Final Project - Intro to Machine Learning

Amir Voloshin

12/08/2022

Election Data

(Q1)

```
## [1] 32177     5
```

```
## # A tibble: 0 x 5
## # ... with 5 variables: state <chr>, county <chr>, candidate <fct>,
## #   party <fct>, total_votes <dbl>
```

```
##  [1] "Delaware"             "District of Columbia" "Florida"
##  [4] "Georgia"              "Hawaii"               "Idaho"
##  [7] "Illinois"             "Indiana"              "Iowa"
## [10] "Kansas"               "Kentucky"             "Louisiana"
## [13] "Maine"                "Maryland"             "Massachusetts"
## [16] "Michigan"             "Minnesota"            "Mississippi"
## [19] "Missouri"             "Montana"              "Nebraska"
## [22] "Nevada"               "New Hampshire"        "New Jersey"
## [25] "New Mexico"           "New York"             "North Carolina"
## [28] "North Dakota"         "Ohio"                 "Oklahoma"
## [31] "Oregon"               "Pennsylvania"         "Rhode Island"
## [34] "South Carolina"       "South Dakota"         "Tennessee"
## [37] "Texas"                "Utah"                 "Vermont"
## [40] "Virginia"             "Washington"           "West Virginia"
## [43] "Wisconsin"            "Wyoming"              "Alabama"
## [46] "Alaska"               "Arkansas"             "California"
## [49] "Colorado"             "Connecticut"          "Arizona"
```

The election.raw data set has dimensions of 32177 x 5, and there are no missing values. There are indeed 51 unique values in the state column, indicating that the data contains all the states and a federal district.

Census data

(Q2)

```
## [1] 3220   37
```

```
## # A tibble: 1 x 37
##   CountyId State  County  TotalPop   Men Women Hispanic White Black Native Asian
##      <dbl> <chr>  <chr>      <dbl> <dbl> <dbl>    <dbl> <dbl> <dbl>  <dbl> <dbl>
## 1    15005 Hawaii Kalawa~       86    41    45      4.7  20.9     0      0  29.1
## # ... with 26 more variables: Pacific <dbl>, VotingAgeCitizen <dbl>,
## #   Income <dbl>, IncomeErr <dbl>, IncomePerCap <dbl>, IncomePerCapErr <dbl>,
## #   Poverty <dbl>, ChildPoverty <dbl>, Professional <dbl>, Service <dbl>,
## #   Office <dbl>, Construction <dbl>, Production <dbl>, Drive <dbl>,
## #   Carpool <dbl>, Transit <dbl>, Walk <dbl>, OtherTransp <dbl>,
## #   WorkAtHome <dbl>, MeanCommute <dbl>, Employed <dbl>, PrivateWork <dbl>,
## #   PublicWork <dbl>, SelfEmployed <dbl>, FamilyWork <dbl>, ...
```

```
## [1] 3220    2
```

```
## [1] 4633    2
```

The dimensions of the census data set are 3220 x 37, and there is one observation with a missing value. The election.raw has 51 distinct values in the State column, while the census data set has 52 distinct values in the State column. The difference between these two data sets is that the census data set includes Puerto Rico as a State, while the election.raw data set does not. In addition, census has a total of 3220 distinct counties, while election.raw has 4633 distinct counties. This means there is likely some discrepency in the election.raw dataset because there are a total of 3243 counties in the US (including counties of territoriesterritories).

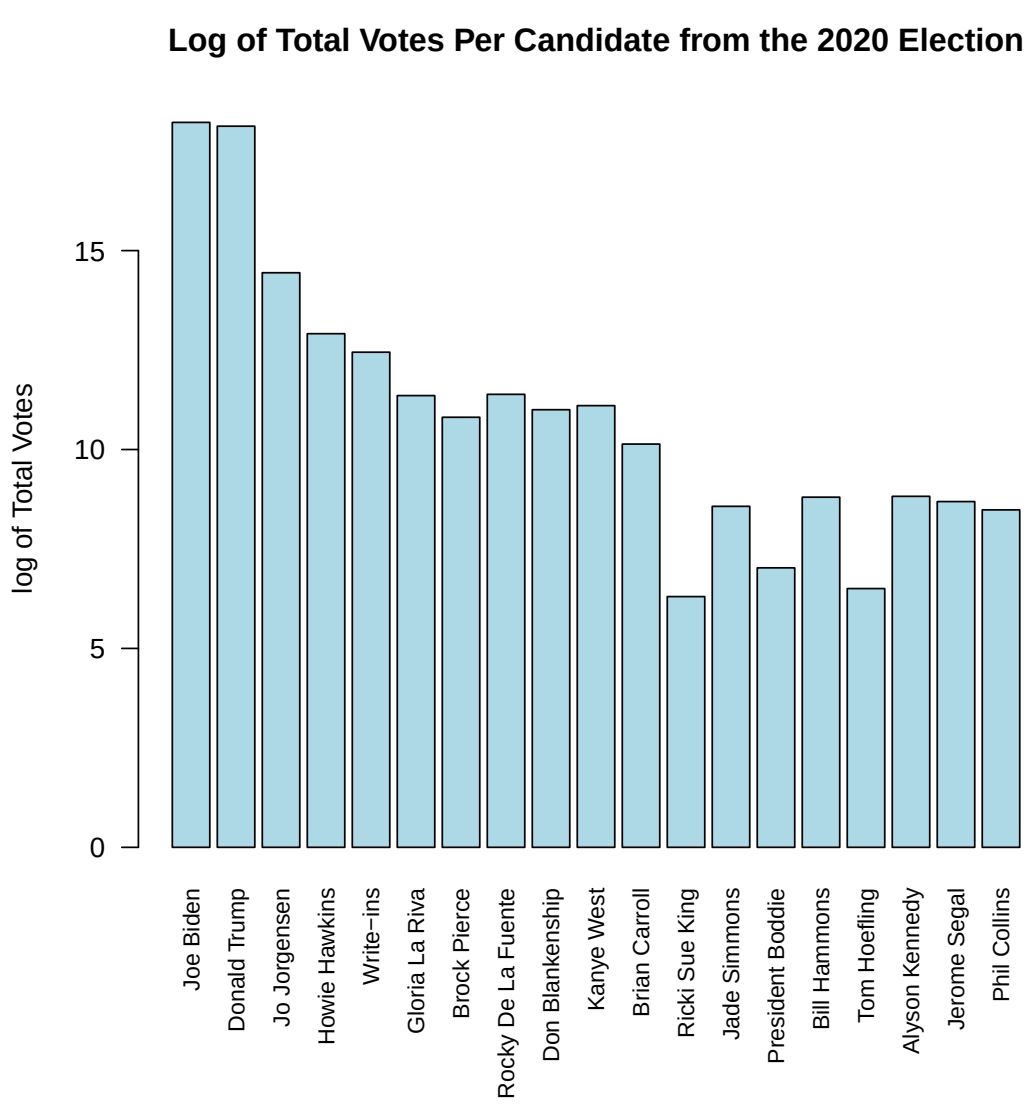
Data Wrangling

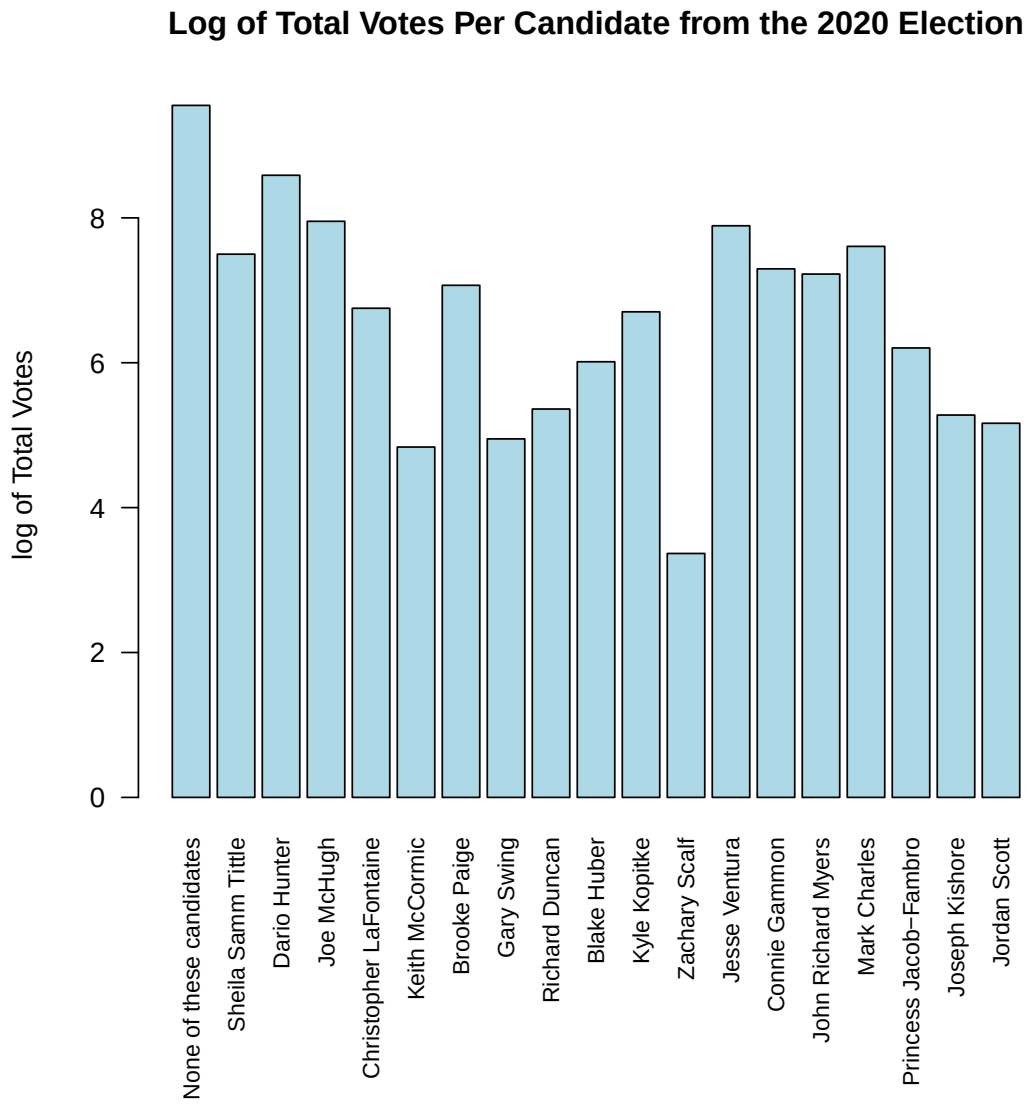
(Q3) See Source Code

(Q4)

All Named Candidates Printed Below:

```
##  [1] Joe Biden               Donald Trump            Jo Jorgensen
##  [4] Howie Hawkins           Write-ins               Gloria La Riva
##  [7] Brock Pierce            Rocky De La Fuente      Don Blankenship
## [10] Kanye West              Brian Carroll           Ricki Sue King
## [13] Jade Simmons            President Boddie        Bill Hammons
## [16] Tom Hoefling            Alyson Kennedy          Jerome Segal
## [19] Phil Collins            None of these candidates Sheila Samm Tittle
## [22] Dario Hunter            Joe McHugh              Christopher LaFontaine
## [25] Keith McCormic          Brooke Paige            Gary Swing
## [28] Richard Duncan          Blake Huber             Kyle Kopitke
## [31] Zachary Scalf           Jesse Ventura           Connie Gammon
## [34] John Richard Myers      Mark Charles            Princess Jacob-Fambro
## [37] Joseph Kishore          Jordan Scott
## 38 Levels: Alyson Kennedy Bill Hammons Blake Huber ... Zachary Scalf
```


Log of Total Votes Per Candidate from the 2020 Election
log of Total Votes
0
5
10
15
Joe Biden
Donald Trump
Jo Jorgensen
Howie Hawkins
Write−ins
Gloria La Riva
Brock Pierce
Rocky De La Fuente
Don Blankenship
Kanye West
Brian Carroll
Ricki Sue King
Jade Simmons
President Boddie
Bill Hammons
Tom Hoefling
Alyson Kennedy
Jerome Segal
Phil Collins

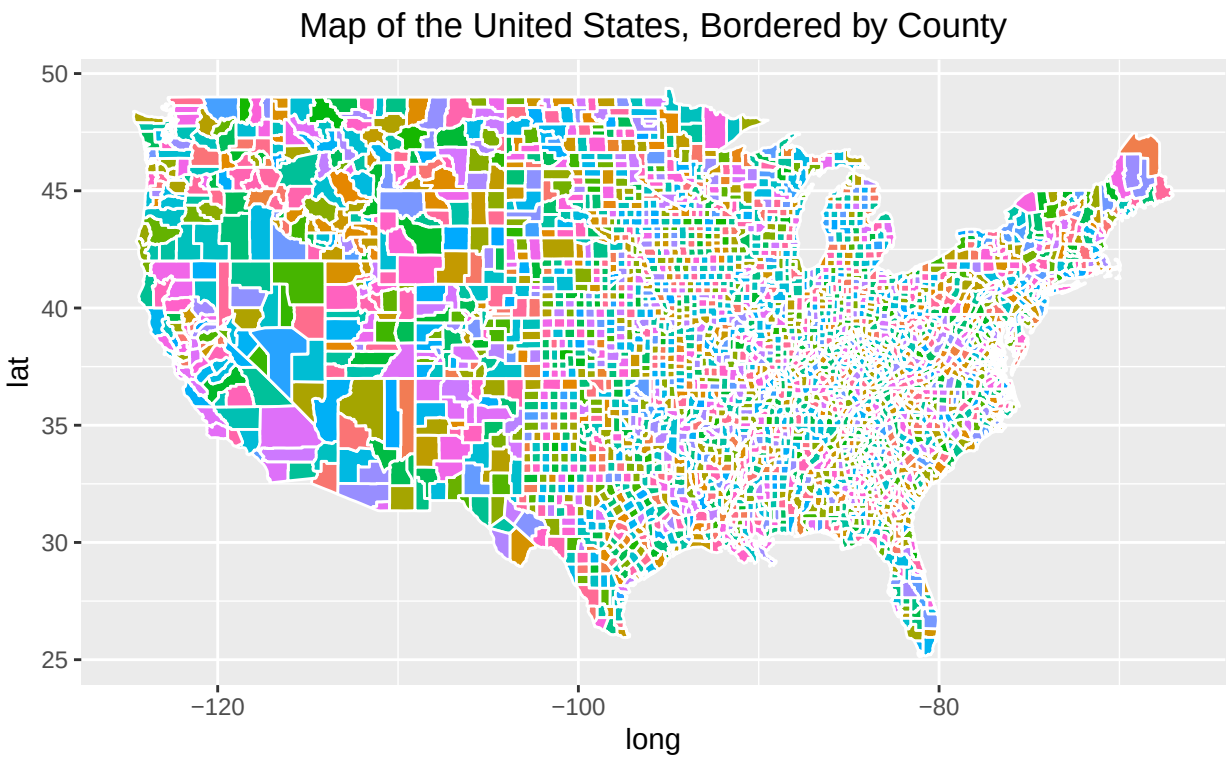


There were 36 named candidates in the 2020 election, 38 total but one is “Write-ins” and the other is “None of these candidates”.

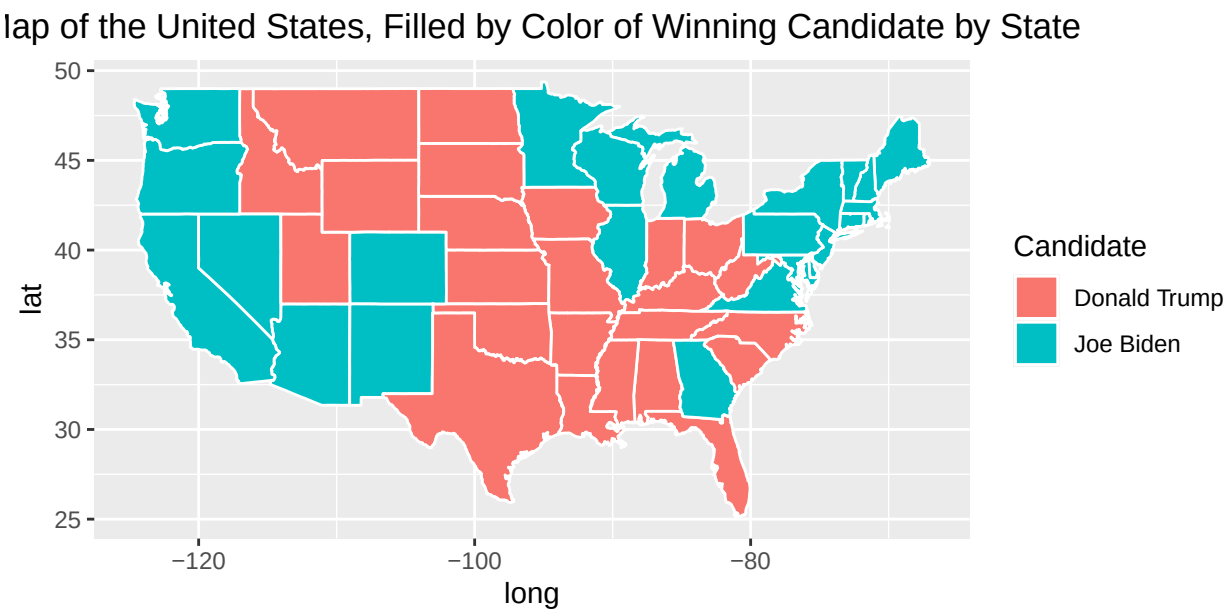
(Q5) See Source Code

Visualization

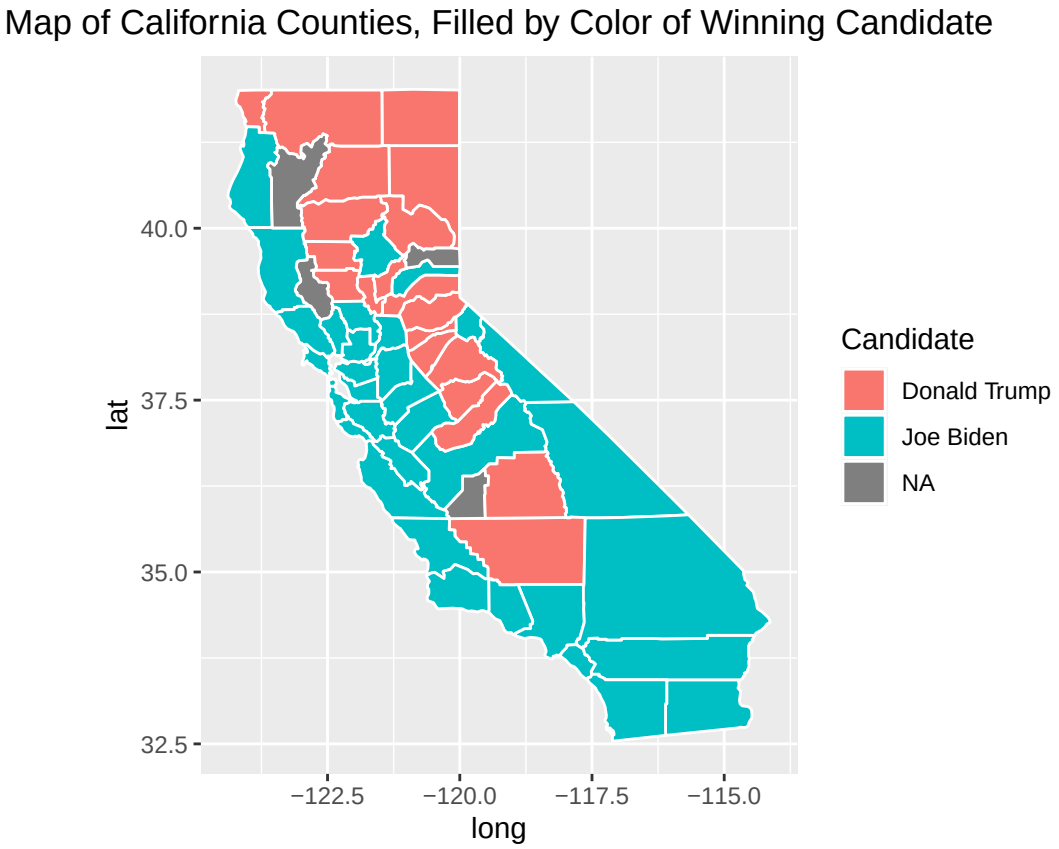
(Q6)



(Q7)


lap of the United States, Filled by Color of Winning Candidate by State
50
45
40
35
30
25
lat
−120
−100
−80
long
Candidate
Donald Trump
Joe Biden

(Q8)


Map of California Counties, Filled by Color of Winning Candidate
40.0
37.5
35.0
32.5
lat
−122.5
−120.0
−117.5
−115.0
long
Candidate
Donald Trump
Joe Biden
NA

(Q9)

(Q10)

First Five Rows of the Census Data

```
## # A tibble: 5 x 21
##   CountyId State   County         TotalPop   Men VotingAgeCitizen Income Poverty
##      <dbl> <chr>   <chr>             <dbl> <dbl>            <dbl>  <dbl>   <dbl>
## 1     1001 Alabama Autauga County    55036 0.489            0.745  55317    13.7
## 2     1003 Alabama Baldwin County   203360 0.489            0.764  52562    11.8
## 3     1005 Alabama Barbour County    26201 0.533            0.774  33368    27.2
## 4     1007 Alabama Bibb County       22580 0.543            0.782  43404    15.2
## 5     1009 Alabama Blount County     57667 0.494            0.737  47412    15.6
```

```
## # ... with 13 more variables: Professional <dbl>, Service <dbl>, Office <dbl>,
## #   Drive <dbl>, Carpool <dbl>, Transit <dbl>, WorkAtHome <dbl>,
## #   MeanCommute <dbl>, Employed <dbl>, PrivateWork <dbl>, SelfEmployed <dbl>,
## #   FamilyWork <dbl>, Minority <dbl>
```

Removed the Following Variables:
- ChildPoverty because correlation with Poverty is close to 1, 0.93823316.
- Women because it is a linear combination of Men and TotalPop.
- White because it is a linear combination of Minority and TotalPop.
- Unemployment, because it is colineared with Employment.
- OtherTransp because it is a linear combination with Walking, Bus, Carpool, etc.
- Production because it is colineared with Professional, Service, Office, and Employed.

Dimensionality Reduction

(Q11)

Sorted Principle Component Values

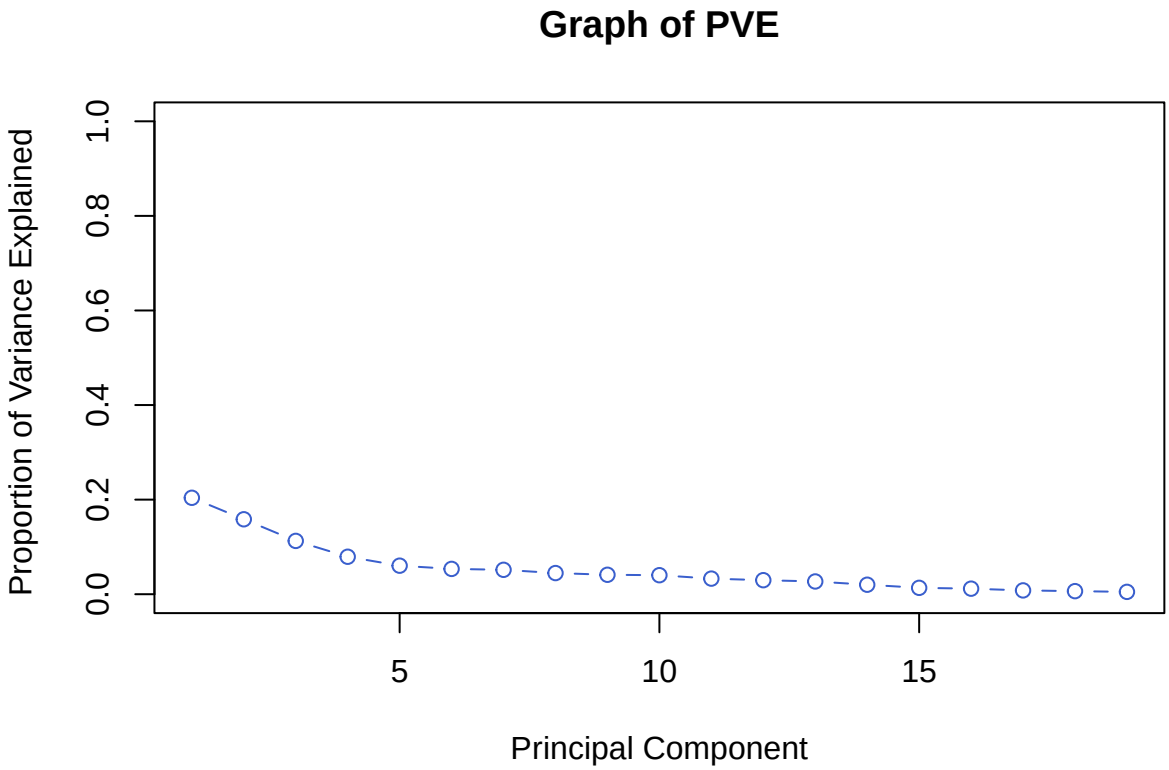
```
##  [1] 0.42144094 0.41451888 0.40193430 0.33839513 0.30538426 0.26742470
##  [7] 0.24741192 0.21682937 0.18456370 0.11786739 0.10835798 0.10276412
## [13] 0.10067037 0.08734112 0.07775389 0.07350481 0.03475860 0.02242102
## [19] 0.01775263
```

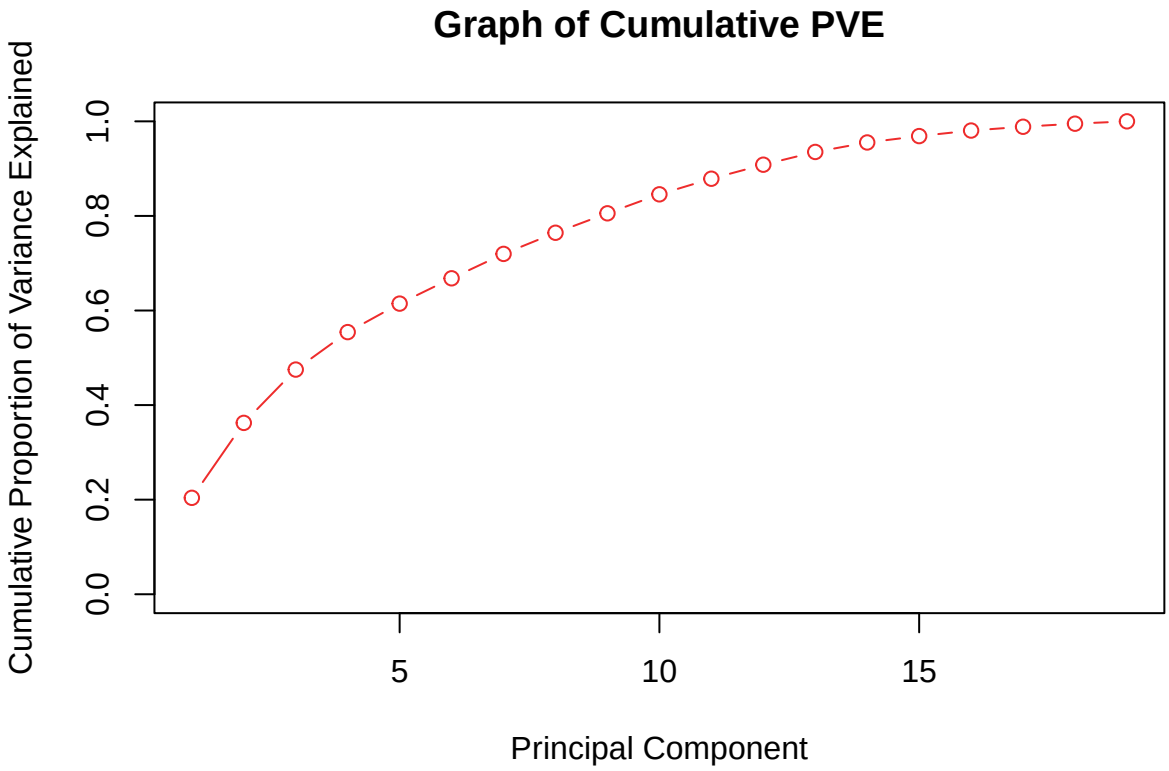
I center and scale my features prior to running PCA, because numerically, all of these predictors are on significantly different scales which means if I ran PCA with data that isn't all on the same scale I would have very poor results.

The three features with the largest absolute values of the first principal component are Employed, Poverty, and Income Their loading values are 0.42144094, 0.41451888, and 0.40193430 respectively.

TotalPop, Men, Income, Professional, Transit, WorkAtHome, Employed, PrivateWork, SelfEmployed, and FamilyWork all have negative loading values. Negative loading values are simply coefficients of some predictors which make the linear combination of a given principle component.

(Q12)


Graph of PVE
Proportion of Variance Explained
0.0
0.2
0.4
0.6
0.8
1.0
5
10
15
Principal Component




```
## [1] 12
```

The minimum number of principle components needed to capture 90% of the data is 12 principle components.

Clustering

(Q13)

```
## census.cut
##    1    2    3    4    5    6    7    8    9   10
## 2121  968   23    5   51    1   11   29    6    4

## pc.cut
##    1    2    3    4    5    6    7    8    9   10
## 2963  184    2   34    8   16    1    2    8    1

## # A tibble: 184 x 21
##    CountyId State      County     TotalPop   Men VotingAgeCitizen Income Poverty
##       <dbl> <chr>      <chr>         <dbl> <dbl>            <dbl>  <dbl>   <dbl>
##  1     1073 Alabama    Jefferson~   659460 0.474            0.745  49321    17.6
##  2     1089 Alabama    Madison C~   353213 0.489            0.750  61318    13.6
##  3     1097 Alabama    Mobile Co~   414328 0.478            0.748  45802    19.3
##  4     2020 Alaska     Anchorage~   298225 0.511            0.717  82271     8.1
```

```
##  5     4021 Arizona    Pinal Cou~   405537 0.522           0.705  52628    15.5
##  6     5119 Arkansas   Pulaski C~   392848 0.479           0.733  48850    17.3
##  7     6041 California Marin Cou~   260814 0.489           0.708 104703     8.1
##  8     6053 California Monterey ~   433168 0.510           0.535  63249    14.7
##  9     6061 California Placer Co~   374985 0.488           0.735  80488     8.2
## 10     6077 California San Joaqu~   724153 0.497           0.611  57813    17.1
## # ... with 174 more rows, and 13 more variables: Professional <dbl>,
## #   Service <dbl>, Office <dbl>, Drive <dbl>, Carpool <dbl>, Transit <dbl>,
## #   WorkAtHome <dbl>, MeanCommute <dbl>, Employed <dbl>, PrivateWork <dbl>,
## #   SelfEmployed <dbl>, FamilyWork <dbl>, Minority <dbl>
```

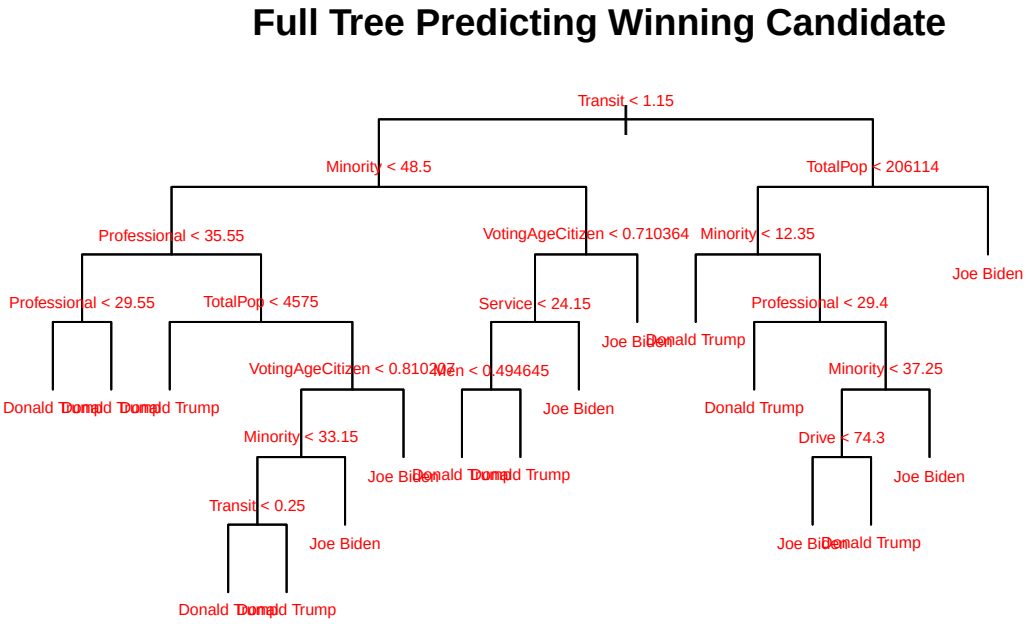
After cutting the tree into 10 clusters with the original data, I find that Santa Barbara County is in a cluster with 2120 other counties-making it very hard to interpret this cluster. With the first two principal components of pc.county, after cutting the tree into 10 clusters, Santa Barbara county is in a cluster with only 183 other counties. This makes it much more feasible to interpret the results of a cluster with only 184 observations as opposed to one with 2121 observations, so the second approach seems to cluster Santa Barbara more appropriately. It is important to note though that there are other clusters with very many observations, so while the second approach is better for Santa Barbara County, it may be useless if we are looking at other counties. In the cluster from the principle component method, Santa Barbara County is appears to be clustered with counties who have similar demographics for Employment, Income, and Minorities.

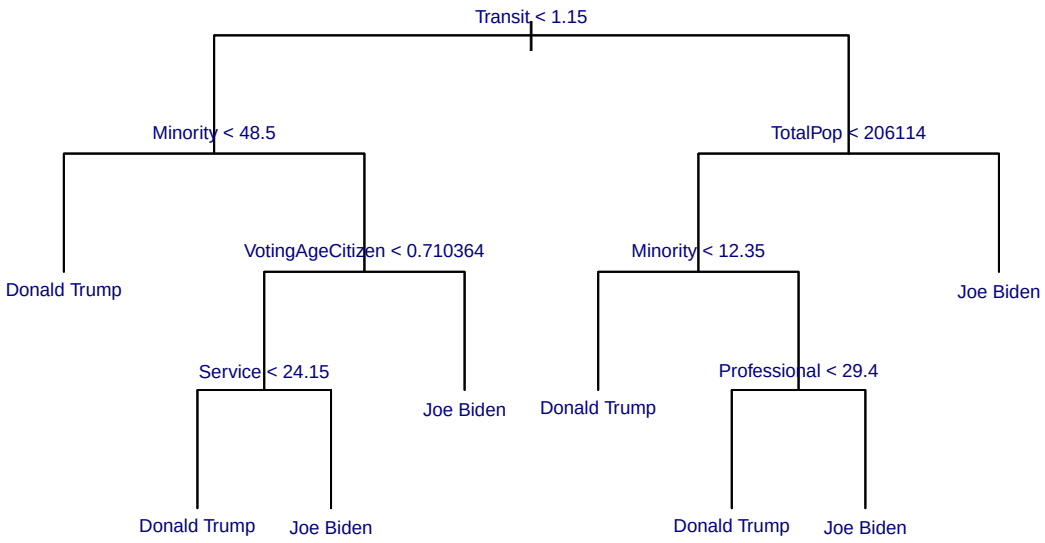
Classification

(Q14)

We need to remove the predictor party from the election.cl data set because all of the other predictors are numerical data, while party is a factor. If we left party in the election.cl data set we would not be able to run models from the data set. We are trying to predict which candidate will win which county from the census data, and the candidates party is not necessary to do so.

(Q15)


Full Tree Predicting Winning Candidate
Transit < 1.15
Minority < 48.5
TotalPop < 206114
Professional < 35.55
VotingAgeCitizen < 0.710364
Minority < 12.35
Joe Biden
Professional < 29.55
TotalPop < 4575
Service < 24.15
Professional < 29.4
Joe Biden
Donald Trump
VotingAgeCitizen < 0.810207
Men < 0.494645
Minority < 37.25
Donald Trump
Joe Biden
Donald Trump
Minority < 33.15
Drive < 74.3
Joe Biden
Joe Biden
Transit < 0.25
Joe Biden
Joe Biden
Donald Trump
Donald Trump




```
##          train.error test.error
## tree      0.08927336  0.1077348
## logistic          NA         NA
## lasso             NA         NA
```

The un-pruned tree has many more terminal nodes than the pruned tree, 17 and 8 terminal nodes respectively. The most impactful predictors for predicting the winning candidate for a county are Transit, Minority, and TotalPop. So for example, in a county where Transit is less than 1.15 and Minority is less than 48.5, Donald Trump is predicted to win. Another example, in a county where Transit is greater than 1.15, and the Total Population is greater than 206114, Joe Biden is predicted to win.

(Q16)

Implementing Logistic Regression

```
##          train.error test.error
## tree      0.08927336 0.10773481
## logistic  0.08166090 0.06906077
## lasso             NA         NA
```

The significant variables in my logistic regression model are TotalPop, VotingAgeCitizen, Poverty, Professional, Drive, Carpool, Employed, PrivateWork, and Minority. These significant variables are similar to the most significant variables from the decision tree. e to the power of these coefficients are the odds ratio which are associated with the prediction of the candidate. A unit increase will cause change by a factor of

$e^{coefficient}$. If there is an increase of the Employed coefficient which is 21.11 by 1, the odds of the outcome change by a factor of $e^{21.11}$. Similarly, if the VotingAgeCitizen coefficient which is 20.99 is increase by 1, the odds of the outcome change by a factor of $e^{20.99}$.
With the data I have, there is no linear combination of predictors which can perfectly predict the winning candidate.

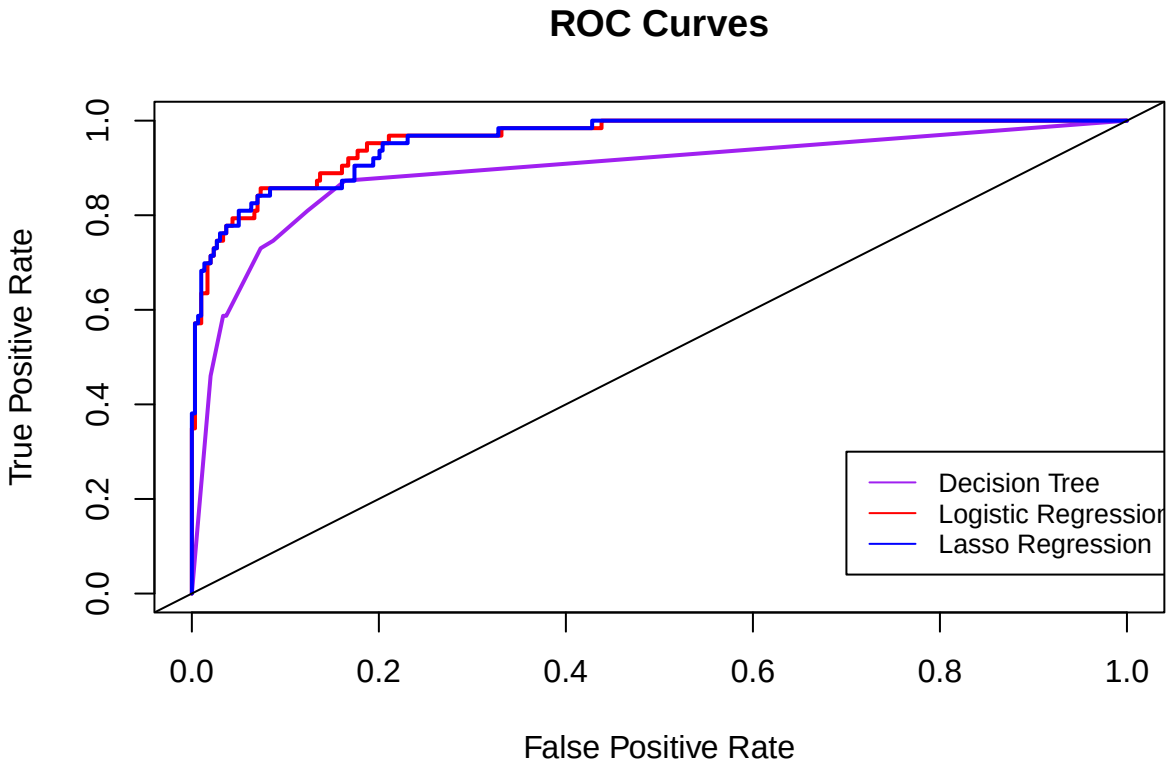
(Q17)

Implementing Lasso Regression

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                               s0
## votes             .
## TotalPop           1.811899e-06
## Men               -7.420482e+00
## VotingAgeCitizen   1.675239e+01
## Income             2.698796e-06
## Poverty            7.404152e-02
## Professional       1.462143e-01
## Service            1.660650e-01
## Office             4.886950e-02
## Drive             -1.043104e-01
## Carpool           -9.465524e-02
## Transit            8.958890e-02
## WorkAtHome         .
## MeanCommute        5.095082e-04
## Employed           1.788102e+01
## PrivateWork        4.555833e-02
## SelfEmployed      -6.237864e-02
## FamilyWork        -2.416082e-01
## Minority           1.105791e-01
```

The optimal value of lambda from the cross validation is 0.0026. The predictors which were found to be most meaningful are Employed and VotingAgeCitizen, and these are consistent with the most significant variables in logistic and lasso regression. I have two predictors which are zero, votes and WorkAtHome. These two predictors in my logistic regression model are not very significant, so the lasso penalizing them makes sense here.

(Q18)



Based on my classification results, Lasso Regression was shown to be the most accurate model for predicting candidate winner by county. The Logistic Regression model was a very close second to Lasso, while the decision tree model was quite far. The benefit of the decision tree is it's interpretability, it is very easy to visually see how the tree, quite literally, makes decisions. Although the disadvantage of the decision tree is it's accuracy compared to logistic and lasso regression. With Logistic and Lasso Regression, their disadvantage is their interpretability in the decision making, although the final result is very easy to understand. And of course their benefit is their low test error rates. I think the different classifiers are more appropriate for answering different questions about the election. Is it very likely that for a certain question a decision tree model will out perform a Logistic and Lasso Regression model.

Taking it Further

(Q19)

Implementing K-Nearest Neighbor Model to Predict Winning Candidate

```
##          train.error test.error
## tree      0.08927336 0.10773481
## logistic  0.08166090 0.06906077
## lasso     0.08650519 0.06629834
## KNN       0.15017301 0.12707182
## SVM              NA         NA
```

Implementing a Support Vector Machine Model to Predict Winning Candidate

```
##          train.error test.error
## tree      0.08927336 0.10773481
## logistic  0.08166090 0.06906077
## lasso     0.08650519 0.06629834
## KNN       0.15017301 0.12707182
## SVM       0.07474048 0.06629834
```

My KNN model had the worst train and test error rates out of all of my other classification models, likely because of the curse of dimensionality. Although it was the worst model, it's test error rate is not terrible. As we can see right above, the Support Vector Machine model had the lowest error rates out of all the classification methods used. It has the lowest train error rate, and matches Lasso Regression for the lowest test error rate.

(Q20)

Implementing Linear Regression to predict total votes for each candidate by county
Mean Squared Error's and Error Rate's Printed Below, Train and Test Respectively

```
## [1] 577871762
```

```
## [1] 2417858062
```

```
## [1] 0.4960265
```

```
## [1] 0.4827586
```

Linear Regression is a way of predicting how many votes a candidate will receive by county. Although I would personally prefer to use a classification method for predicting who will win a county, or state. Classification provides a quicker direct answer (ie who won), whereas with Regression some further wrangling is required to see who received the most votes. Furthermore, error rate with classification is much easier to interpret than MSE in Regression. Especially with very large data, MSE can get really large and is hard to interpret, while error rate is on a simple scale. An additional problem with Regression is that some candidates are predicted negative votes, which is obviously not possible. Of course, each method for their own use, if I am trying to get a number for total votes, go with regression. If I am only interested in the winner, then classification. Regression and Classification can complement each other if we want to validate the results beyond a test set (seeing if in fact the candidate with the most votes was predicted to win).

In order to get good interpretation of Regression, we would need to convert the votes into winner for each county, and classify the winner by county by the candidate which had the most votes. I did this, calculating train and error rates, and the results were not good, 0.4960265 and 0.4827586 respectively. The Linear Regression model is a pure linear model, and the relationship between the predictors and the number of votes is probably non-linear, which is probably a big factor for lack of performance. Another effect is that it is possible to get negative values for predicted votes (indeed I got negative values), which simply are not possible for this question. With that being said, it is understandable that linear regression is not a good fit for this type of question.

Q20 continued. . .

Classification with Dimension Reduction (PCA)

Implementing Logistic and Lasso Regression with Reduced Dimensions

```
##          train.error test.error train.error(pca) test.error(pca)
## logistic  0.08166090 0.06906077        0.1093426      0.08839779
## lasso     0.08650519 0.06629834        0.1660900      0.13259669
```

```
## 12 x 1 sparse Matrix of class "dgCMatrix"
##                s0
## PC1  .
## PC2  2.776852e-06
## PC3  6.565870e-09
## PC4  .
## PC5  1.323916e-05
## PC6  .
## PC7  2.322215e-05
## PC8  .
## PC9  .
## PC10 .
## PC11 .
## PC12 .
```

After re-running logistic and lasso regression with 12 principle components, which account for 90% of the variation, I have not gotten more accurate models. As shown above, the logistic regression test error rates are not too far off, but the model with the original data is still better. The same conclusion can be made with lasso regression, although the lasso regression model with PCA is much worse, from a test error rate of 0.06629834 with th original data, to 0.13259669 with PCA. Lasso actually removes some of the most important PC's because of its penalizing parameter. This specific model only has 4 out of 12 predictors with nonzero value, meaning a lot of information is lost. This process of taking away information by PCA and losing information by lasso increased the bias too much, ultimately leading to a poor model.

(Q21)

Overall Insights and Conclusion The ultimate goal of this project (other than experience) was to produce models which accurately predict who won a given county in the 2020 US Presidential election. To reach this goal, I did some data wrangling, some visualizations, and ran various different types of models. The data wrangling to is to make the raw data useful and valid for modeling. The visualizations were to understand what type of data I am dealing with, and to picture how it can be used for modeling. And finally, the models were created to predict which candidate won a given county.

The test error rates of each type of model I created are shown below.

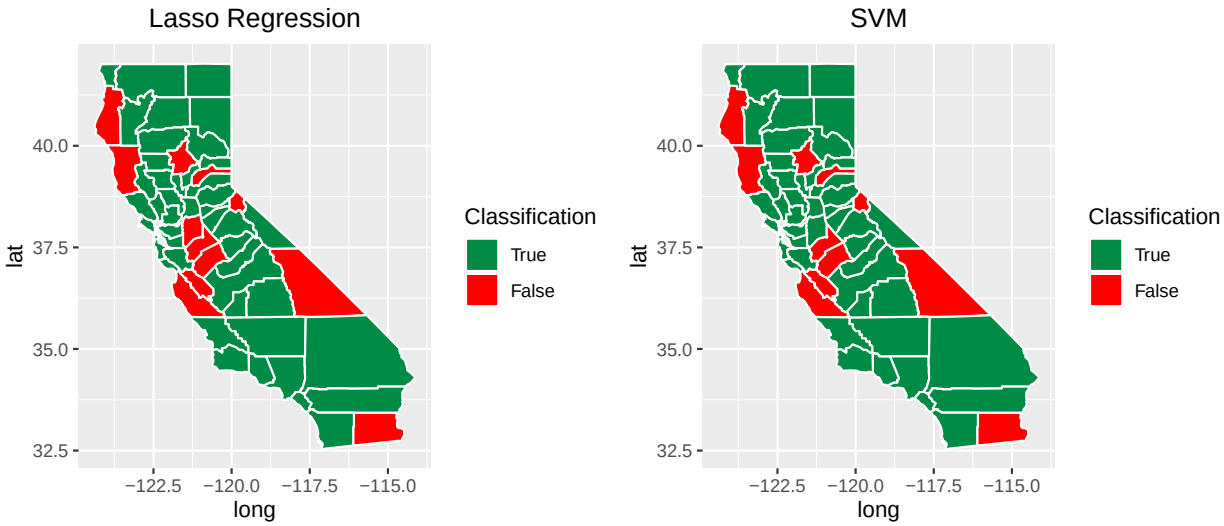
As we can see above, out of all the modeling methods, Linear Regression has the worst performance in predicting the winning candidate. On the flip side, my Support Vector Machine model was the most accurate in predicting the winning candidate among all the methods, with Lasso and Logistic Regression in a close second and third.

If we look closely, with the exception of the Decision Tree model, all test error rates are smaller than their respective training error rate. A possible explanation for this can be that the model is not overfit, which is a really good sign. A second reason is our test data set is 4 times smaller than our training data set. What this means is that when predicting with the test data, the model is actually predicting less observations, and sequentially there is less room for failure.

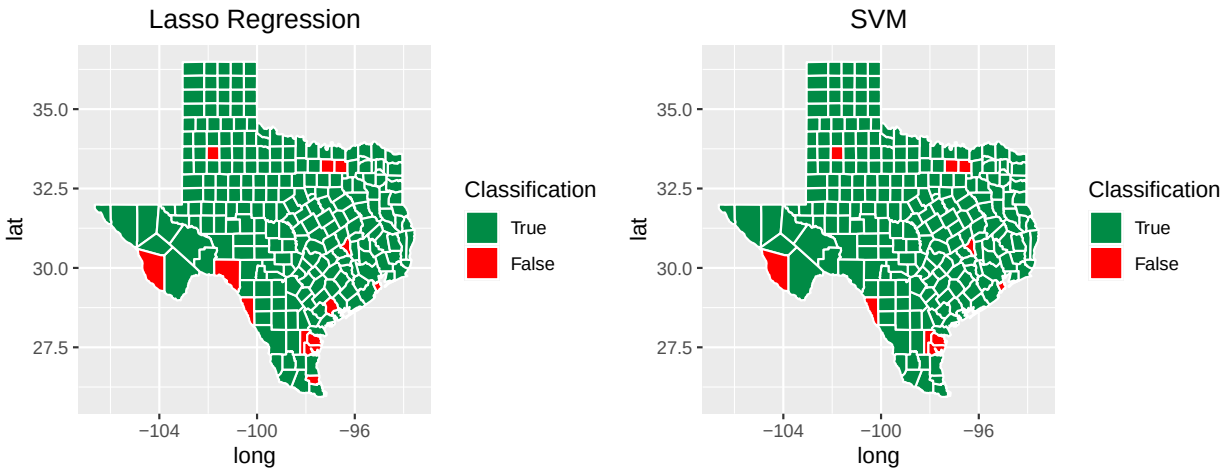
The two models with the lowest test error rate were Lasso Regression, and Support Vector Machine. So I decided to make maps of a few states visualizing where these two models misclassified the winning candidate. My train and test set were randomly selected, so not all of the counties of a given state are in the test set. So in order to make a complete map I combined the misclassifications from both the train and test set.

Below are maps indicating the counties Lasso Regression and Support Vector Machine misclassified in California, Texas, and Florida.

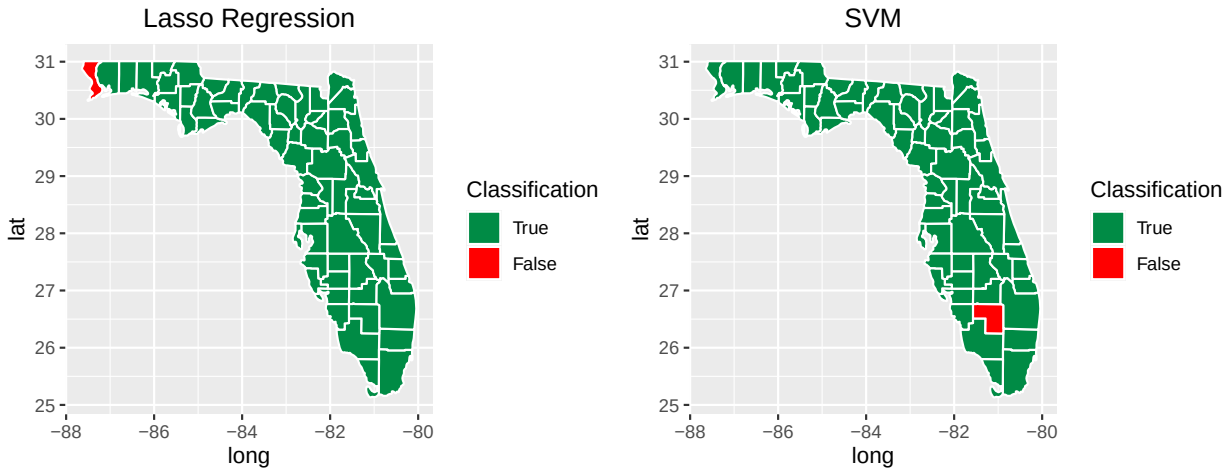
Misclassified Counties in California by Method



Misclassified Counties in Texas by Method



Misclassified Counties in Florida by Method



Relative to Florida and Texas, California has the most misclassified between the three. This suggests that my SVM and Lasso models were not great at predicting counties specifically in California.

Final Thoughts

A few of the models I created were very accurate, although there is a very important part to remember in this project is that all of the models were trained with the actual results from the 2020. In the real world, we would want to predict the 2020 elections before the elections were held, and with out the actual data from the election. So ideally, we would take data from prior elections, and a comnination of census data during those prior elections along with updated 2020 census data. The only issue with this is that we will have different candidates to predict, so we may have to classify based on political party.

Thank you for reading and analyzing!