Econ 106

Lecture 18

slides adapted from: https://github.com/dlab-berkeley/R-Geospatial-Fundamentals/tree/master/docs

https://datacarpentry.org/r-raster-vector-geospatial/index.html

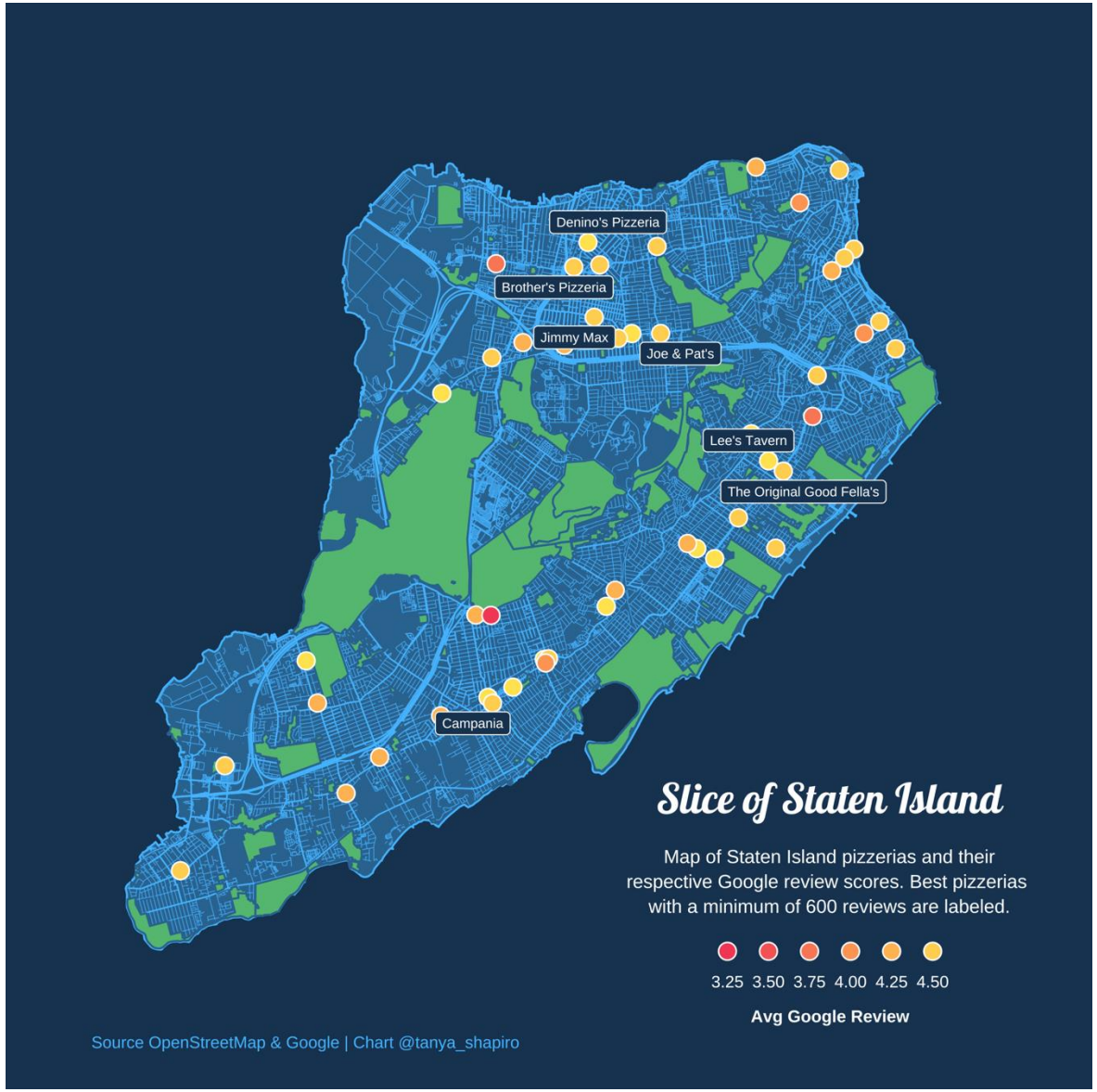
Updates

- MS #3 is due Sunday, 11:59pm
- No class on Wednesday
- Week 10: Visualization for Communication

https://pollev.com/vsovero

#30DayMapChallenge

- On the map:
  - locations of pizzerias (points)
  - road network
  - parks



https://www.tanyashapiro.com/more/30daymapchallenge

Outline for today

- loading maps from TIGRIS
- Annotations
- Map projections
- Spatial Queries:
  - measuring area
  - measuring distance

Recap: Map of California Counties

- We read in a shapefile (CA_Counties) using st_read()
- We plot sf objects using geom_sf()

```
ggplot(data=CA_Counties) +
geom_sf()
```

Need a Map of the US?

- The US Census Bureau has a large collection of geographies of the United States:
  - state
  - county
  - census tract
  - and so on

Need a Map of the US?

- We can use the tigris package to load the data directly into R

```
library(tigris)
```

tigris package

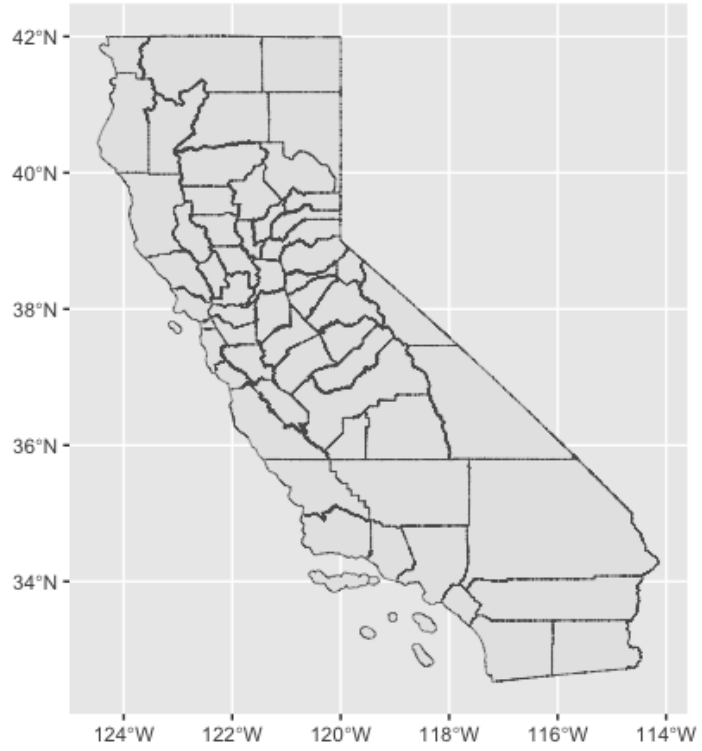
- Choose the function name that corresponds to the geography you want (list available here)
- Arguments:
  - If you don’t want all the US, specify the State and/or county
- Output: a sf object with California counties

```
library(tigris)

CA_Counties_Tigris<-counties("California")
```

Map of California Counties

- We can map this data using ggplot() and geom_sf() because it's a sf object

```
ggplot(data=CA_Counties_Tigris) +
  geom_sf()
```



Borders for United States and Territories

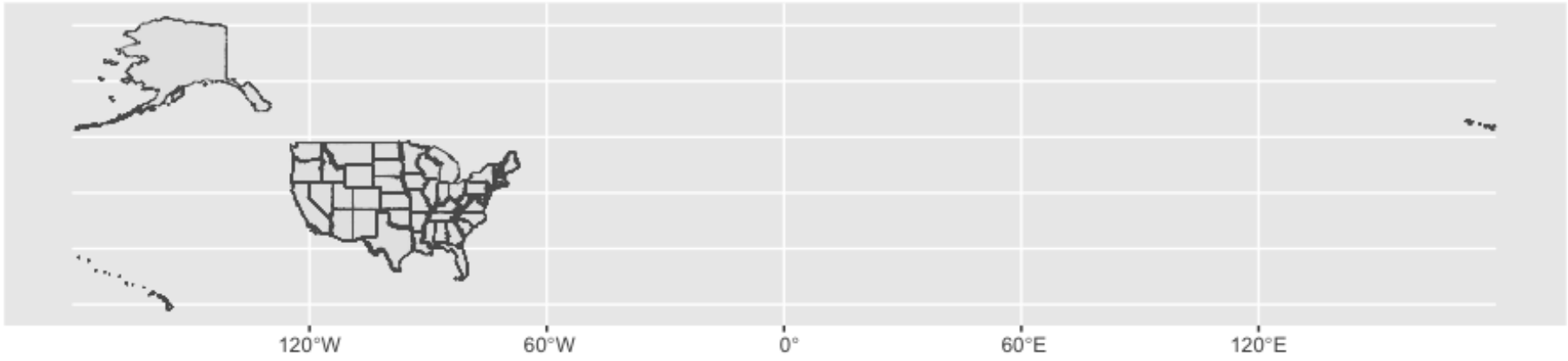
- Leave the argument blank if you want geography for the entire US

```
US_States_Tigris<-states()
```

Map of the United States and Territories

- Map doesn't look great because it includes US territories

```
ggplot(data=US_States_Tigris) +
geom_sf()
```



Map of the United States

- We can filter out territories (region 9)
- It's also helpful to relocate the non-contiguous states

```
only_states <- US_States_Tigris%>%
  filter(REGION != "9") %>%
  shift_geometry()
```

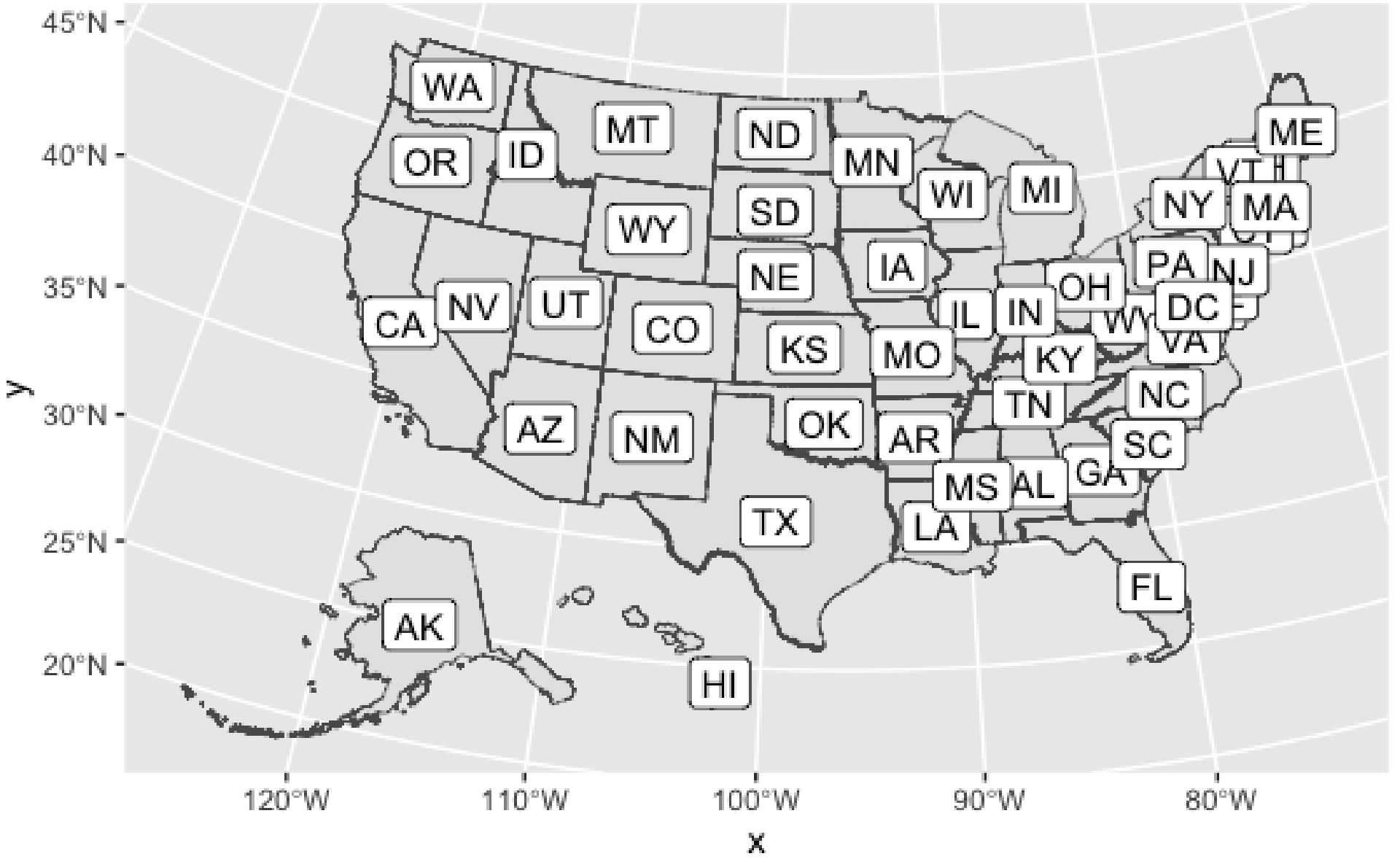
Map of the United States

```
ggplot(data=only_states) +
    geom_sf()
```

Annotations for sf

- **geom_sf_label**() – add annotations to a geom_sf
- Arguments:
  - **label**
- Remember to use aes() when referencing variable names

```
ggplot(data=only_states) +
geom_sf()+
geom_sf_label(aes(label=STUSPS))
```


45°N
40°N
35°N
30°N
25°N
20°N
y
120°W
110°W
100°W
90°W
80°W
x
WA
OR
CA
ID
NV
MT
WY
UT
AZ
CO
NM
ND
SD
NE
KS
OK
TX
MN
IA
MO
AR
LA
WI
IL
MS
MI
IN
KY
TN
AL
OH
WV
GA
NY
PA
VA
NC
SC
FL
ME
VT
MA
NJ
DC
AK
HI

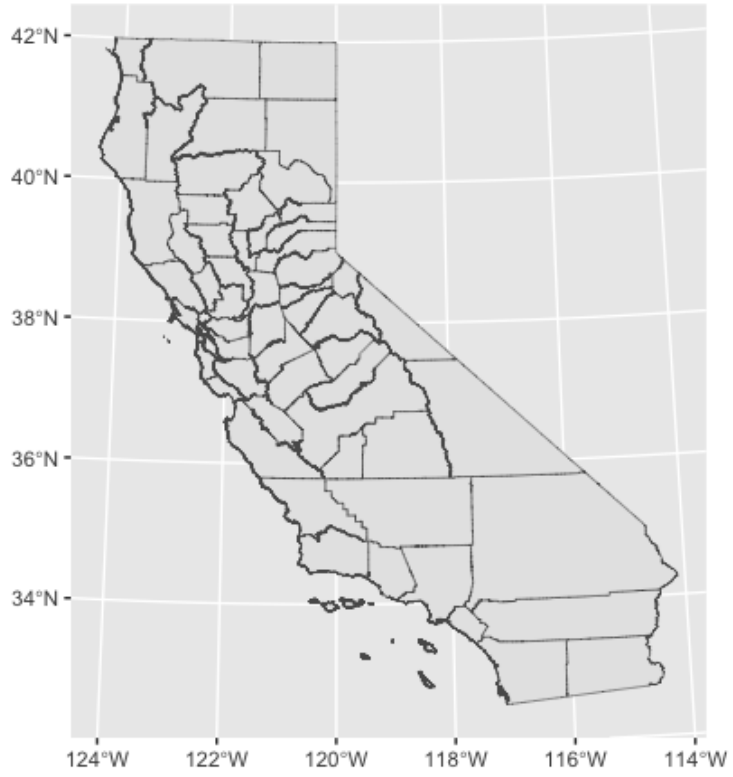
Class Exercise

- create a data frame of the Los Angeles school district in California
- create a data frame of CSU campuses in Los Angeles county
- create a data frame of Los Angeles County
- Plot all three on a map:
  - Los Angeles county
  - LAUSD (purple)
  - CSU's (green)

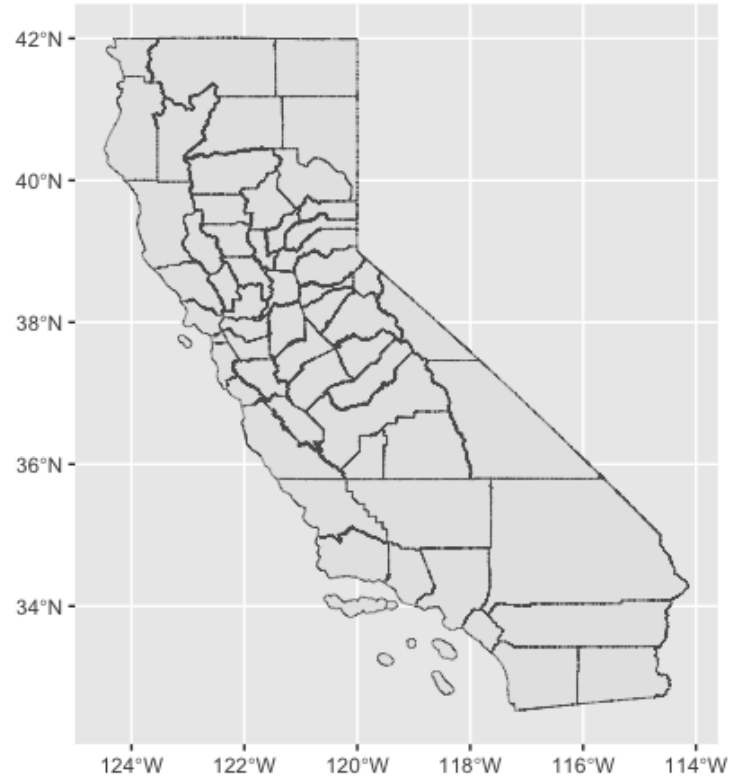
https://pollev.com/vsovero

Spot the Differences

```
ggplot(data=CA_Counties) +
    geom_sf()
```

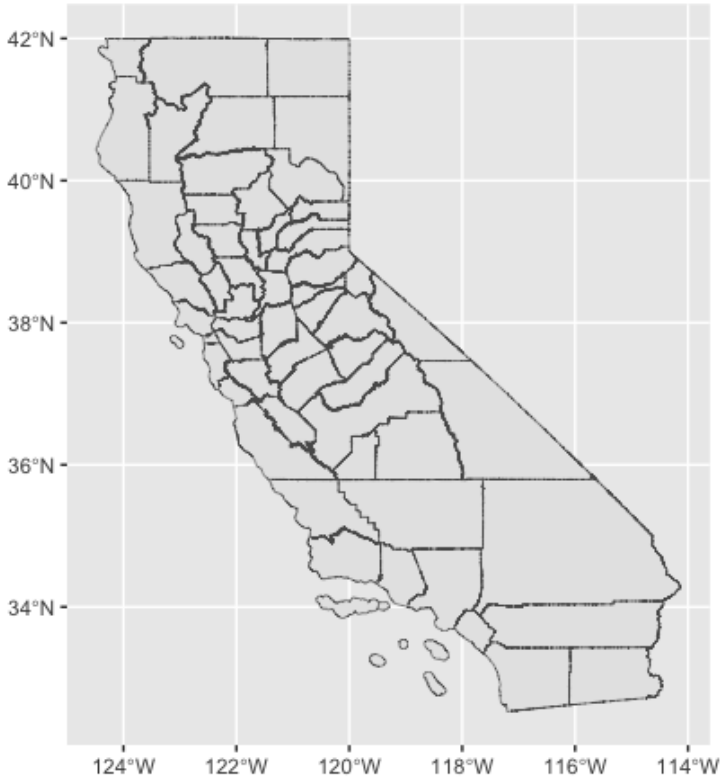



```
ggplot(data=CA_Counties_Tigris) +
    geom_sf()
```



Coordinate Reference System

- The maps look different because the geometries use different **coordinate reference systems**.



Coordinate Reference Systems (CRS)

- A CRS describes how the coordinates in a geospatial dataset relate to locations on the surface of the earth.
- A geographic CRS uses a 3D model of the shape of the earth
- A projected CRS consists of:
  - a geographic CRS
  - a specific **map projection** used to transform geographic coordinates from a curved to a flat surface

Common Projections

- **Web Mercator (3857)**
  - Preserves direction/angle/shape but distorts area and distance.
- **U.S. National Atlas (Albers) Equal Area (2163)**
  - Preserves area but distorts direction/angle/shape and distance.
- **California (Albers) Equal Area (3310)**
  - Equal Area projection optimized to California
  - Preserves area but distorts direction/angle/shape and distance.

Checking the CRS

- We can use st_crs() to check the CRS
- CA_Counties uses a projected CRS:
  - geographic CRS: NAD83
  - projection: California Albers
- CA_Counties_Tigris uses a geographic CRS: NAD83

```
> st_crs(CA_Counties)
Coordinate Reference System:
  User input: NAD83 / California Albers
  wkt:
PROJCRS["NAD83 / California Albers",
    BASEGEOGCRS["NAD83",
        DATUM["North American Datum 1983",
            ELLIPSOID["GRS 1980",6378137,298.257222101,
                LENGTHUNIT["metre",1]]],
        PRIMEM["Greenwich",0,
            ANGLEUNIT["degree",0.0174532925199433]],
        ID["EPSG",4269]],
    CONVERSION["California Albers",
        METHOD["Albers Equal Area",
```

```
> st_crs(CA_Counties_Tigris)
Coordinate Reference System:
  User input: NAD83
  wkt:
GEOGCRS["NAD83",
    DATUM["North American Datum 1983",
        ELLIPSOID["GRS 1980",6378137,298.257222101,
            LENGTHUNIT["metre",1]]],
    PRIMEM["Greenwich",0,
        ANGLEUNIT["degree",0.0174532925199433]],
    CS[ellipsoidal,2],
        AXIS["latitude",north,
            ORDER[1],
            ANGLEUNIT["degree",0.0174532925199433]],
```

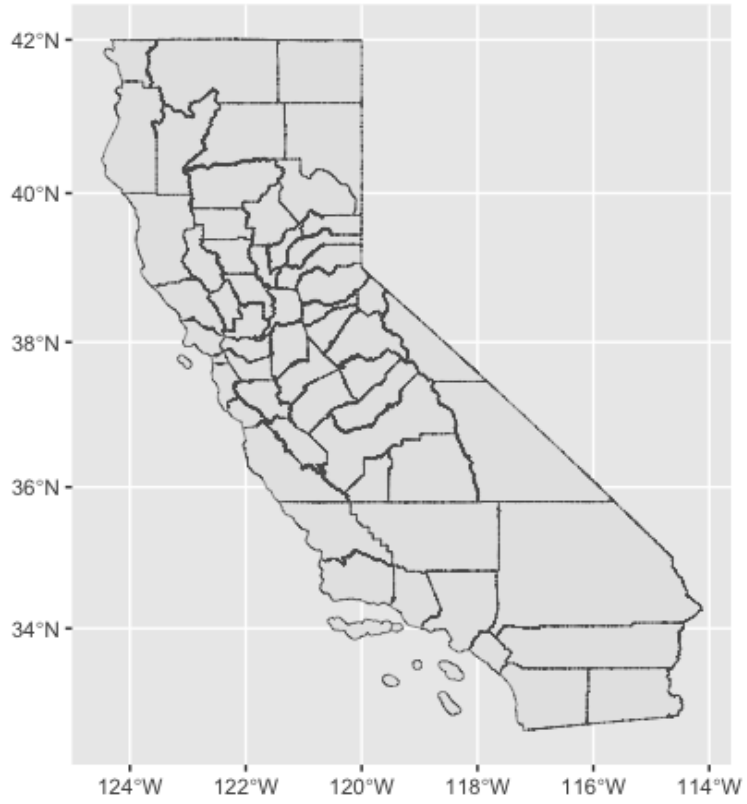
Transforming the CRS

- To change a CRS, we need to **project** (or re-project) our data using st_transform()
- Arguments:
  - The sf object
  - The CRS (four digit code)

```
CA_Counties_Tigris_Mercator<-
st_transform(CA_Counties_Tigris, crs=3857)
```

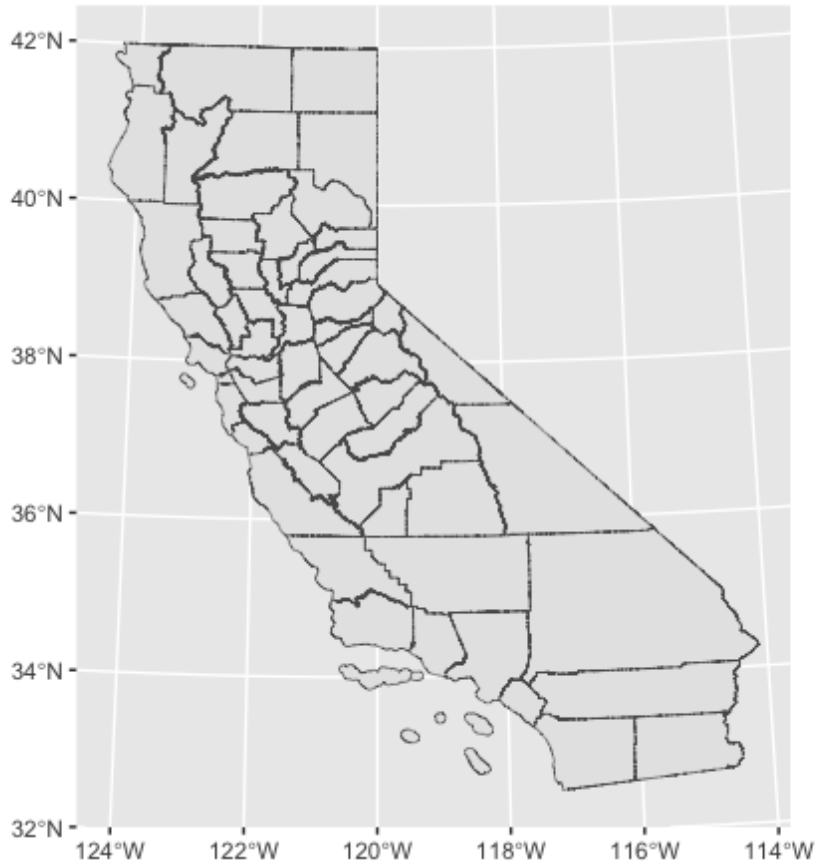
Mercator Projection

```
CA_Counties_Tigris_Mercator<-
st_transform(CA_Counties_Tigris, crs=3857)

ggplot(data=CA_Counties_Tigris_Mercator) +
        geom_sf()
```



CA Albers Projection

```
CA_Counties_Tigris_Albers_CA<-
st_transform(CA_Counties_Tigris, crs=3310)

ggplot(data=CA_Counties_Tigris_Albers_CA) +
        geom_sf()
```



Class Exercise

- Transform LA county to the Mercator Projection
- Transform LA county to the Albers Projection

Beyond Mapping: Spatial Queries

- Queries are software operations that allow us to ask questions of our data and which return data metrics, subsets or new data objects.
- The basic types of **spatial queries** are:
  - Spatial Measurement Queries
  - Spatial Relationship Queries

Beyond Mapping: Spatial Queries

Spatial Measurement Queries

- What is feature A's **area**?
  - *What is the area of Alameda County?*
- What is feature A's **length**?
  - *What is the length of the BART train line between Walnut Creek and Rockridge?*
- What is feature A's **distance** from feature B?
  - *What is the distance between Berkeley High School and Berkeley BART Station?*

Spatial Relationship Queries

- Is feature A **within** feature B?
  - *What schools are in Berkeley?*
- Does feature A **intersect** with feature B?
  - *What in what cities is Tilden Regional Park located?
- Does feature A **cross** feature B?
  - *Does the BART line cross into Albany?*

Measuring Area

- We will use st_area() to calculate the area of Riverside County
- Arguments:
  - the sf object
- Output: area printed to the console

```
st_area(Riverside_County_Albers)
```

```
18921433643 [m^2]
```

Area Measurements under different Projections

- The calculated area of Riverside county is very different depending on the CRS
- Albers: distorts shape, preserves area and distance
- Mercator: preserves shape, distorts area and distance

```
st_area(Riverside_County_Albers)
```

```
18921433643 [m^2]
```

```
st_area(Riverside_County_Mercator)
```

```
27426748703 [m^2]
```

Changing the units

- The default unit of measurement is square meters
- we can use set_units() from the units package to change the unit of measurement to square miles
- Arguments:
  - the measure you want converted
  - the unit of measurement
- Output: Area of Riverside County in square miles

```
set_units(st_area(Riverside_County_Albers), mi^2)
```

```
7303.084 [mi^2]
```

Class Exercise

- Calculate the area of LA county in the Mercator Projection (sq miles)
- Calculate the area of LA county to the Albers Projection (sq miles)

Measuring Distance

- What is the distance between UCR and CSU San Bernadino?
- First, make sure that your simple features (UCR and CSUSB are using the same CRS (CA Albers)

```
CSUSB<-CSU_sf%>%
  filter(IALIAS=="Cal State San Bernardino")

CSUSB_Albers<-st_transform(CSUSB, crs=3310)
```

Measure Distance

- we will use st_distance() to measure distance between two sf objects

```
st_distance(UCR_Albers, CSUSB_Albers)

Units: [m]
         [,1]
[1,] 23316.02

set_units(st_distance(UCR_Albers, CSUSB_Albers),mi)

Units: [mi]
         [,1]
[1,] 14.48791
```

Measuring Distance between all the UC's and all the CSU's

- Can we do it? Sure.
- Output will be a matrix of all the pairwise combinations of UC's and CSU's
- Hard to read the output that's printed to the console

```
CSU_Albers<-st_transform(CSU_sf, crs=3310)
UC_Albers<-st_transform(UC_sf, crs=3310)
```

```
set_units( st_distance(UC_Albers, CSU_Albers),mi)
```

```
Units: [mi]
          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]     [,9]     [,10]     [,11]
 [1,] 198.68028 246.97558  80.65708 363.52333 375.53819 361.93581 130.24696 356.12420 156.50786 368.66805  18.60597
 [2,] 231.75624 264.33181  85.39446 384.53130 388.78777 378.94535  82.34036 376.96555 162.15410 387.07589  63.13645
 [3,] 196.91959 137.69986 316.64743  22.22725  47.32378  28.24103 474.78004  28.14841 243.27143  16.26030 364.09993
 [4,] 151.85055  95.88027 274.05436  25.31031  64.55075  35.61511 433.50435  17.83219 202.78535  34.55405 319.71528
 [5,] 210.53132 138.78428 315.07573  52.25354  14.48791  28.96170 469.41170  53.86962 238.63033  32.67075 367.31109
 [6,] 257.56390 201.27443 380.22267  82.77708  90.02923  88.05261 537.97892  90.02441 306.39753  78.68583 427.45246
 [7,] 197.40039 249.93902  89.45233 364.99942 378.98515 364.47434 139.75519 357.66827 163.14657 370.82225  23.30097
 [8,]  76.36671  76.94940 221.90838 104.48921 145.15507 118.49566 383.05174  98.75151 165.38839 118.24912 255.54610
 [9,] 140.84311 200.24578  75.81441 311.48520 329.52519 313.00268 189.19287 304.29510 128.58761 318.54002  45.73422
[10,] 143.28797 157.43604  25.92998 278.26472 280.27025 271.11418 180.39916 270.66620  53.64856 279.66315  91.59737
          [,12]     [,13]     [,14]     [,15]     [,16]     [,17]     [,18]     [,19]     [,20]     [,21]     [,22]
 [1,] 365.67316 348.40508 326.09514  65.74962  13.70872 229.1567 457.48879  15.75076  42.43156  39.50378 435.26458
```

Measuring Distance between UC's and CSU's

- First, save our output to an object (distance_matrix)
- Next, assign the names of the schools to the rows and columns
- Finally, save it as a data frame

```
distance_matrix<-set_units( st_distance(UC_Albers, CSU_Albers) , mi)

rownames(distance_matrix)=c(UC_Albers$INSTNM)
colnames(distance_matrix)=c(CSU_Albers$INSTNM)


distance_data<-data.frame(distance_matrix)
```

Measuring Distance between UC's and CSU's

```
distance_matrix<-set_units( st_distance(UC_Albers, CSU_Albers) , mi)

rownames(distance_matrix)=c(UC_Albers$INSTNM)
colnames(distance_matrix)=c(CSU_Albers$INSTNM)

distance_data<-data.frame(distance_matrix)
```

	California.Polytechnic.State.University.San.Luis.Obispo	California.State.University.Bakersfield	California.State.University.Stanislaus	C
University of California-Berkeley	198.68028 [mi]	246.97558 [mi]	80.65708 [mi]	3
University of California-Davis	231.75624 [mi]	264.33181 [mi]	85.39446 [mi]	3
University of California-Irvine	196.91959 [mi]	137.69986 [mi]	316.64743 [mi]	2
University of California-Los Angeles	151.85055 [mi]	95.88027 [mi]	274.05436 [mi]	2
University of California-Riverside	210.53132 [mi]	138.78428 [mi]	315.07573 [mi]	5
University of California-San Diego	257.56390 [mi]	201.27443 [mi]	380.22267 [mi]	8
University of California-San Francisco	197.40039 [mi]	249.93902 [mi]	89.45233 [mi]	3
University of California-Santa Barbara	76.36671 [mi]	76.94940 [mi]	221.90838 [mi]	1
University of California-Santa Cruz	140.84311 [mi]	200.24578 [mi]	75.81441 [mi]	3
University of California-Merced	143.28797 [mi]	157.43604 [mi]	25.92998 [mi]	2

Make it Tidy

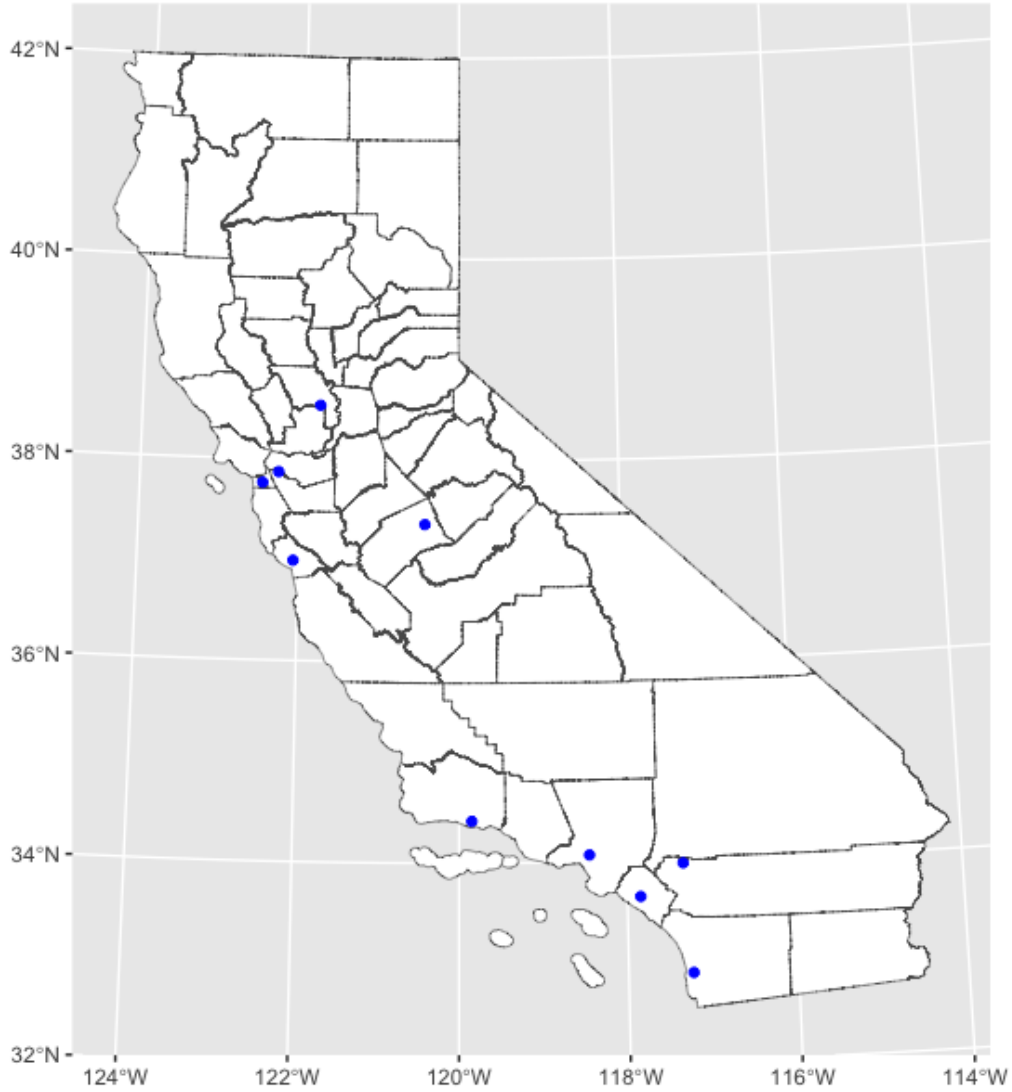
- Things you will need to use:
  - row_names_to_columns()
  - pivot_longer()

Filter

	UC campus	CSU campus	Distance
1	University of California–Berkeley	California.Polytechnic.State.University.San.Luis.Obispo	198.68028 [mi]
2	University of California–Berkeley	California.State.University.Bakersfield	246.97558 [mi]
3	University of California–Berkeley	California.State.University.Stanislaus	80.65708 [mi]
4	University of California–Berkeley	California.State.University.Chancellors.Office	363.52333 [mi]
5	University of California–Berkeley	California.State.University.San.Bernardino	375.53819 [mi]
6	University of California–Berkeley	California.State.Polytechnic.University.Pomona	361.93581 [mi]
7	University of California–Berkeley	California.State.University.Chico	130.24696 [mi]
8	University of California–Berkeley	California.State.University.Dominguez.Hills	356.12420 [mi]
9	University of California–Berkeley	California.State.University.Fresno	156.50786 [mi]
10	University of California–Berkeley	California.State.University.Fullerton	368.66805 [mi]
11	University of California–Berkeley	California.State.University.East.Bay	18.60597 [mi]
12	University of California–Berkeley	California.State.University.Long.Beach	365.67316 [mi]
13	University of California–Berkeley	California.State.University.Los.Angeles	348.40508 [mi]
14	University of California–Berkeley	California.State.University.Northridge	326.09514 [mi]
15	University of California–Berkeley	California.State.University.Sacramento	65.74962 [mi]

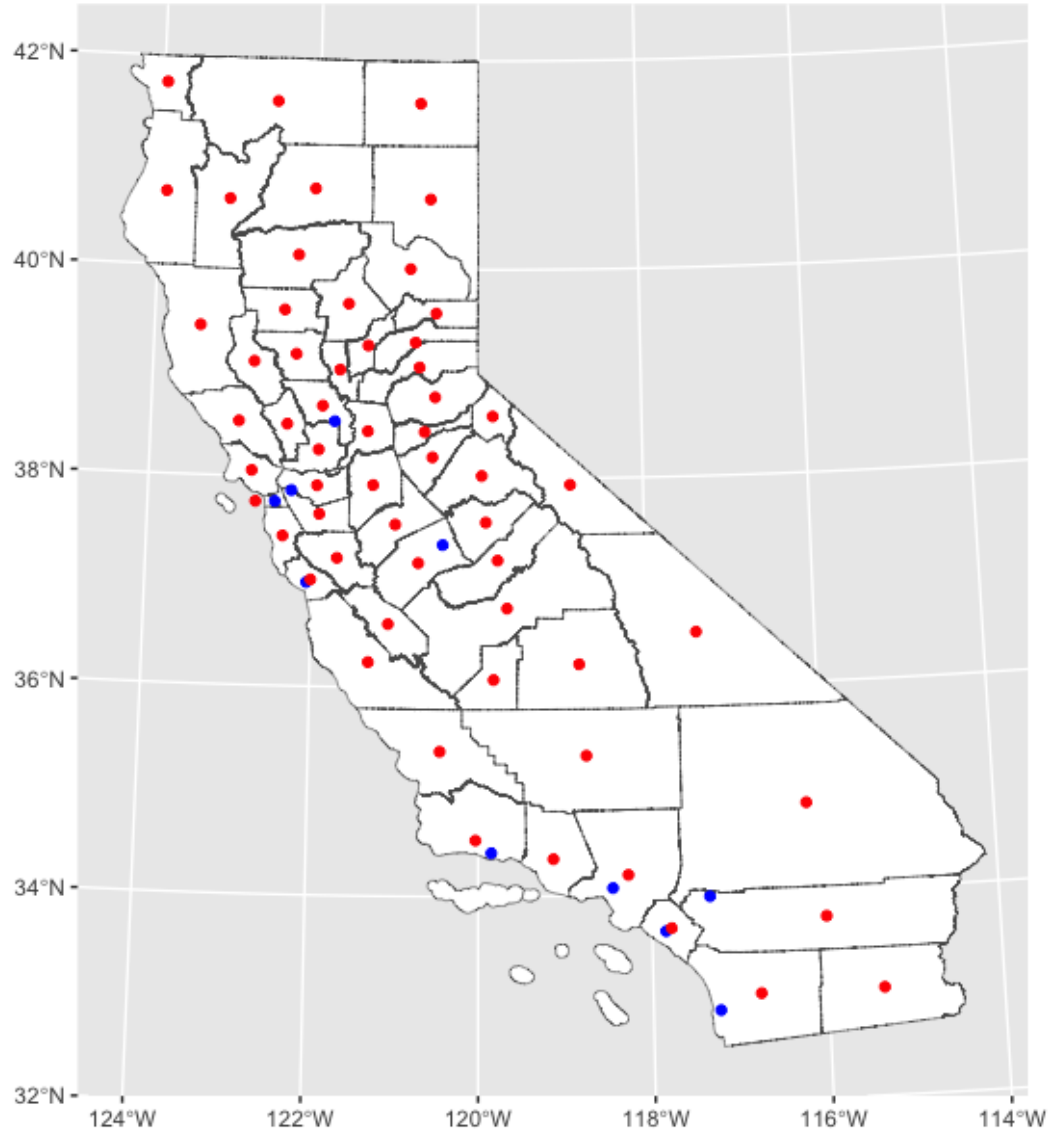
Measuring Distance between Counties and UC Campuses

- Let say we wanted to measure the distance between every county in California and every UC campus
- **Problem**: Distance needs to be measured between two points, counties are polygons
- What do we do?



Measuring Distance between Counties and UC Campuses

- **Problem**: Distance needs to be measured between two points, counties are polygons
- **Solution**: find the centroid (center) of each county
- Blue: UC Campuses
- Red: County Centroids



County Centroids

```
CA_Counties_Centroids<-st_centroid(CA_Counties_Tigris_Albers_CA)
```

- We will use st_centroid() to find the centroid of each county
- Arguments: a simple features object (polygons)
- Output: a simple features object (points)

Class Exercise

- Create a data frame that measures the distance between every county centroid and every UC campus
- What is the distance of LA county to all of the UC campuses?
- How many campuses are within 100 miles of the centroid of LA County?