Election Forecasting Revisited

```r
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.4.2
```

```
## -- Attaching packages ---------------------------------------------------------------------------------
```

```
## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  2.0.1     v dplyr   0.7.8
## v tidyr   0.8.0     v stringr 1.3.1
## v readr   1.1.1     v forcats 0.3.0
```

```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

```
## Warning: package 'tibble' was built under R version 3.4.4
```

```
## Warning: package 'tidyr' was built under R version 3.4.3
```

```
## Warning: package 'purrr' was built under R version 3.4.4
```

```
## Warning: package 'dplyr' was built under R version 3.4.4
```

```
## Warning: package 'forcats' was built under R version 3.4.3
```

```
## -- Conflicts ------------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(ggmap)
```

```
## Google Maps API Terms of Service: https://cloud.google.com/maps-platform/terms/.
```

```
## Please cite ggmap if you use it: see citation("ggmap") for details.
```

```r
library(maps)
```

```
## Warning: package 'maps' was built under R version 3.4.4
```

```
##
## Attaching package: 'maps'
```

```
## The following object is masked from 'package:purrr':
##
##     map
```

```r
library(caTools)
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 3.4.3
```

In the recitation from Unit 3, we used logistic regression on polling data in order to construct US presidential election predictions. We separated our data into a training set, containing data from 2004 and 2008 polls, and a test set, containing the data from 2012 polls. We then proceeded to develop a logistic regression model to forecast the 2012 US presidential election.

In this homework problem, we'll revisit our logistic regression model from Unit 3, and learn how to plot the output on a map of the United States. Unlike what we did in the Crime lecture, this time we'll be plotting predictions rather than data!

First, load the ggplot2, maps, and ggmap packages using the library function. All three packages should be installed on your computer from lecture, but if not, you may need to install them too using the install.packages function.

Then, load the US map and save it to the variable statesMap, like we did during the Crime lecture:

```
states_map = map_data("state")
```

The maps package contains other built-in maps, including a US county map, a world map, and maps for France and Italy.

1.1) Drawing a Map of the US

If you look at the structure of the statesMap data frame using the str function, you should see that there are 6 variables. One of the variables, group, defines the different shapes or polygons on the map. Sometimes a state may have multiple groups, for example, if it includes islands. **How many different groups are there?**

The variable "order" defines the order to connect the points within each group, and the variable "region" gives the name of the state.

Submit

```
glimpse(states_map)
```

```
## Observations: 15,537
## Variables: 6
## $ long      <dbl> -87.46201, -87.48493, -87.52503, -87.53076, -87.5708...
## $ lat       <dbl> 30.38968, 30.37249, 30.37249, 30.33239, 30.32665, 30...
## $ group     <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ order     <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1...
## $ region    <chr> "alabama", "alabama", "alabama", "alabama", "alabama...
## $ subregion <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
```

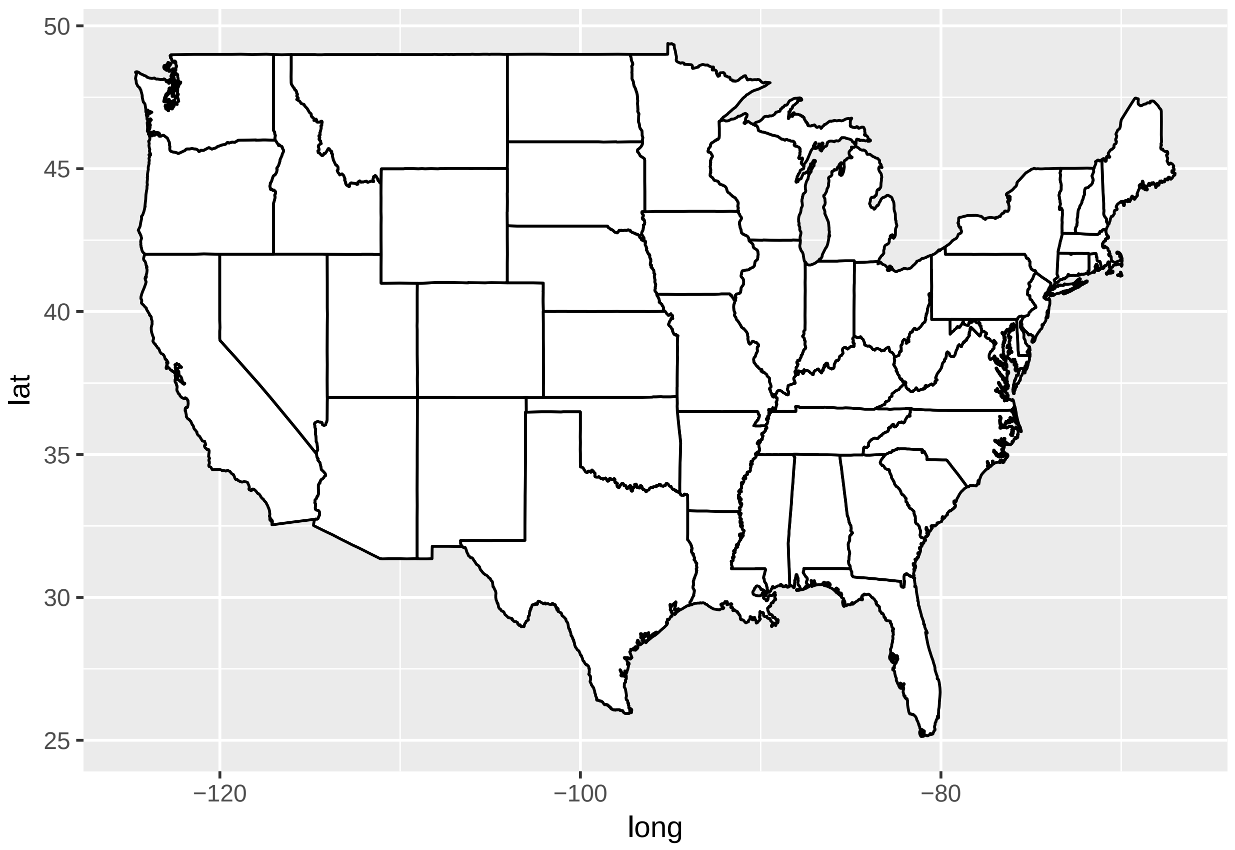
```
length(table(states_map$group))
```

```
## [1] 63
```

1.2) Drawing a Map of the US

You can draw a map of the United States by typing the following in your R console:

```
ggplot(states_map, aes(x = long, y = lat, group = group)) +
  geom_polygon(fill = "white", color = "black")
```



We specified two colors in geom_polygon – fill and color. Which one defined the color of the outline of the states?

2.1) Coloring the States by Predictions

Now, let's color the map of the US according to our 2012 US presidential election predictions from the Unit 3 Recitation. We'll rebuild the model here, using the dataset PollingImputed.csv. Be sure to use this file so that you don't have to redo the imputation to fill in the missing values, like we did in the Unit 3 Recitation.

Load the data using the read.csv function, and call it "polling". Then split the data using the subset function into a training set called "Train" that has observations from 2004 and 2008, and a testing set called "Test" that has observations from 2012.

```r
polling = read.csv('PollingImputed.csv')
```

```r
glimpse(polling)
```

```
## Observations: 145
## Variables: 7
## $ State      <fct> Alabama, Alabama, Alaska, Alaska, Arizona, Arizona,...
## $ Year       <int> 2004, 2008, 2004, 2008, 2004, 2008, 2012, 2004, 200...
## $ Rasmussen  <int> 11, 21, 19, 16, 5, 5, 8, 7, 10, 13, -11, -27, -12, ...
## $ SurveyUSA  <int> 18, 25, 21, 18, 15, 3, 5, 5, 7, 21, -11, -24, -14, ...
## $ DiffCount  <int> 5, 5, 1, 6, 8, 9, 4, 8, 5, 2, -8, -5, -6, 9, -15, -...
## $ PropR      <dbl> 1.0000000, 1.0000000, 1.0000000, 1.0000000, 1.00000...
## $ Republican <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, ...
```

```r
train = filter(polling, Year <= 2008)
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

```
test = filter(polling, Year > 2008)
```

Note that we only have 45 states in our testing set, since we are missing observations for Alaska, Delaware, Alabama, Wyoming, and Vermont, so these states will not appear colored in our map.

Then, create a logistic regression model and make predictions on the test set using the following commands:

```
mod2 = glm(Republican ~ SurveyUSA + DiffCount,
           data = train,
           family = 'binomial')
```

```
test_pred = predict(mod2,
                    newdata = test,
                    type = 'response')
```

TestPrediction gives the predicted probabilities for each state, but let's also create a vector of Republican/Democrat predictions by using the following command:

```
test_pred_binary = as.numeric(test_pred > 0.5)
```

Now, put the predictions and state labels in a data.frame so that we can use ggplot:

```
pred_df = data.frame(test_pred, test_pred_binary, test$State)
```

To make sure everything went smoothly, answer the following questions.

For how many states is our binary prediction 1 (for 2012), corresponding to Republican?

```
table(test$Year == 2012, pred_df$test_pred_binary)
```

```
##
##         0  1
##   TRUE 23 22
```

What is the average predicted probability of our model (on the Test set, for 2012)?

```
mean(pred_df$test_pred)
```

```
## [1] 0.4852626
```

Explanation

You can create the data frame predictionDataFrame by running the following lines of R code:

polling = read.csv("PollingImputed.csv")

Train = subset(polling, Year < 2012)

Test = subset(polling, Year == 2012)

mod2 = glm(Republican~SurveyUSA+DiffCount, data=Train, family="binomial")

TestPrediction = predict(mod2, newdata=Test, type="response")

TestPredictionBinary = as.numeric(TestPrediction > 0.5)

predictionDataFrame = data.frame(TestPrediction, TestPredictionBinary, Test$State)

You can answer the two questions with the functions table(TestPredictionBinary) and mean(TestPrediction).

2.2) Coloring the States by Predictions

Now, we need to merge "predictionDataFrame" with the map data "statesMap", like we did in lecture. Before doing so, we need to convert the Test.State variable to lowercase, so that it matches the region variable in statesMap. Do this by typing the following in your R console:

```
pred_df$region = tolower(pred_df$test.State)
```

Now, merge the two data frames using the following command:

```
prediction_map = merge(states_map, pred_df, by = "region")
```

Lastly, we need to make sure the observations are in order so that the map is drawn properly, by typing the following:

```
prediction_map = prediction_map[order(prediction_map$order),]
```

How many observations are there in predictionMap?

```
glimpse(prediction_map)
```

```
## Observations: 15,034
## Variables: 9
## $ region           <chr> "arizona", "arizona", "arizona", "arizona", "...
## $ long             <dbl> -114.6374, -114.6431, -114.6030, -114.5744, -...
## $ lat              <dbl> 35.01918, 35.10512, 35.12231, 35.17961, 35.23...
## $ group            <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, ...
## $ order            <int> 204, 205, 206, 207, 208, 209, 210, 211, 212, ...
## $ subregion        <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ test_pred        <dbl> 0.9739028, 0.9739028, 0.9739028, 0.9739028, 0...
## $ test_pred_binary <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ test.State       <fct> Arizona, Arizona, Arizona, Arizona, Arizona, ...
```

How many observations are there in statesMap?

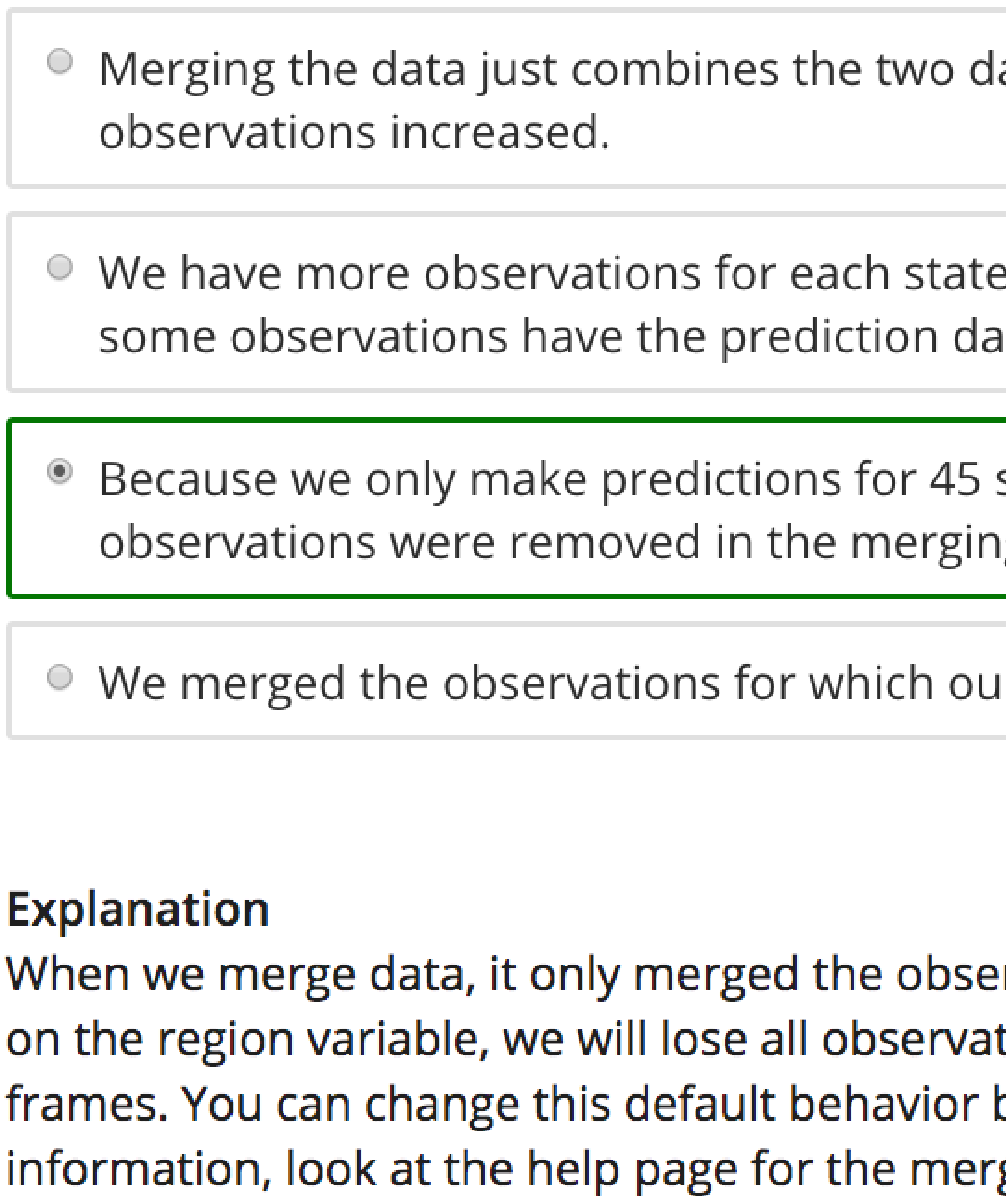
```
glimpse(states_map)
```

```
## Observations: 15,537
## Variables: 6
## $ long      <dbl> -87.46201, -87.48493, -87.52503, -87.53076, -87.5708...
## $ lat       <dbl> 30.38968, 30.37249, 30.37249, 30.33239, 30.32665, 30...
## $ group     <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ order     <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1...
## $ region    <chr> "alabama", "alabama", "alabama", "alabama", "alabama...
## $ subregion <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
```

2.3) Coloring the States by Predictions

When we merged the data in the previous problem, it caused the number of observations to change. Why? Check out the help page for merge by typing ?merge to help you answer this question.

```
include_graphics('2.3.png')
```



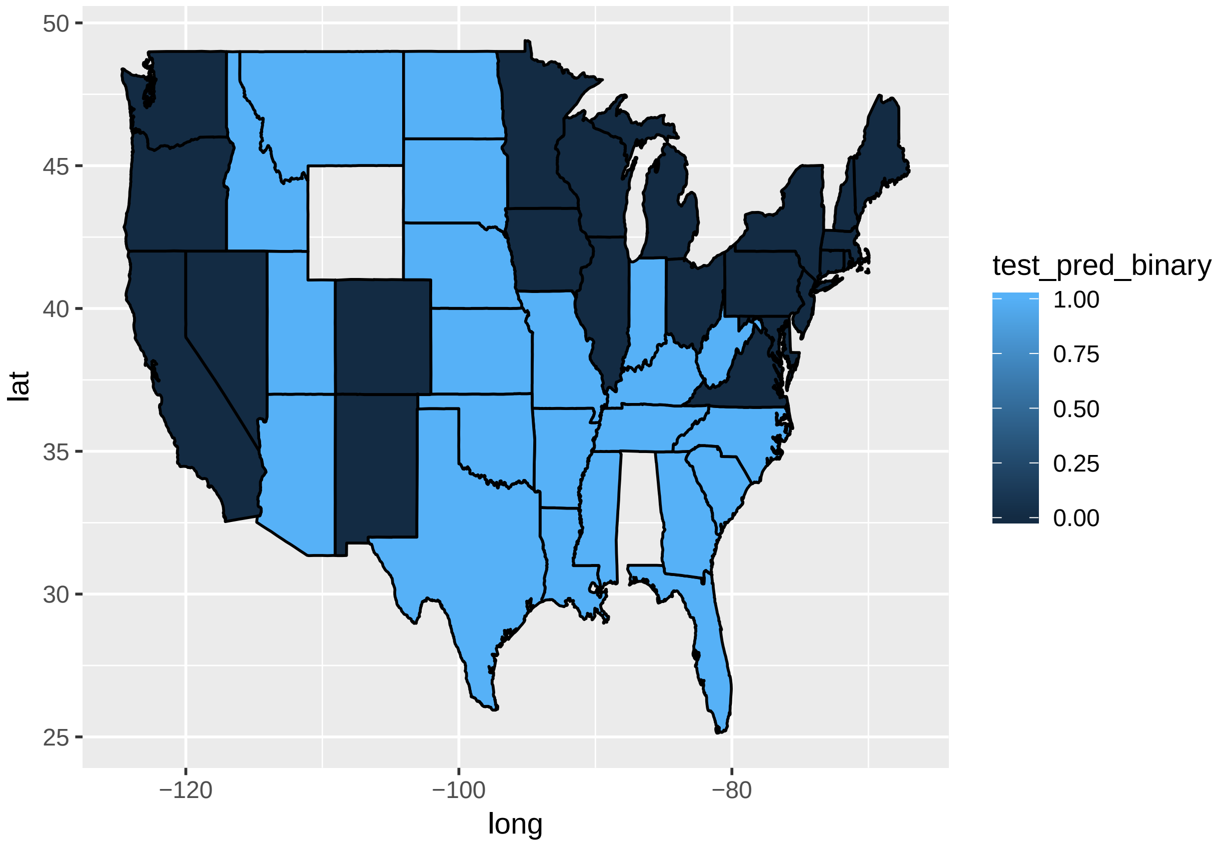
- ○ Merging the data just combines the two da
  observations increased.
- ○ We have more observations for each state
  some observations have the prediction da
- [✓ Because we only make predictions for 45 s
  observations were removed in the mergin]
- ○ We merged the observations for which ou

Explanation
When we merge data, it only merged the obse
on the region variable, we will lose all observat
frames. You can change this default behavior b
information, look at the help page for the merg

2.4) Coloring the States by Predictions

Now we are ready to color the US map with our predictions! You can color the states according to our binary predictions by typing the following in your R console:

```
ggplot(prediction_map, aes(x = long, y = lat, group = group, fill = test_pred_binary)) +     geom_polyg
```



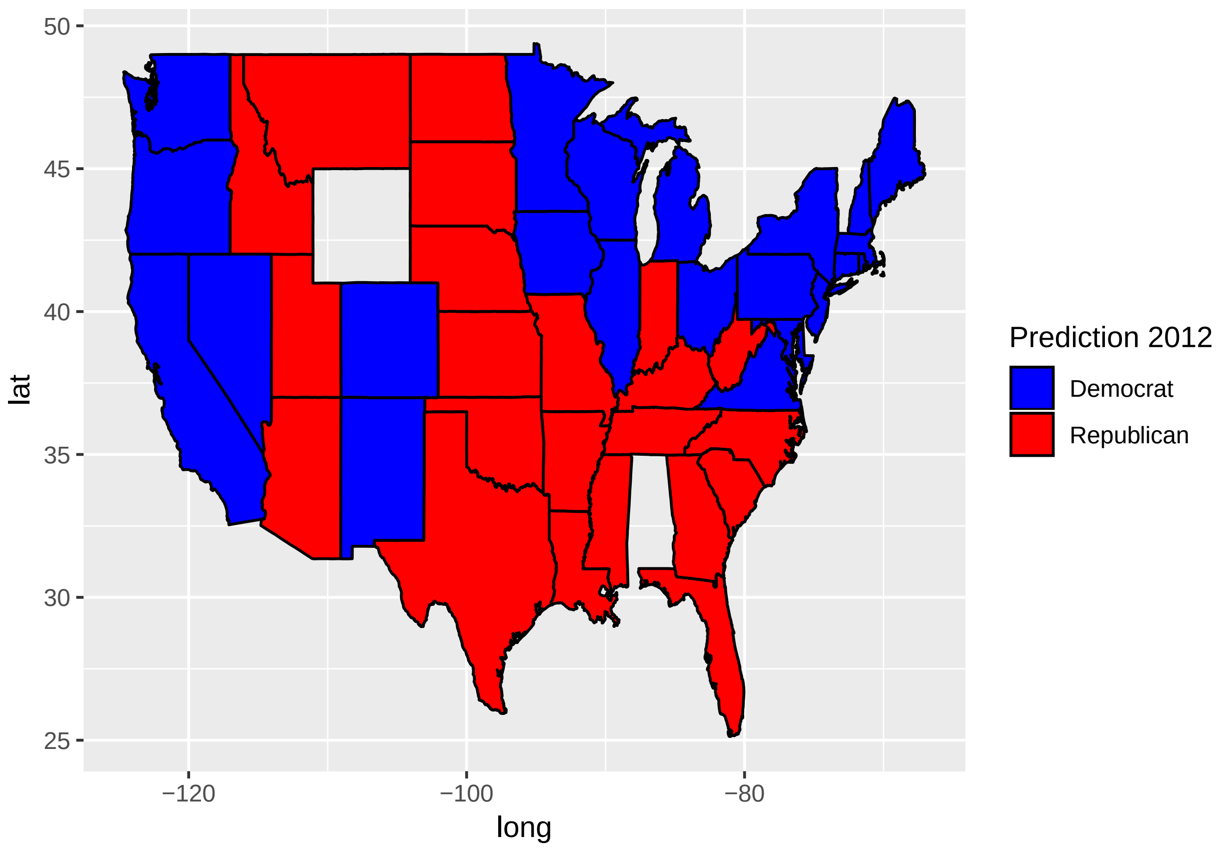
The states appear light blue and dark blue in this map. Which color represents a Republican prediction?

- Light blue

2.5) Coloring the States by Predictions

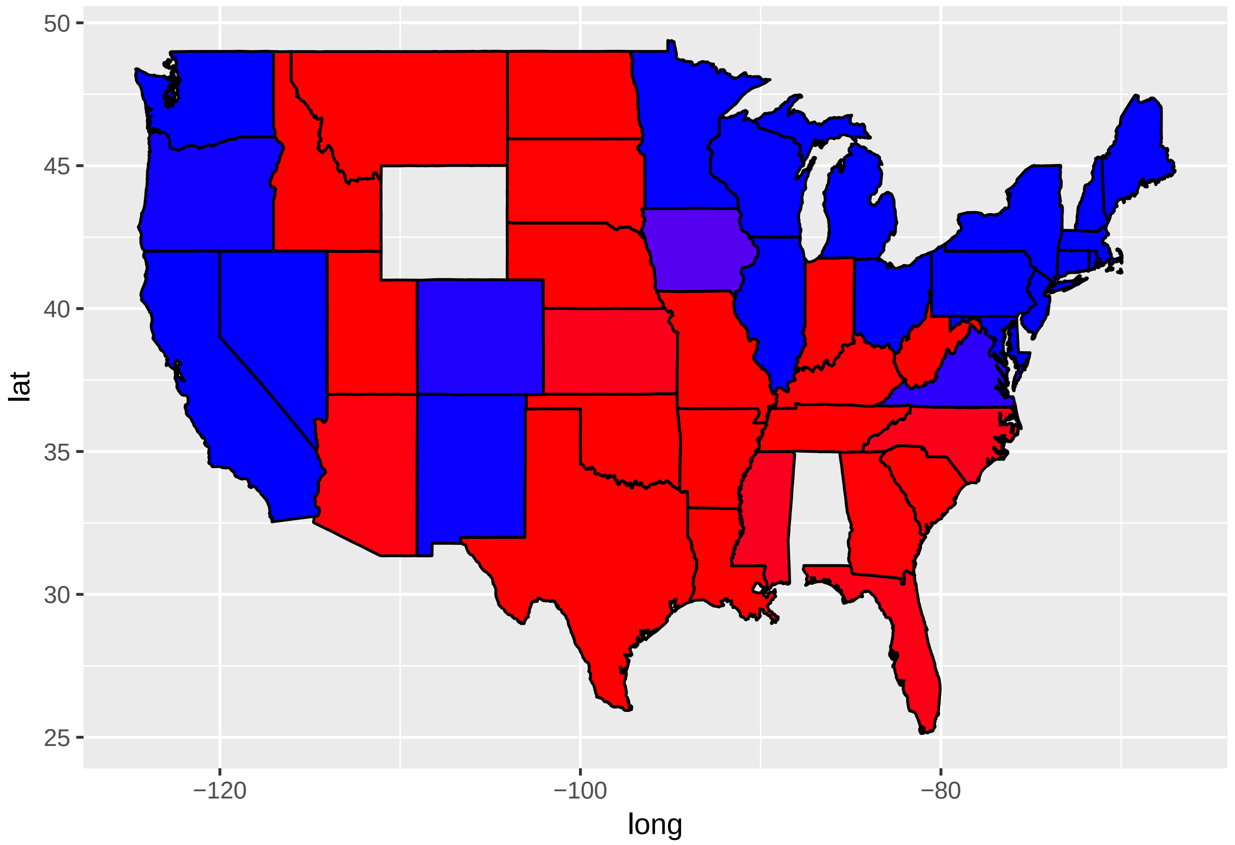
We see that the legend displays a blue gradient for outcomes between 0 and 1. However, when plotting the binary predictions there are only two possible outcomes: 0 or 1. Let's replot the map with discrete outcomes. We can also change the color scheme to blue and red, to match the blue color associated with the Democratic Party in the US and the red color associated with the Republican Party in the US. This can be done with the following command:

```
ggplot(prediction_map, aes(x = long, y = lat, group = group, fill = test_pred_binary)) +    geom_polygon
```



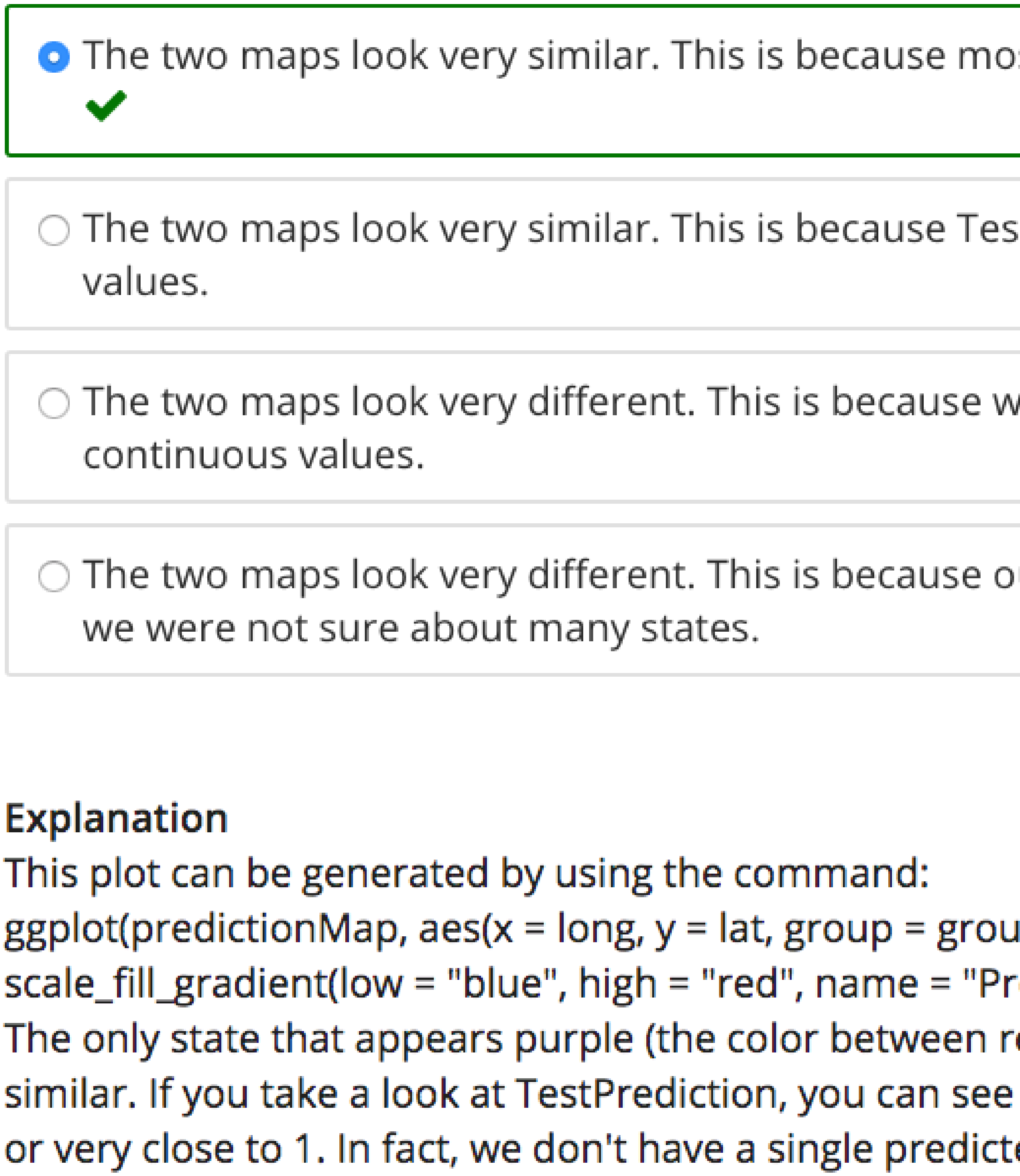
Alternatively, we could plot the probabilities instead of the binary predictions. Change the plot command above to instead color the states by the variable TestPrediction. You should see a gradient of colors ranging from red to blue. Do the colors of the states in the map for TestPrediction look different from the colors of the states in the map with TestPredictionBinary? Why or why not?

```
ggplot(prediction_map, aes(x = long, y = lat, group = group, fill = test_pred)) +   geom_polygon(color =
```



NOTE: If you have a hard time seeing the red/blue gradient, feel free to change the color scheme, by changing the arguments low = “blue” and high = “red” to colors of your choice (to see all of the color options in R, type colors() in your R console). You can even change it to a gray scale, by changing the low and high colors to “gray” and “black”.

```r
include_graphics('2.5.png')
```



[✓ The two maps look very similar. This is because mo]

The two maps look very similar. This is because Tes
values.

The two maps look very different. This is because w
continuous values.

The two maps look very different. This is because o
we were not sure about many states.

Explanation

This plot can be generated by using the command:
ggplot(predictionMap, aes(x = long, y = lat, group = grou
scale_fill_gradient(low = "blue", high = "red", name = "Pr
The only state that appears purple (the color between r
similar. If you take a look at TestPrediction, you can see
or very close to 1. In fact, we don't have a single predict

3.1) Understanding the Predictions

In the 2012 election, the state of Florida ended up being a very close race. It was ultimately won by the Democratic party. **Did we predict this state correctly or incorrectly?** To see the names and locations of the different states, take a look at the World Atlas map here.

- Incorrecly

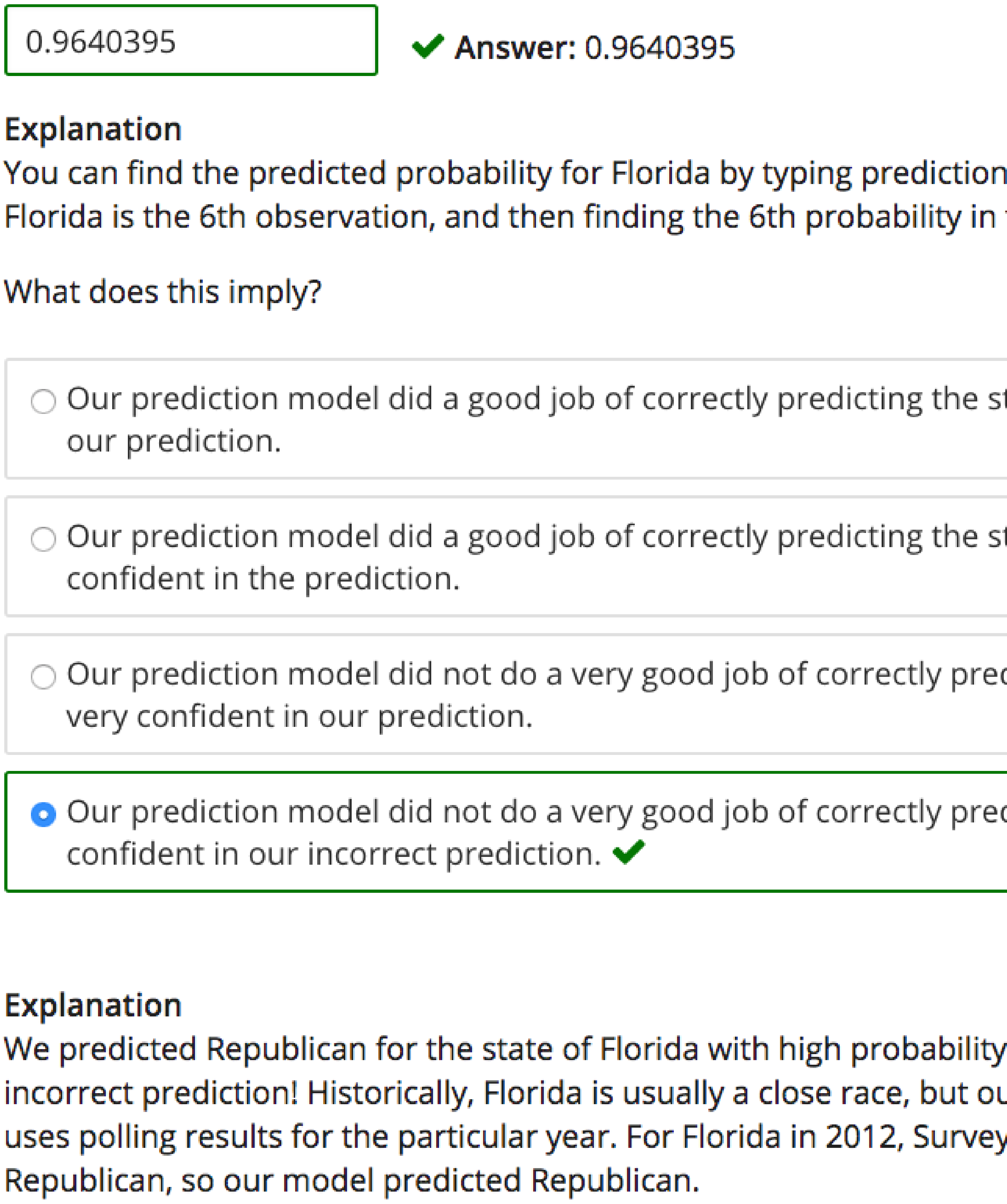
3.2) Understanding the Predictions

What was our predicted probability for the state of Florida?

```
pred_df[pred_df$test.State == 'Florida',]
```

```
##   test_pred test_pred_binary test.State  region
## 6 0.9640395                1    Florida florida
```

```
include_graphics('3.2.png')
```



0.9640395 ✓ **Answer:** 0.9640395

Explanation

You can find the predicted probability for Florida by typing prediction

Florida is the 6th observation, and then finding the 6th probability in

What does this imply?

- ○ Our prediction model did a good job of correctly predicting the st our prediction.
- ○ Our prediction model did a good job of correctly predicting the st confident in the prediction.
- ○ Our prediction model did not do a very good job of correctly pred very confident in our prediction.
- [✓ Our prediction model did not do a very good job of correctly pred confident in our incorrect prediction.] ✓

Explanation

We predicted Republican for the state of Florida with high probability incorrect prediction! Historically, Florida is usually a close race, but ou uses polling results for the particular year. For Florida in 2012, Survey Republican, so our model predicted Republican.

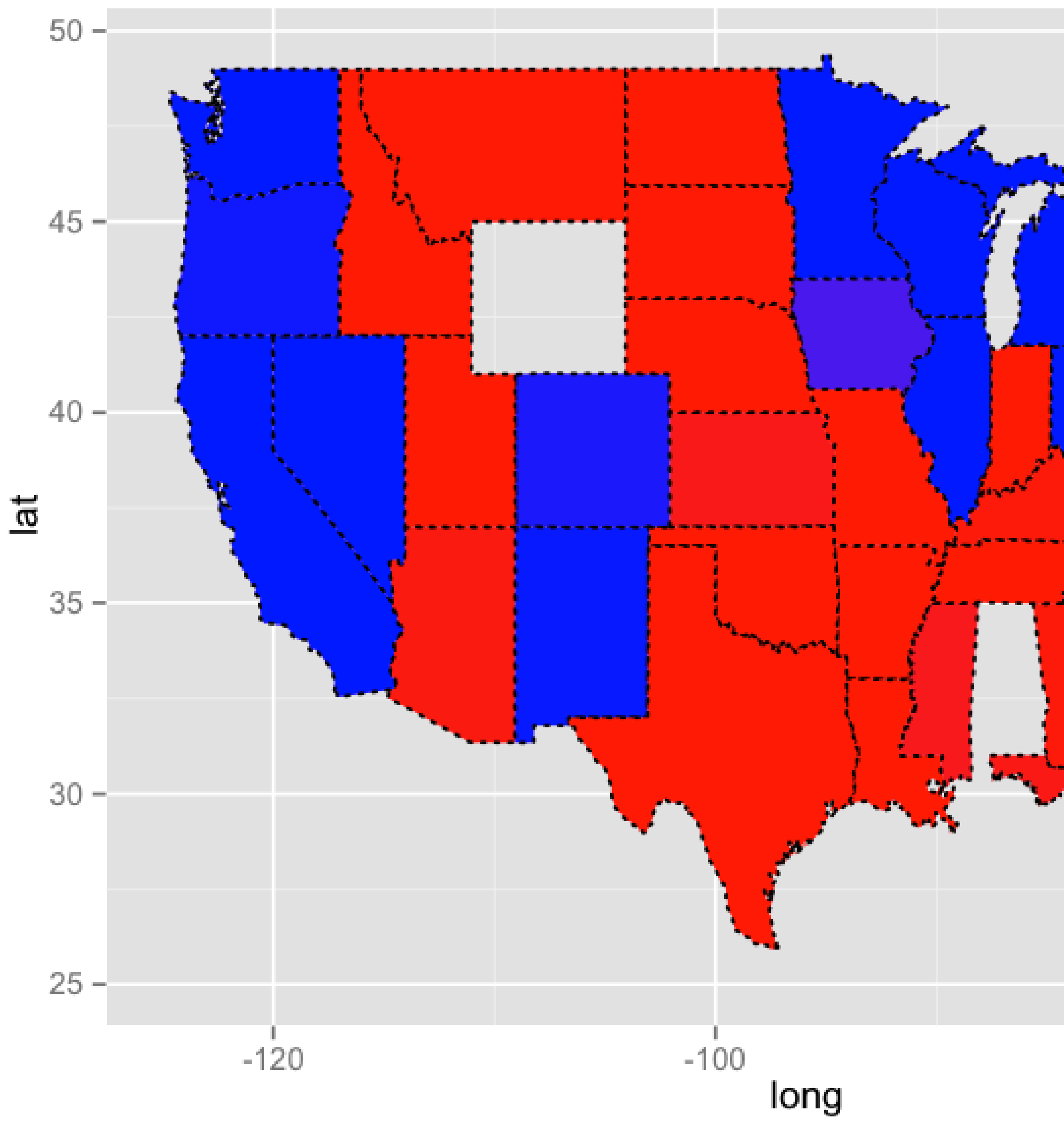
Part 4: Parameter Settings

In this part, we'll explore what the different parameter settings of geom_polygon do. Throughout the problem, use the help page for geom_polygon, which can be accessed by ?geom_polygon. To see more information about a certain parameter, just type a question mark and then the parameter name to get the help page for that parameter. Experiment with different parameter settings to try and replicate the plots!

We'll be asking questions about the following three plots:

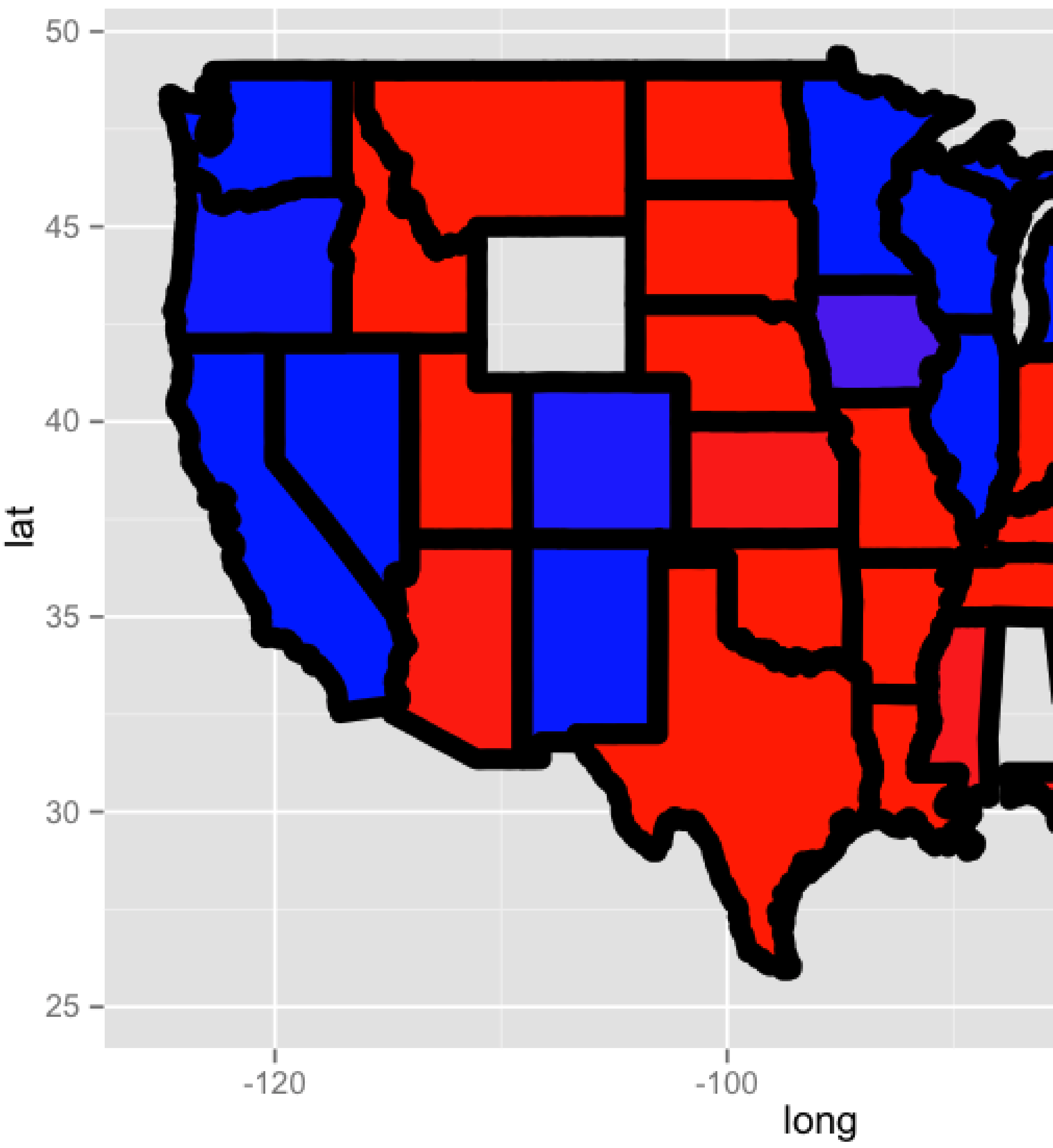
```r
include_graphics('plot1.png')
```

Plot (1)



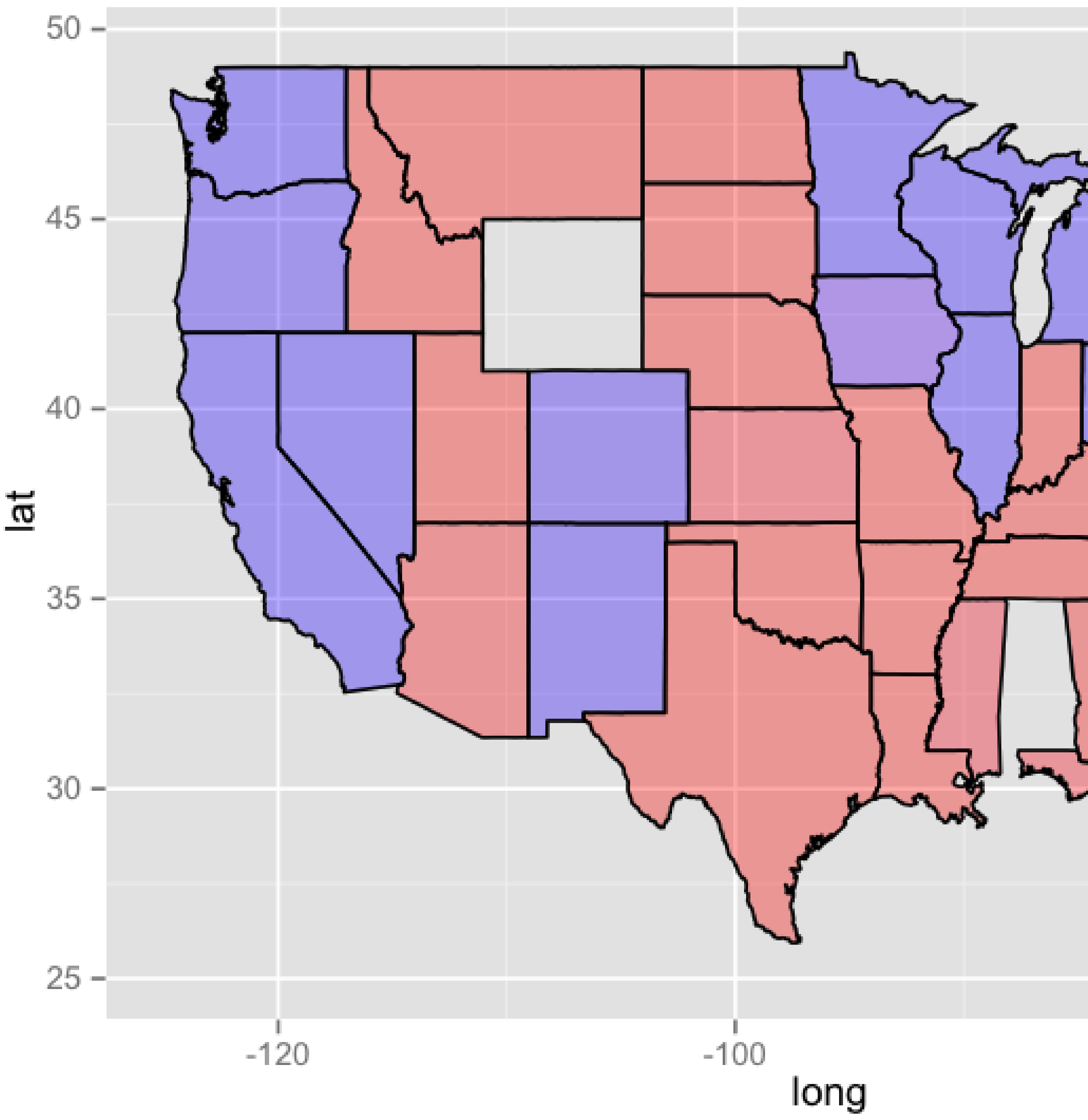
```r
include_graphics('plot2.png')
```

Plot (2)



```r
include_graphics('plot3.png')
```

Plot (3)

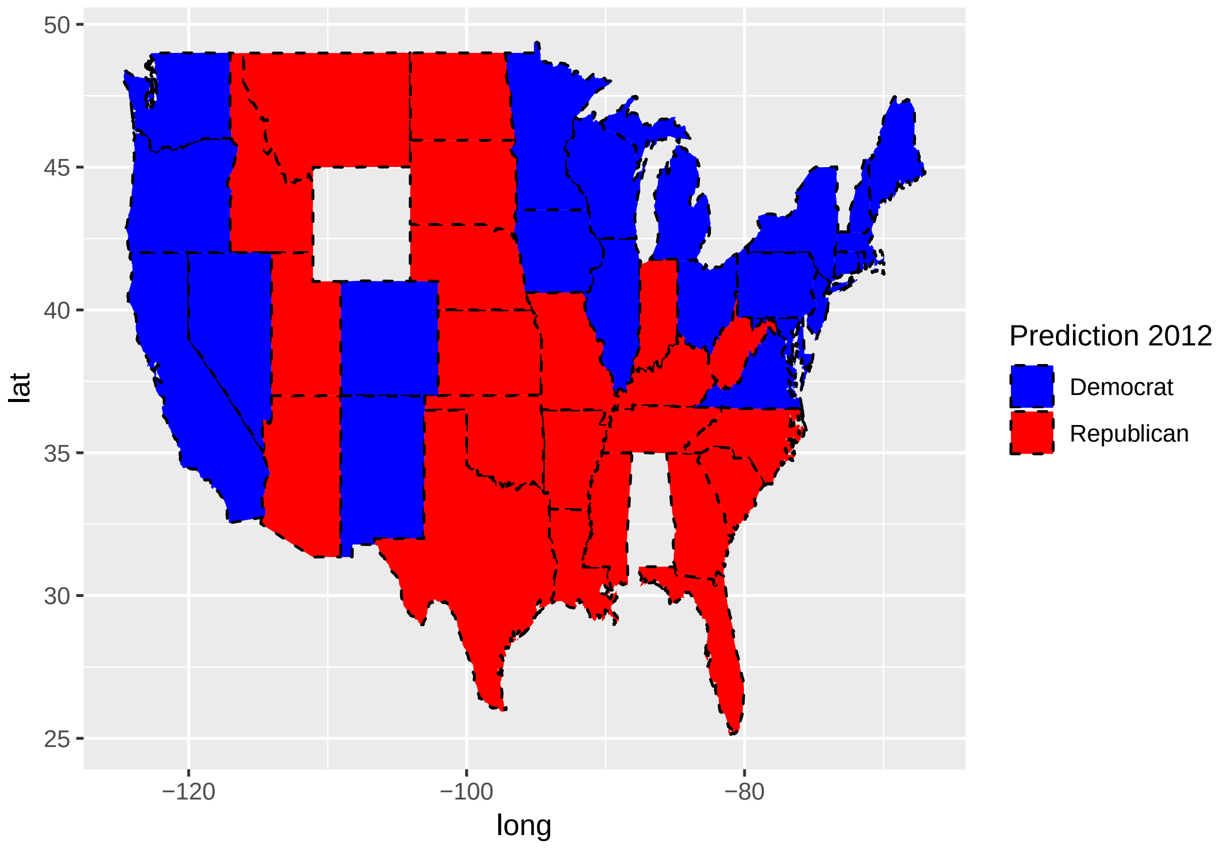


4.1) Parameter settings

Plots (1) and (2) were created by changing different parameters of geom_polygon from their default values.

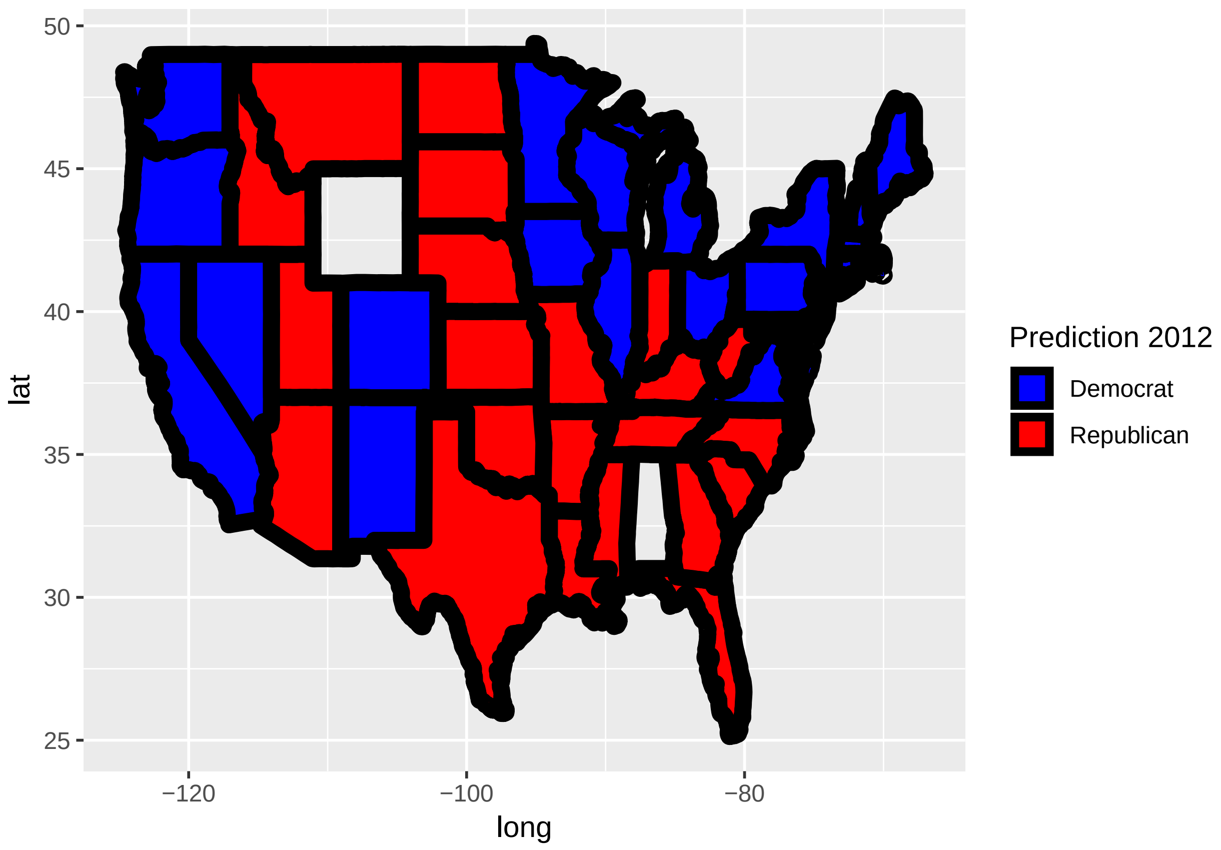
What is the name of the parameter we changed to create plot (1)?

```
ggplot(prediction_map, aes(x = long, y = lat, group = group, fill = test_pred_binary)) +   geom_polygon
```

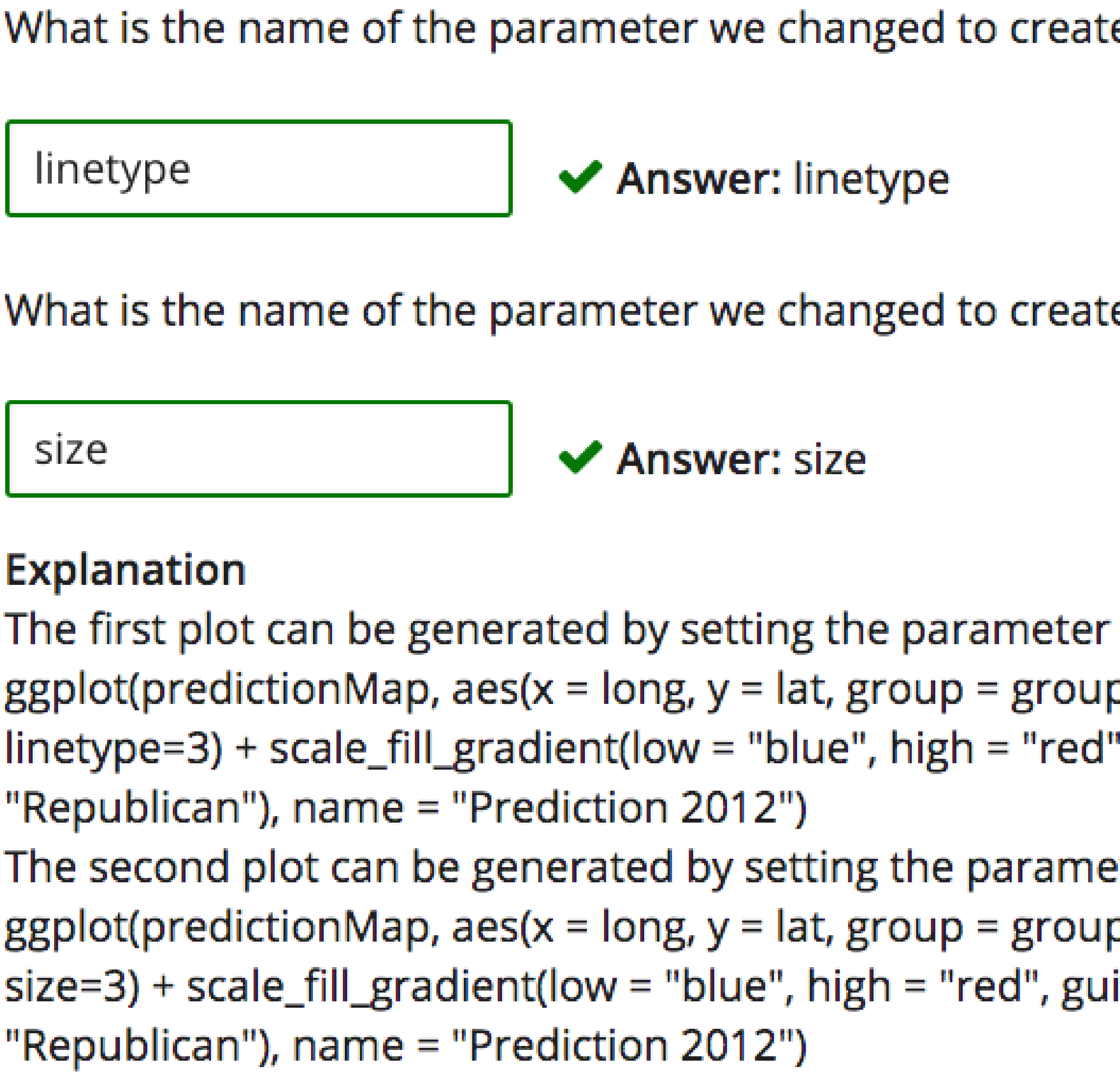


What is the name of the parameter we changed to create plot (2)?

```
ggplot(prediction_map, aes(x = long, y = lat, group = group, fill = test_pred_binary)) +   geom_polygon
```



```
include_graphics('4.1.png')
```


What is the name of the parameter we changed to creat

linetype

✓ **Answer:** linetype

What is the name of the parameter we changed to creat

size

✓ **Answer:** size

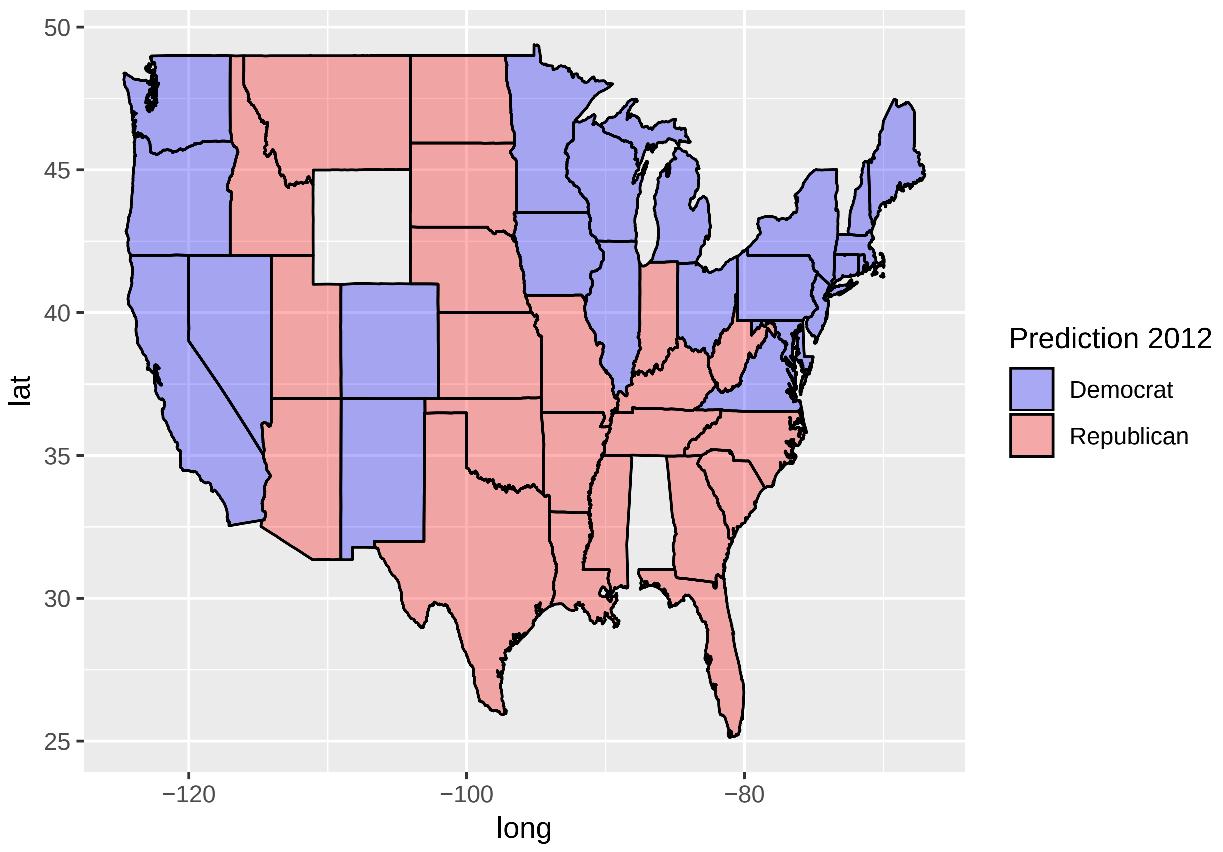
Explanation

The first plot can be generated by setting the parameter
ggplot(predictionMap, aes(x = long, y = lat, group = grou
linetype=3) + scale_fill_gradient(low = "blue", high = "red"
"Republican"), name = "Prediction 2012")
The second plot can be generated by setting the parame
ggplot(predictionMap, aes(x = long, y = lat, group = grou
size=3) + scale_fill_gradient(low = "blue", high = "red", gu
"Republican"), name = "Prediction 2012")

4.2) Parameter settings

Plot (3) was created by changing the value of a different geom_polygon parameter to have value 0.3. Which parameter did we use?

```r
ggplot(prediction_map, aes(x = long, y = lat, group = group, fill = test_pred_binary)) +   geom_polygon
```



50
45
40
35
30
25
lat
−120
−100
−80
long
Prediction 2012
Democrat
Republican