Map Visualization

Mike Chapple

1/21/2020

This week, we will explore the use of Google's map and geocoding APIs with our data. We first need to register a Google API key. You should change the value below to a key that you obtain for your own use.

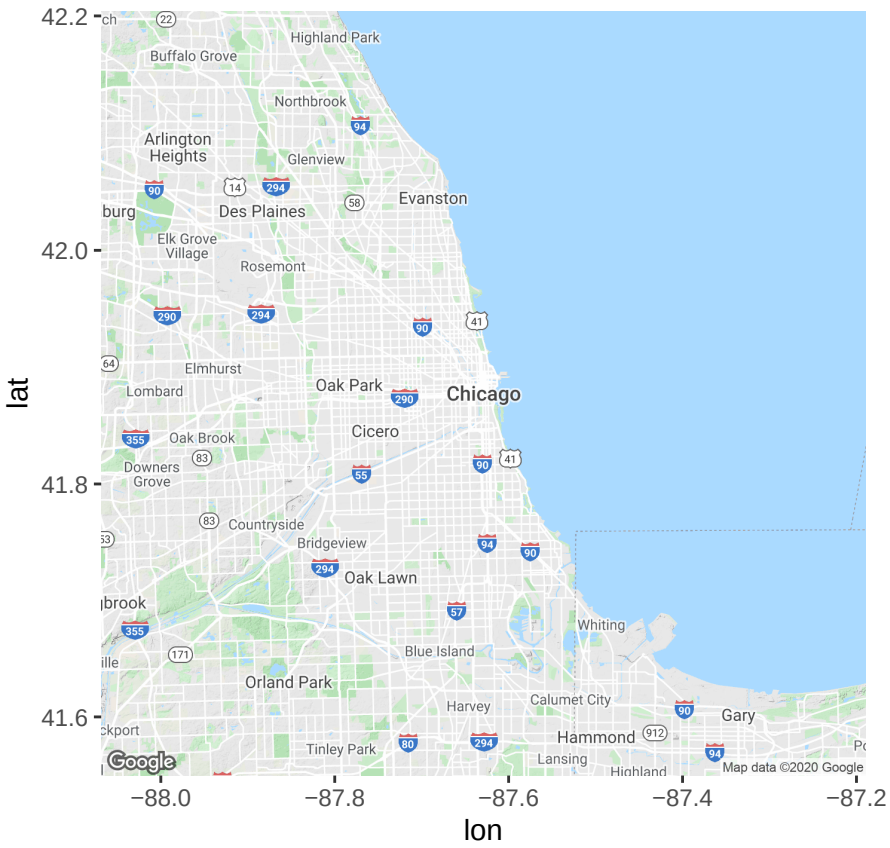
Working with Map Data

Let's begin by building a map of Chicago with the qmap function.

```
qmap("Chicago, IL", zoom=18)
```

And now I'll try with the get_map function.

```
chicago_map <- get_map("Chicago, IL", zoom=10)
ggmap(chicago_map)
```



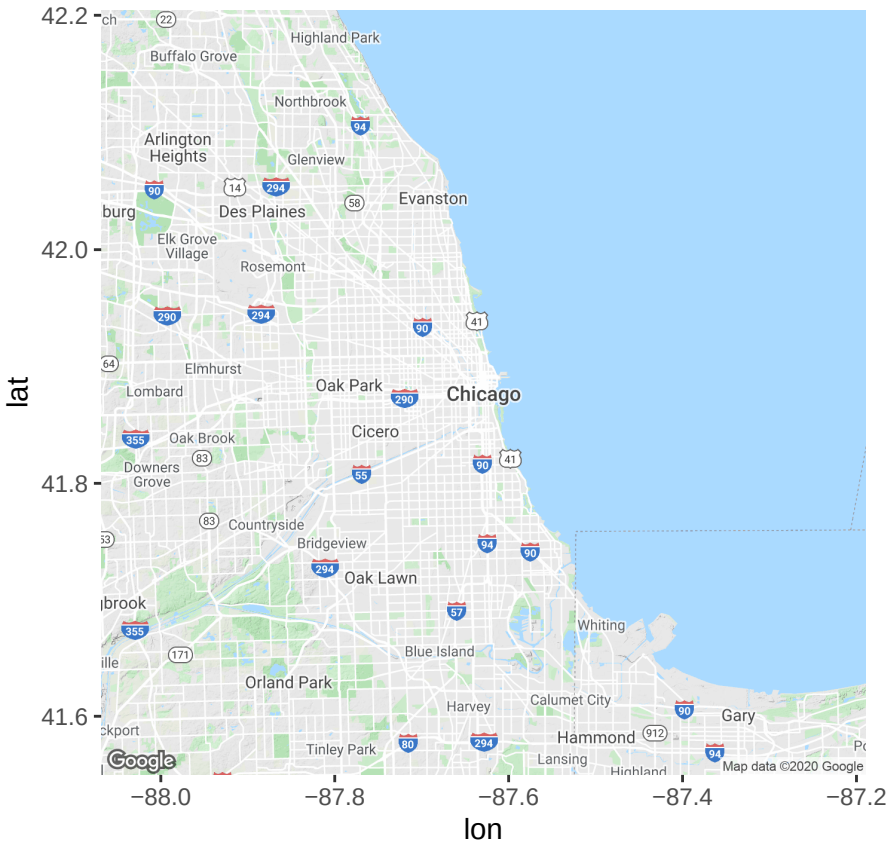
Geocoding

Geocoding is the process of converting a location to a latitude and longitude. Let's try a few examples.

```
chicago <- geocode("Chicago, IL")
campus <- geocode("224 S Michigan Ave, Chicago, IL")
mendoza <- geocode("Mendoza College of Business")
whitehouse <- geocode("White House")
```

Let's get a map of the Chicago campus now.

```
chicago_map <- get_map(chicago)
ggmap(chicago_map)
```



I can also wrap the get_map() call in a call to ggmap() and save the step of storing the map in a variable.

```
ggmap(get_map(mendoza, zoom=17))
```

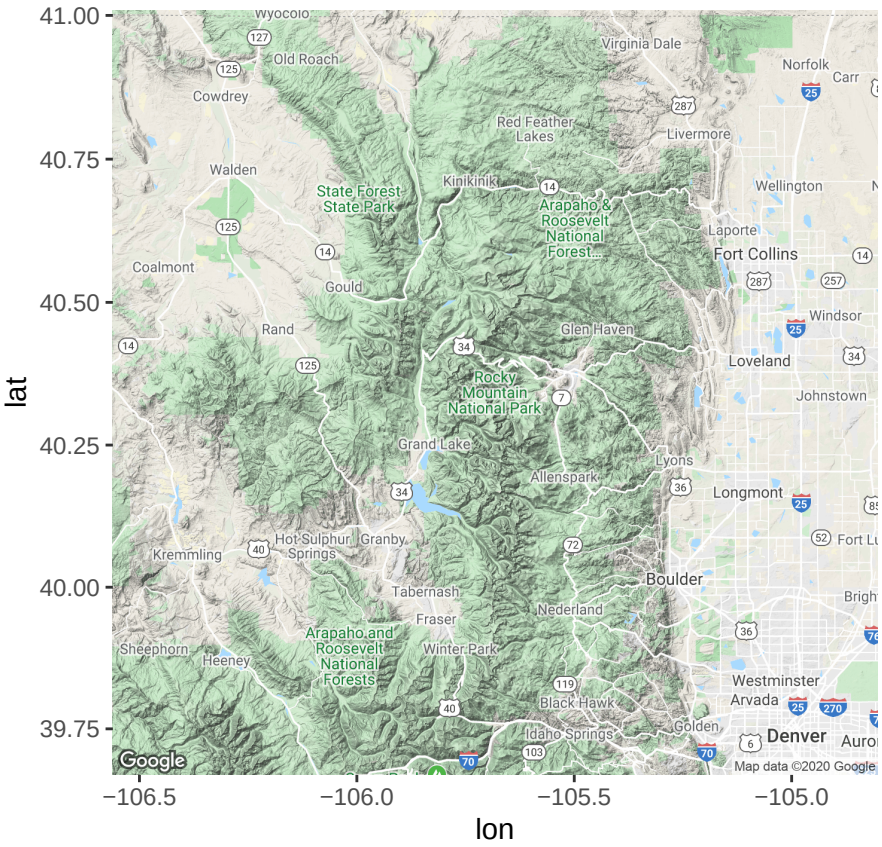
Changing Map Types

We can also change the type of our maps.

Terrain Maps

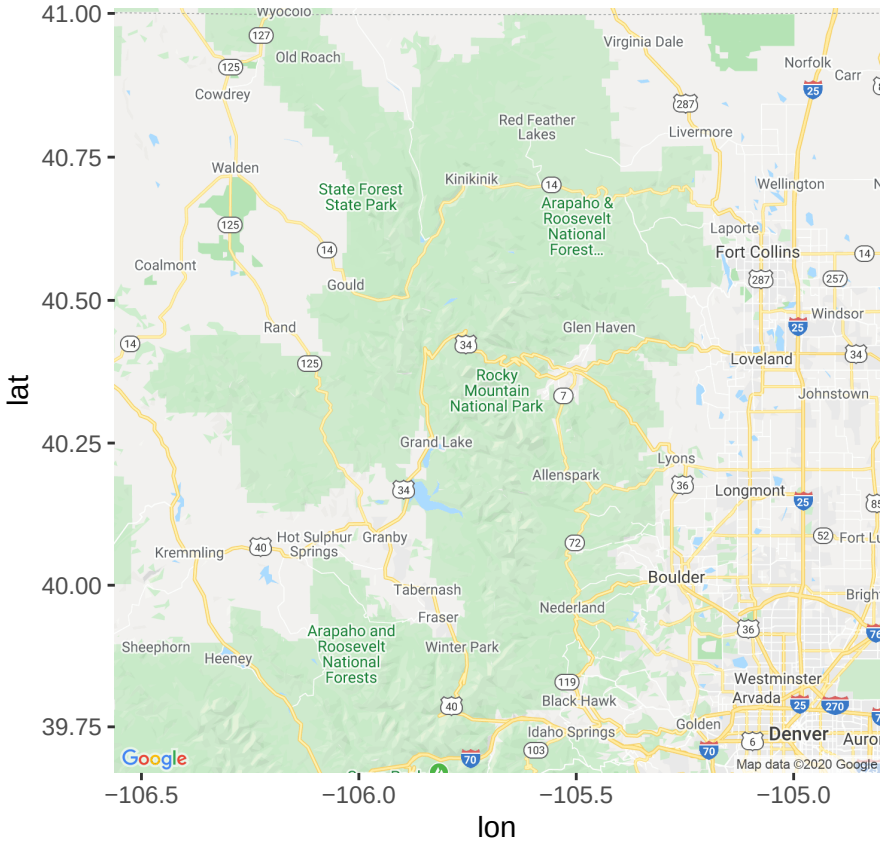
The default map type is terrain.

```
rockies<-geocode("Rocky Mountain National Park")
ggmap(get_map(rockies, zoom=9, maptype = "terrain"))
```



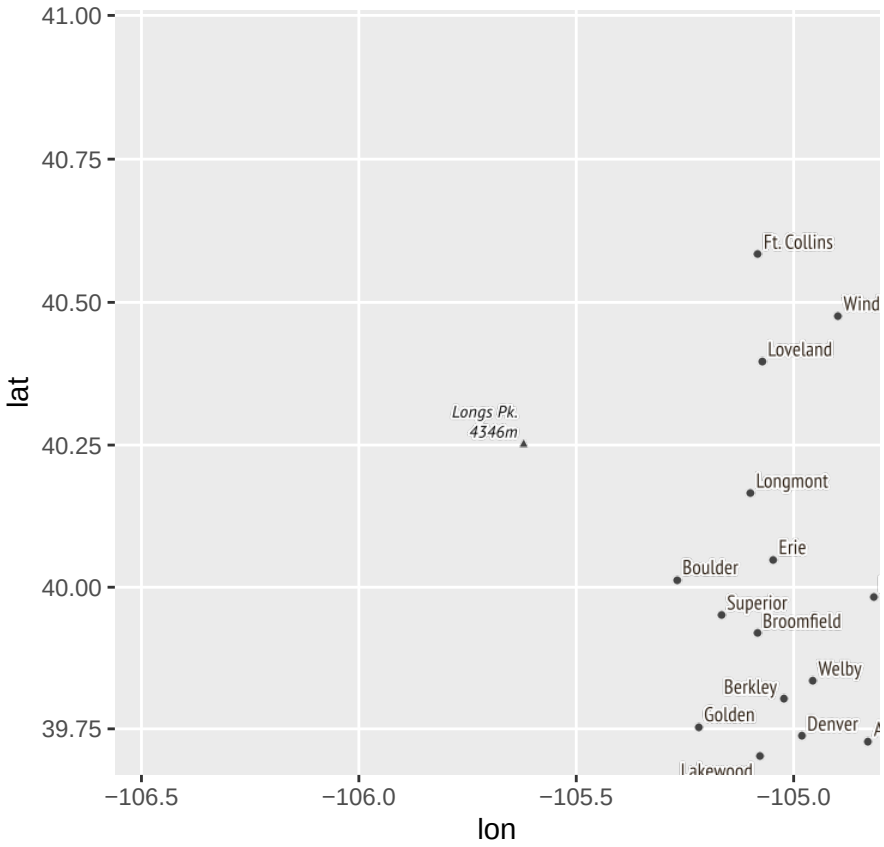
Road Maps

```
ggmap(get_map(rockies, zoom=9, maptype = "roadmap"))
```



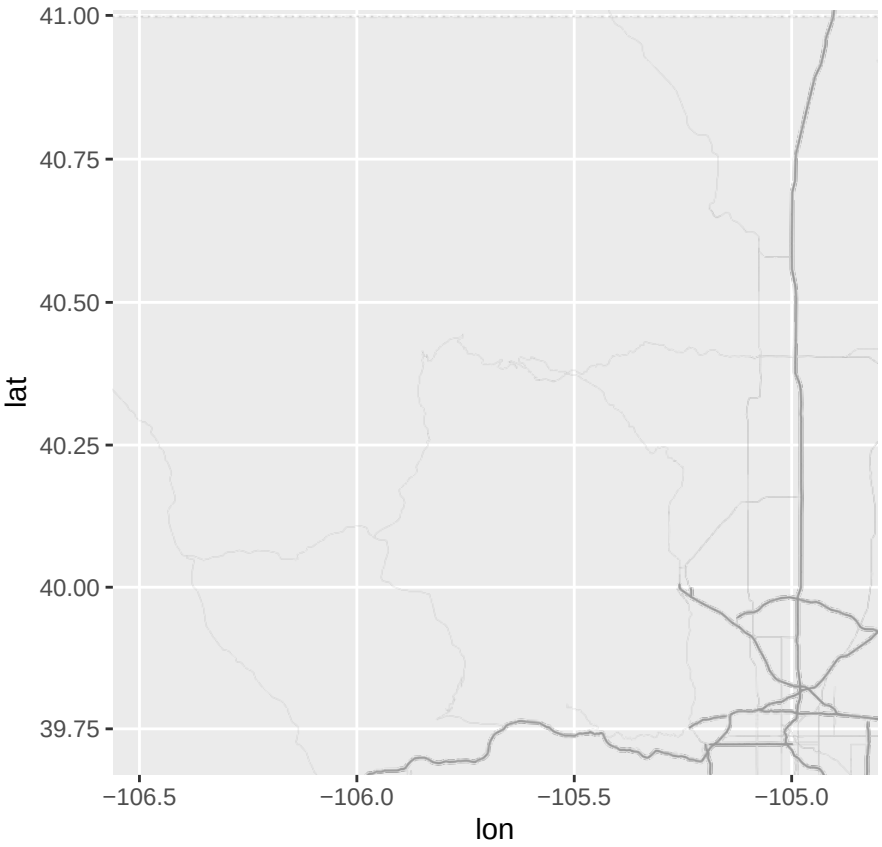
Terrain Labels

```
ggmap(get_map(rockies, zoom=9, maptype='terrain-labels'))
```



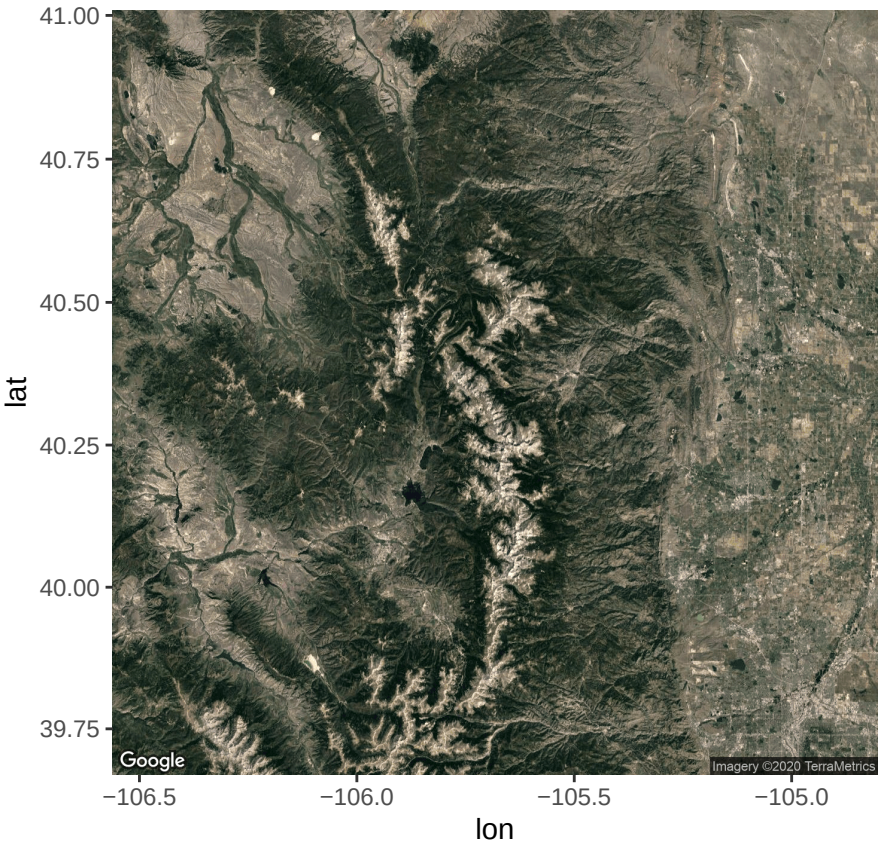
Terrain Lines

```
ggmap(get_map(rockies, zoom=9, maptype='terrain-lines'))
```



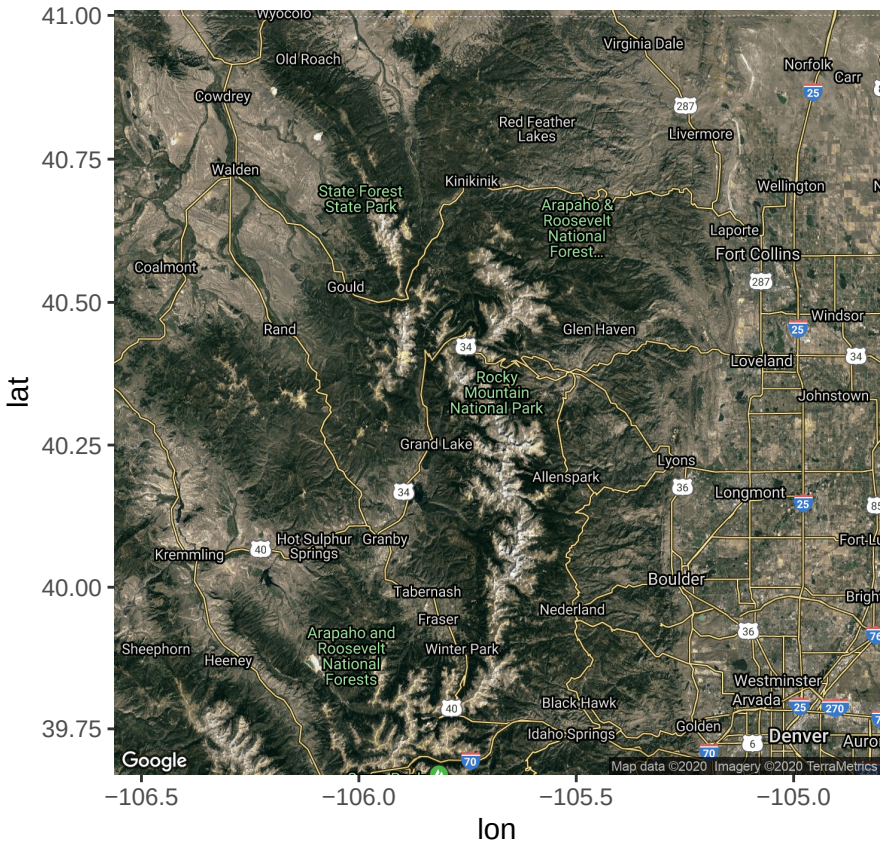
Satellite

```
ggmap(get_map(rockies, zoom=9, maptype='satellite'))
```



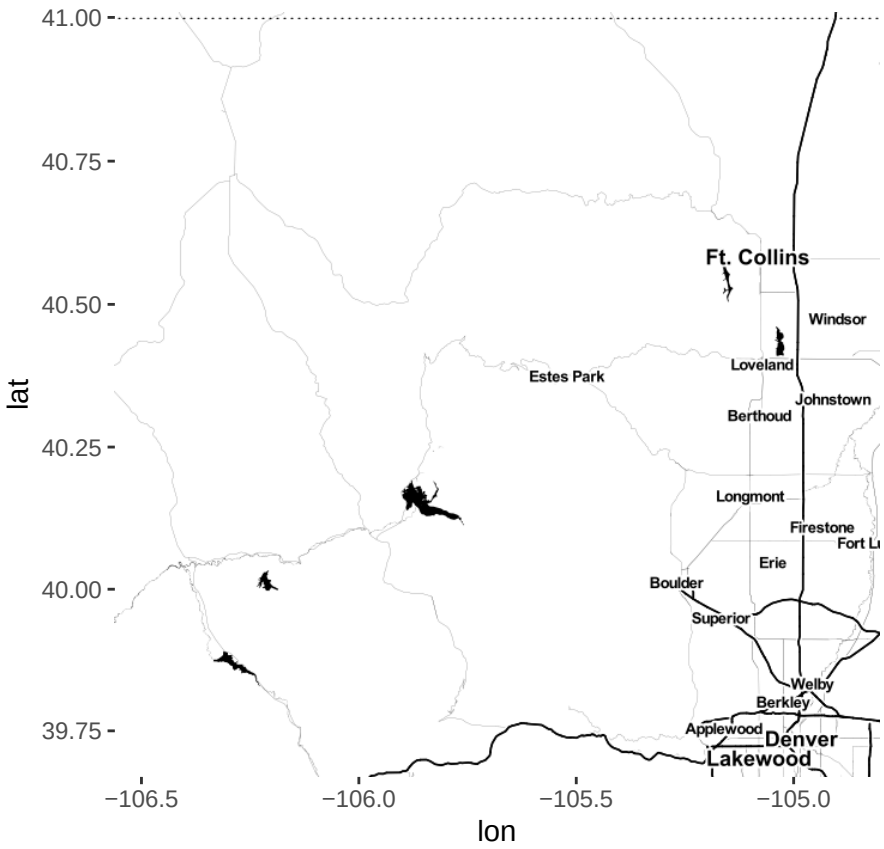
Hybrid

```
ggmap(get_map(rockies, zoom=9, maptype='hybrid'))
```



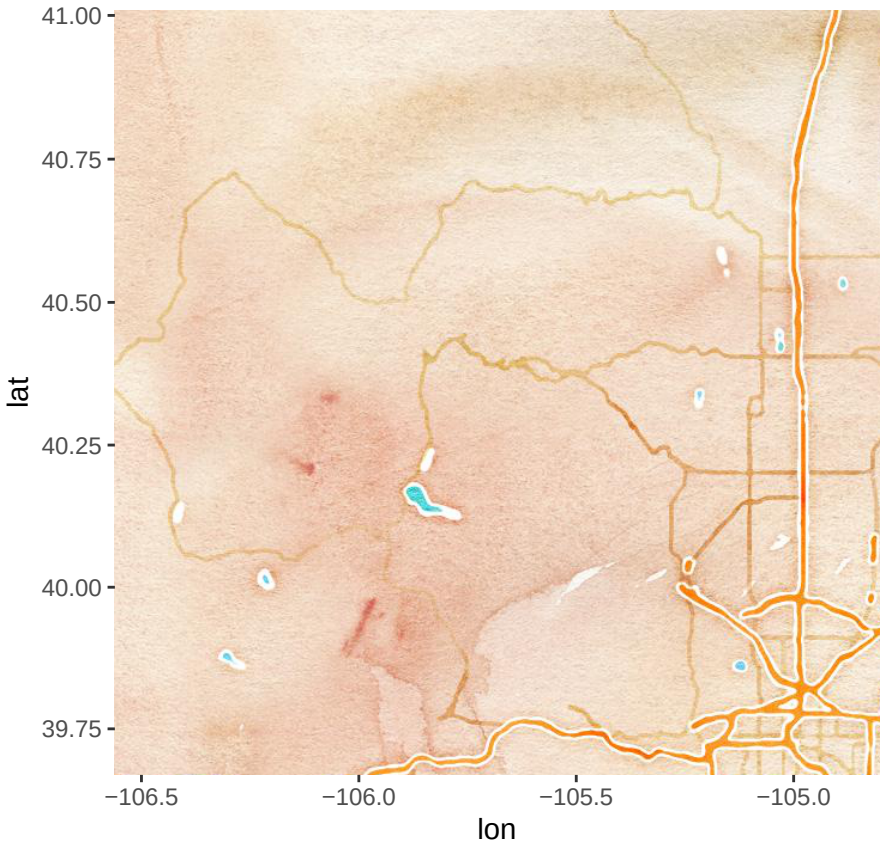
Toner

```
ggmap(get_map(rockies, zoom=9, maptype='toner'))
```



Watercolor

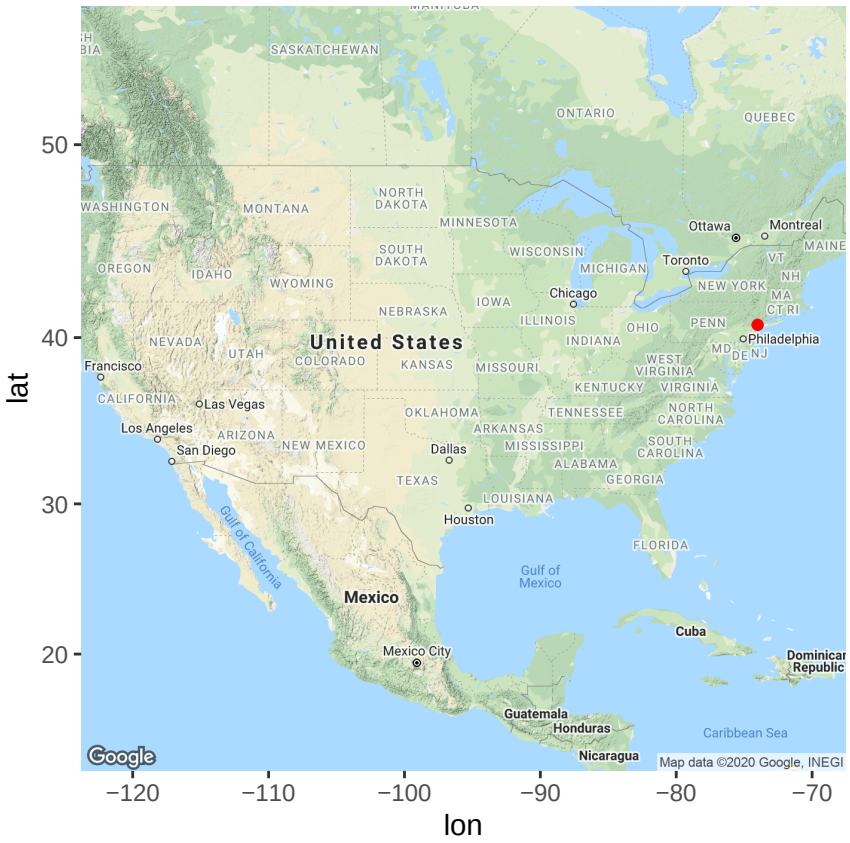
```
ggmap(get_map(rockies, zoom=9, maptype='watercolor'))
```



Plotting Points on a Map

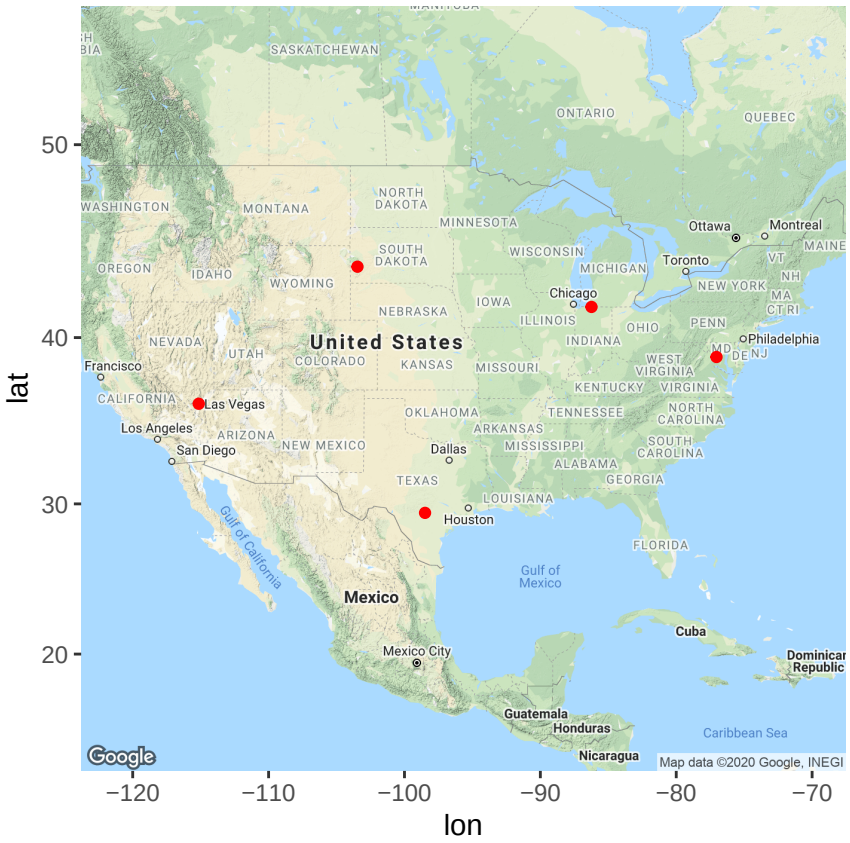
Let's try plotting New York City on a map of the United States.

```
usa <- geocode('United States')
nyc <- geocode('New York City')
ggmap(get_map(usa, zoom=4)) +
  geom_point(mapping=aes(x=lon, y=lat), color="red",data=nyc)
```



Geocoding and Plotting a Vector

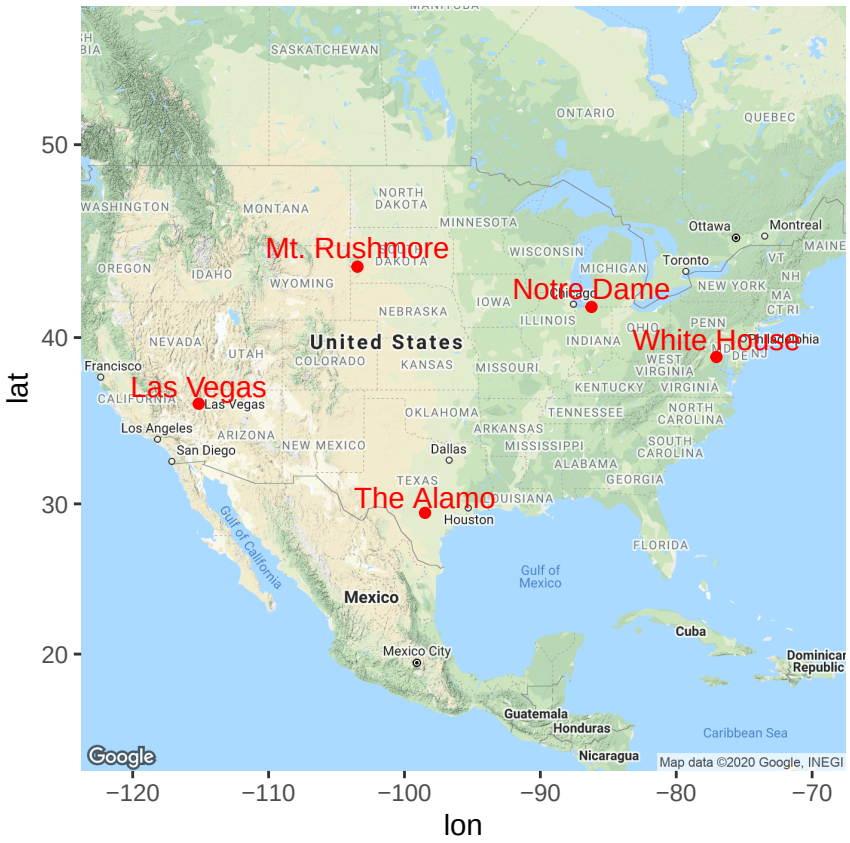
```
placenames <- c("Las Vegas", "White House", "Notre Dame", "Mt. Rushmore", "The Alamo")
locations <- geocode(placenames)
places <- tibble(name=placenames, lat=locations$lat,lon=locations$lon)
ggmap(get_map(usa, zoom=4)) +
  geom_point(mapping=aes(x=lon, y=lat), color="red",data=places)
```



Adding Labels

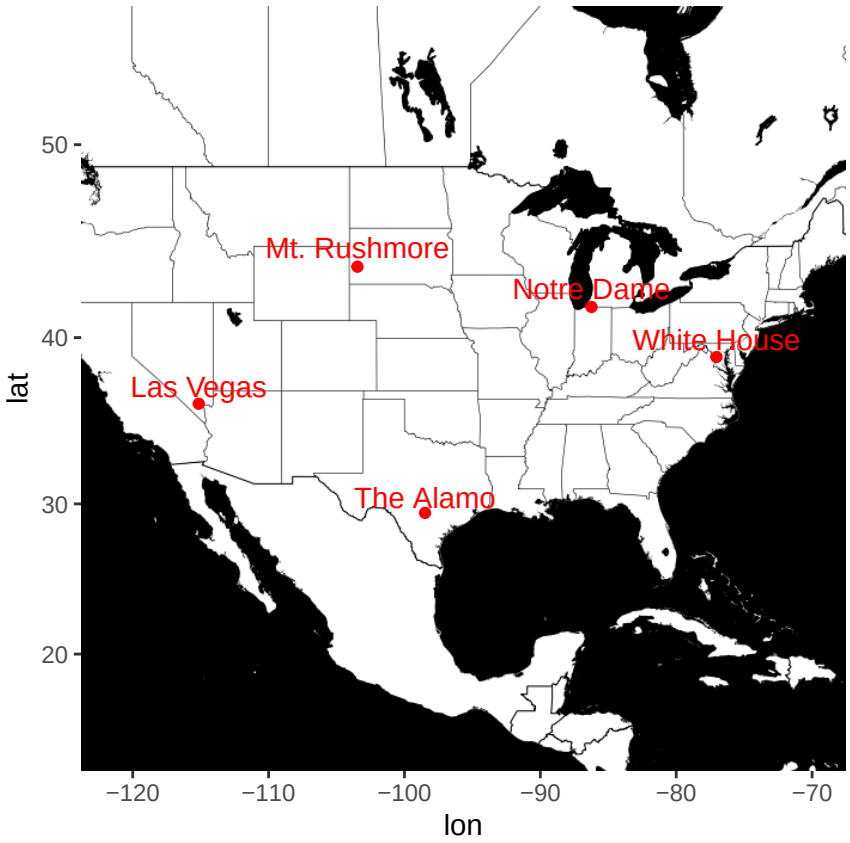
We can add text to our points with the geom_text geometry.

```
ggmap(get_map(usa, zoom=4)) +
  geom_point(mapping=aes(x=lon, y=lat), color="red",data=places) +
  geom_text(mapping=aes(x=lon, y=lat, label=name), color="red", nudge_y=1, data=places)
```



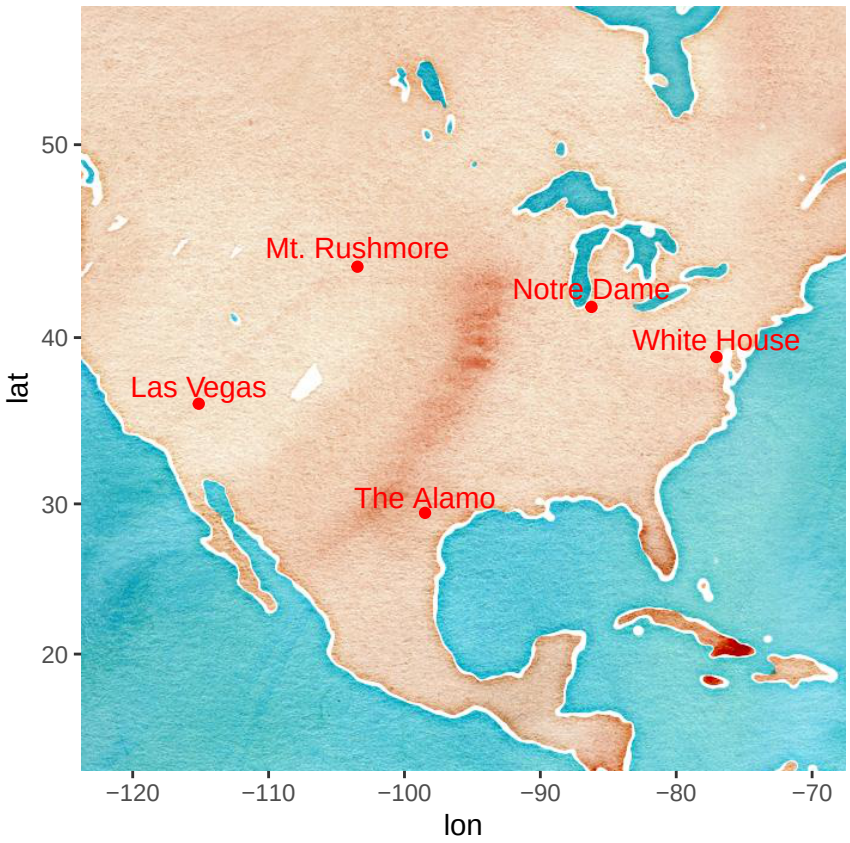
Let's see if using a toner background map would declutter this a bit.

```
ggmap(get_map(usa, zoom=4, maptype = "toner-background")) +
  geom_point(mapping=aes(x=lon, y=lat), color="red",data=places) +
  geom_text(mapping=aes(x=lon, y=lat, label=name), color="red", nudge_y=1, data=places)
```



And what would this look like in watercolor?

```
ggmap(get_map(usa, zoom=4, maptype = "watercolor")) +
  geom_point(mapping=aes(x=lon, y=lat), color="red",data=places) +
  geom_text(mapping=aes(x=lon, y=lat, label=name), color="red", nudge_y=1, data=places)
```

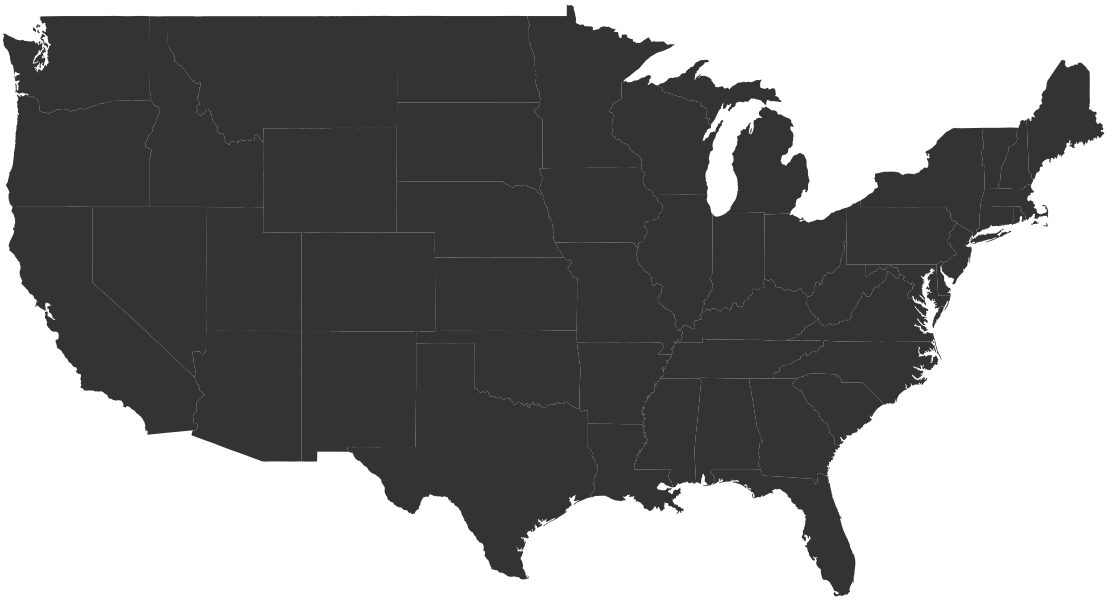


Building a Map Manually

We can always build our own map from scratch. Let's give that a try.

```
states <- map_data("state")
```

```
ggplot(data=states, mapping=aes(x=long, y=lat, group=group)) +
  geom_polygon() +
  coord_map() +
  theme(axis.ticks=element_blank(), axis.text=element_blank(), axis.title=element_blank()) +
  theme(panel.background=element_blank())
```



Choropleth Maps

Let's load the college dataset from Data Management.

```
college <- read_csv('https://s3.amazonaws.com/chapple-datasets/college.csv')
```

```
## Parsed with column specification:
## cols(
##   id = col_double(),
##   name = col_character(),
##   city = col_character(),
##   state = col_character(),
##   region = col_character(),
##   highest_degree = col_character(),
##   control = col_character(),
##   gender = col_character(),
##   admission_rate = col_double(),
##   sat_avg = col_double(),
##   undergrads = col_double(),
##   tuition = col_double(),
##   faculty_salary_avg = col_double(),
##   loan_default_rate = col_character(),
##   median_debt = col_double(),
##   lon = col_double(),
```

```
##   lat = col_double()
## )
```

```
college <- college %>%
  mutate(state=as.factor(state), region=as.factor(region),
         highest_degree=as.factor(highest_degree),
         control=as.factor(control), gender=as.factor(gender),
         loan_default_rate=as.numeric(loan_default_rate))
```

```
## Warning: NAs introduced by coercion
```

```
college_summary <- college %>%
  group_by(state) %>%
  summarize(schools=n())
```

I can use the setNames function along with R's built-in state.name and state.abb vectors to clean up the fact that one table has full state names and the other has abbreviations.

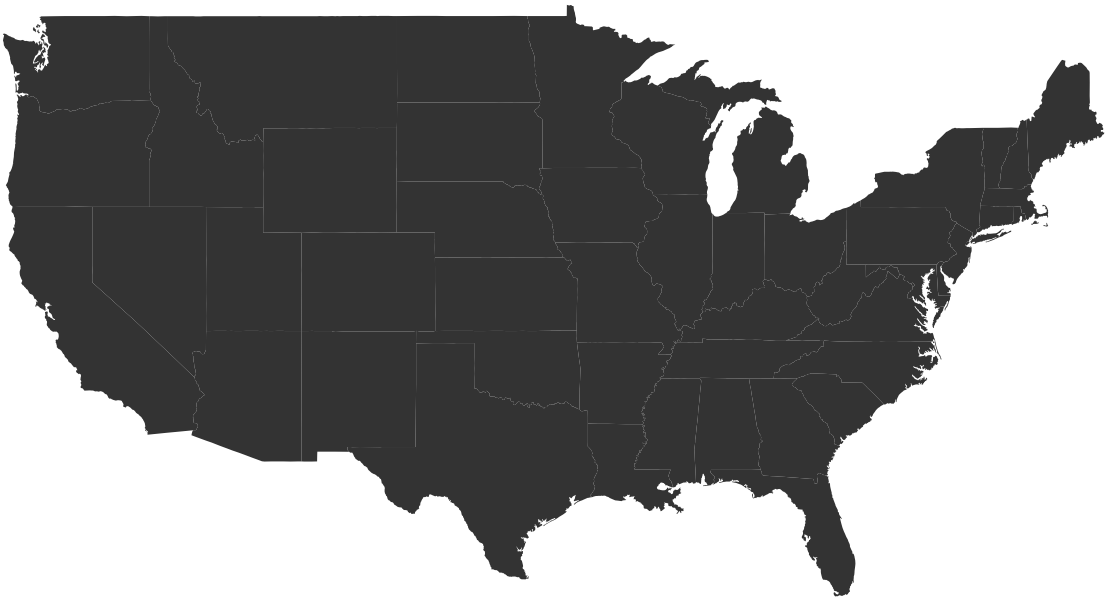
```
college_summary <- college_summary %>%
  mutate(region=as.character(setNames(str_to_lower(state.name), state.abb)[as.character(state)]))

college_summary <- college_summary %>%
  mutate(region=ifelse(as.character(state)=="DC", "district of columbia",region))

mapdata <- merge(states, college_summary, by="region")
```

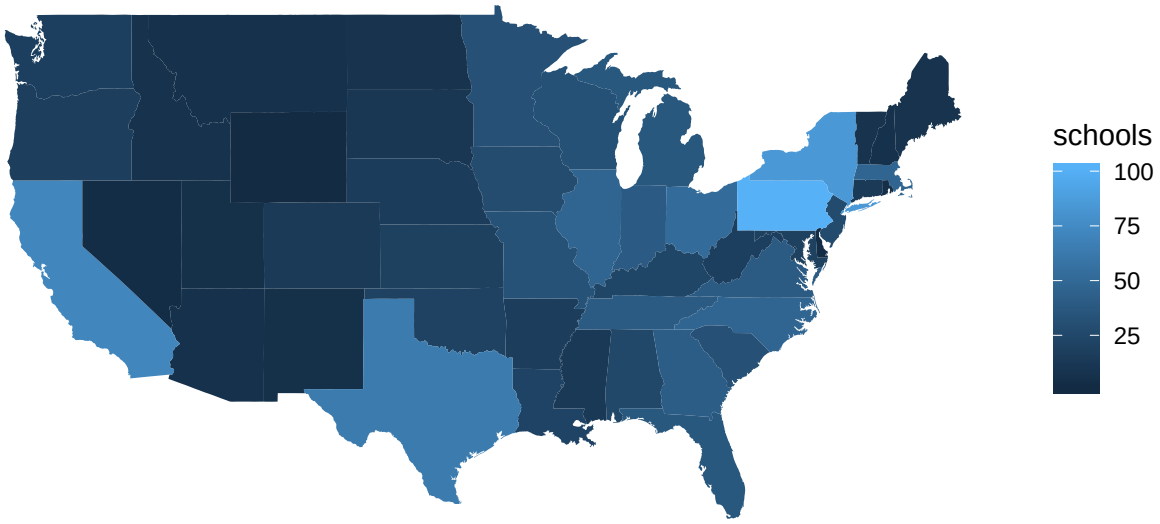
Let's start with a basic state map.

```
ggplot(mapdata) +
  geom_polygon(aes(x=long,y=lat,group=group)) +
  coord_map() +
  theme(plot.background=element_blank(), panel.background = element_blank(), axis.title=element_blank()
```



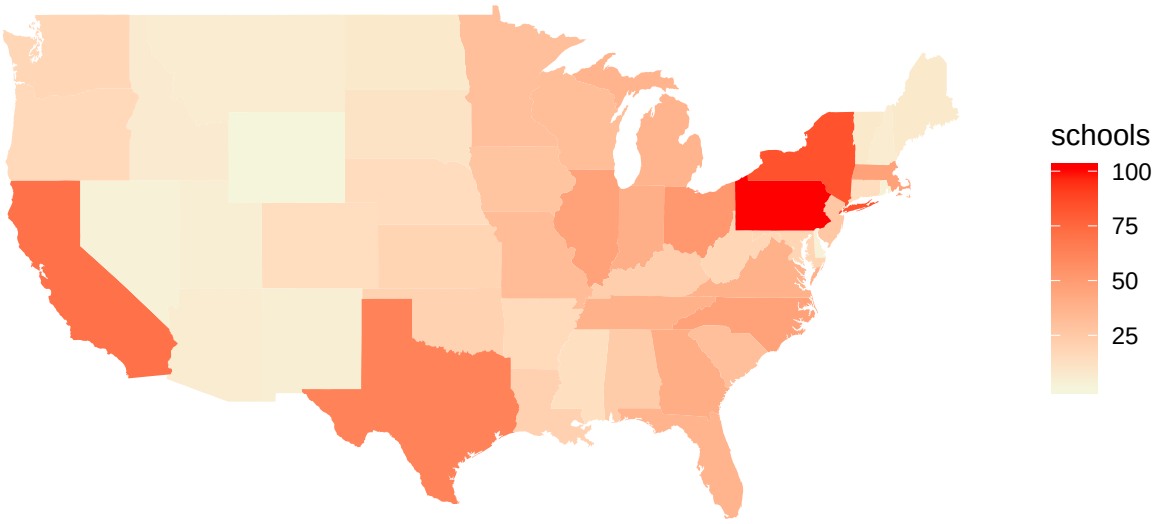
Then we can adjust the fill aesthetic based upon the schools variable.

```
ggplot(mapdata) +
  geom_polygon(aes(x=long,y=lat,group=group, fill=schools)) +
  coord_map() +
  theme(plot.background=element_blank(), panel.background = element_blank(), axis.title=element_blank()
```



And, finally, let's use a more traditional color scale.

```
ggplot(mapdata) +
  geom_polygon(aes(x=long,y=lat,group=group, fill=schools)) +
  coord_map() +
  theme(plot.background=element_blank(), panel.background = element_blank(), axis.title=element_blank()
  scale_fill_gradient(low = "beige", high = "red")
```


schools
100
75
50
25