Using R For Mapping with Shapefiles

Turning a Shapefile into a Presentable Figure

First Few Steps: Library And Reading in files

The first thing I need to do in order to create a map with a shapefile involves downloading the file. There are a few places on the web where you can find geographic shapefiles, and I picked up this shapefile from http://www.census.gov/cgi-bin/geo/shapefiles/index.php but http://www.data.gov also has great resources for shapefiles, if you can't find what you're looking for on the census' webpage.

I'm going to read in the data right here

```
library(maptools)
setwd("/Users/bjr/GitHub/MapProj/MapDemo")
shpin <- readShapePoly("tl_2016_20_sldl/tl_2016_20_sldl.shp")
```

One of the minuses of ggplot2 is that it absolutely needs a data.frame for plotting, and shapefiles come in with a unique structure, as seen above. The fortify() command transforms a shapefile or other kind of map into a dataframe suitable for plotting.

```
library(ggplot2)
shpfort <- fortify(shpin, region = "GEOID")
head(shpfort)
```

```
##        long      lat order  hole piece    id   group
## 1 -95.12256 37.03277     1 FALSE     1 20001 20001.1
## 2 -95.12249 37.03327     2 FALSE     1 20001 20001.1
## 3 -95.12224 37.03465     3 FALSE     1 20001 20001.1
## 4 -95.12198 37.03595     4 FALSE     1 20001 20001.1
## 5 -95.12194 37.03622     5 FALSE     1 20001 20001.1
## 6 -95.12189 37.03650     6 FALSE     1 20001 20001.1
```

This is just me making up some data to serve as coloration, but if you wanted there to be real data, you'd just need to read it in and merge it with your map data.frame. You do need to make sure you have a way of identifying which map goes where though!

```
shpID <- unique(shpfort$id)
prefcols <- data.frame("colid" = shpID, "col" = sample(1:100, length(shpID), replace = TRUE))
```

And here we merge the two dataframes together. Now if the id is the same, you can use the by argument (following the form by = "idGoesHere"), instead of by.x and by.y like I do below.

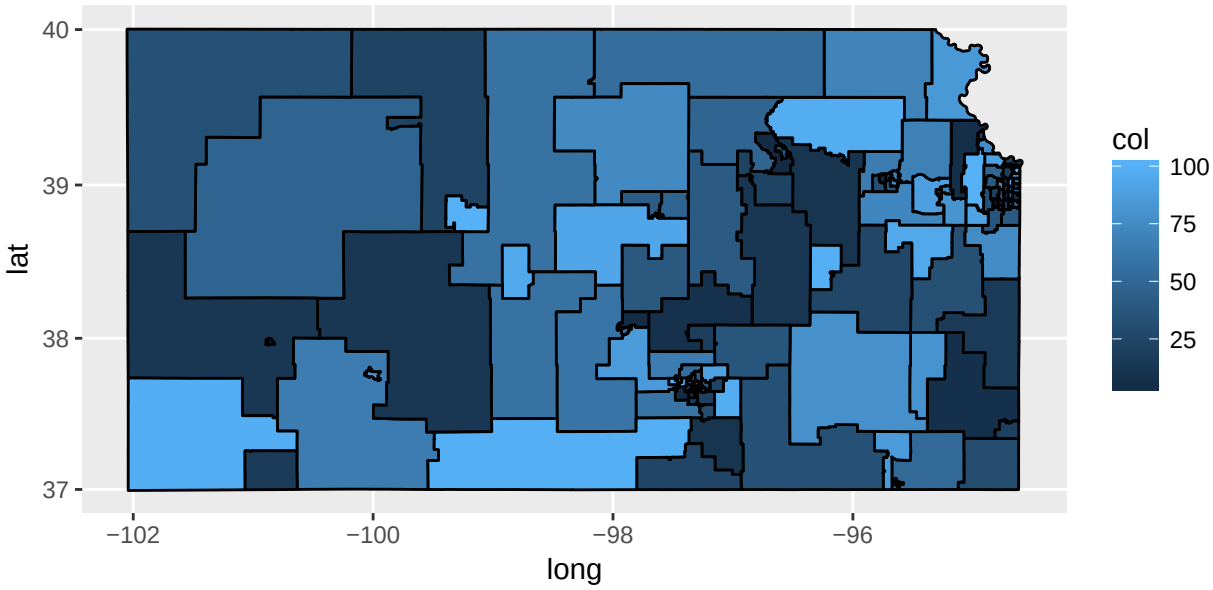
```
shpfort <- merge(x = shpfort, y = prefcols, by.x = "id", by.y = "colid", all.x = TRUE)
head(shpfort)
```

```
##      id      long      lat order  hole piece   group col
## 1 20001 -95.12256 37.03277     1 FALSE     1 20001.1  29
## 2 20001 -95.12249 37.03327     2 FALSE     1 20001.1  29
## 3 20001 -95.12224 37.03465     3 FALSE     1 20001.1  29
## 4 20001 -95.12198 37.03595     4 FALSE     1 20001.1  29
## 5 20001 -95.12194 37.03622     5 FALSE     1 20001.1  29
## 6 20001 -95.12189 37.03650     6 FALSE     1 20001.1  29
```

Making the Map

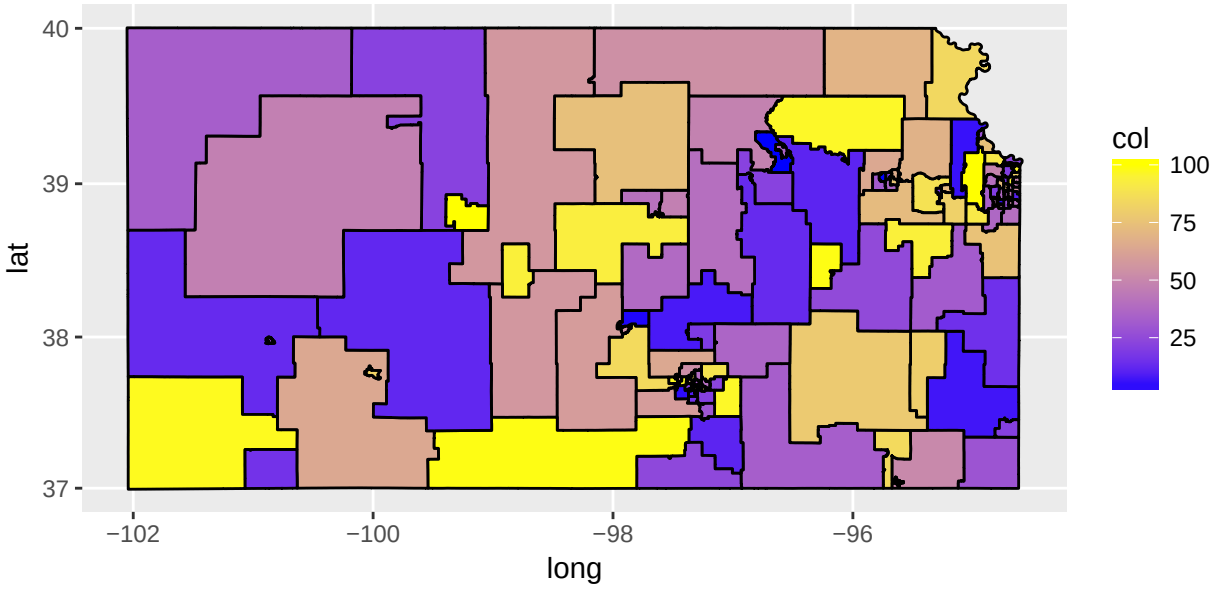
This is the initial ggplot command. Note that we start with ggplot, and add layers like geom_polygon() and coord_map()
coord_map() give us a mercator projection by default. It also resolves aspect ratio problems very easily.

```
shp <- ggplot(data = shpfort) + geom_polygon(aes(x = long, y = lat, group = group, fill = col, colour =
shp
```



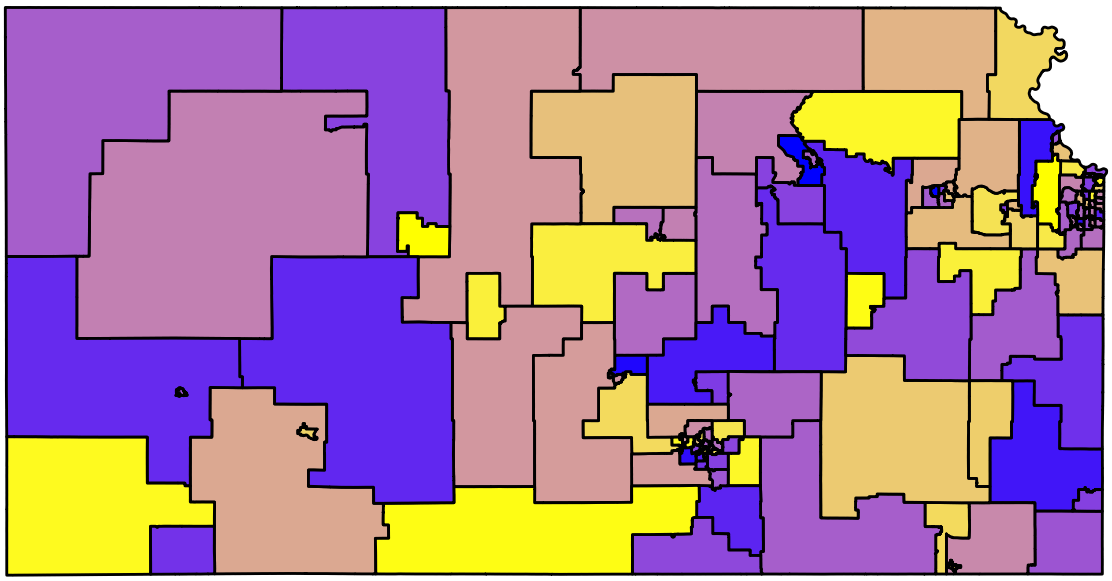
Change the colors to the ones I want. There are many ways to change colors in ggplot2, but this one lets me set a gradient between the limits I prefer.

```
shp <- shp + scale_fill_gradient(high = "Yellow", low = "Blue" )
shp
```



This command is optional, but it gets rid of the axes, background, ticks, and legend, which I don't need for this plot.

```
shp <- shp + theme(panel.background = element_blank(),
                   axis.title.x = element_blank(),
                   axis.title.y = element_blank(),
                   axis.text.x = element_blank(),
                   axis.text.y = element_blank(),
                   axis.ticks = element_blank(),
                   legend.position = "none")
shp
```



But I do want a title, so I put it together using the labs command, and I add a subtitle as well. I want the title centered, so I use the theme command again to get that, but this is once again, optional.

```
shp <- shp + labs(title = "Ben's Mapping Demonstration:", subtitle = "State Legislative Districts in Ka
shp
```

Ben's Mapping Demonstration:

State Legislative Districts in Kansas