Portfolio Project 2 Writeup

https://rpubs.com/brattonm/1269018

Molly Bratton

```
knitr::opts_chunk$set(warning = FALSE, message = FALSE, echo = TRUE)
library(tidyverse)
library(ggplot2)
library(dplyr)
library(maps)
library(mapproj)
library(patchwork)
```

Setting up to explore which parts of the U.S. struggle with weather prediction and determine possible reasons why

```
cities <- read_csv("data/forecast_cities.csv")
outlook <- read_csv("data/outlook_meanings.csv")
weather <- read_csv("data/weather_forecasts.csv")
```

Your goal is to learn which areas of the U.S. struggle with weather prediction and explore possible reasons why. Specifically, you will focus on the error in high and low temperature forecasting, and may wish to also consider precipitation and outlook.

- observed - forecast in weather data to find the error
- sort by largest
- could use long lat to create a map colored by the error

```
#need to use distinct (city, state) because some cities have the same name
```

```
#join the datasets together using full join to keep all info
cities <- cities %>%
  full_join(weather)

#load map data of the U.S.
states <- map_data("state")
```

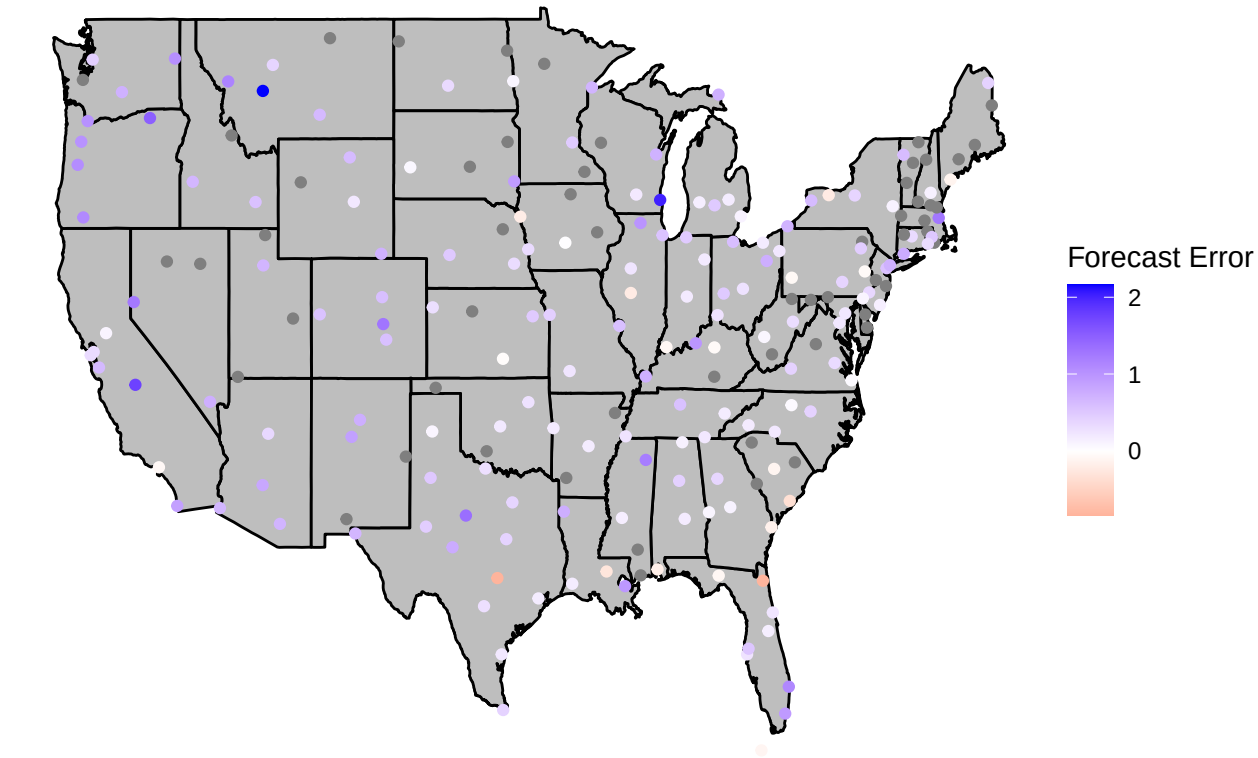
```
#group by city and then create a mean error for each city as well as keeping other variables using medi
cities_smaller <- cities %>%
  filter(lon > -130 & lat > 24) %>% #remove points from Alaska and Hawaii to have a simplier map
  group_by(city, state) %>%
  summarize(
    error_mean = mean(observed_temp - forecast_temp, na.rm = TRUE),
    lat = median(lat),
    lon = median(lon),
    elevation = median(elevation),
    distance_to_coast = median(distance_to_coast),
    wind = median(wind),
```

```
    percip = median(avg_annual_precip)
  )
```

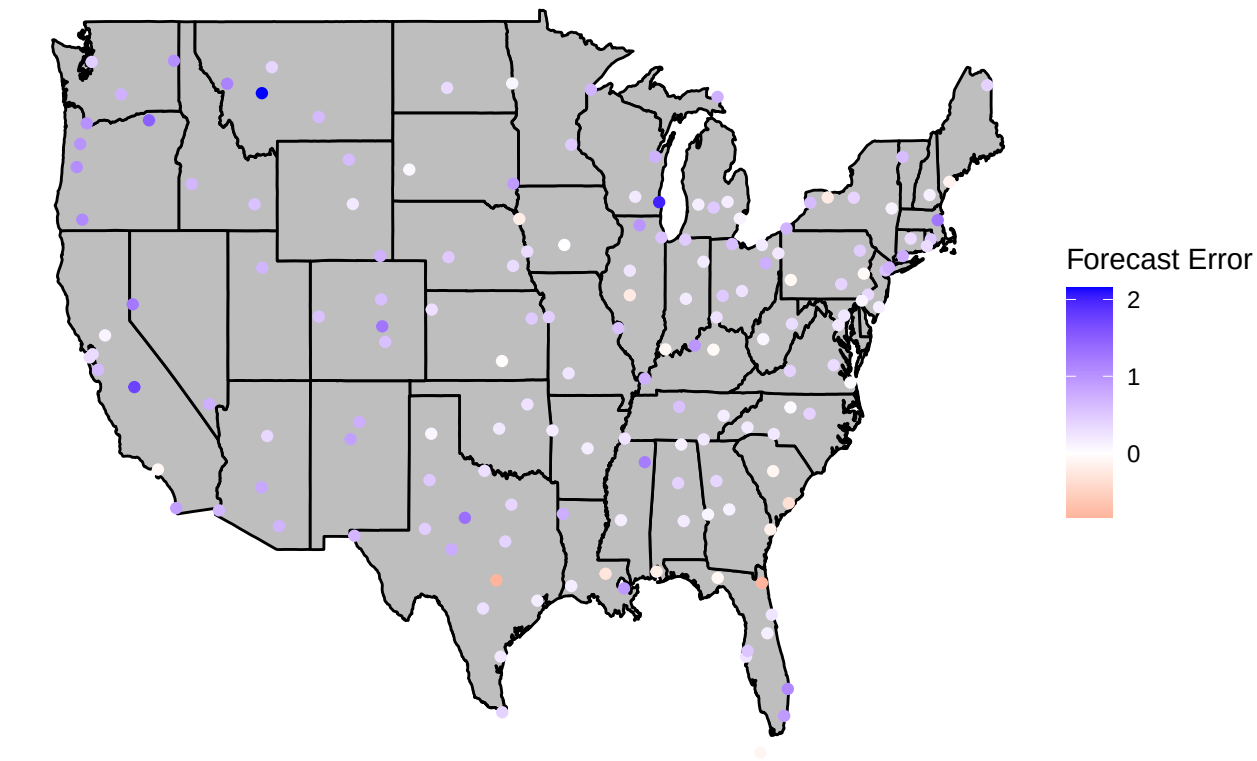
Create maps to see if there are geographical patterns

I explored data from the National Weather Service that details weather forecasts and observations in U.S. cities, as well as data about the cities' locations. I started by combining these data sets to be able to have rows that included both the weather info and the location info so that the data could be made into a graph. I then decided to group the data by city because I found that the overall data was too overwhelming, and hard to interpret because there were numerous points for each city. I then took the mean error by subtracting the expected temperature from the observed temperature, so that I would have the mean error for each city to look for any geographical/spatial patterns. I chose to use the void theme in my maps because I did not think the latitude and longitude lines on the graph were necessary, and made the graphs overly confusing. I also chose to use a color palette with white as the middle tone, because when I tried to just use a high and low color, it was difficult to tell which values were close to 0 because there were also negative values.

```
#create the map with points for error in each city
states %>%
  ggplot(aes(x=long, y=lat, group=group)) +
  geom_polygon(color="black", fill="gray") +
  geom_point(data = cities_smaller, aes(x = lon, y = lat, group = FALSE, color = error_mean)) +
  scale_color_gradient2(low = "red", mid = "white", high = "blue") +
  labs(
    title = "Mean Forecast Error in U.S. Cities",
    color = "Forecast Error"
  ) +
    theme_void()
```



```r
#make the same map but without the NA values
states %>%
  ggplot(aes(x=long, y=lat, group=group)) +
  geom_polygon(color="black", fill="gray") +
  geom_point(data = na.omit(cities_smaller), aes(x = lon, y = lat, group = FALSE, color = error_mean)) +
  scale_color_gradient2(low = "red", mid = "white", high = "blue") +
  labs(
    title = "Mean Forecast Error in U.S. Cities Without Null Values",
    color = "Forecast Error"
  ) +
    theme_void()
```

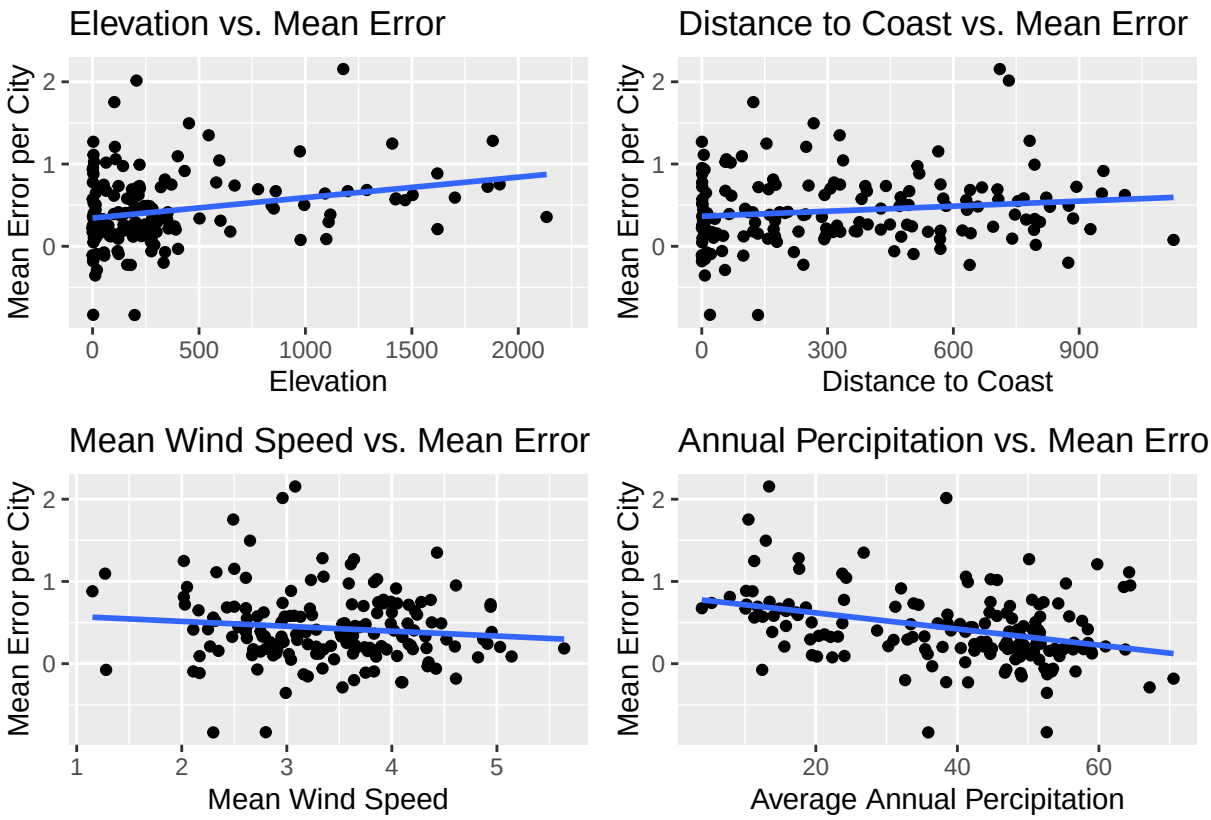


After plotting the mean city forecast errors, I found a slight spatial trends that the West coast seems to have higher errors, and trouble with forecasting. Although I saw this slight trend within my first map, I wanted to explore some of the other variables in the data set, such as elevation, distance to coast, wind speed, and annual precipitation.

Create scatterplots and linear regression models

```r
#create several scatterplots using variables from each city vs. error mean to look for any clear trends
p1 <- ggplot(cities_smaller, aes(x = elevation, y = error_mean)) +
  geom_point() +
  geom_smooth(method=lm, se=FALSE) +
  labs(x = "Elevation", y = "Mean Error per City", title = "Elevation vs. Mean Error")

p2 <- ggplot(cities_smaller, aes(x = distance_to_coast, y = error_mean)) +
  geom_point() +
  geom_smooth(method=lm, se=FALSE) +
```

```
  labs(x = "Distance to Coast", y = "Mean Error per City", title = "Distance to Coast vs. Mean Error")

p3 <- ggplot(cities_smaller, aes(x = wind, y = error_mean)) +
  geom_point() +
  geom_smooth(method=lm, se=FALSE) +
  labs(x = "Mean Wind Speed", y = "Mean Error per City", title = "Mean Wind Speed vs. Mean Error")

p4 <- ggplot(cities_smaller, aes(x = percip, y = error_mean)) +
  geom_point() +
  geom_smooth(method=lm, se=FALSE) +
  labs(x = "Average Annual Percipitation", y = "Mean Error per City", title = "Annual Percipitation vs.

(p1 + p2) / (p3 + p4)
```




```
#make regression models to check for significance
elevation_model <- lm(error_mean ~ elevation, data = cities_smaller)
summary(elevation_model)
```

```
##
## Call:
## lm(formula = error_mean ~ elevation, data = cities_smaller)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.22783 -0.23609 -0.08838  0.20691  1.62017
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 3.430e-01  4.251e-02   8.070 1.62e-13 ***
## elevation   2.488e-04  7.196e-05   3.457 0.000701 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4282 on 159 degrees of freedom
##   (58 observations deleted due to missingness)
## Multiple R-squared:  0.0699, Adjusted R-squared:  0.06405
## F-statistic: 11.95 on 1 and 159 DF,  p-value: 0.0007011
```

```
coast_model <- lm(error_mean ~ distance_to_coast, data = cities_smaller)
summary(coast_model)
```

```
##
## Call:
## lm(formula = error_mean ~ distance_to_coast, data = cities_smaller)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.22791 -0.25992 -0.03778  0.21447  1.64467
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.3648001  0.0518661   7.034 5.65e-11 ***
## distance_to_coast 0.0002044  0.0001167   1.751   0.0819 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4398 on 159 degrees of freedom
##   (58 observations deleted due to missingness)
## Multiple R-squared:  0.01891,    Adjusted R-squared:  0.01274
## F-statistic: 3.065 on 1 and 159 DF,  p-value: 0.08193
```

```
wind_model <- lm(error_mean ~ wind, data = cities_smaller)
summary(wind_model)
```

```
##
## Call:
## lm(formula = error_mean ~ wind, data = cities_smaller)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.33240 -0.25599 -0.06512  0.21525  1.70427
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.63303    0.14697   4.307 2.88e-05 ***
## wind        -0.05927    0.04217  -1.405    0.162
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4412 on 159 degrees of freedom
##   (58 observations deleted due to missingness)
## Multiple R-squared:  0.01227,    Adjusted R-squared:  0.006057
## F-statistic: 1.975 on 1 and 159 DF,  p-value: 0.1619
```

```r
percip_model <- lm(error_mean ~ percip, data = cities_smaller)
summary(percip_model)
```

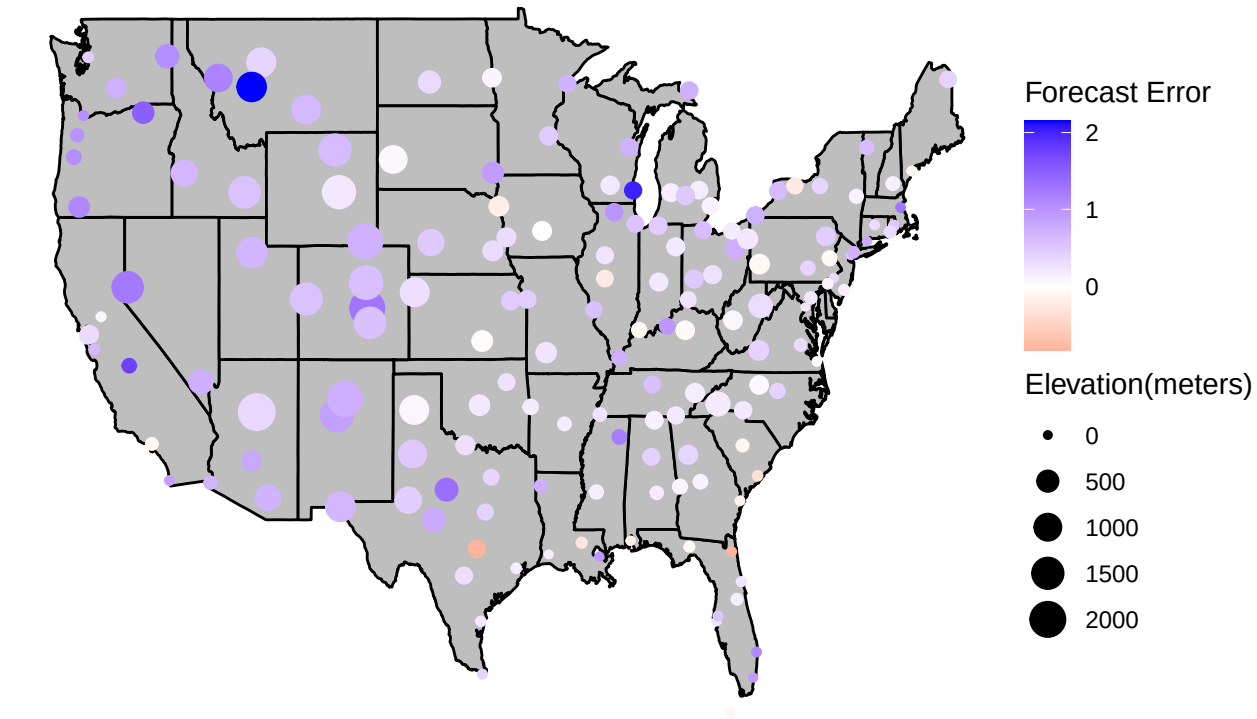
```
##
## Call:
## lm(formula = error_mean ~ percip, data = cities_smaller)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.29793 -0.22734 -0.05072  0.17469  1.57722
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.813714   0.086086   9.452  < 2e-16 ***
## percip      -0.009792   0.002045  -4.789 3.81e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4151 on 159 degrees of freedom
##   (58 observations deleted due to missingness)
## Multiple R-squared:  0.1261, Adjusted R-squared:  0.1206
## F-statistic: 22.94 on 1 and 159 DF,  p-value: 3.807e-06
```

The new concept I chose to use was linear regression models because I felt a bit overwhelmed with trying to find a reason why there were forecasting errors, so I thought that if I looked at linear regression models for several variables, I would be able to see which patterns had a significant relationship with the mean error for the U.S. cities in the data set. When we look at the summary tables for these linear regression models, we can see that the p-values for the elevation model and the precipitation models are statistically significant, while the wind and distance to coast coefficients are not statistically significant. Regarding the elevation model, for a 1 meter increase in elevation, the mean error will increase by an estimated 2.461e-04. Although this is a very small change, there are areas in the U.S. that have elevations in the 1000s and 2000s, meaning that these areas would have a much larger mean error within this regression model. Additionally, for every 1 inch increase in average annual precipitation, there is an estimated 0.009689 decrease in mean error.

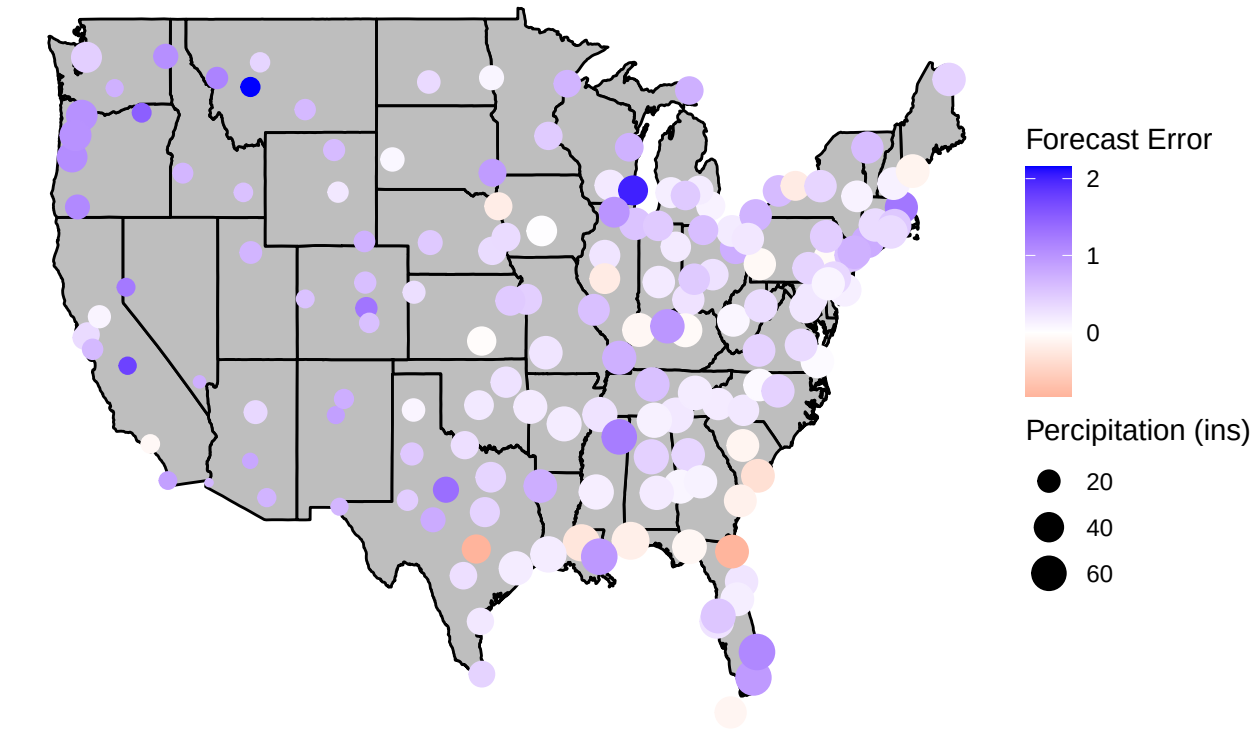
Remake maps using significant variables

I then recreated the maps using significant variables as the size of the points.

```r
# I chose to graph the two significant trends on maps using the size variable
states %>%
  ggplot(aes(x=long, y=lat, group=group)) +
  geom_polygon(color="black", fill="gray") +
  geom_point(data = na.omit(cities_smaller), aes(x = lon, y = lat, group = FALSE, color = error_mean, si
  scale_color_gradient2(low = "red", mid = "white", high = "blue") +
  labs(
    title = "Mean Forecast Error in U.S. Cities",
    subtitle = "Colored by Mean Error and Sized by Elevation",
    color = "Forecast Error",
    size = "Elevation(meters)"
  ) +
    theme_void()
```




```
states %>%
  ggplot(aes(x=long, y=lat, group=group)) +
  geom_polygon(color="black", fill="gray") +
  geom_point(data = na.omit(cities_smaller), aes(x = lon, y = lat, group = FALSE, color = error_mean, si
  scale_color_gradient2(low = "red", mid = "white", high = "blue") +
  labs(
    title = "Mean Forecast Error in U.S. Cities",
    subtitle = "Colored by Mean Error and Sized by Average Annual Percipitation",
    color = "Forecast Error",
    size = "Percipitation (ins)"
  ) +
    theme_void()
```



Create Interactive Graph

I also chose to implement another new concept with an interactive map. I was having some trouble seeing the how the different variables in the data interacted with each other in each city, so I thought that having an interactive map that the user can hover over each city to see the mean forecast error, as well as the two significant variables from from exploration before (elevation and precipitation), and the user can also zoom into certain areas of the U.S. to get more specific. Overall, I found that areas in the U.S. that have lower elevations tend to have greater errors within weather forecasting, and areas in the U.S. that have higher annual precipitation tend to have smaller forecasting errors, and there seems to be a slight pattern of the Western United States struggling more with forecasting.

```
#Have to look at the interactive plot on the website, it will not work in the PDF
library(plotly)

cities_interactive <- cities_smaller %>%
  mutate(mytext = paste(
    "City: ", city, "\n",
    "Error: ", error_mean, "\n",
    "Elevation: ", elevation, "\n",
    "Precipitation: ", percip,
    sep = ""
  ))

graph <- states %>%
  ggplot(aes(x=long, y=lat, group=group)) +
  geom_polygon(color="black", fill="gray") +
  geom_point(data = na.omit(cities_interactive), aes(x = lon, y = lat, group = FALSE, color = error_mean
```

```
  scale_color_gradient2(low = "red", mid = "white", high = "blue") +
  labs(
    title = "Interactive Map of Mean Forecast Error in U.S. Cities",
    color = "Forecast Error"
  ) +
    theme_void()

graph <- ggplotly(graph, tooltip = "text")
graph
```