first_maps

Pete

10/07/2014

This is my attempt to work through a mix of Lincoln's historic mapping tutorial: http://dh-r.lincolnmullen.com/mapping.html

```
library(rgdal)
```

```
## Loading required package: sp
## rgdal: version: 0.9-1, (SVN revision 518)
## Geospatial Data Abstraction Library extensions to R successfully loaded
## Loaded GDAL runtime: GDAL 1.7.3, released 2010/11/10
## Path to GDAL shared files: /usr/share/gdal/1.7
## GDAL does not use iconv for recoding strings.
## Loaded PROJ.4 runtime: Rel. 4.7.1, 23 September 2009, [PJ_VERSION: 470]
## Path to PROJ.4 shared files: (autodetected)
```

```
library(sp)
library(rgeos)
```

```
## rgeos version: 0.3-8, (SVN revision 460)
##  GEOS runtime version: 3.2.2-CAPI-1.6.2
##  Polygon checking: TRUE
```

```
library(maptools)
```

```
## Checking rgeos availability: TRUE
```

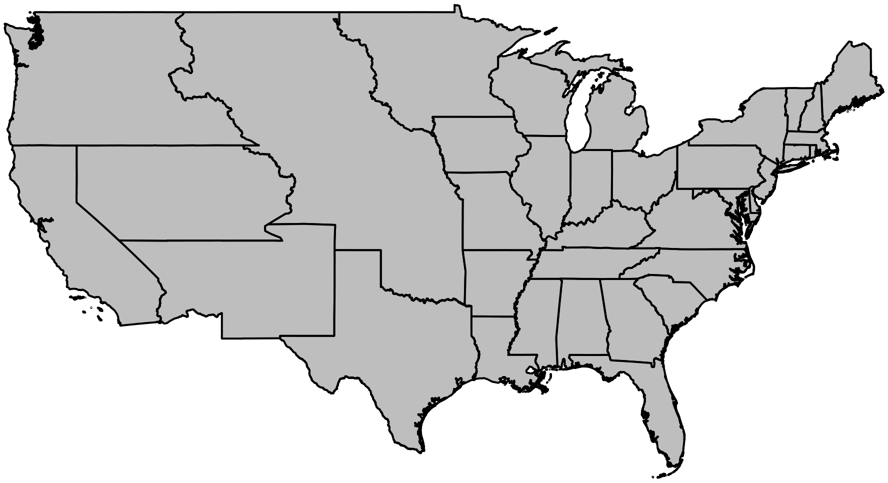
```
library(ggmap)
```

```
## Loading required package: ggplot2
```

```
library(ggplot2)

#load libraries
#First plot
map_sp <- readOGR("nhgis-shp/", "state_1850")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "nhgis-shp/", layer: "state_1850"
## with 37 features and 7 fields
## Feature type: wkbPolygon with 2 dimensions
```

```
plot(map_sp, col = "grey")
title("United States, 1850")
```

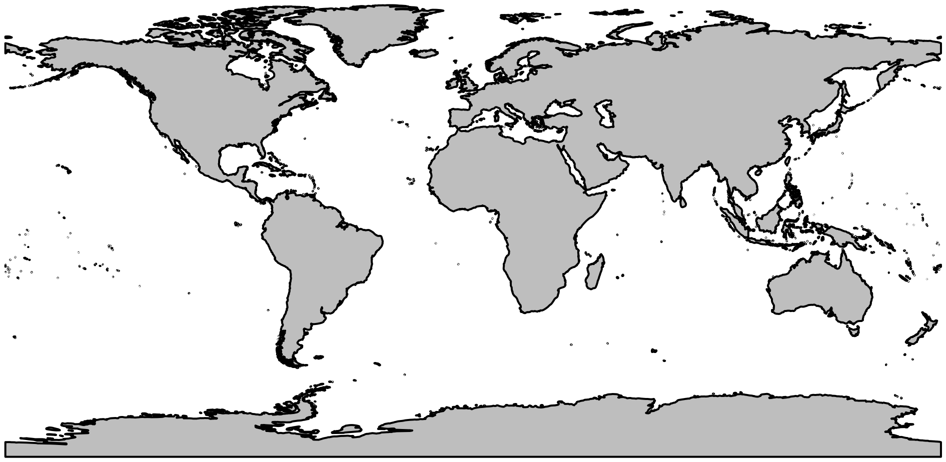


United States, 1850

```
#Second plot - Whitney was starting to get a little more into DH at this point.
earth <- readOGR("naturalearth/50m_physical/", "ne_50m_land")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "naturalearth/50m_physical/", layer: "ne_50m_land"
## with 1420 features and 2 fields
## Feature type: wkbPolygon with 2 dimensions
```

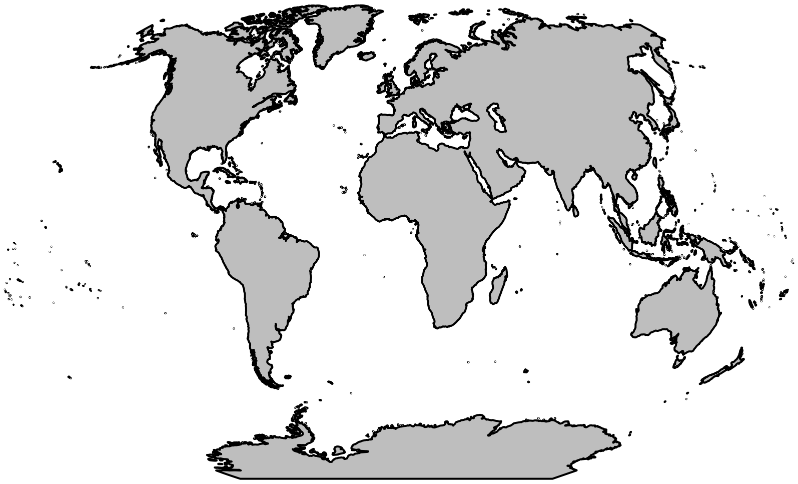
```
plot(earth, col = "gray"); title("Whitney's Queendom")
```



Whitney's Queendom

```
#Second plot transformed into a new projection
winkel <- spTransform(earth, CRS("+proj=wintri"))
plot(winkel, col="gray")
title("Whitney's world according to Oswald Winkel")
```

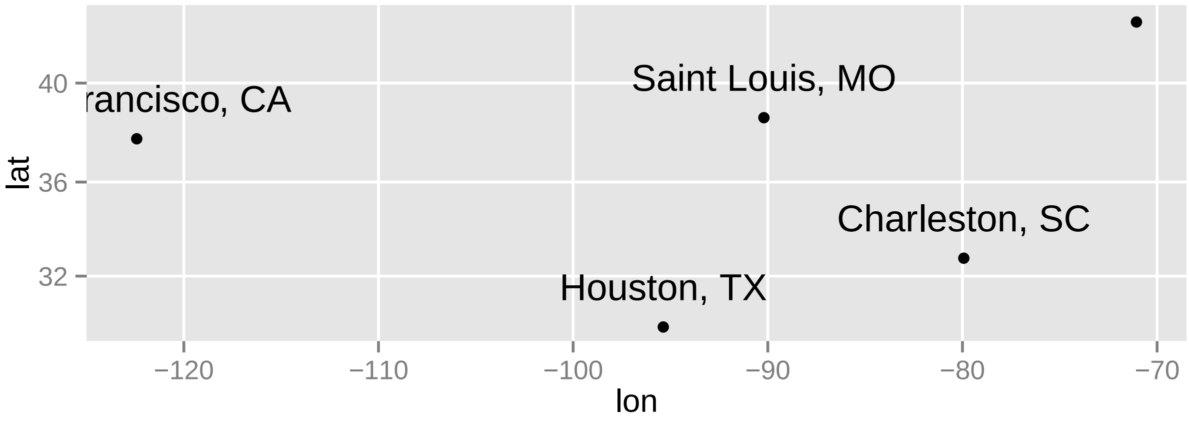
Whitney's world according to Oswald Winkel



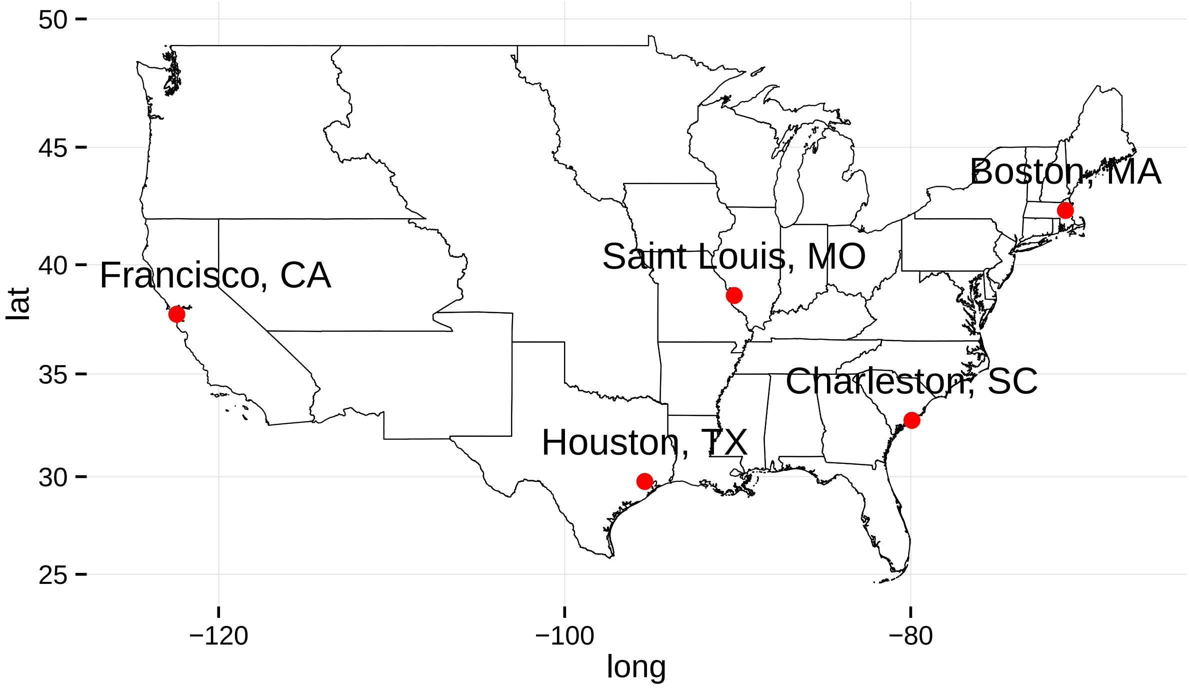
```r
#Created a list of cities
cities <- data.frame(name = c("Saint Louis, MO", "San Francisco, CA", "Boston, MA", "Charleston, SC", "

#Gave them a lat/long via ggmap's google map function and binded those fields to my cities DF
cities_geocoded <- geocode(cities$name)
```

```
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Saint+Louis,+MO&sens
## Google Maps API Terms of Service : http://developers.google.com/maps/terms
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=San+Francisco,+CA&se
## Google Maps API Terms of Service : http://developers.google.com/maps/terms
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Boston,+MA&sensor=fa
## Google Maps API Terms of Service : http://developers.google.com/maps/terms
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Charleston,+SC&senso
## Google Maps API Terms of Service : http://developers.google.com/maps/terms
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Houston,+TX&sensor=f
## Google Maps API Terms of Service : http://developers.google.com/maps/terms
```

```r
cities <- cbind(cities, cities_geocoded)

#Drew a plot of my cities in space using lat/long and labeled with the "name" column.
ggplot(cities, aes(x=lon, y = lat)) +
  geom_point() +
  geom_text(aes(label=name), vjust = -1) +
  coord_map()
```



```
#turn the sp into a data frame object.
map_df <- fortify(map_sp, region = "GISJOIN")
#And make a map with ggplot
map_1850 <- ggplot() +
  geom_map(data = map_df,
           map = map_df,
  aes(x=long, y=lat, group = group, map_id = id),
  fill = "white",
  color = "black",
  size = 0.2) +
coord_map() +
  theme_minimal()
#Add the "Cities" Data to our 1850 map. Voila!
map_1850 +
  geom_point(data = cities, aes(x = lon, y = lat), color = "red", size = 3) +
  geom_text(data = cities, aes(x = lon, y = lat, label=name), vjust = -1)
```

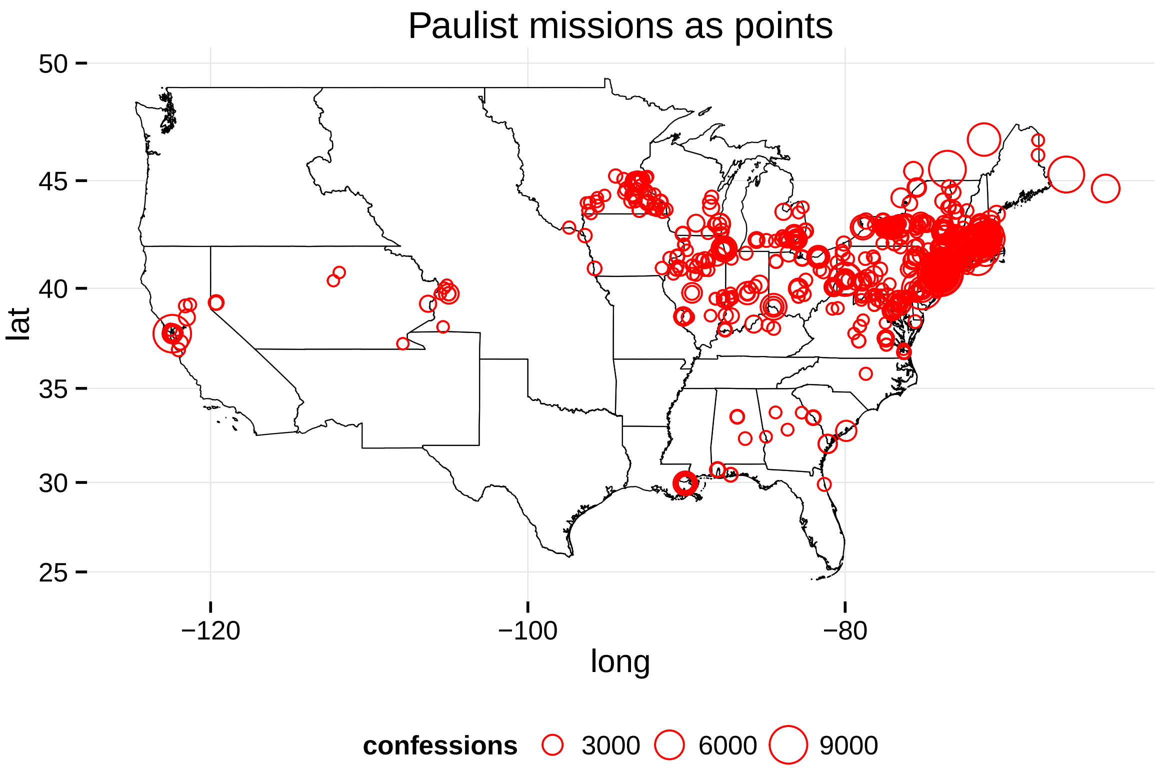


```
#Now a plot of the Paulist missions on the 1850 map:
#adding new packages
#devtools::install_github("lmullen/historydata")
library(dplyr)
```

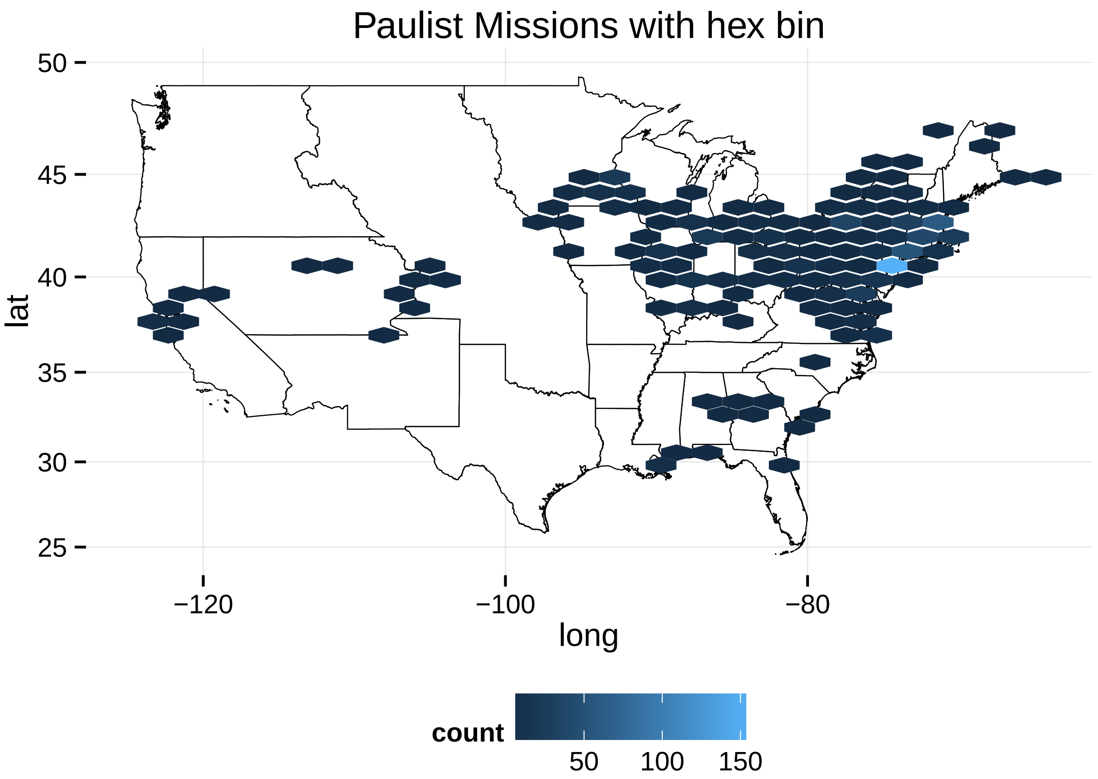
```
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:rgeos':
##
##     intersect, setdiff, union
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(historydata)

#adding paulist missions data frame
data(paulist_missions)

#Mapping Paulist missions as points.
map_1850 +
  geom_point(data = paulist_missions,
             aes(x=long, y = lat, size = confessions),
             color="red", shape = 1) +
  theme(legend.position="bottom") +
  scale_size(range = c(2, 8)) +
  ggtitle("Paulist missions as points")
```

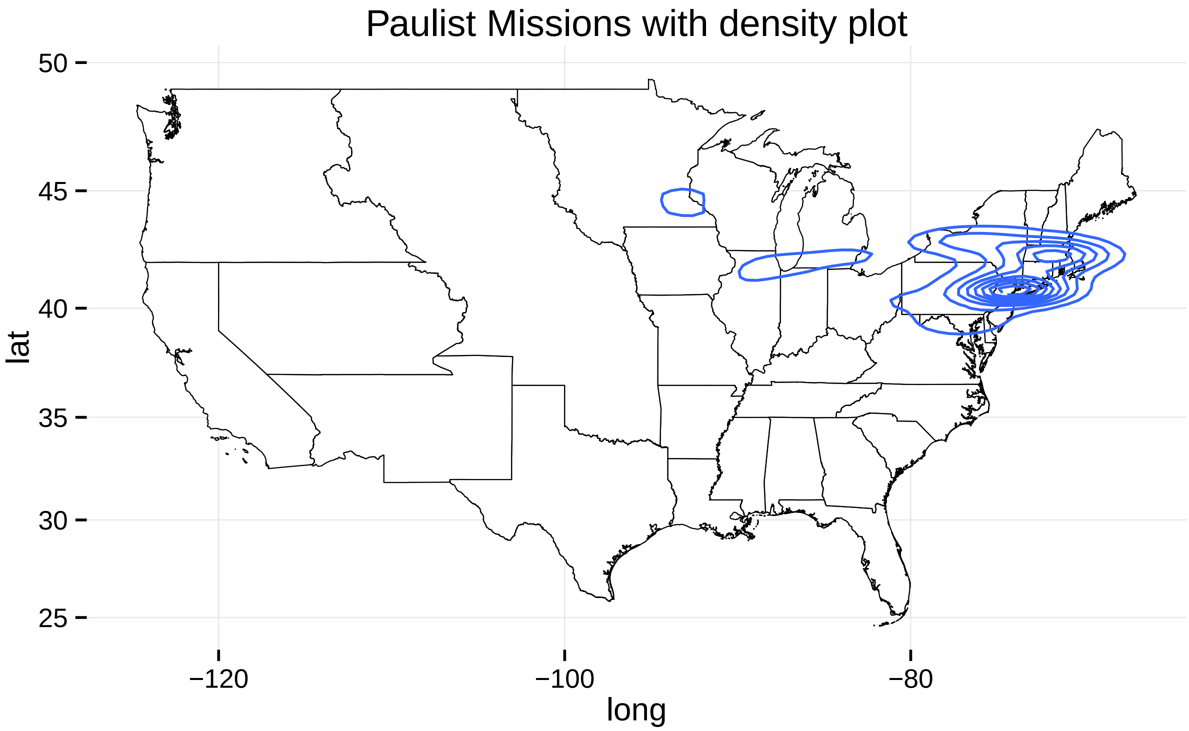
```
## Warning: Removed 6 rows containing missing values (geom_point).
```



```
#Paulist missions with Density instead of points
map_1850 +
  geom_hex(data = paulist_missions,
           aes(x = long, y = lat)) +
  theme(legend.position = "bottom") +
  ggtitle("Paulist Missions with hex bin")
```



```
#Paulist missions with density plot
map_1850 +
  geom_density2d(data = paulist_missions,
           aes(x = long, y = lat)) +
  theme(legend.position = "bottom") +
  ggtitle("Paulist Missions with density plot")
```

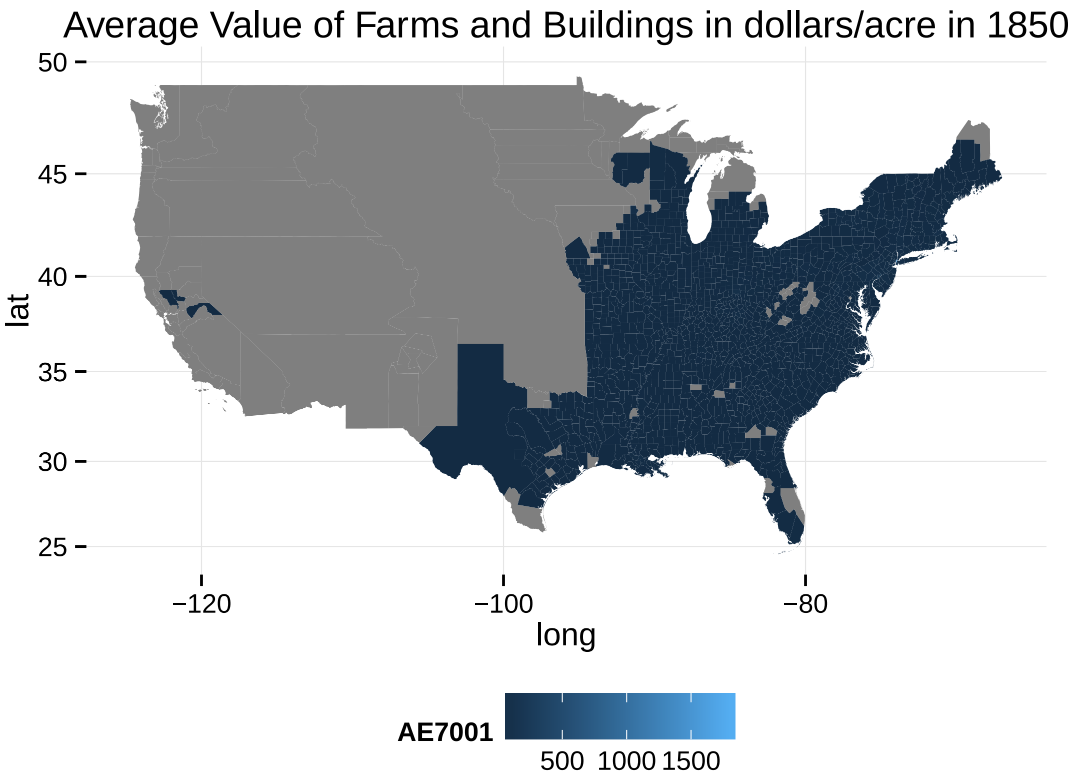


Exploring Chloropleths

```
#Adding Libraries
library(rgdal)
library(sp)
library(rgeos)
library(maptools)
library(ggmap)
library(ggplot2)
library(historydata)
library(dplyr)

#add us county data and farm data
counties_1850_sp <- readOGR("nhgis-shp/", "US_county_1850")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "nhgis-shp/", layer: "US_county_1850"
## with 1632 features and 20 fields
## Feature type: wkbPolygon with 2 dimensions
```

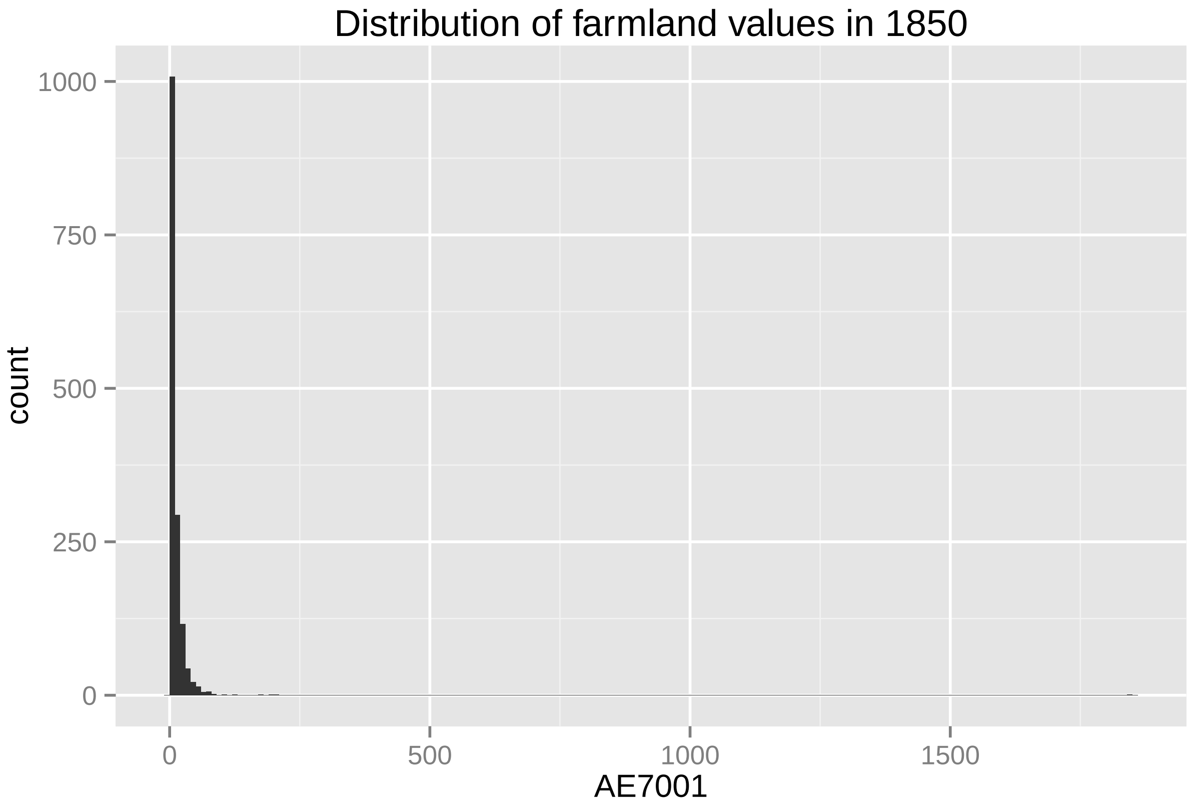
```
counties_1850_df <- fortify(counties_1850_sp, region ="GISJOIN")
farms_1850 <- read.csv("nhgis0003_csv/nhgis0003_ds11_1850_county.csv", stringsAsFactors = FALSE)
#merge the tables at GISJOIN/ID fields. I had to change the GISJOIN column to id in "farms_1850" to get
colnames(farms_1850)[1]<-"id"
farms_merged <- counties_1850_df %>%
  left_join(farms_1850, by = "id")
#Plot this map
ggplot(data = farms_merged,
       aes(x = long, y = lat, group = group, fill = AE7001, map_id = id)) +
  geom_map(map = farms_merged) +
  ggtitle("Average Value of Farms and Buildings in dollars/acre in 1850") +
  coord_map() +
  theme_minimal() +
  theme(legend.position = "bottom")
```


Average Value of Farms and Buildings in dollars/acre in 1850
50
45
40
35
30
25
lat
−120
−100
−80
long
AE7001
500
1000
1500

```
#Explore the farmland values table to improve map.
summary(farms_1850$AE7001)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0     4.0     7.0    12.1    12.0  1850.0
```

```
ggplot(data = farms_1850, aes (x = AE7001)) + geom_histogram(binwidth = 10) +
  ggtitle("Distribution of farmland values in 1850")
```



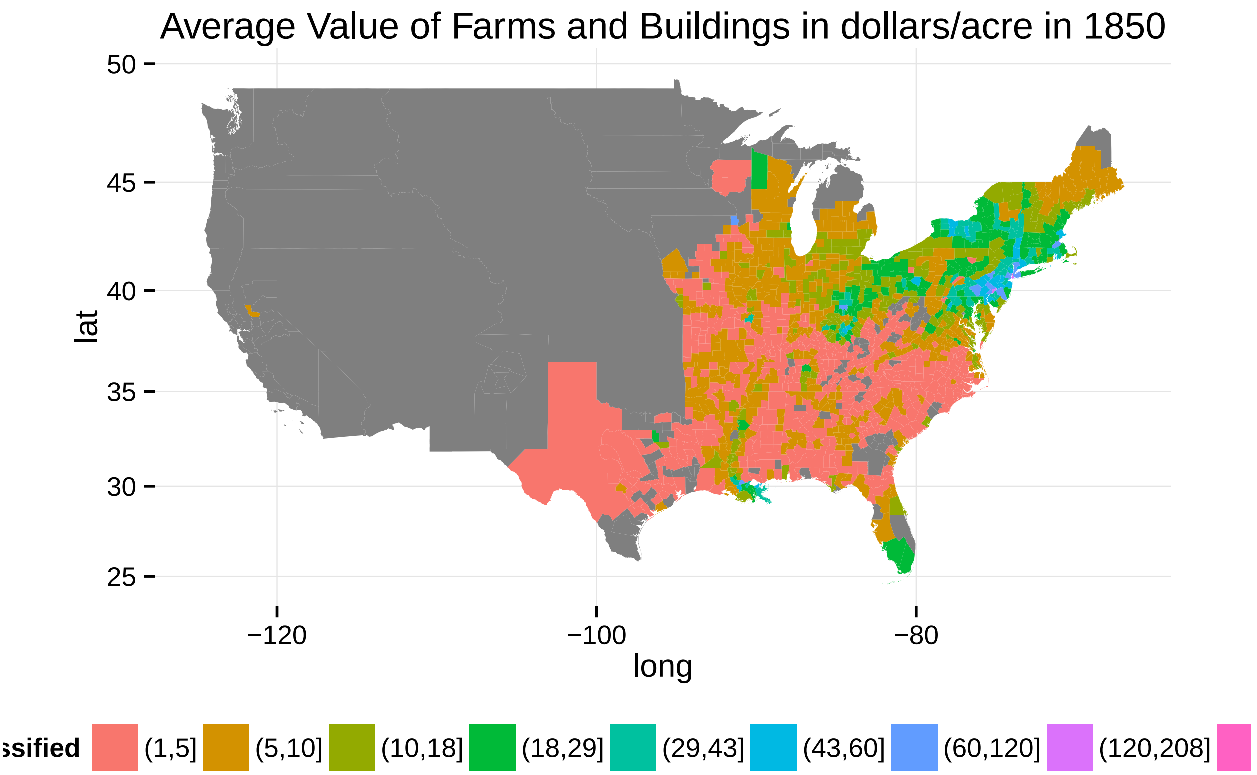
```
# Break values into bins with classInt package
#install.packages("classInt")
library(classInt)
classIntervals(farms_1850$AE7001, 9, "pretty")
```

```
## style: pretty
##   one of 6.503e+10 possible partitions of this variable into 10 classes
##      [0,200)    [200,400)    [400,600)    [600,800)   [800,1000) [1000,1200)
##         1515            1            0            0            0            0
## [1200,1400) [1400,1600) [1600,1800) [1800,2000]
##            0            0            0            1
```

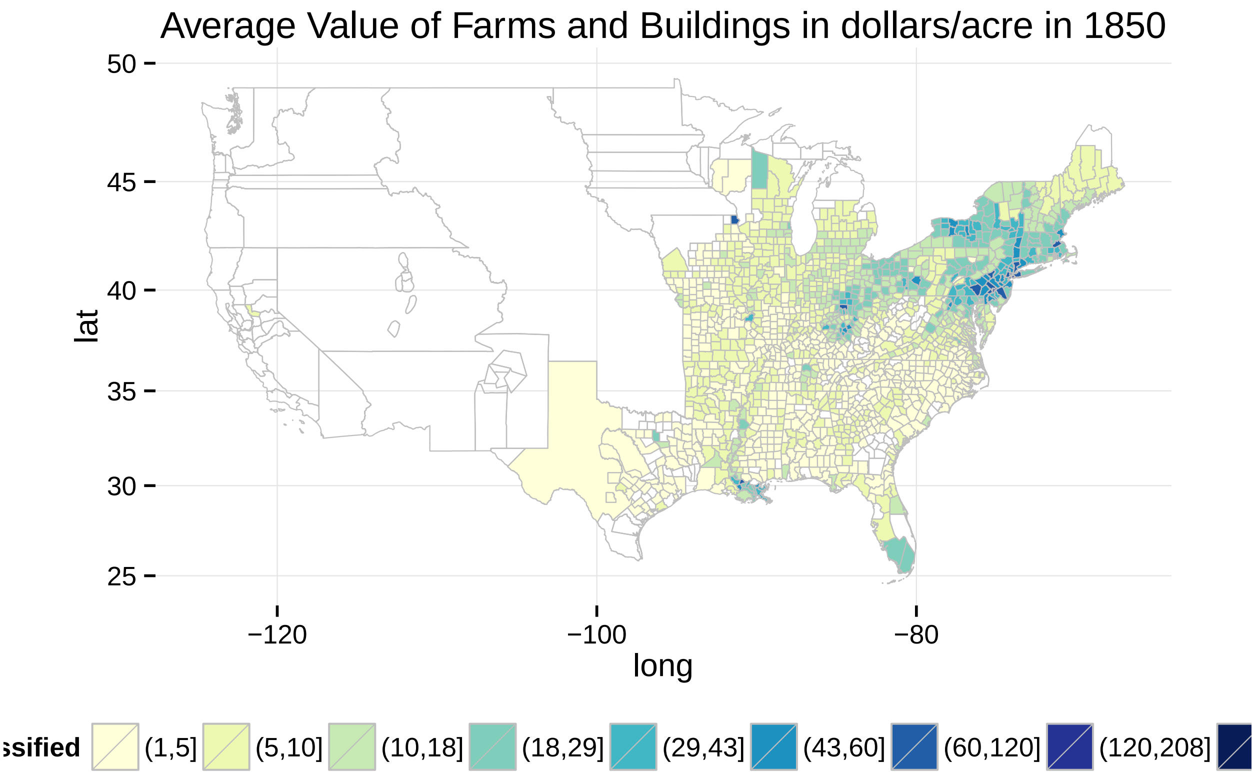
```
intervals <- classIntervals(farms_1850$AE7001, 9, "jenks")
head(cut(farms_1850$AE7001, breaks = intervals$brks))
```

```
## [1] (1,5] (1,5] (1,5] (1,5] (1,5] (1,5]
## 9 Levels: (1,5] (5,10] (10,18] (18,29] (29,43] (43,60] ... (208,1.85e+03]
```

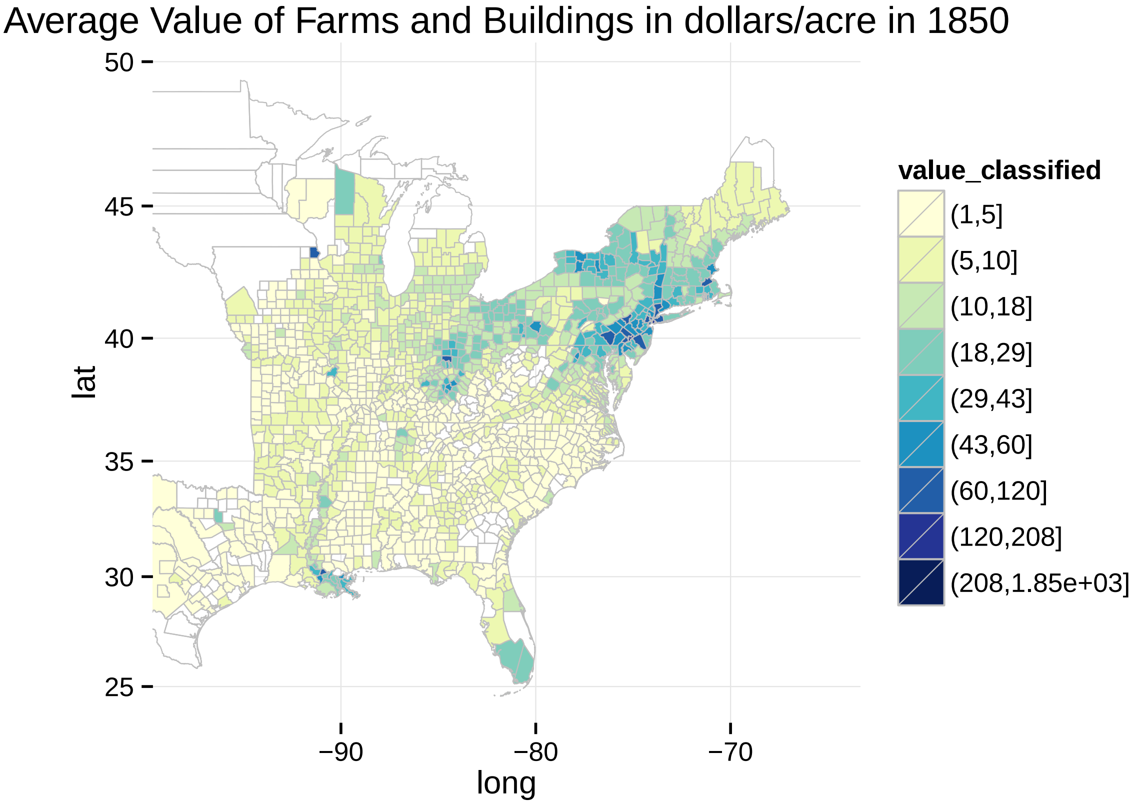
```
#merge it back into our farms dataframe
farms_1850 <- farms_1850 %>%
  mutate(value_classified = cut(AE7001, intervals$brks))
farms_merged <- counties_1850_df %>%
  left_join(farms_1850, by = "id")
#Now redo the map. Pretty!
ggplot(data = farms_merged,
       aes(x = long, y = lat, group = group,
           fill = value_classified, map_id = id)) +
  geom_map(map = farms_merged) +
  ggtitle("Average Value of Farms and Buildings in dollars/acre in 1850") +
  coord_map() +
  theme_minimal() +
  theme(legend.position = "bottom")
```



```
#Make the map prettier with newer, better colors
#install.packages("RColorBrewer")
library(RColorBrewer)
ggplot(data = farms_merged,
       aes(x = long, y = lat, group = group, fill = value_classified, map_id = id)) +
  geom_map(map = farms_merged, color="gray", size = 0.2) +
  ggtitle("Average Value of Farms and Buildings in dollars/acre in 1850") +
  coord_map() +
  theme_minimal() +
  theme(legend.position = "bottom") +
  scale_fill_brewer(palette = "YlGnBu")
```



```r
#And limited x/y axis to show only Eastern US, move the legend to the side.
ggplot(data = farms_merged,
       aes(x = long, y = lat, group = group,
           fill = value_classified, map_id = id)) +
  geom_map(map = farms_merged, color="gray", size = 0.2) +
  ggtitle("Average Value of Farms and Buildings in dollars/acre in 1850") +
  coord_map() +
  theme_minimal() +
  scale_fill_brewer(palette = "YlGnBu") +
  xlim(-98, -65)
```



Now that's a map!