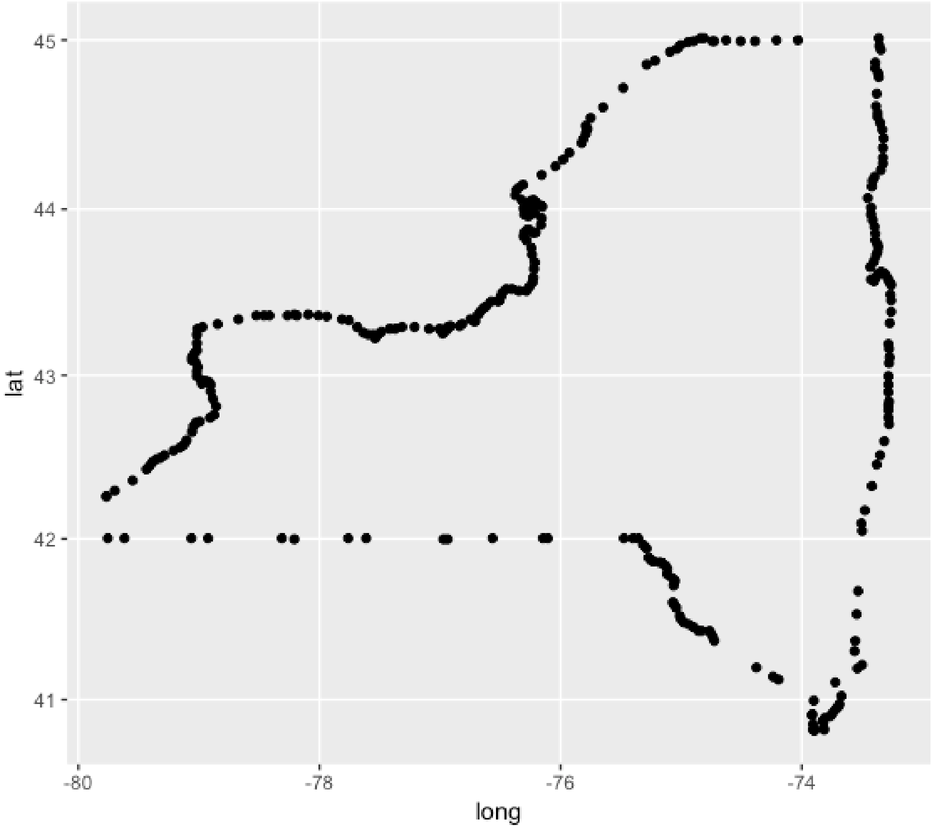
Zoom Into NY

```
NYData <- state_geomDF %>%
          filter(region=="new york")

ggplot(NYData) +
      geom_point(aes(x=long,y=lat)) +
      coord_map()
```



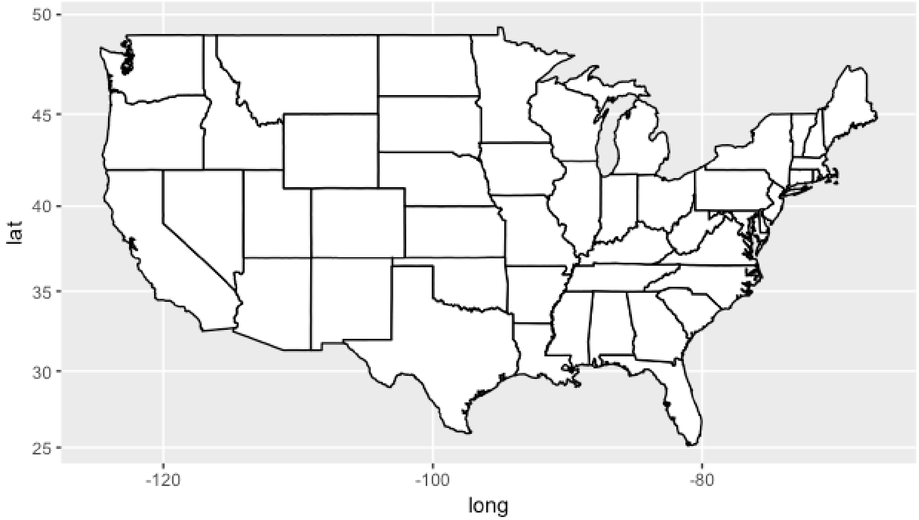
Zoom Into a Part of NY

```
state_geomDF  %>% filter(group==35) %>%
    ggplot()  +
        geom_point(aes(x=long,y=lat)) +
        coord_map()
```



Creating a Simple Map

```
map.simple <- ggplot(state_geomDF) +
    geom_polygon(color="black", fill="white",
             aes(x=long,y=lat, group=group)) +
    coord_map()
map.simple
```



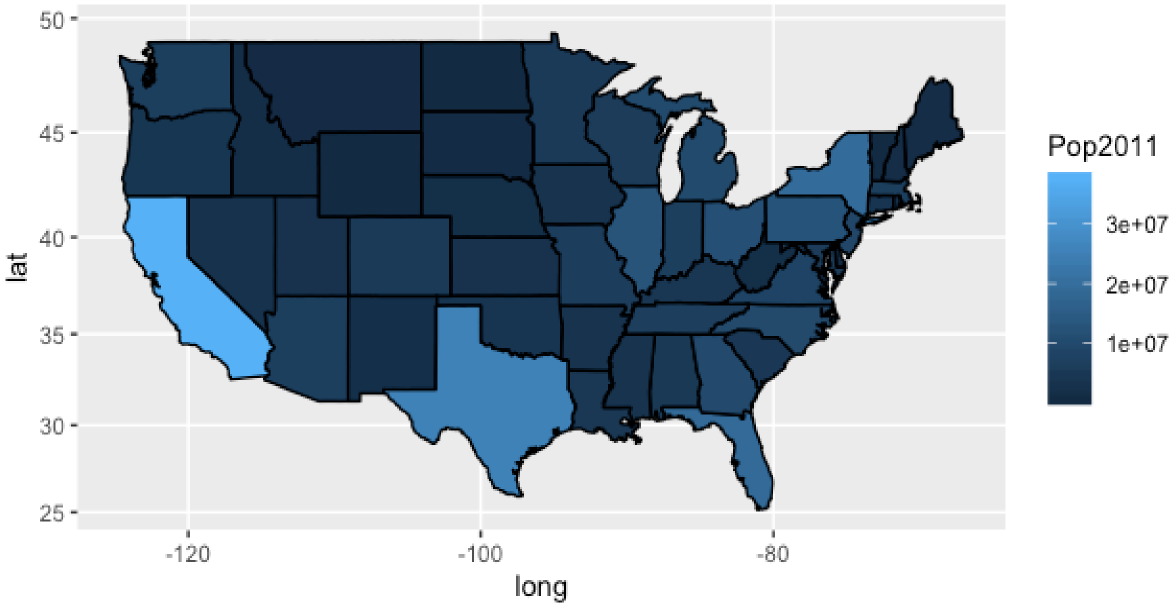
```
Longitude     Latitude     Group      Order      Region     Subregion
1 -87.46201   30.38968     1          1          alabama    <NA>
2 -87.48493   30.37249     1          2          alabama    <NA>
3 -87.52503   30.37249     1          3          alabama    <NA>
```

Creating a DataFrame With Geometry

```
#Create a df with state center & population
usData <- data.frame(stateName=state.name,  area=state.area)
usData$centerX=state.center$x
usData$centerY=state.center$y

#Make sure everything is lowercase
usData$stateName <- tolower(usData$stateName)

#Combine dataframes using the merge function
usDataWithGeom <- merge(usData, state_geomDF,  by.x="stateName",by.y="region")

#Don’t lose the order for the points in polygon
usDataWithGeom <- usDataWithGeom %>%  arrange(order)
```

Creating a Filled Map (Based on Area)

```
ggplot(usDataWithGeom) +
  geom_polygon(color="black",
            aes(x=long,y=lat, group=group,
                    fill=area)) +
  coord_map()
```



Question

What is a choropleth and what is it good for?

Visualization With Maps Using R (cont.)

School of Information Studies
Syracuse University

Answer

What is a choropleth and what is it good for?

Choropleth maps are used to represent statistical data through various shading patterns or symbols on predetermined geographic areas (i.e., countries).

They are good at utilizing data to easily represent variability of the desired measurement, across a region.

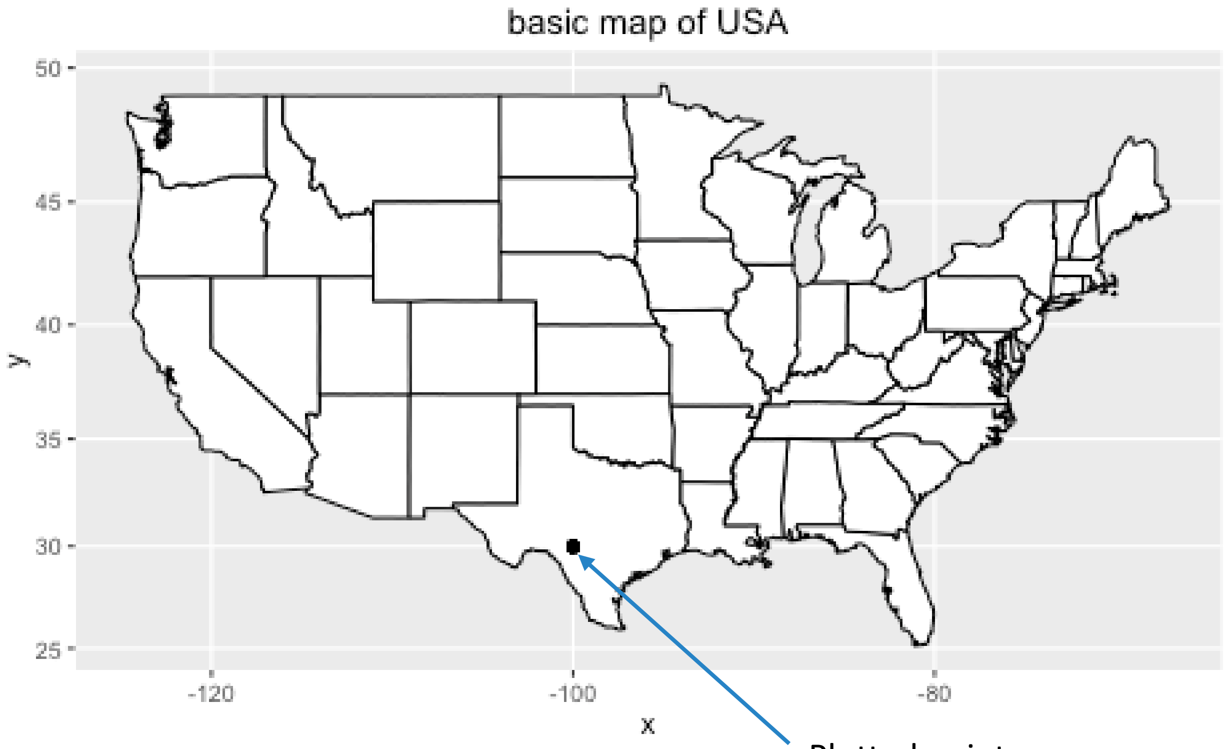
Source: https://www.arcgis.com/apps/MapJournal/index.html

Creating Layers on Maps

School of Information Studies
Syracuse University

Add a Point to the Map

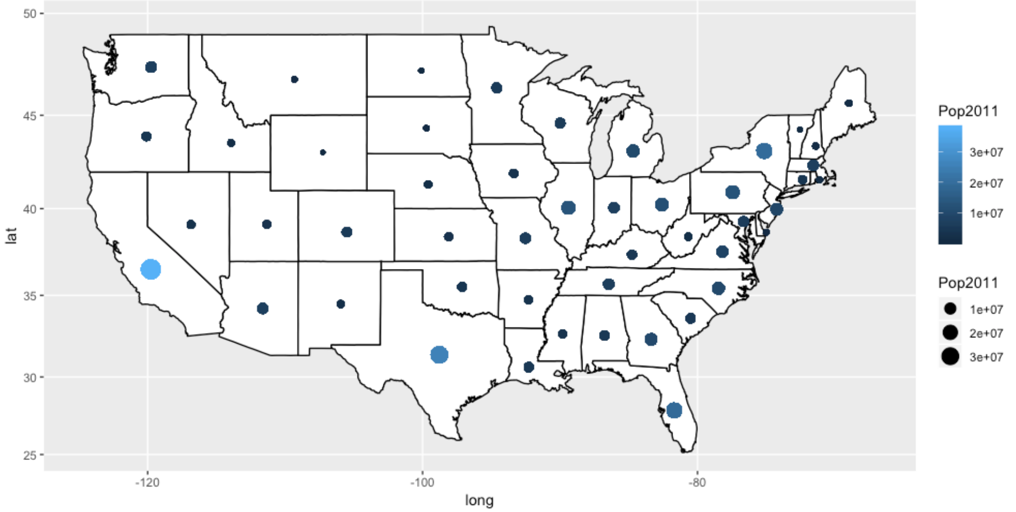
```
map.simple + geom_point(aes(x = -100, y = 30))
```



Plotted point

Add Points to the Map

```
#Add a points layer:
# color and size showing population
ggplot(usDataWithGeom) +
   geom_polygon(color="black",
fill="white",
            aes(x=long,y=lat, group=group)) +
   geom_point(aes(x=centerX,y=centerY,
            color=area,size=area)) +
coord_map()
```



Question

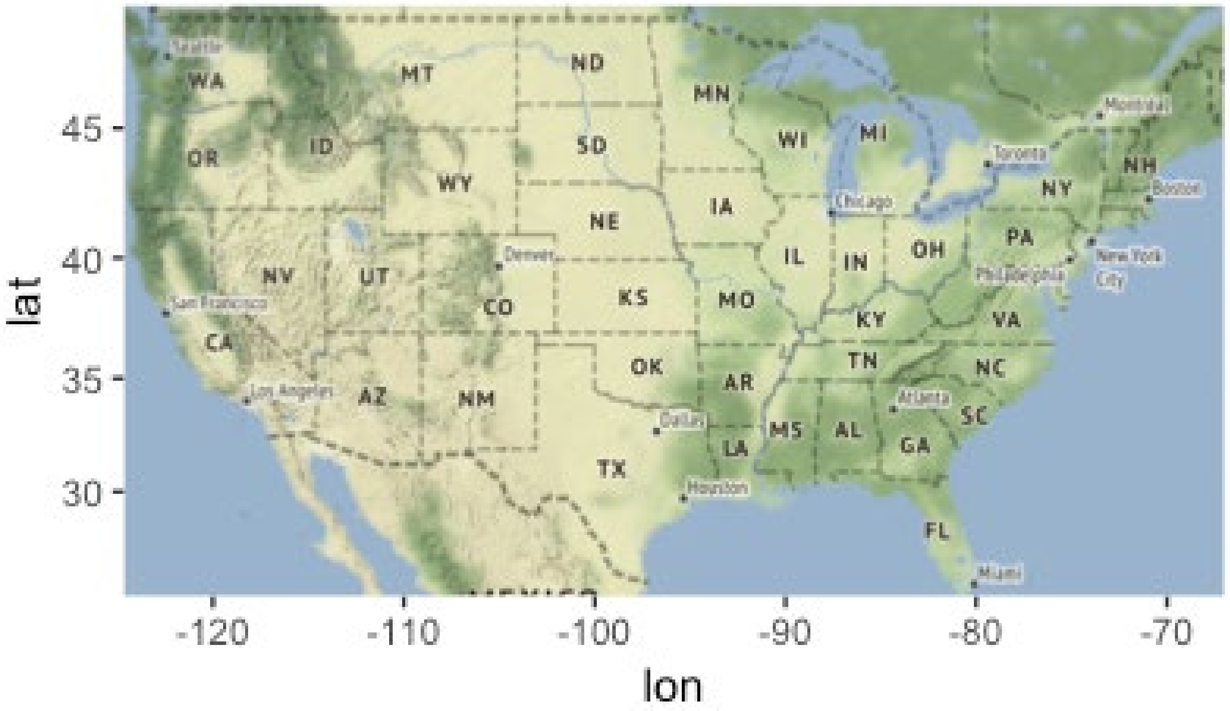
Give an example of two data series that could be plotted 1) as the color of choropleths; and 2) as the size of dots.

Creating Layers on Maps (cont.)

School of Information Studies
Syracuse University

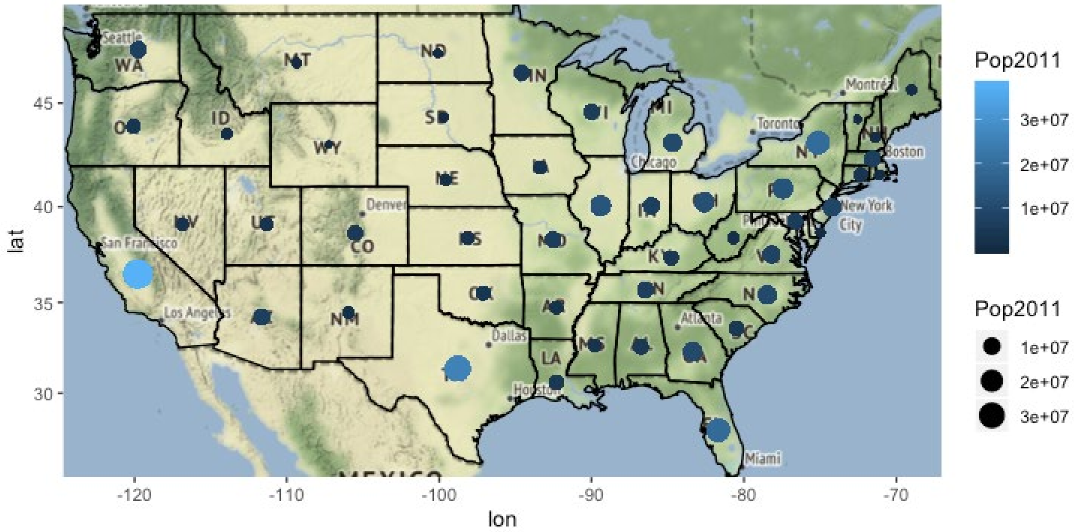
Create an Image (Raster) Map

```
#Put a map image behind the visualizaiton
bb <- c(left = min(usDataWithGeom$long),
        bottom = min(usDataWithGeom$lat),
        right = max(usDataWithGeom$long),
        top = max(usDataWithGeom$lat))

# get the map  - Note: If the zoom is too large,
#     it will take a long time to load the maps.
library(ggmap)
map <- get_stamenmap(bbox = bb,  zoom=4)

#Show the map using ggmap
ggmap(map)
```



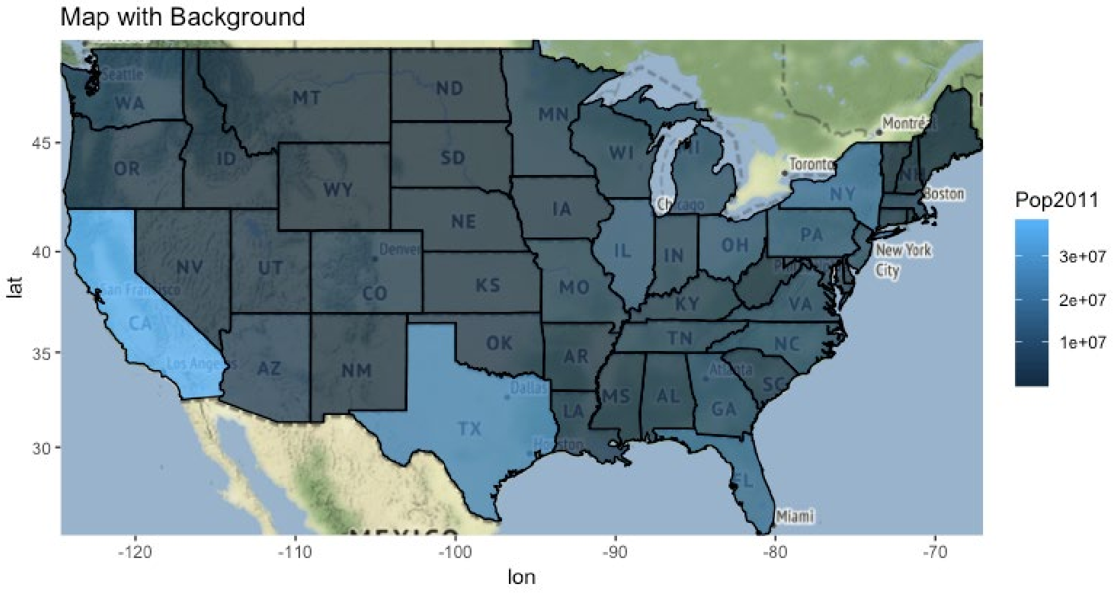
Add a Layer to the Raster Map

```
#Add a points layer with color=population
ggmap(map) +
  geom_polygon(data= usDataWithGeom,
               color="black", fill="NA",
               aes(x=long,y=lat, group=group)) +
  geom_point(data= usDataWithGeom,
      aes(x=centerX,y=centerY, color=area,
      size=area))
```



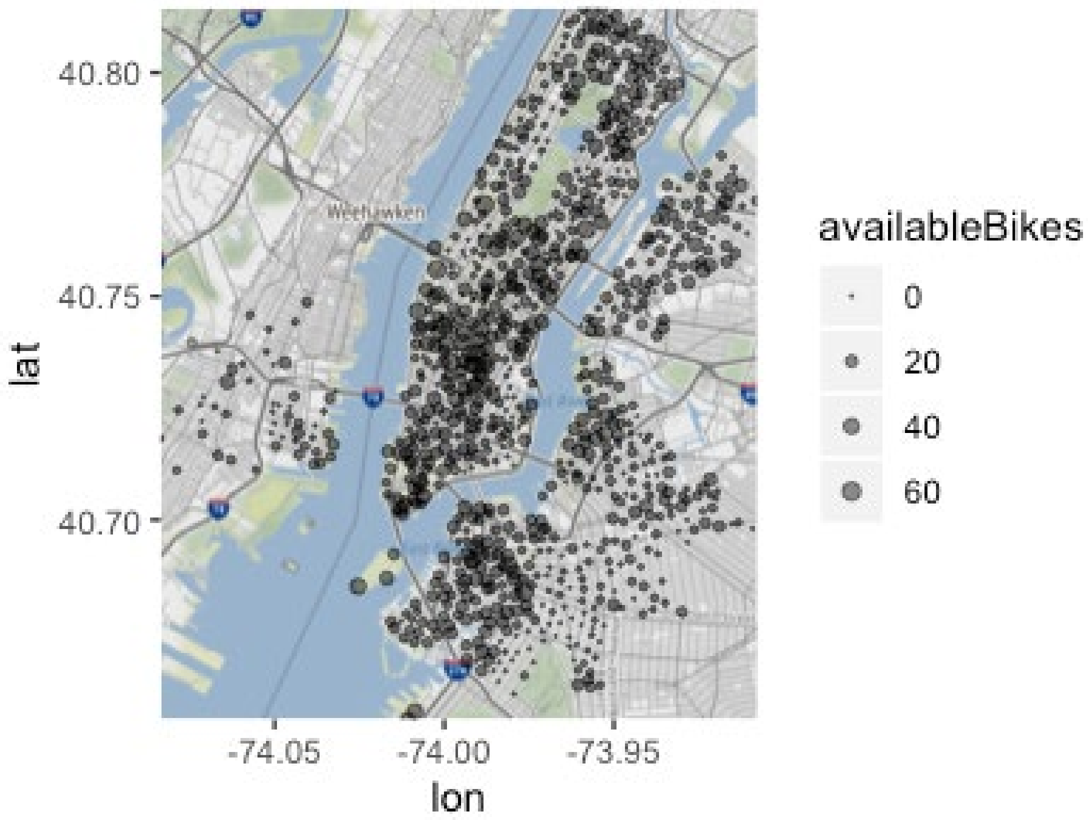
Add a Choropleth to the Raster Map

```
ggmap(map) +
 geom_polygon(data=
usDataWithGeom,
          color="black", alpha=0.8,
          aes(x=long,y=lat, group=group,
          fill=area)) +
 ggtitle("Map with Background")
```



Zoom on the Map (Visualize the Bike Data)

```
#Get the bounding box for the map
bb <- c(left = min(allBikeData$longitude),
        bottom = min(allBikeData$latitude),
        right = max(allBikeData$longitude),
        top = max(allBikeData$latitude))

#Get the new background map—note zoom level
mapNY <- get_stamenmap(bbox = bb, zoom=12)

#Visualize the points and map, scaling the points
ggmap(mapNY) +
  geom_point(data=allBikeData,
             alpha=0.5, color="black",
             aes( x=longitude,  y=latitude,
size=availableBikes))  +
```


40.80
40.75
40.70
lat
-74.05
-74.00
-73.95
lon
Weehawken
availableBikes
0
20
40
60