

Visualization with Maps

Professor Jeff Saltz
Professor Jeff Stanton

Copyright 2021: Jeffrey Saltz and Jeffrey Stanton; please do not upload.

School of Information Studies
school.syr.edu SYRACUSE UNIVERSITY

Summary of Previous Learning

What you should know and be able to do at this point :

1. List major skills needed by data scientists
2. Describe a DS project with domain analysis and SMEs
3. Explain basic concepts of data modeling
4. Explain and use a data frame in R
5. Define the most common descriptive statistics and calculate them with R using appropriate functions
6. Demonstrate the development of a simple function in R
7. Describe the effects of randomness on sampling
8. Create and interpret a sampling distribution including defining the law of large numbers and the central limit theorem
9. Visualize a distribution and interpret a histogram
10. Describe multiple strategies for accessing external data from R
11. Use ggplot to visualize data

Data Science in the Real World

School of Information Studies
school.syr.edu SYRACUSE UNIVERSITY

ATMs

<http://www.licong.com/png-automated-teller-machine-commercial-bank-money-cx-274507/>

ATMs

<http://www.licong.com/png-automated-teller-machine-commercial-bank-money-cx-274507/>

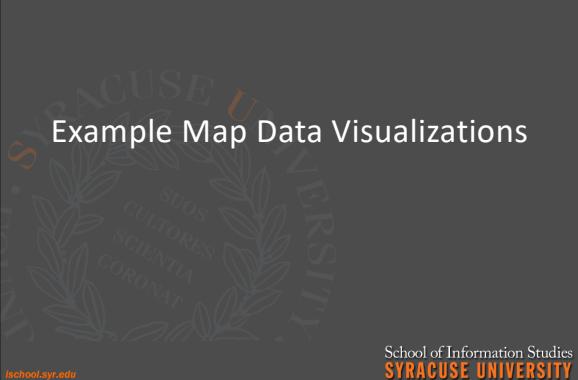
A University & Data Science?

The United States
of College Football

Goals for this Week

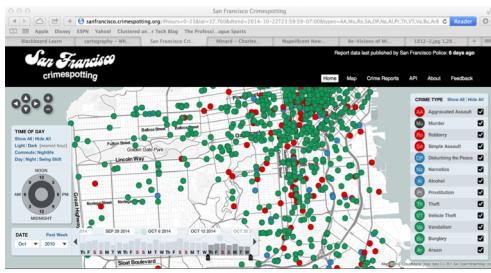
- Demonstrate the integration of **disparate data sources** and formats producing a representative information model for decision-making
- Utilize **ggplot for GIS**
- Plot **geographic** and **numerical** data within one visualization

Example Map Data Visualizations



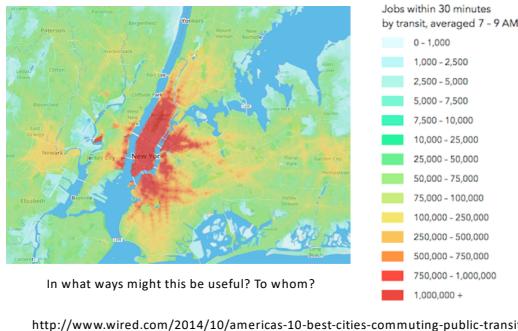
Data Mapping Example 1

<http://sanfrancisco.crimespotting.org>



In what ways might this be useful? To whom?

Data Mapping Example 2



<http://www.wired.com/2014/10/americas-10-best-cities-commuting-public-transit/>

[View Details](#) | [Edit](#) | [Delete](#)

Data Mapping Example 3



In what ways might this be useful? To whom?

Question

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

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)
- 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 is mid-Pacific (longitude lines are half circles)

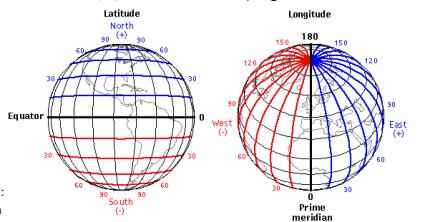


Image credit: 60n95w.com

Map Projections

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

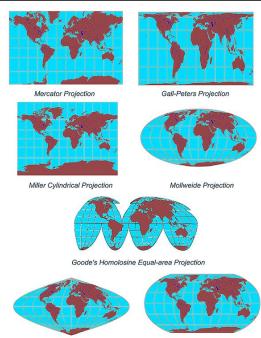


Image credit: geoawesomeness.com

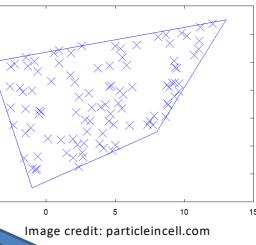
All Projections Create Distortions

- Map Distortion in Context: Alaska vs Lower 48



Four Major Classes of Spatial Data

- Point – individual point, usually (x,y), or collection of points
- Line – An ordered collection of points, assumed to be connected
- Polygon – An area enclosed by lines
- Raster or Grid – a collection of locations, usually organized in a rectangular lattice



Note: a filled polygon is called a choropleth.

Vector vs Raster

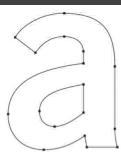


Image credit: Seeka Creative

- Vector graphics work by drawing lines between points
- Raster graphics treat every image as a grid of pixels
- Both types of maps exist: "Shapefile" maps contain vector graphics info; "Tile maps" contain raster graphics
- We will use ggplot2 to make vector-based maps

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.



ID	Type	Population	
1	A	Primary School	280
2	A	Primary School	350
3	A	Primary School	360
4	A	Middle School	360
5	A	Middle School	450
6	A	Secondary School	720
7	B	Primary School	420
8	B	Primary School	400
9	B	Primary School	400
10	B	Middle School	570
11	B	Middle School	570
12	B	Secondary School	1000

Image credit: [Mapbox](#)

Mapping in R

School of Information Studies
SYRACUSE UNIVERSITY

[school.syr.edu](#)

Polygon Data Using map_data()

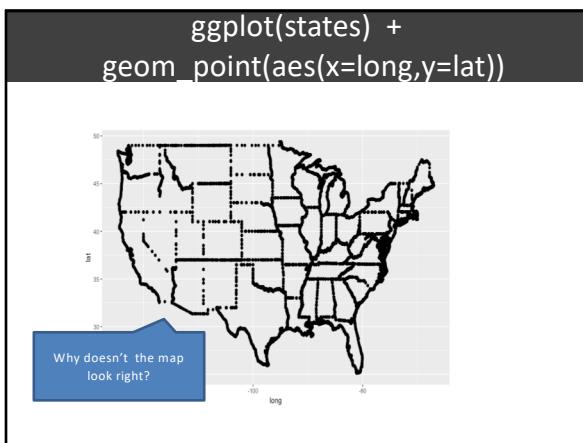
```
# Shapes of states
states <- map_data("state")
```

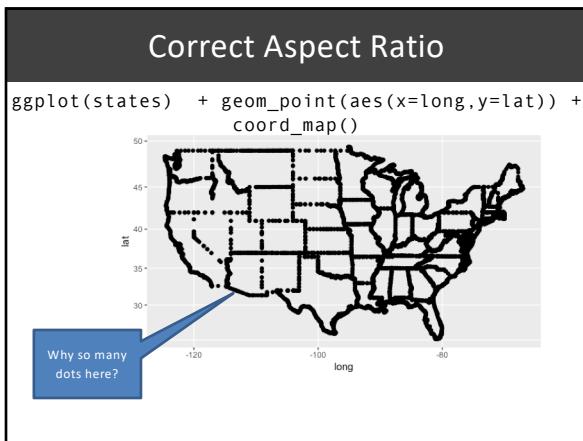
```
head(states)
```

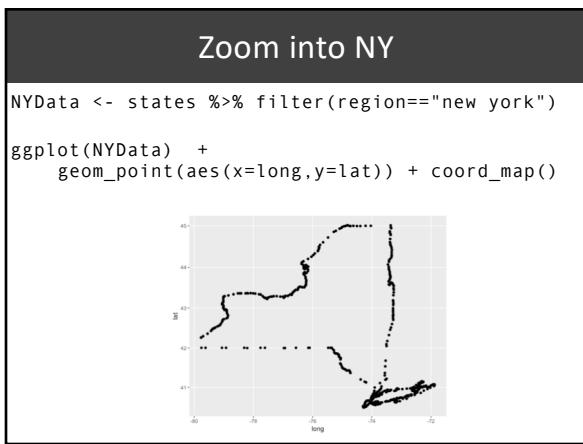
group is a bunch of points that belong together

long	lat	group	order	region	subregion
-87.46201	30.38968	1	1	alabama	<NA>
-87.48493	30.37249	1	2	alabama	<NA>
-87.52503	30.37249	1	3	alabama	<NA>
-87.53076	30.33239	1	4	alabama	<NA>
-87.57087	30.32665	1	5	alabama	<NA>
-87.58806	30.32665	1	6	alabama	<NA>

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

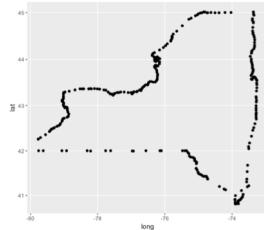






Zoom into a Part of NY

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



Creating a Simple Map

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



	long	lat	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

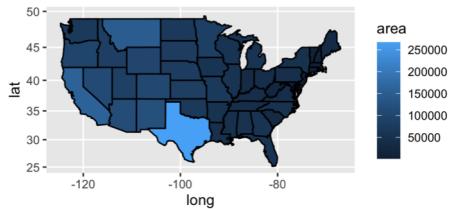
#Make sure everything is lowercase
usData$stateName <- tolower(usData$stateName)

#Combine dataframes using the merge function
usDataWithGeom <- merge(usData,
                         states, 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()
```

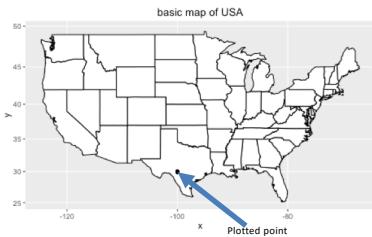


Question

- What is a choropleth and what is it good for?

Add a Point to the Map

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



Get State Centers

```
#create a dataframe with state centers and population
stateCenter <- data.frame(
  state=tolower(state.name),
  x=state.center$x,
  y=state.center$y)

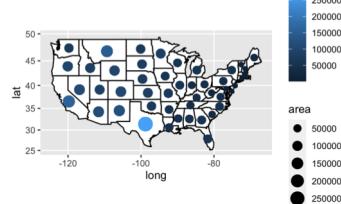
#merge with the the dataframe we have been using
dfStatesWithCenter <- merge(usDataWithGeom,
  stateCenter, by.x="stateName", by.y="state")

#don't forget to sort on the order attribute
dfStatesWithCenter <- dfStatesWithCenter %>%
  arrange(order)
```

Note: "points on the map" is often done via geocoding

Add Points to the Map

```
#add a points layer: color and size showing population
ggplot(dfStatesWithCenter) +
  geom_polygon(color="black", fill="white",
    aes(x=long,y=lat, group=group)) +
  geom_point(aes(x=x,y=y, 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.

Create an Image (Raster) Map

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

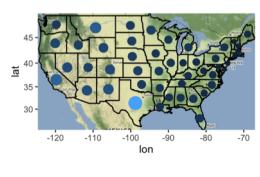
#actually get the map
# note if the zoom is too large,
# it will take a long time to load
# the maps across the internet
library(ggmap)
map <- get_stamenmap(bbox = bb,
                      zoom=4)

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



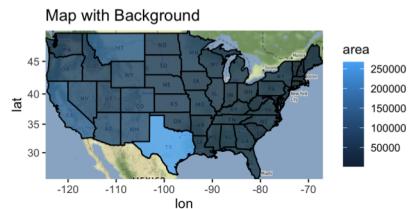
Add a Layer to the Raster Map

```
#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")
```

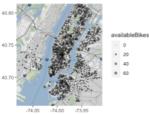


Zoom on the Map (Visualize the Bike Data)

```
#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 & map, scaling the points
ggmap(mapNY) +
  geom_point(data=allBikeData,
             alpha=0.5, color="black",
             aes( x=longitude,
                  y=latitude,
                  size=availableBikes)) +
  scale_size(range=c(0, 2))
```



Zoom on the Map (Visualize the Bike Data)

