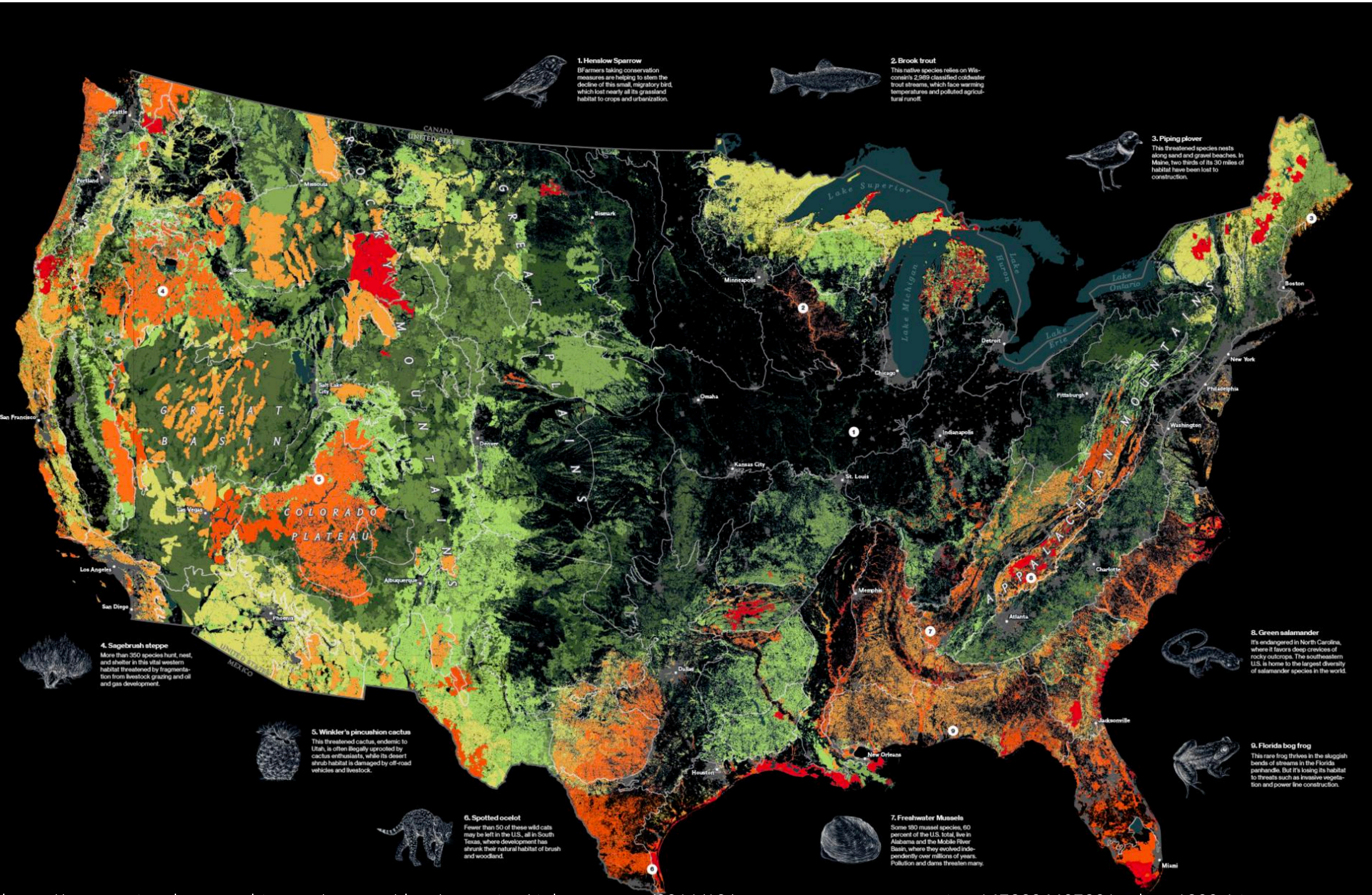
Spatial Data in R

E. Anne Chambers



1. Henslow Sparrow
BFarmers taking conservation measures are helping to stem the decline of this small, migratory bird, which lost nearly all its grassland habitat to crops and urbanization.

2. Brook trout
This native species relies on Wisconsin's 2,989 classified coldwater trout streams, which face warming temperatures and polluted agricultural runoff.

3. Piping plover
This threatened species nests along sand and gravel beaches. In Maine, two thirds of its 30 miles of habitat have been lost to construction.

4. Sagebrush steppe
More than 350 species hunt, nest, and shelter in this vital western habitat threatened by fragmentation from livestock grazing and oil and gas development.

5. Winkler's pincushion cactus
This threatened cactus, endemic to Utah, is often illegally uprooted by cactus enthusiasts, while its desert shrub habitat is damaged by off-road vehicles and livestock.

6. Spotted ocelot
Fewer than 50 of these wild cats may be left in the U.S., all in South Texas, where development has shrunk their natural habitat of brush and woodland.

7. Freshwater Mussels
Some 180 mussel species, 60 percent of the U.S. total, live in Alabama and the Mobile River Basin, where they evolved independently over millions of years. Pollution and dams threaten many.

8. Green salamander
It's endangered in North Carolina, where it favors deep crevices of rocky outcrops. The southeastern U.S. is home to the largest diversity of salamander species in the world.

9. Florida bog frog
This rare frog thrives in the sluggish bends of streams in the Florida panhandle. But it's losing its habitat to threats such as invasive vegetation and power line construction.

https://www.nationalgeographic.com/content/dam/magazine/rights-exempt/2016/12/green-spaces-og.ngsversion.1479234407991.adapt.1900.1.jpg

At the most basic level…

- Vector data and raster data

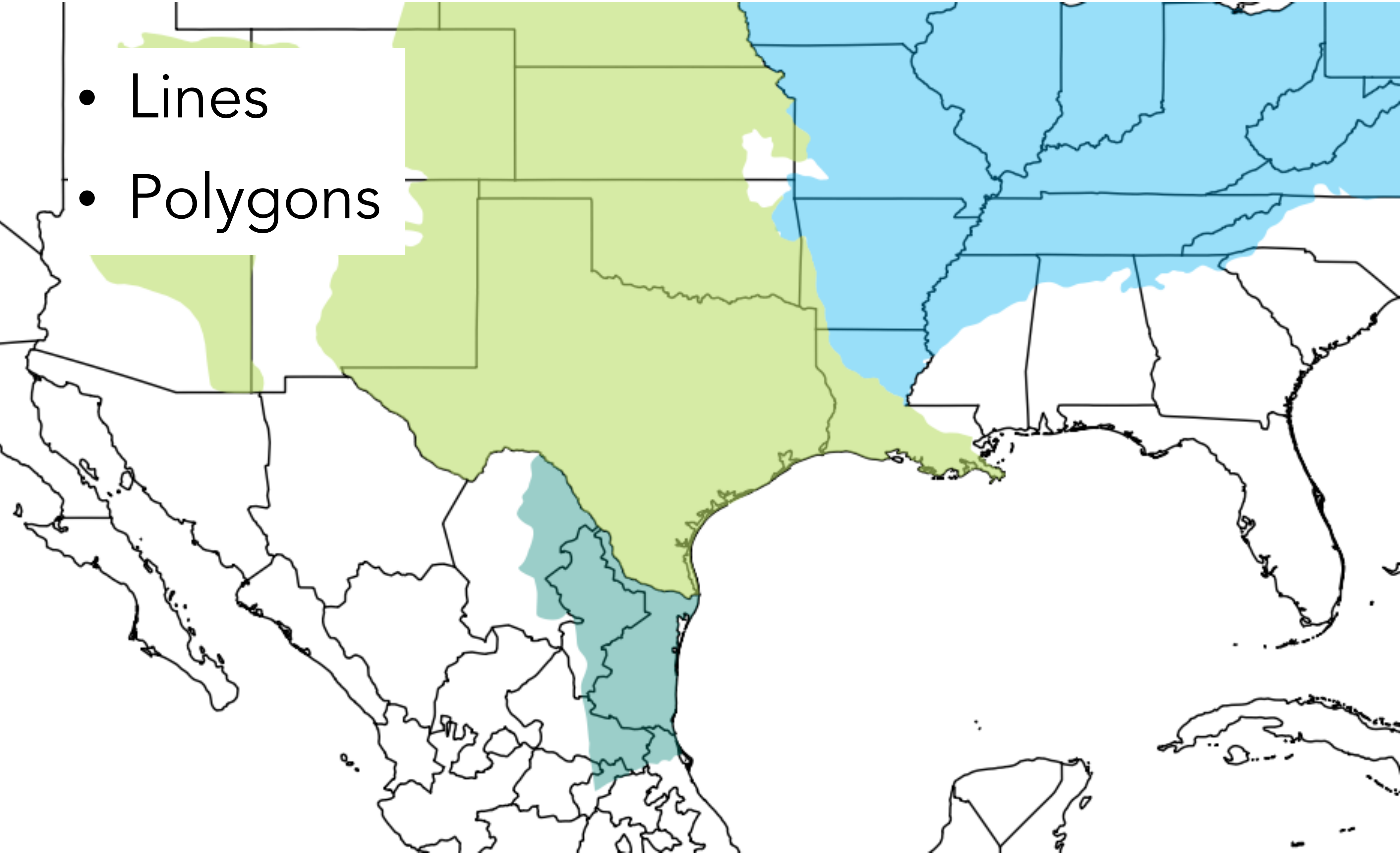
- Vector data:
  - Lines
  - Polygons
  - Points

How does this extend to mapping?

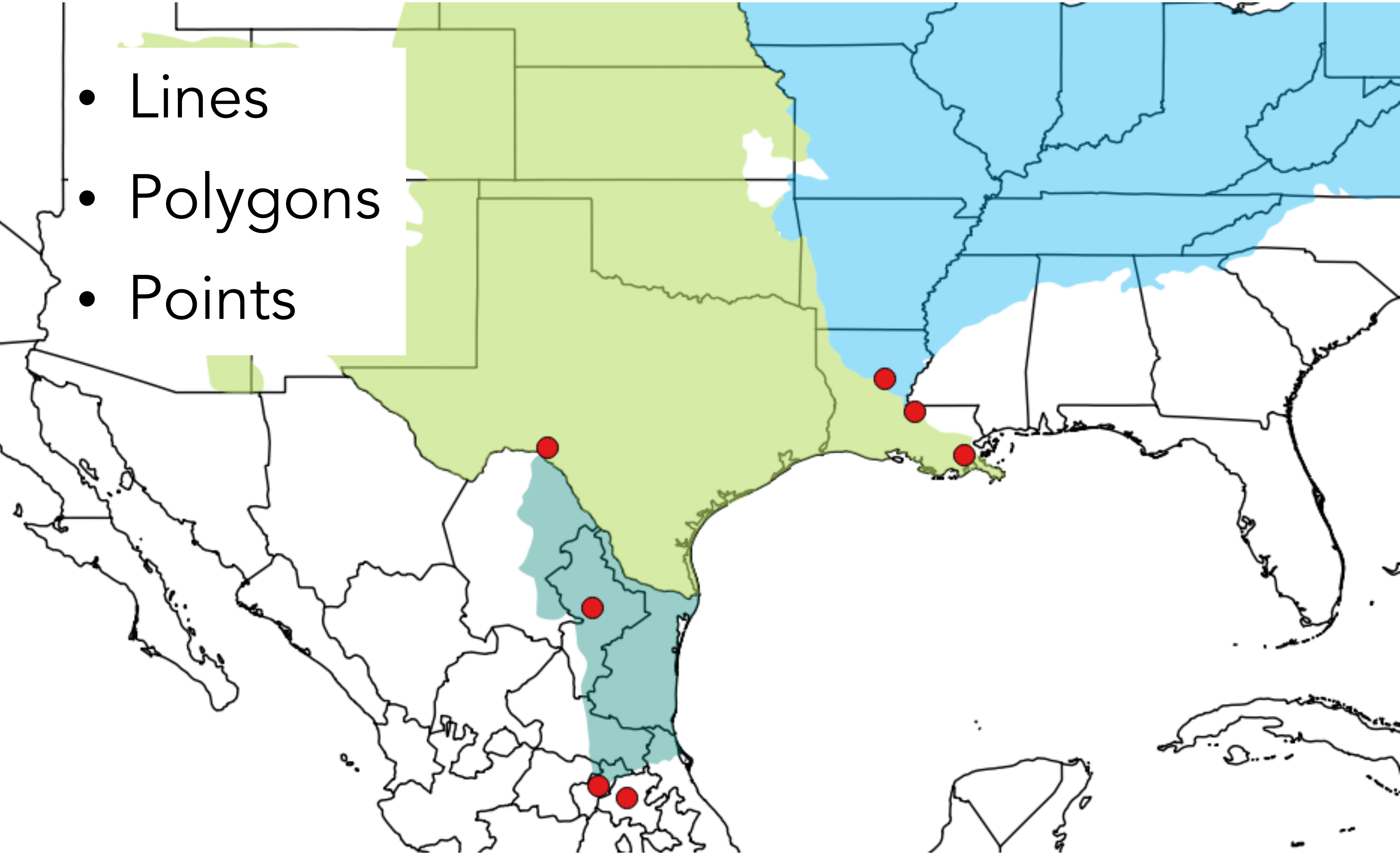
- Lines

How does this extend to mapping?

- Lines
- Polygons



How does this extend to mapping?



- Lines
- Polygons
- Points

Spatial data in R

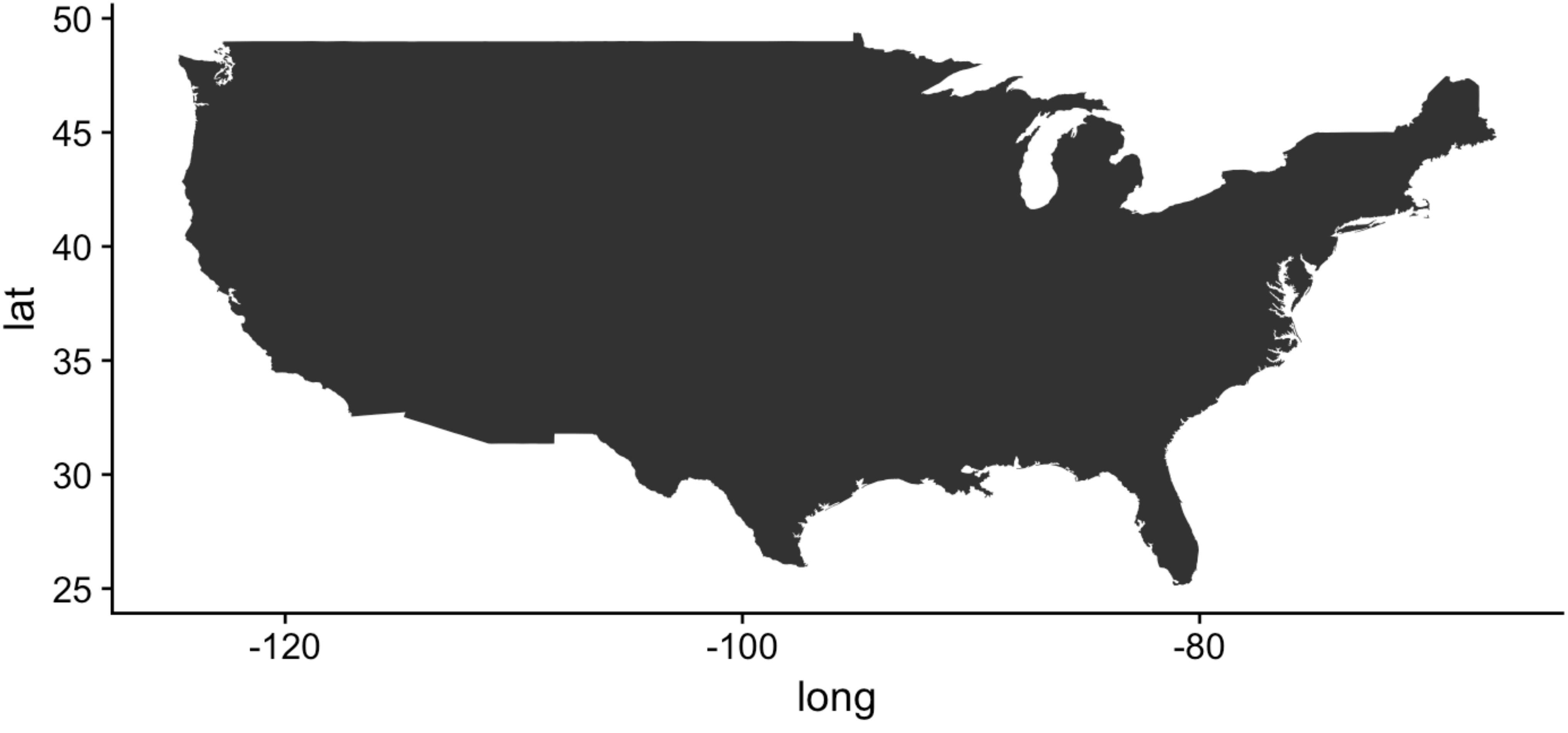
- Many packages

- Information on some packages that handle spatial data:
- https://cran.r-project.org/web/views/Spatial.html

- Useful cheatsheet for key functions:
- http://www.maths.lancs.ac.uk/~rowlings/Teaching/UseR2012/cheatsheet.html

Think of mapping like any other figure

- x and y coordinates are longitude and latitude



Exercise 1:
Built-in spatial mapping with ggplot

- ggplot can create beautiful maps

- Using same syntax as we’ve already learned

Exercise 1

- Let’s go back to the map of the U.S.

50
45
40
35
30
25
lat
-120
-100
-80
long

```
> ggusa <- map_data(“usa")
```

Function is part of ggplot2, creates data frame of map data

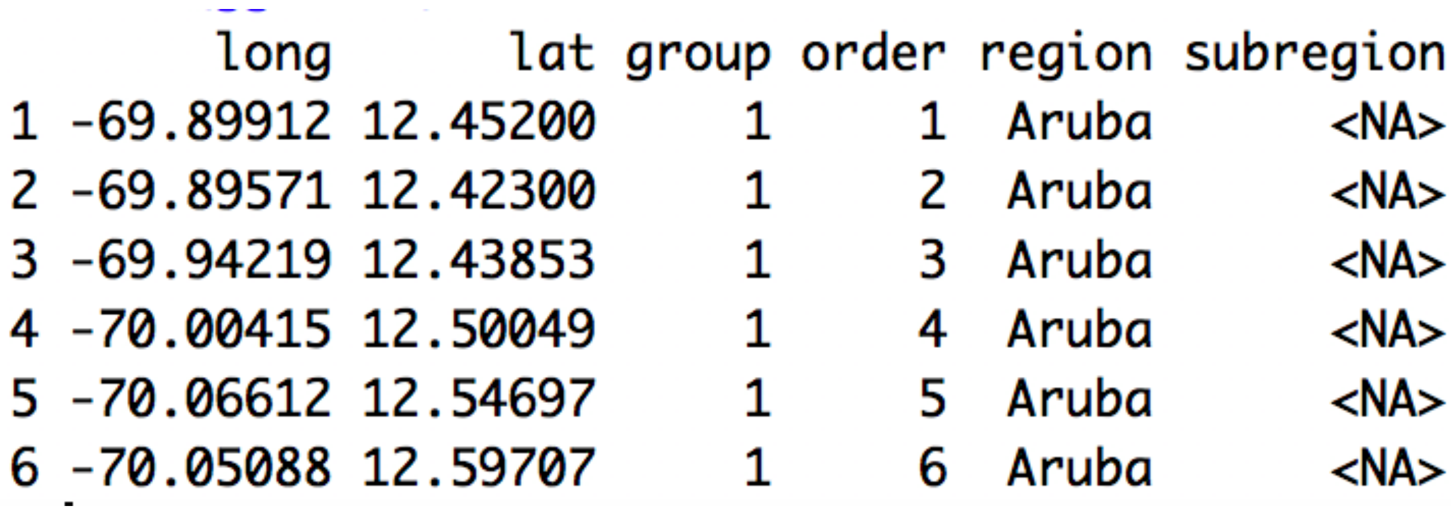
Name of maps provided

Exercise 1

```
> ggusa <- map_data(“usa")
```

Let’s take a look at what this dataframe looks like:

```
> head(ggusa)
```



```
       long      lat group order region subregion
1 -69.89912 12.45200     1     1  Aruba      <NA>
2 -69.89571 12.42300     1     2  Aruba      <NA>
3 -69.94219 12.43853     1     3  Aruba      <NA>
4 -70.00415 12.50049     1     4  Aruba      <NA>
5 -70.06612 12.54697     1     5  Aruba      <NA>
6 -70.05088 12.59707     1     6  Aruba      <NA>
```

Exercise 1

```
> ggusa <- map_data(“usa")
```

```
> head(ggusa)
```

Now, let’s use ggplot to build the map:

```
> ggplot(ggusa,
```

Exercise 1

```
> ggusa <- map_data(“usa")
```

```
> head(ggusa)
```

Now, let’s use ggplot to build the map:

```
> ggplot(ggusa, aes(x=long, y=lat,
group=group))
```

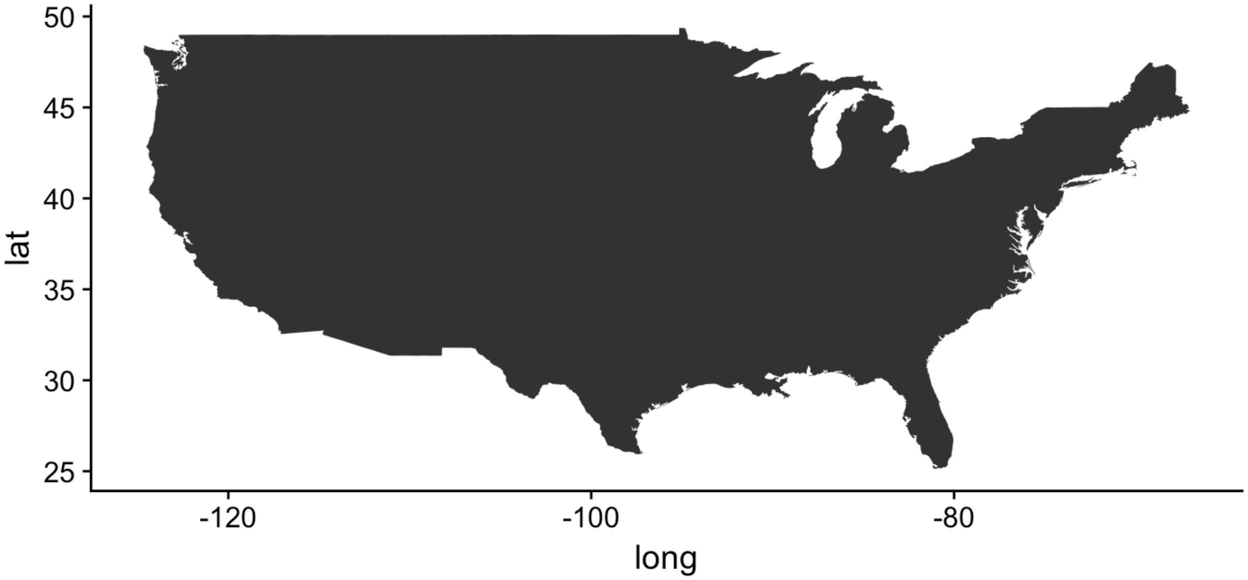
Exercise 1

```
> ggusa <- map_data(“usa")
```

```
> head(ggusa)
```

Now, let’s use ggplot to build the map:

```
> ggplot(ggusa, aes(x=long, y=lat,
  group=group)) +
     geom_polygon()
```



Exercise 1

```
> ggusa <- map_data(“usa")
```

```
> head(ggusa)
```

We can also add on some fun other items to this:

```
> ggplot(ggusa, aes(x=long, y=lat,
  group=group)) +
     geom_polygon() +
     theme_nothing()
```

Exercise 1

```
> ggusa <- map_data(“usa")
```

```
> head(ggusa)
```

We can also add on some fun other items to this:

```
> ggplot(ggusa, aes(x=long, y=lat,
  group=group)) +
      geom_polygon() +
      theme_nothing()
```

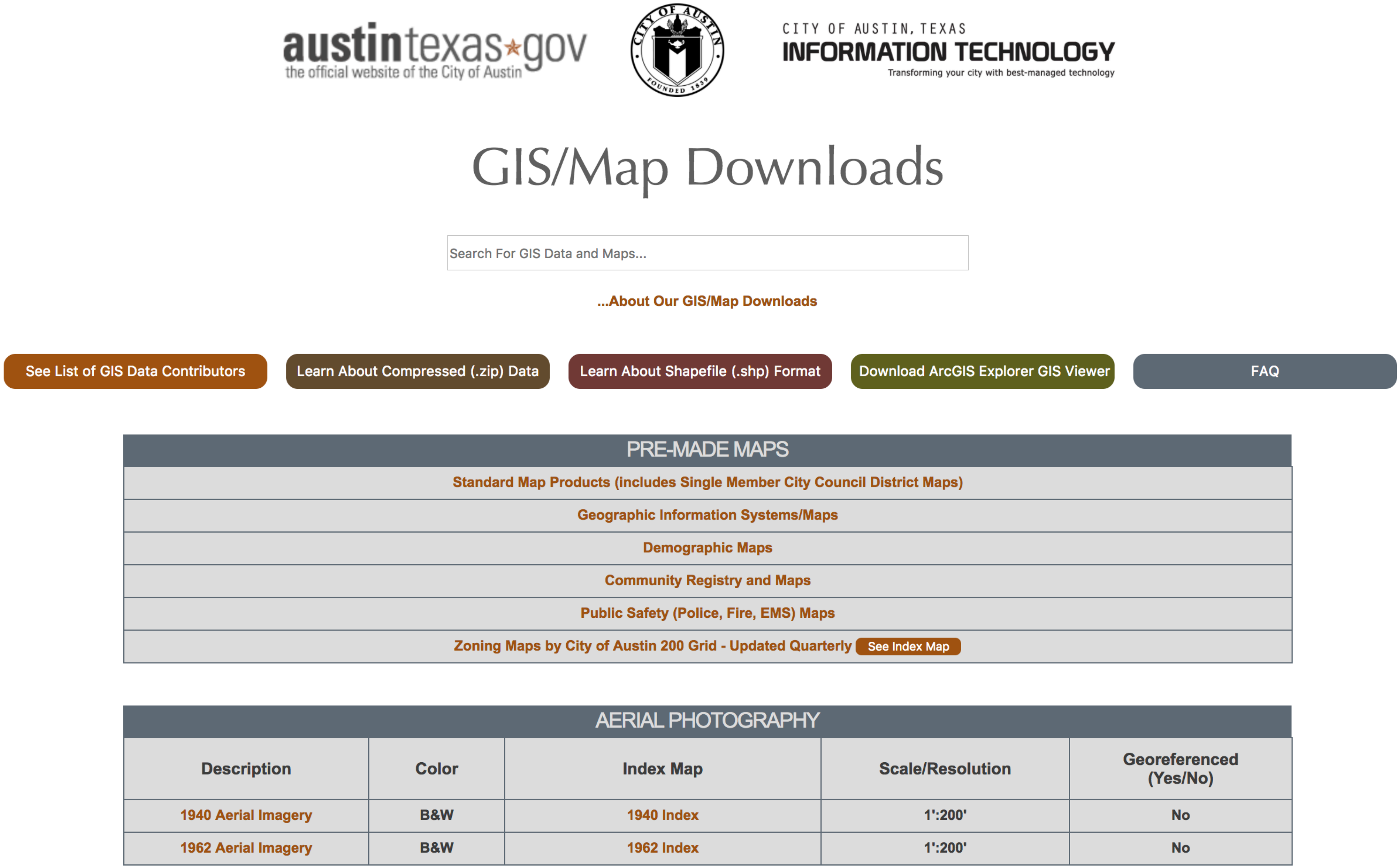
Exercise 1

What happens when our data aren't formatted for entry into ggplot?

Most of the time, spatial data are stored as **shapefiles (.shp).**

Keep all four files (.prj, .dbf, .shp, .shx) together!

Exercise 2: Austin zipcodes + roads


austintexas★gov
the official website of the City of Austin
CITY OF AUSTIN, TEXAS
INFORMATION TECHNOLOGY
Transforming your city with best-managed technology
Search For GIS Data and Maps...
See List of GIS Data Contributors
Learn About Compressed (.zip) Data
Learn About Shapefile (.shp) Format
Download ArcGIS Explorer GIS Viewer
FAQ
See Index Map

GIS/Map Downloads

...About Our GIS/Map Downloads

PRE-MADE MAPS
Standard Map Products (includes Single Member City Council District Maps)
Geographic Information Systems/Maps
Demographic Maps
Community Registry and Maps
Public Safety (Police, Fire, EMS) Maps
Zoning Maps by City of Austin 200 Grid - Updated Quarterly

AERIAL PHOTOGRAPHY

Description	Color	Index Map	Scale/Resolution	Georeferenced (Yes/No)
1940 Aerial Imagery	B&W	1940 Index	1':200'	No
1962 Aerial Imagery	B&W	1962 Index	1':200'	No

Exercise 2

- Two important components to making spatial data readable in ggplot2:

- 1. Reading in shape files in R:

```
> readOGR(dsn = “shapefiles”, layer = “name”)
```

Within the ***rgdal*** package

Data source name; directory that holds shapefile

Name of shapefile (without .shp suffix)

- Two important components to making spatial data readable in ggplot2:

- 1. Reading in shape files in R:

```
> readOGR(dsn = “shapefiles”, layer = “name”)
```

- 2. Using fortify() to make shapefile a dataframe

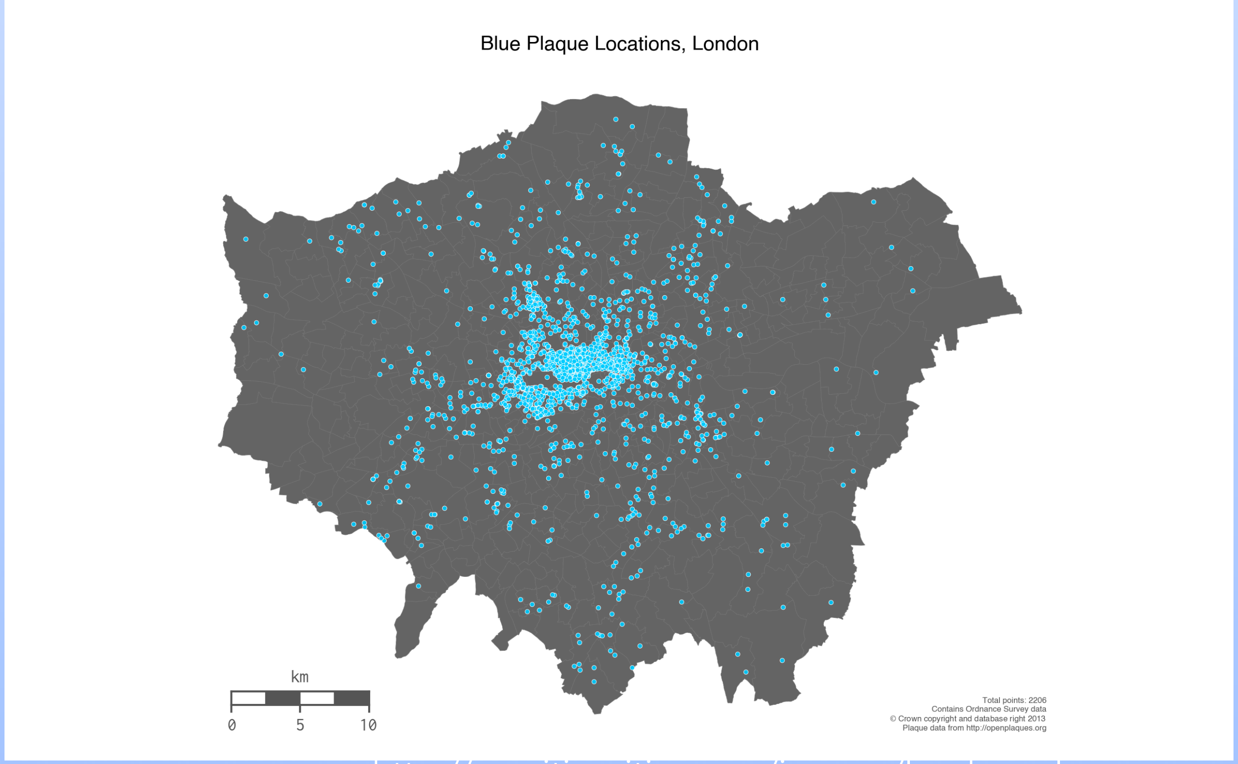
```
> fortify(data, region=“region”)
```

Shapefile

Unique records within shapefile

Exercise 2

How do I import raw point data and plot it onto my spatial data/map?



Blue Plaque Locations, London

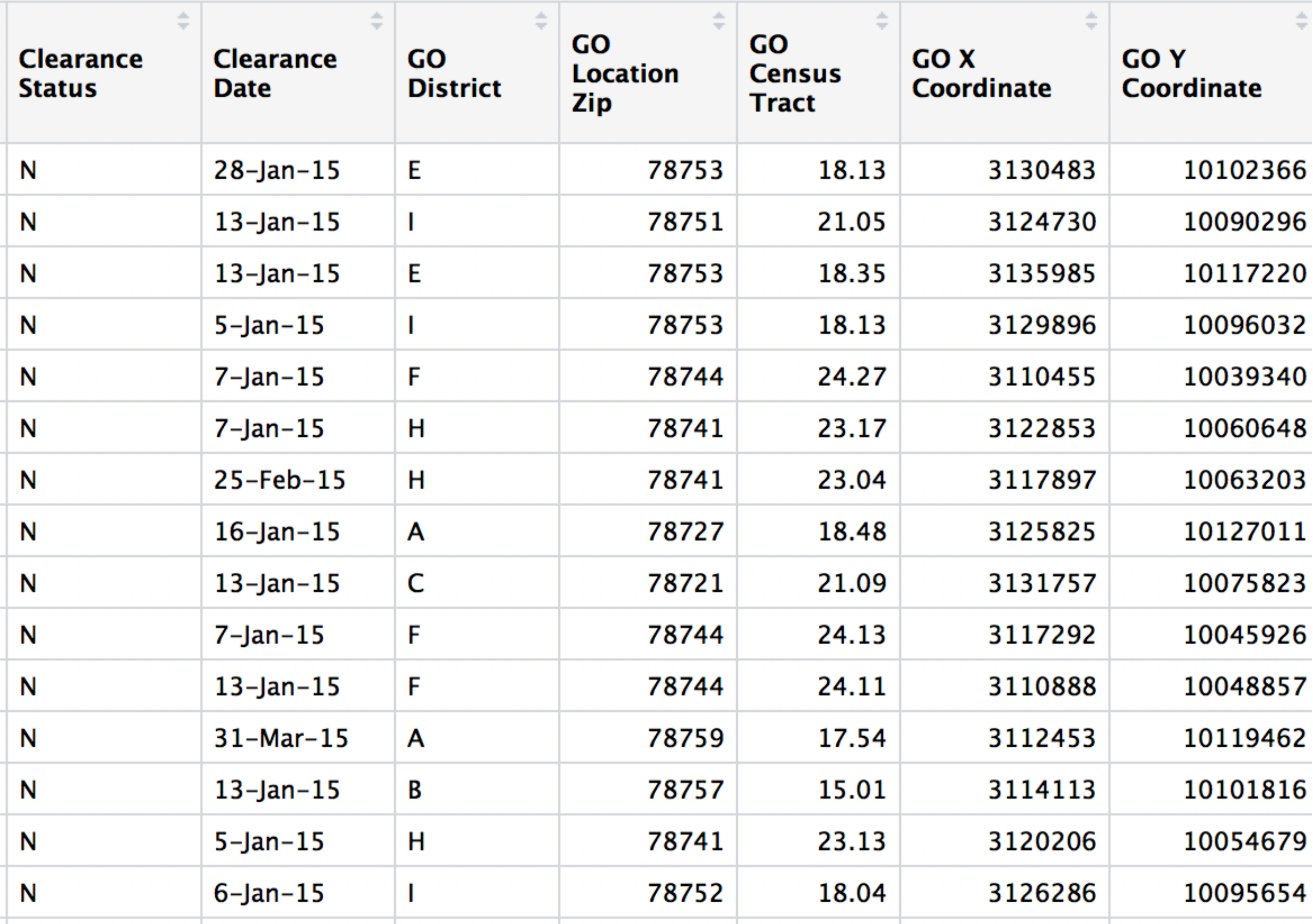
Total points: 2206
Contains Ordnance Survey data
© Crown copyright and database right 2013
Plaque data from http://openplaques.org

http://sensitivecities.com/images/london_plaques.png

Exercise 3: 2015 Austin crime data

```
> data <- read_csv(“data/
Annual_Crime_Dataset_2015.csv”)

> View(data)
```

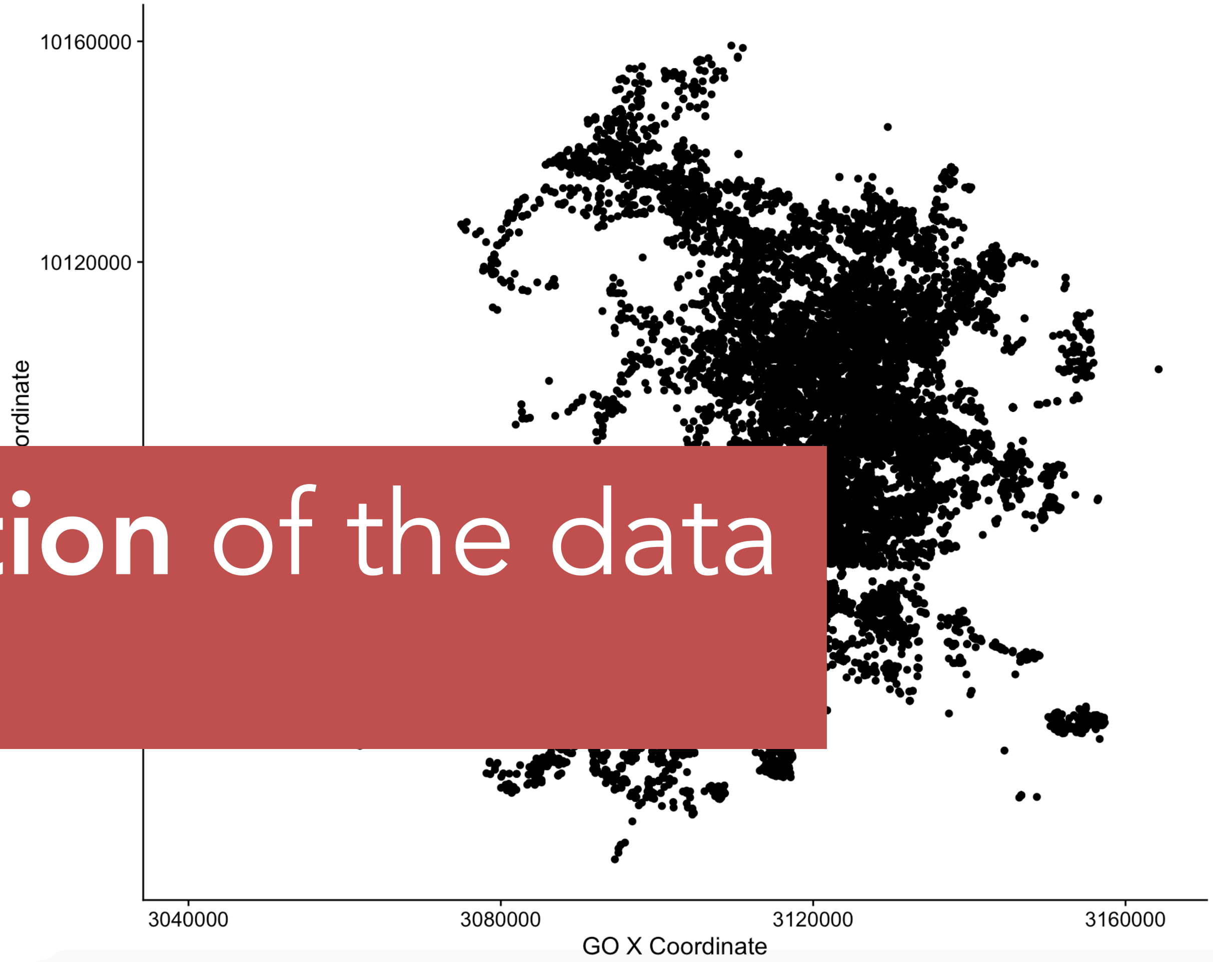


Clearance Status	Clearance Date	GO District	GO Location Zip	GO Census Tract	GO X Coordinate	GO Y Coordinate
N	28-Jan-15	E	78753	18.13	3130483	10102366
N	13-Jan-15	I	78751	21.05	3124730	10090296
N	13-Jan-15	E	78753	18.35	3135985	10117220
N	5-Jan-15	I	78753	18.13	3129896	10096032
N	7-Jan-15	F	78744	24.27	3110455	10039340
N	7-Jan-15	H	78741	23.17	3122853	10060648
N	25-Feb-15	H	78741	23.04	3117897	10063203
N	16-Jan-15	A	78727	18.48	3125825	10127011
N	13-Jan-15	C	78721	21.09	3131757	10075823
N	7-Jan-15	F	78744	24.13	3117292	10045926
N	13-Jan-15	F	78744	24.11	3110888	10048857
N	31-Mar-15	A	78759	17.54	3112453	10119462
N	13-Jan-15	B	78757	15.01	3114113	10101816
N	5-Jan-15	H	78741	23.13	3120206	10054679
N	6-Jan-15	I	78752	18.04	3126286	10095654

Exercise 3

One quick thing we can do is plot the data

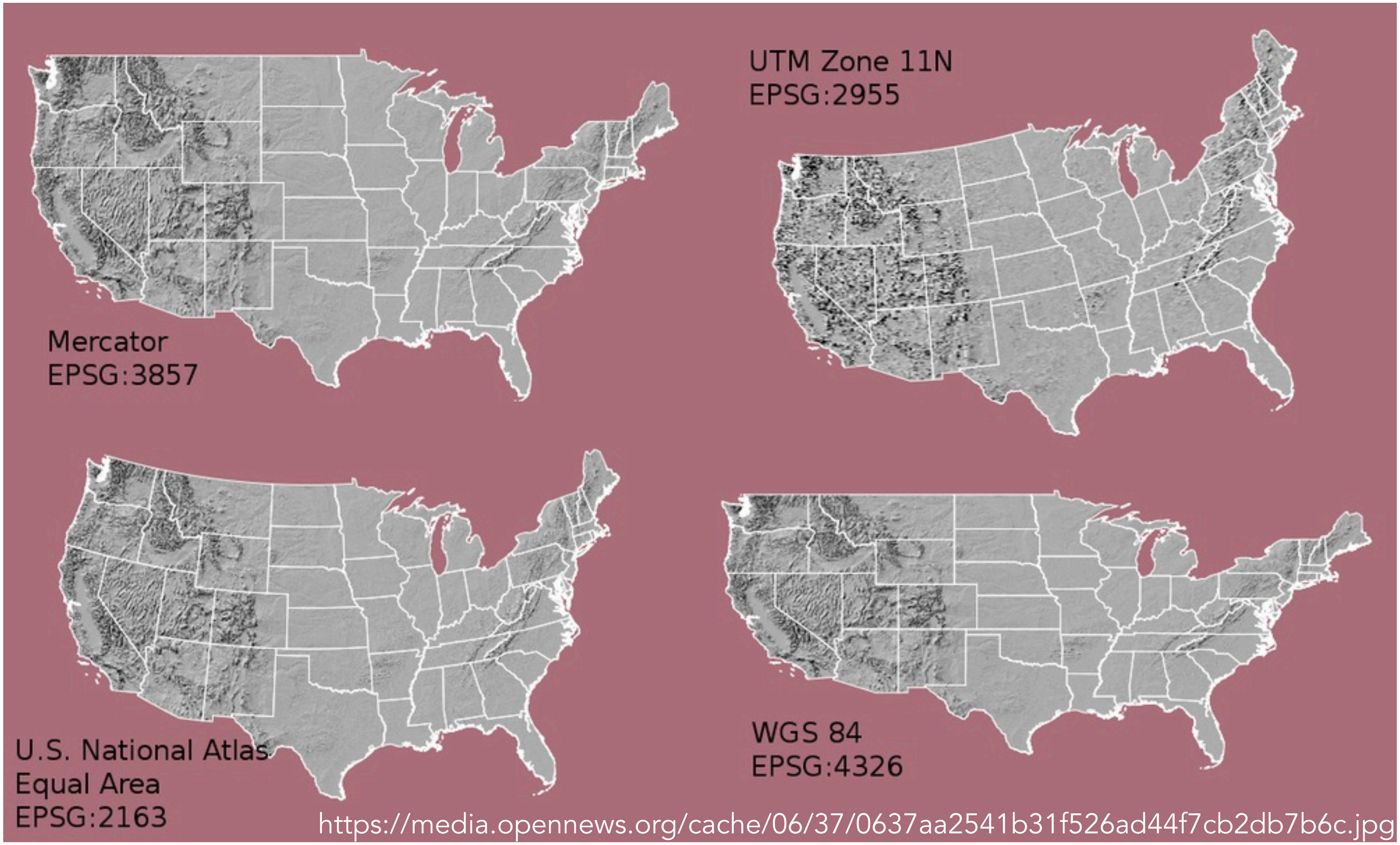
```
> ggplot(data, aes(x=`GO X Coordinate`,
y=`GO Y Coordinate`) +
    geom_point()
```



Projection of the data

Changing projections

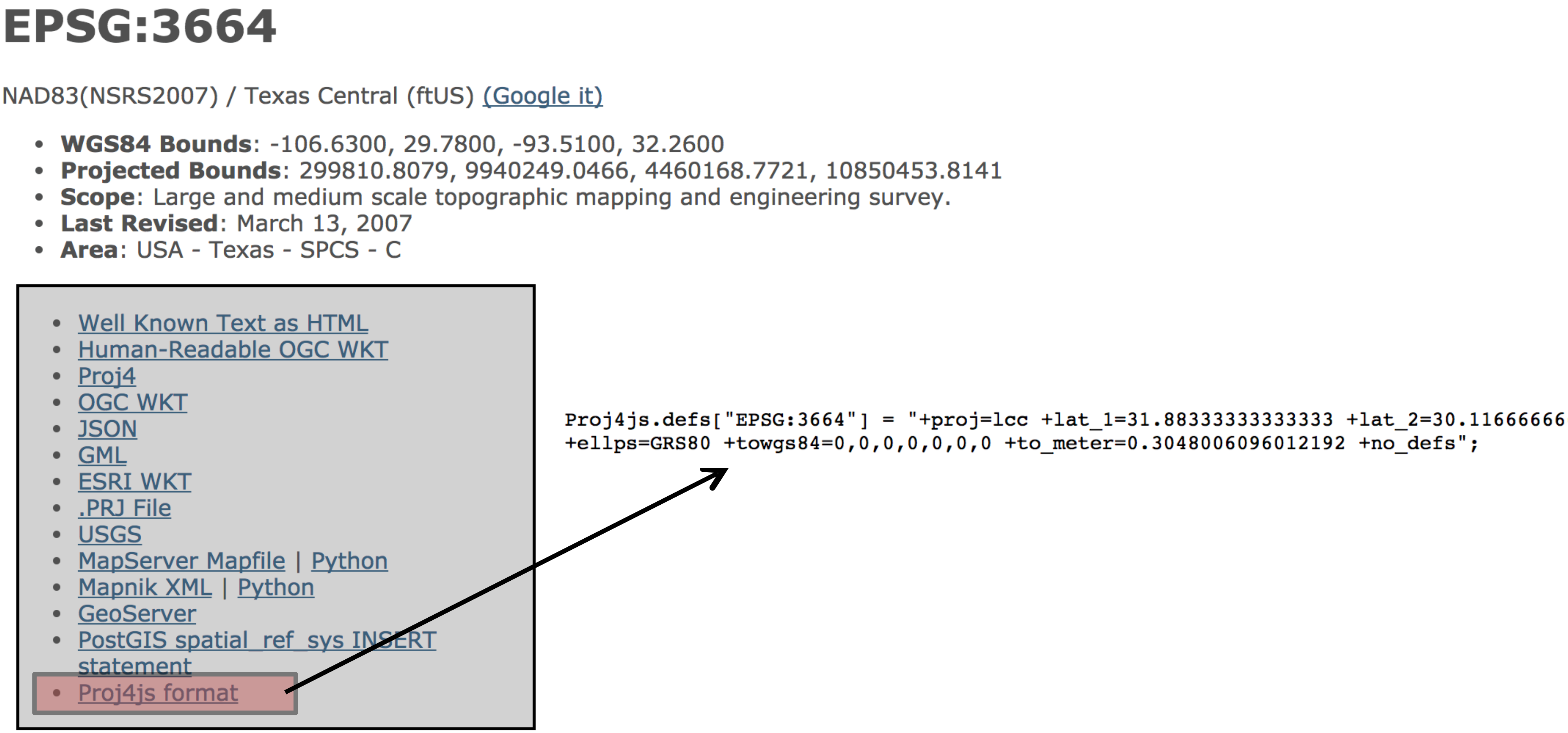
Can't just do a simple transformation of the data



https://media.opennews.org/cache/06/37/0637aa2541b31f526ad44f7cb2db7b6c.jpg

What’s the code for my projection?

spatialreference.org



EPSG:3664

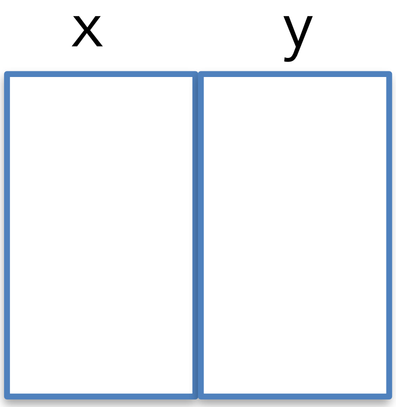
NAD83(NSRS2007) / Texas Central (ftUS) (Google it)

- **WGS84 Bounds**: -106.6300, 29.7800, -93.5100, 32.2600
- **Projected Bounds**: 299810.8079, 9940249.0466, 4460168.7721, 10850453.8141
- **Scope**: Large and medium scale topographic mapping and engineering survey.
- **Last Revised**: March 13, 2007
- **Area**: USA - Texas - SPCS - C

- Well Known Text as HTML
- Human-Readable OGC WKT
- Proj4
- OGC WKT
- JSON
- GML
- ESRI WKT
- .PRJ File
- USGS
- MapServer Mapfile | Python
- Mapnik XML | Python
- GeoServer
- PostGIS spatial_ref_sys INSERT statement
- Proj4js format

```
Proj4js.defs["EPSG:3664"] = "+proj=lcc +lat_1=31.88333333333333 +lat_2=30.1166666
+ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +to_meter=0.3048006096012192 +no_defs";
```

Changing projections



Make data frame containing only coordinates

```
> oldproj <-
data_frame(x=`location x`, y=`location y`)
```

Tell R where to find coordinates

```
> coordinates(oldproj) <- c(‘x’, ‘y’)
```

Specify projection *of existing data*

NAD83 Texas Central (ftUS)

```
> proj4string(oldproj)=CRS(“+init=epsg:3664”)
```

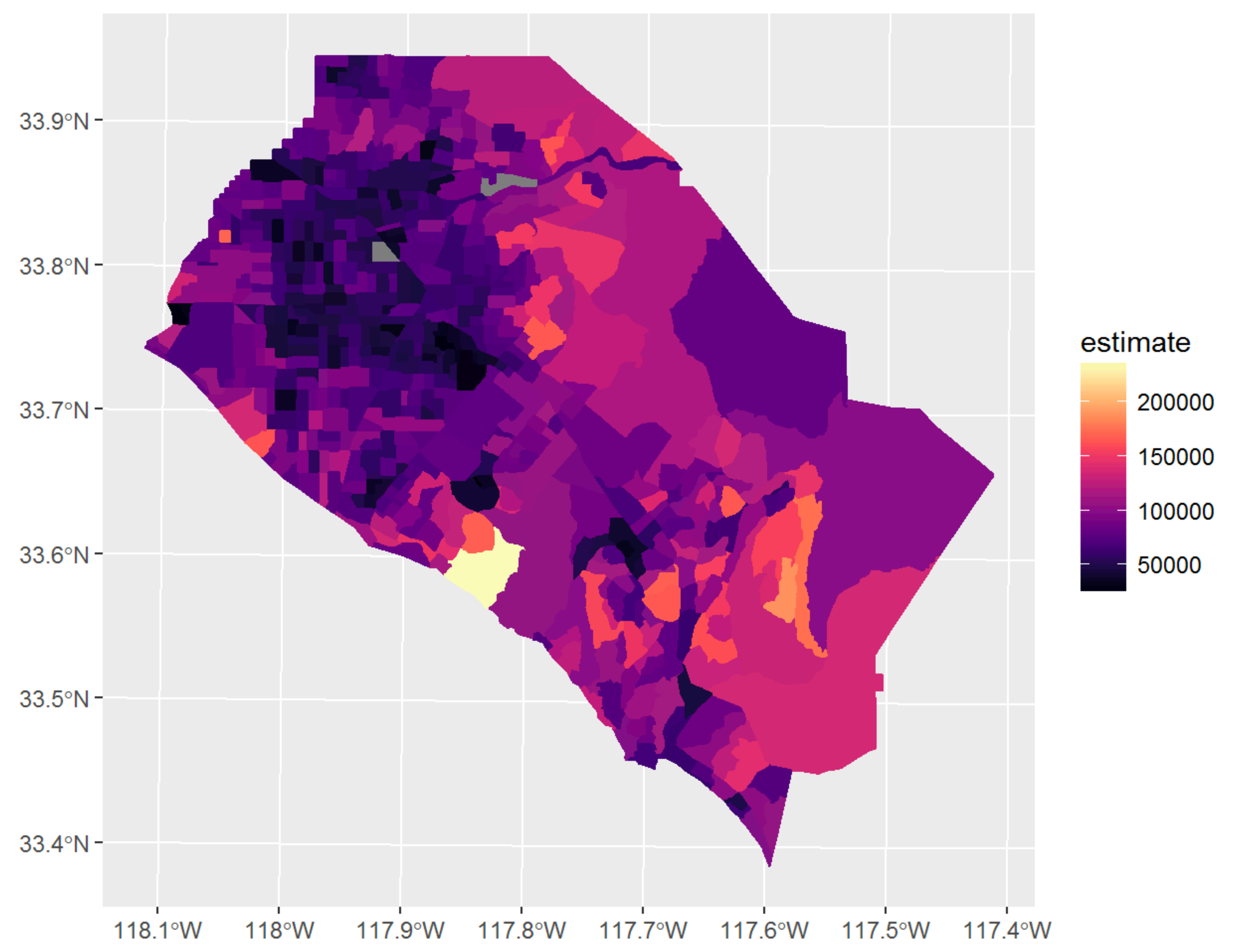
Transform to new projection

WGS84

```
> newproj <- spTransform(oldproj, CRS(“+init=epsg:4326”)
```

Exercise 3

How do I merge my data with polygons on my map?



https://walkerke.github.io/tidycensus/articles/spatial-data_files/figure-html/unnamed-chunk-2-1.png

Exercise 4: Merging data with polygons

Get your data into the right format!

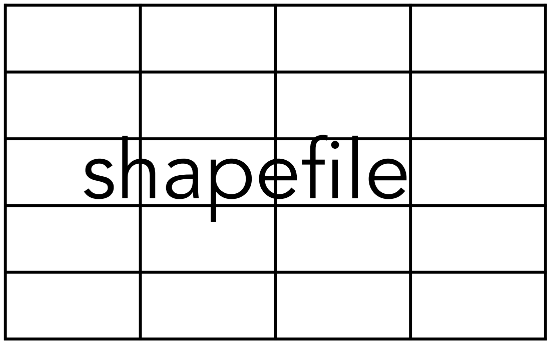
- There needs to be a common column between your fortified ***shapefile*** and your ***data***!
- This common column needs to be the ***same class***

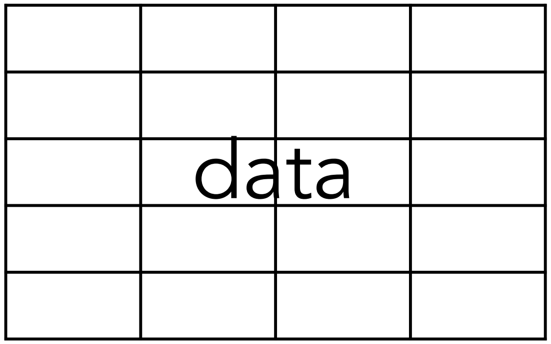
```
> data$column <- as.integer(data$column)
```

Exercise 4

```
> data$column <- as.integer(data$column)
```

To merge two data frames, we can use **joins** in dplyr

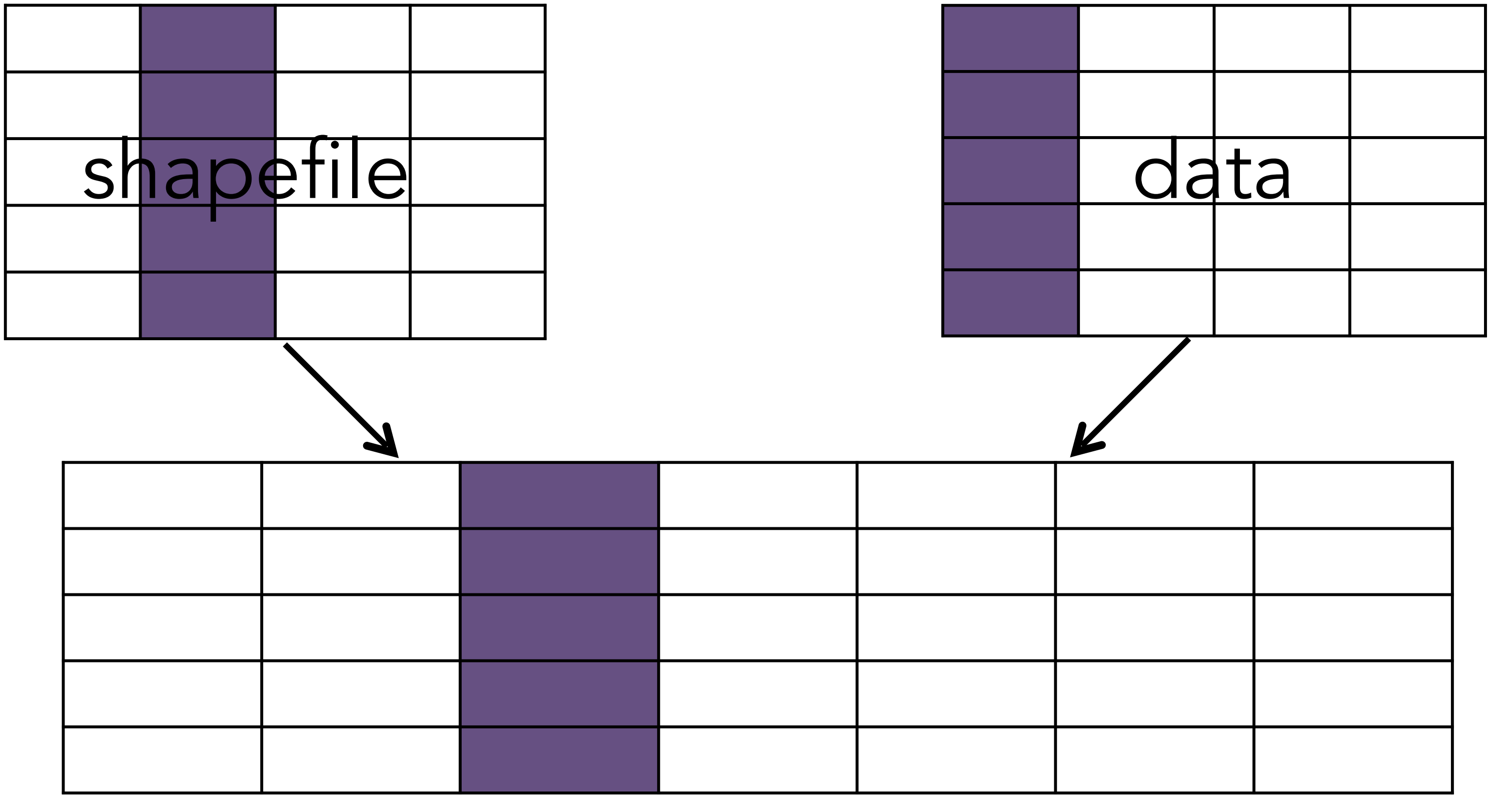




Exercise 4

```
> data$column <- as.integer(data$column)
```

To merge two data frames, we can use **joins** in dplyr



```
> newdata <- left_join(shapefile, data,
by=“commoncolumnname”
```

Exercise 4