PSTAT 131 Final Project

Robin Hollingsworth, Maxine Wang

March 19, 2021

Background

1. What makes voter behavior prediction (and thus election forecasting) a hard problem? According to "The Guardian" article named "How did Nate Silver Predict the US Election", there are many factors which make voter behavior prediction (and thus election forecasting) a difficult problem. First of all there are multiple statistical theories and modeling techniques that must be used and taken into account when determining voter behavior prediction. First, a hierarchical model must be used because the polling data is observed on a state and national level. The poll data is also based on people's thoughts which may change over time, therefore time series must also be used. Shocks must be used in order to account for extra, random/intangible uncertain effects that may change voter decisions. There is also variation between different polls which include error and bias such as sampling error, the quality of the poll, the shy Tory effect, and the house effect. The shy Tory effect regards untruthful responses to surveys, questionnaires, or polls. The house effect is when pollsters attempt to correct the shy Troy effect. Therefore there are a lot of effects that must be estimated.

2. What was unique to Nate Silver's approach in 2012 that allowed him to achieve good predictions? Once again, according to the same article, "How did Nate Silver Predict the US Election", there are many specifics of how Nate Silver's approach in 2012 was unique and allowed him to achieve accurate predictions. First, instead of maximizing the use of many different estimations of unknown variables, Silver used a range of probabilities instead. Silver also used Bayes' Theorem and conditional probability to place weight on each probability associated with the probability of the previous prediction being true. Finally, he used the previous election's actual outcome data to fit and test his model on. This also led him to account for more random factors that made the 2012 and 2008 elections differ.

3. What went wrong in 2016? What do you think should be done to make future predictions better? According to the article, "The Polls Missed Trump. We Asked Pollsters Why.", there are many factors as to what went wrong in the predictions of the 2016 election. Many different types of polls did not predict Trump being elected in 2016. Clinton lost by such a small margin that it could have been due to a polling error, since polling errors are normally 2 to 3 percentage points. Every poll has noise or nonresponse bias which is difficult to quantify, some examples include incorrectly estimating variables like race, gender, etc. Noise such as systematic polling errors occur in all different kinds of polls on the national and state levels as well as individual and aggregate polls. The shy Troy effect, which was previously discussed, was a factor in the prediction errors of the 2012 election. For example, women felt uncomfortable being honest about voting for Trump and the voter turnout was lower than expected for democrats. I think that in the future, in order to make predictions better, it would bode well to not underestimate any candidate and take into account the significzance of smaller margins of error when making predictions. For example, Nate Silver did not settle for maximizing variables, but instead took into account ranges of them to be more accurate and detailed. This can lead to more certainty and account for the smaller bias.

Election Data

4. Report the dimension of `election.raw` after removing rows with `fips=2000`. Provide a reason for excluding them. Please make sure to use the same name `election.raw` before and after removing those observations.

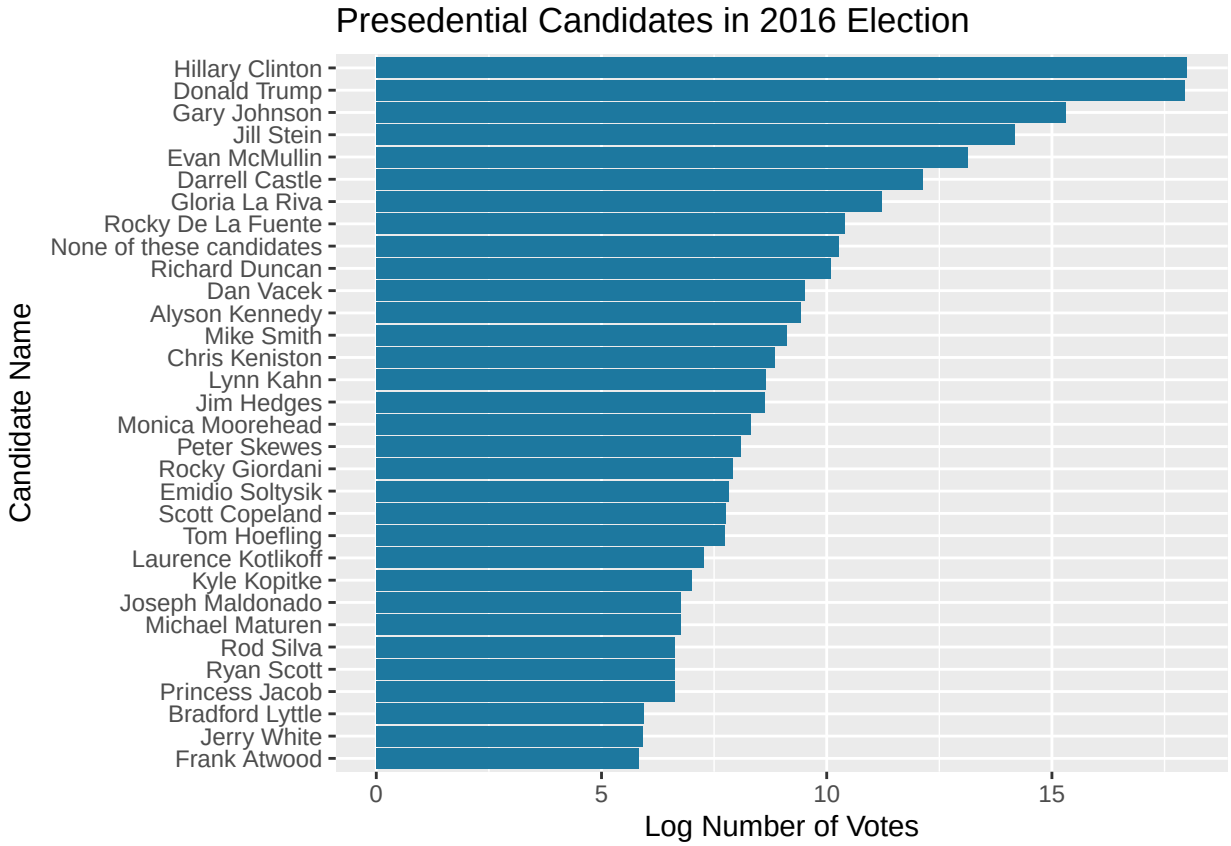
The dimensions of election.raw after removing rows with fips=2000 is 18345 x 5.
We are excluding rows with fips=2000 because these rows do not have a county associated with it (ie: county=NA). If it were a state summary row, then the fips would be the name of the state, but that is not the case. Therefore these rows are invalid and should be removed.

Data Wrangling

5. Remove summary rows from `election.raw` data:

See Rmarkdown for code

6. How many named presidential candidates were there in the 2016 election? Draw a bar chart of all votes received by each candidate. You can split this into multiple plots or may prefer to plot the results on a log scale.



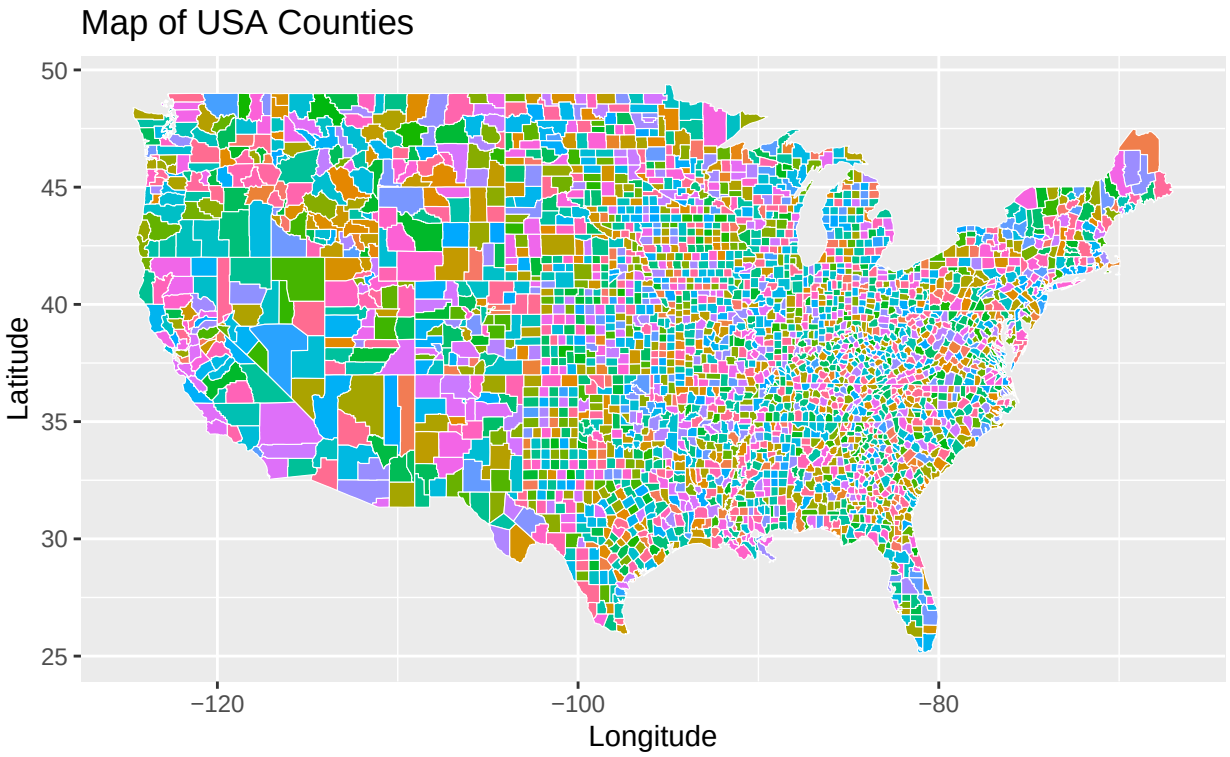
There were 31 named candidates and a *None of these candidates* option in the 2016 election.

7. Create variables `county_winner` and `state_winner` by taking the candidate with the highest proportion of votes.

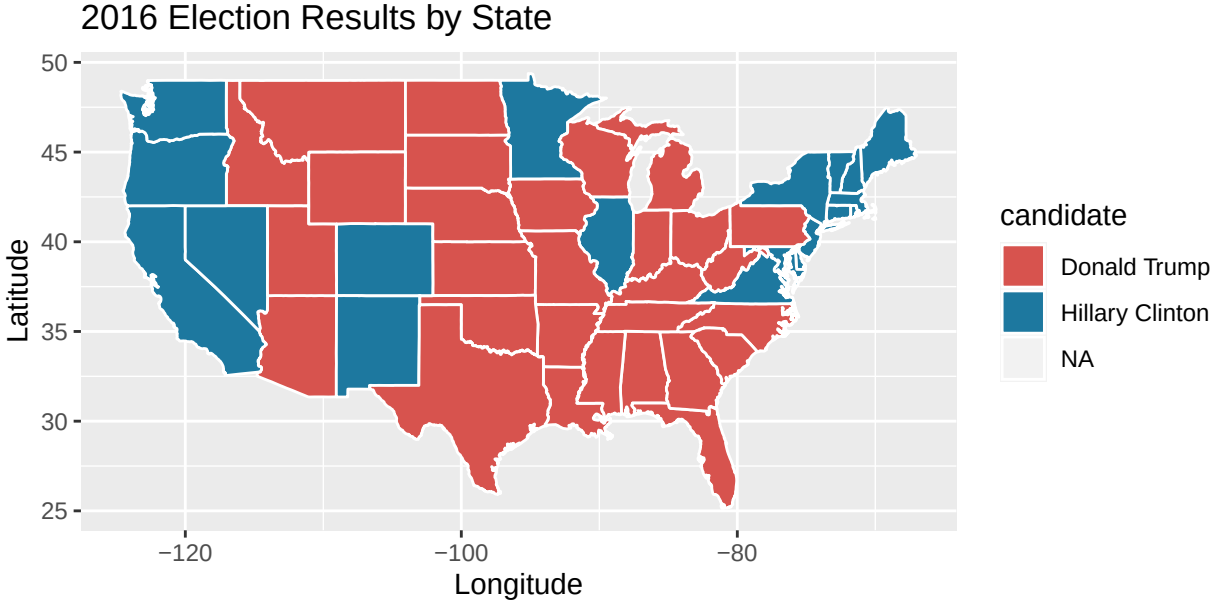
See notebook for code

Visualization

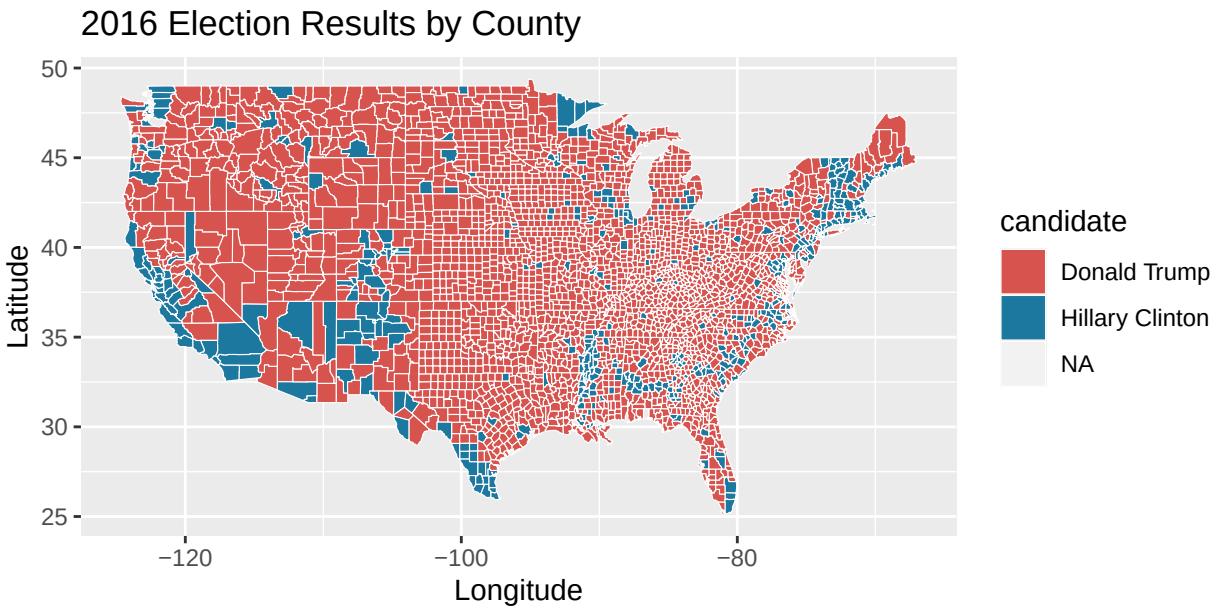
8. Draw a county-level map by creating `counties = map_data("county")`. Color by county.



9. Now color the map by the winning candidate for each state.

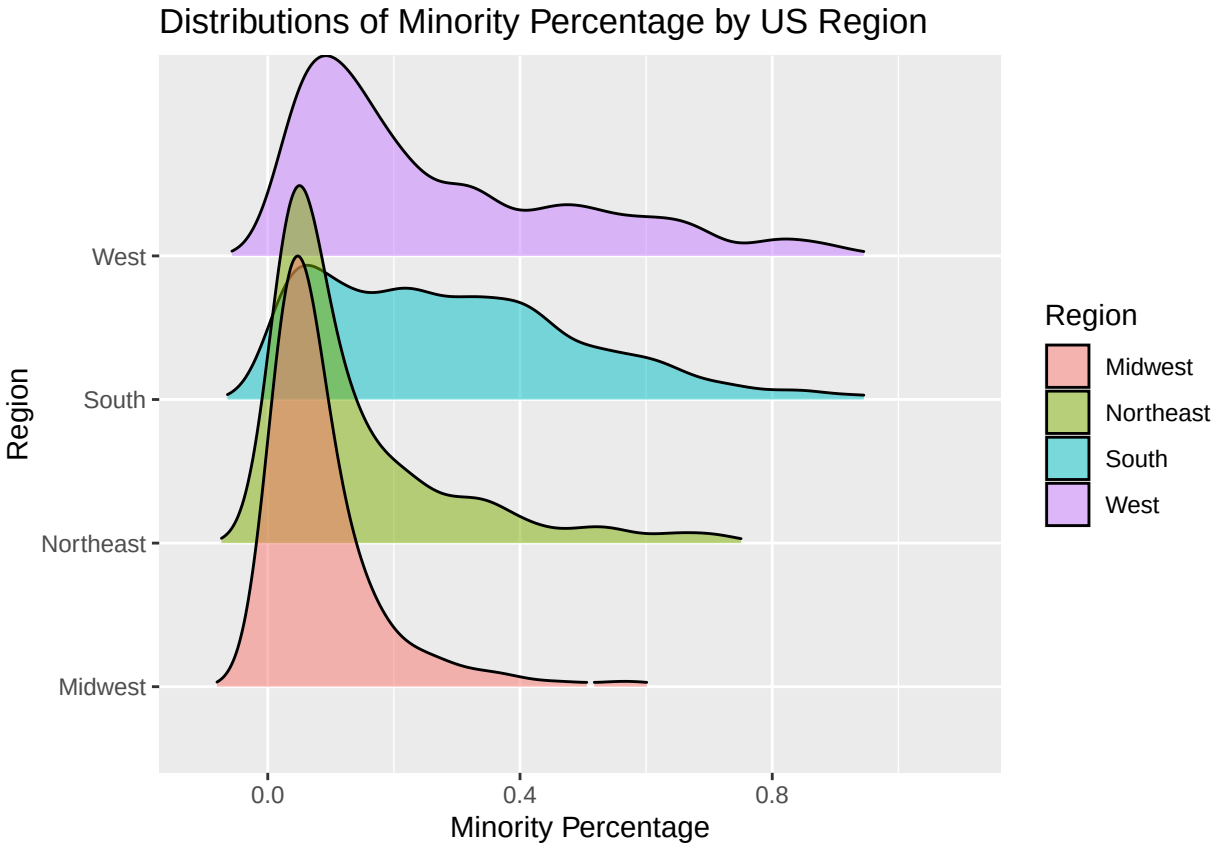


10. The variable `county` does not have fips column. So we will create one by pooling information from `maps::county.fips`.



11. Create a visualization of your choice using census data.

```
## Picking joint bandwidth of 0.0371
```



12. The `census` data contains high resolution information (more fine-grained than county-level). In this problem, we aggregate the information into county-level by computing `TotalPop`-weighted average of each attributes for each county.

Table 1: census.ct

State	County	TotalPop	Men	Women	White	Citizen	Income	IncomeErr
Alabama	Autauga	29848810	2228.750	15410366	348758.47	3393.750	237893387	35760241
Alabama	Baldwin	48458973	3074.645	24762534	523066.63	4764.355	321473552	55043318
Alabama	Barbour	9950462	1610.778	4464551	138345.48	2301.556	98628875	18047627
Alabama	Bibb	36057436	3018.250	16558031	420998.88	4373.750	219748374	31997986
Alabama	Blount	45004651	3168.000	22840854	563338.43	4705.000	296488132	55759313
Alabama	Bullock	15174189	1886.667	7004888	79014.27	2685.667	118499782	32035227

Dimenstionality Reduction

13. Run PCA for both county & sub-county level data.

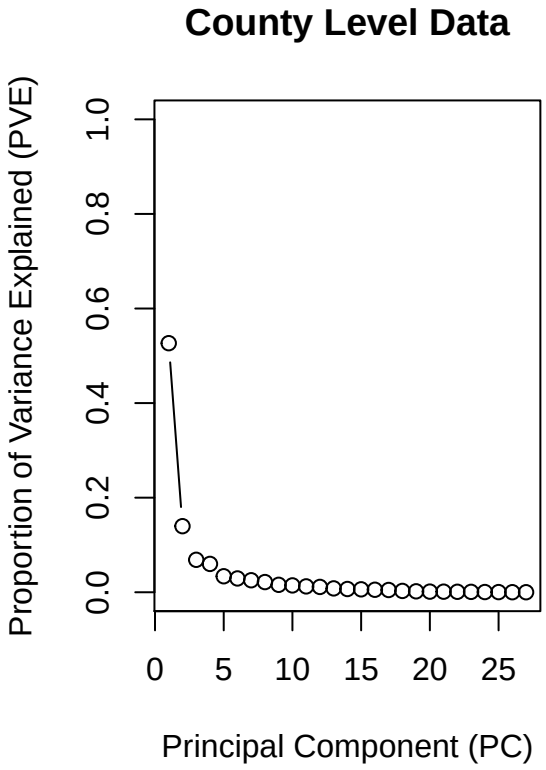
I chose to center and scale the features before running PCA for both the county and sub-county level data because when exploring the means and variances of each covariate for each level of data, it was apparent that the mean was not similar among variables. It was also apparent through exploring the variances of each variable that the variances all differed. Specifically, `Income` had the largest variance for both county and sub-county level data, meaning that if I did not center and scale the features before performing PCA, then the principal component observed would be driven mostly by Income. Therefore, standardizing the mean to 0 and the standard deviation to 1 before performing PCA is necessary.

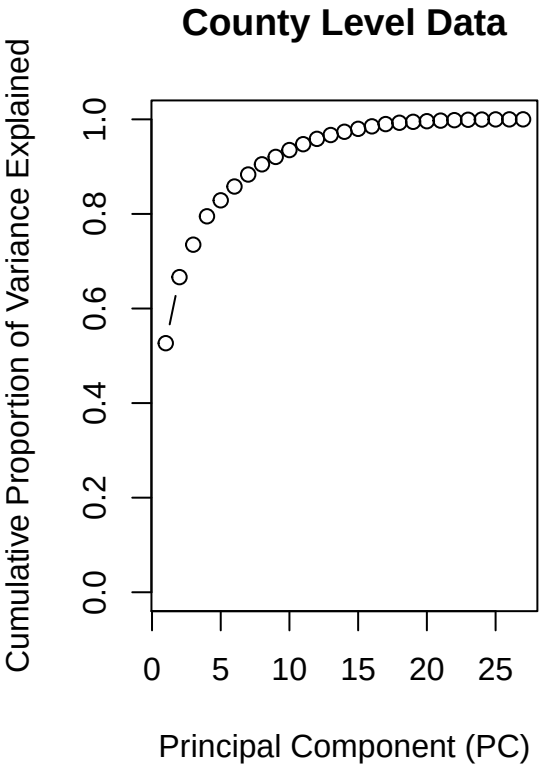
The 3 features with the largest absolute value of eigenvalues for the first principal component for *county* level data are Men, PrivateWork, and Citizen.
The 3 features with the largest absolute value of eigenvalues for the first principal component for *sub-county* level data are IncomePerCap, Professional, and Poverty.

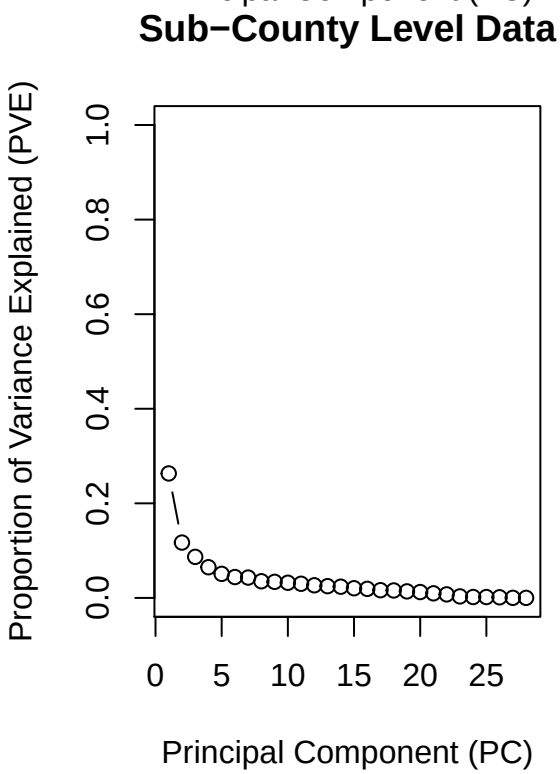
For county level data, TotalPop and FamilyWork have opposite signs. For sub-count level data, PrivateWork and FamilyWork have opposite signs. Therefore they will have a negative correlation. But signs are arbitrary and do not hold meaning, so the absolute value of the correlation is correct. And that is why we want to look at the absolute value of the eigen values for the top 3 features with the largest principal components.

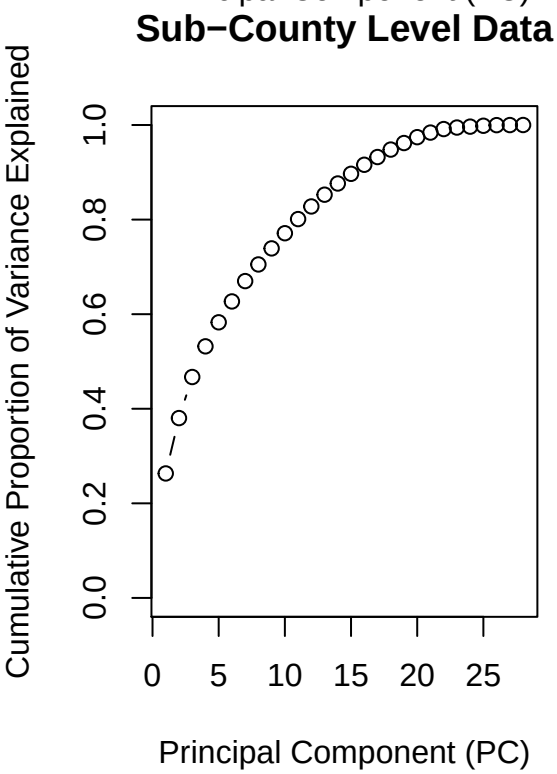
14. Determine the number of minimum number of PCs needed to capture 90% of the variance for both the county and sub-county analyses.

8 PCs are needed to capture 90% of the variance for county analyses. 16 PCs are needed to caputre 90% of the variance for sub-county analyses.



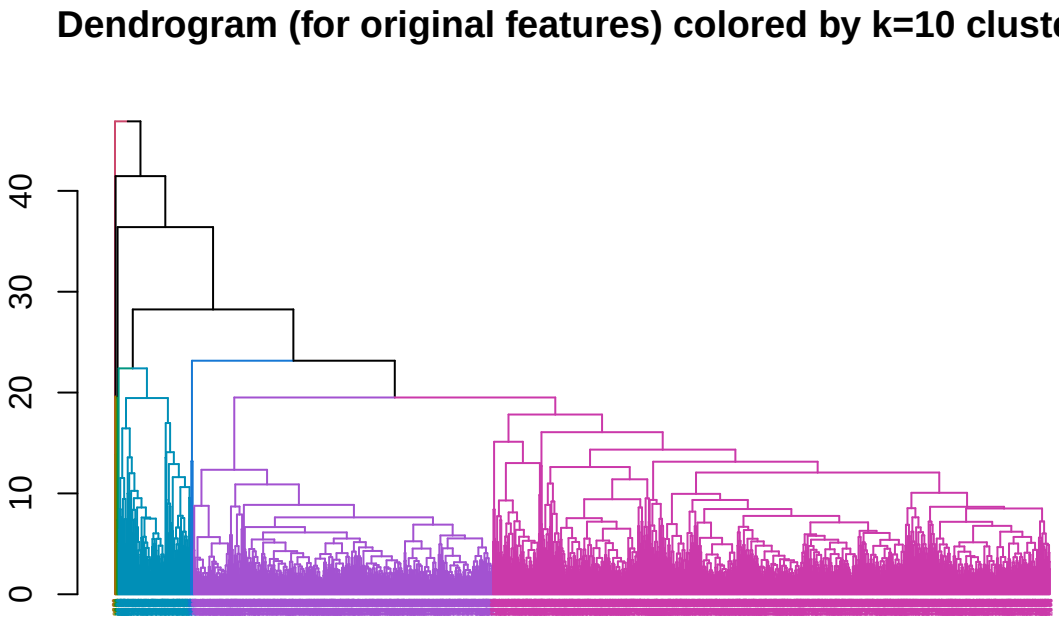


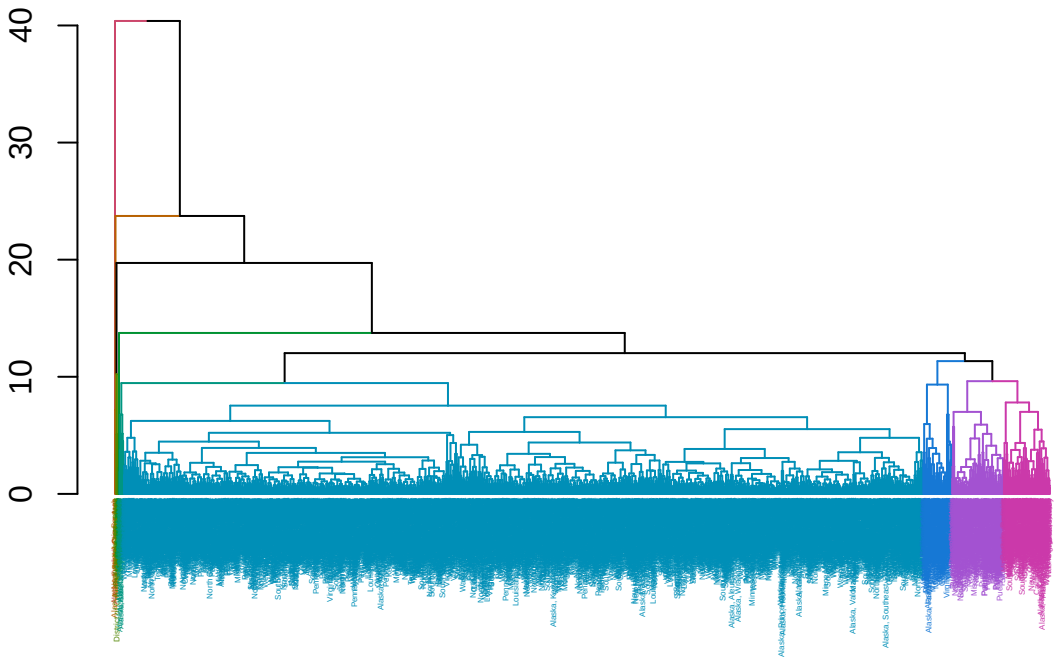




Clustering

15. With census.ct, perform hierarchical clustering with complete linkage.





Looking at the table with the distribution of observations among clusters, the model with the original features does a better job at spreading out the observations to each cluster, although they both have several clusters with a small number of observations. For the dendrogram with using the original features we can see that there are about 3 clusters with a very large number of observations so maybe this is a better value for k. The dendrogram using the first 5 principal components has one cluster which has the majoirty of the observations and 3 or 4 other clusters with the majority of the rest, therefore this is a less helpful model. In the comparison table we can see a lot of disagreement.

Silhouette coefficients closer to 1 mean that the observation is well placed in its cluster. So the model that results in a higher silhouette coefficient for San Mateo County's observation is better. We have found that the silhouette coefficient for the original model is 0.1131399, and the silhouette coefficient for the first 5 PC

model is 0.3575104, so the first 5 PC model places San Mateo County in a more appropriate cluster.

A possible explanations for this could be that dimension reduction and using the first 5 principal components results in a more accurate, fitting clustering result for this data instead of just using the raw features. Also, looking at a different number of clusters, possibly trying to find the optimal number of clusters using a different method, would probably result in a better, more accurate model.

Classification

Decision Tree

16. Decision Tree: train a decision tree by `cv.tree()`.

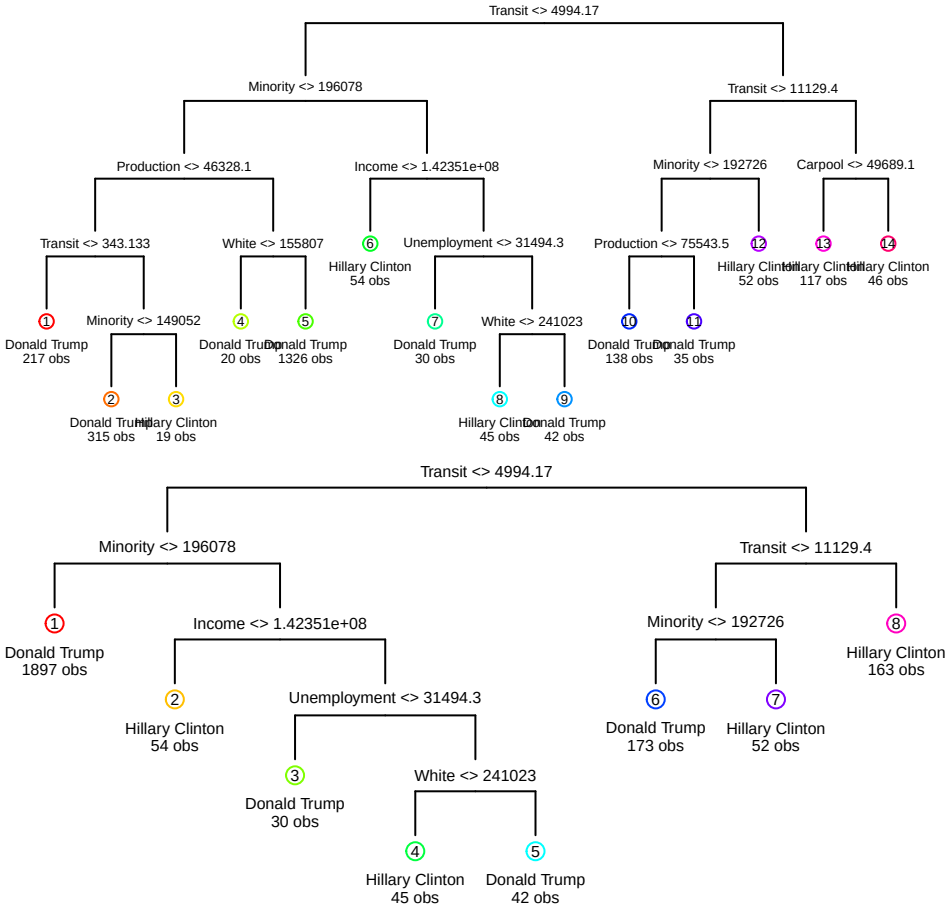


Table 2: Records

	train.error	test.error
tree	0.0859121	0.0747967
logistic	NA	NA
lasso	NA	NA

From both the pruned and the unpruned tree, the first variable that it splits off of is Transit <> 4994.17. The main difference between the pruned and unpruned trees is that the unpruned tree is significantly larger than the pruned tree and splits off more variables. The pruned tree actually only splits based off of `Transit,`

`Minority, Income, Unemployment and White`. From this pruned tree we can see that Hillary Clinton is preferred in counties that are high minority, high transit, and low income. Donald Trump was the winner in counties that are low transit, low minority, high income, high white and low employment.

Logistic Regression (unpenalized)

17. Run a logistic regression to predict the winning candidate in each county.

The significant variables in the logistic regression are `White, Drive, Carpool, Citizen, IncomePerCap, Professional, Service, Production, Employed, Private Work, and Unemployment` at level 0.001. A unit increase for any of these variables corresponds with a multiplicative change in the odds by e raised to the coefficient of that variable. For example, the coefficient for the `Employed` variable is 5.451e-3, meaning that for every unit that `Employed` increases, the logit function increases by 0.005451. The coefficient for the `White` variable is -5.802e-05, meaning that for every unit that `White` increases, the logit function decreases by -5.802e-05. This means that for `Citizen, IncomePerCap, Professional, Service, Production, Employed, Private Work, and Unemployment`, the logit function increases and approaches 1 as these increase and as `White, Drive, and Carpool` increase, the logit function decreases and gets closer to 0.

In terms of candidates, high values of `White, Drive, and Carpool` indicate a more Donald Trump swayed vote and high values of `Citizen, IncomePerCap, Professional, Service, Production, Employed, Private Work, and Unemployment` indicate a more Hillary Clinton vote.

LASSO Logistic Regression (penalized)

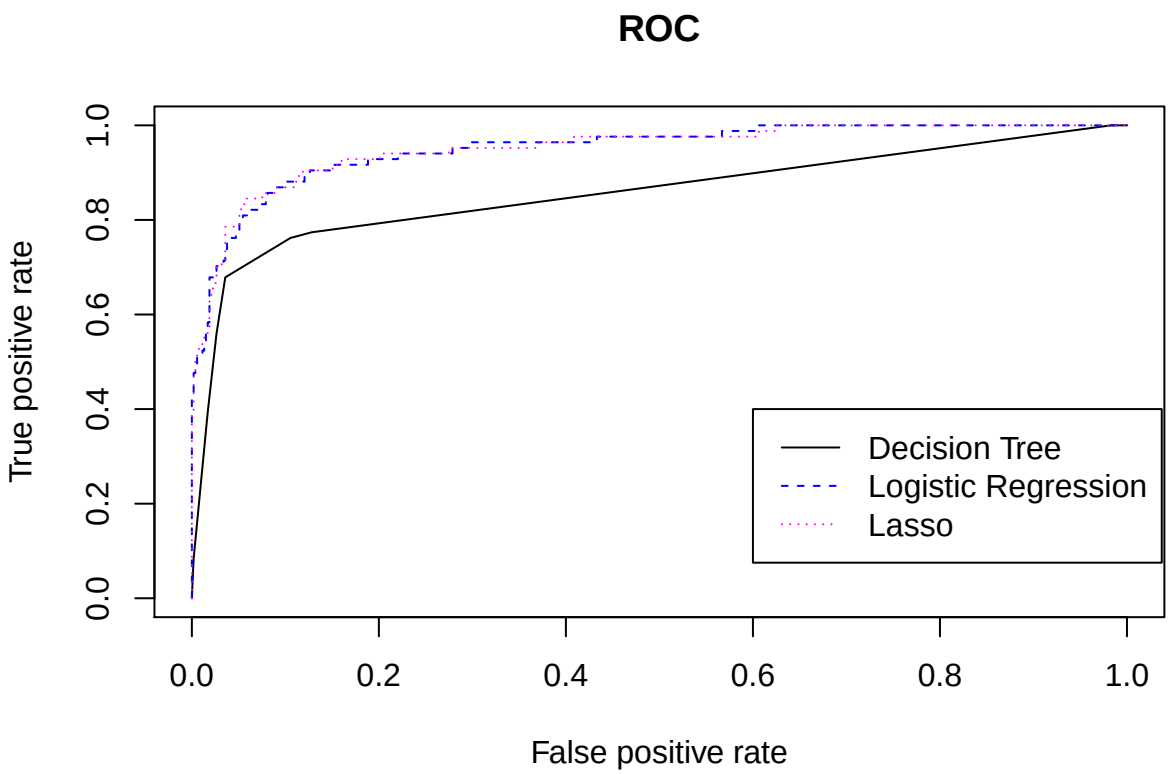
18. You may notice that you get a warning `glm.fit`: fitted probabilities numerically 0 or 1 occurred.

Table 3: Records

	train.error	test.error
tree	0.0859121	0.0747967
logistic	0.0679967	0.0682927
lasso	0.0671824	0.0699187

The optimal lambda value is 5×10^{-4} in cross validation. There are no non-zero coefficients in LASSO regression for the optimal lamba, because ridge regression does not perform variable selection. The unpenalized logistic regression has smaller training and testing errors than the LASSO training and testing errors, but not by much. Both the unpenalized logistic regression and LASSO are better (and have lower training and testing errors) than decision trees. The small difference between LASSO and unpenalized logistic regression can also be seen on the different ROC curves below.

19. Compute ROC curves for the decision tree, logistic regression and LASSO logistic regression using predictions on the test data.



Based on your classification results, I would say that logistic regression is the best method of classification in general, because it has the lowest testing and training errors. Decision trees would probably be the least accurate method of classification, because it has the lowest area under the ROC curve, and the testing and training errors are the highest. As discussed in the last question, LASSO and unpenalized logistic regression are pretty similar, but unpenalized has slightly lower training and testing errors. Since they are very similar, using the penalized logistic regression (LASSO), would probably be safer and yield a better model because it will control how complicated the model is. Since logistic regression. Since decision trees are a hard classification method, it classifies based on decision boundaries and can more easily be split into many levels of classification, while logistic regression is a soft classification method based on probability it is often used for binary classifications. Therefore, in terms of answering different kinds of questions about the election logistic regression is probably better for predicting a final answer about who will win the election: Clinton or Trump. On the other hand decision trees would be more fitting for answering questions about the demographics of voters based on their political affliations or who they voted for.

Taking it Further

KNN

We also took the analysis further by using another classification method: K-Nearest Neighbors. We used cross validation to find the best k to use, which we found was 22. So, the table below shows the knn train and test errors along with the rest of the methods we used. As you can see, the KNN classification method has the highest train error rate, but the second highest test error rate. Therefore, the logistic regression methods (both penalized and not), remain the best classification methods. But KNN may be an alternative that is better than the decision trees classification method.

Table 4: Records

	train.error	test.error
tree	0.0859121	0.0747967
logistic	0.0679967	0.0682927
lasso	0.0671824	0.0699187
knn	0.0802117	0.0861789

Summary of Findings

To outline the process, we first started with using Principal Component Analysis to reduce the dimensions of the data and then used it for hierarchical clustering with complete linkage. Hierarchical clustering was also performed with the original data. Then, several methods of classification were used to model the predictions. We consecutively added the train and test errors to a table in order to compare the methods. The methods used were Decision Trees, Logistic Regression (unpenalized), LASSO Logistic Regression (penalized), and then in addition: K-Nearest Neighbor (KNN).

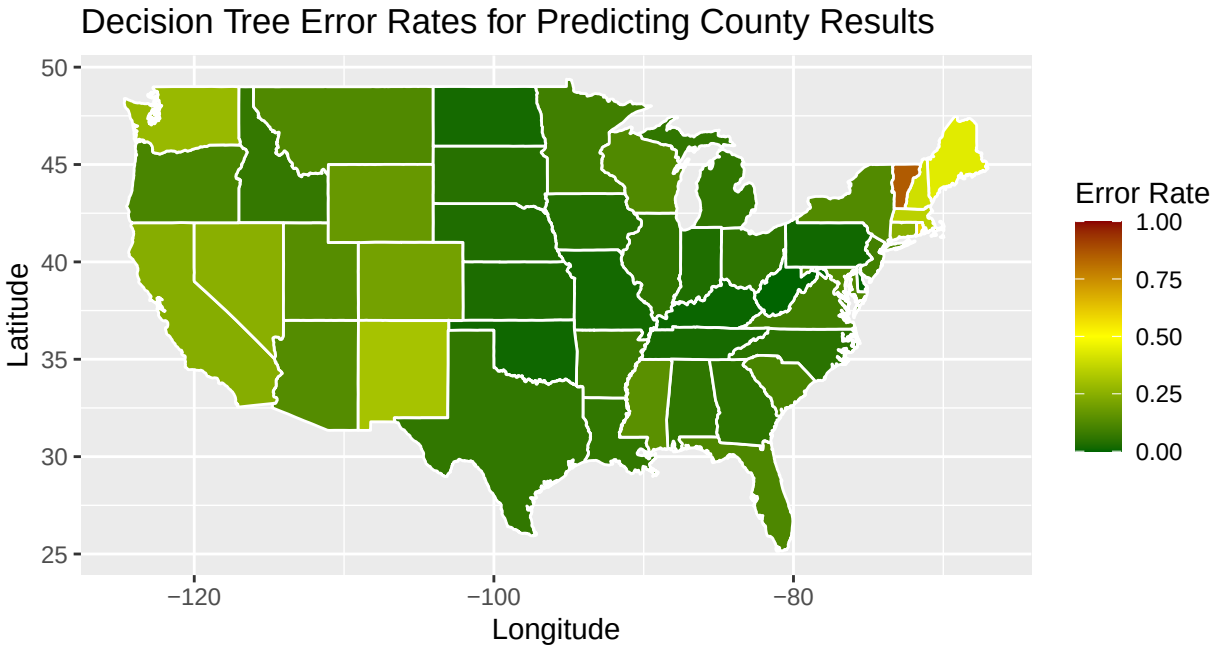
From comparing the test and training errors for each method, we concluded that logistic regression was the best method to use. LASSO (penalized) and unpenalized Logistic Regression had very similar train and test errors. Overall these 2 methods had the lowest errors compared to KNN and Decision Trees.

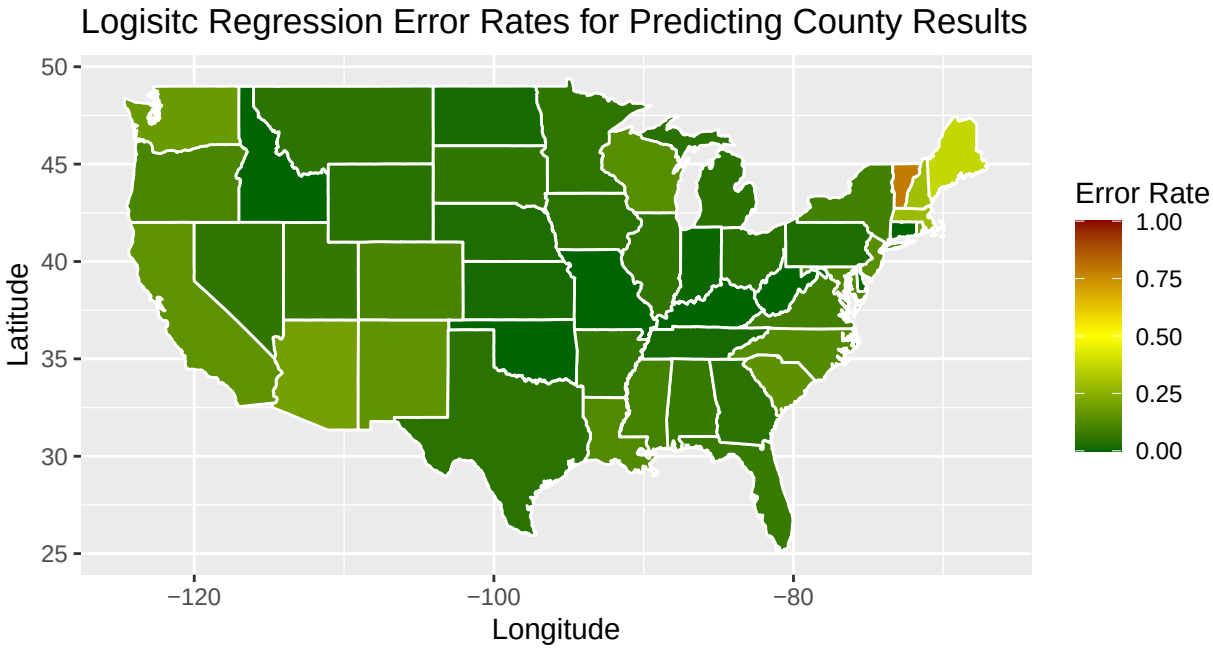
Some of the variables in the logistic regression model were not quite of what we would have anticipated based off our previous knowledge of politics. We were surprised by the signs of coefficients for certain variables. For example, we would have expected a high level of `Employed` resulting the logit function decreasing toward zero and getting closer to predicting Donald Trump. A possible explanation to this phenomenon could be Simpson's Paradox which says that a trend appears when different variables are separated but disappears or reverses when these variables are combined. Because we are looking at these demographic factors together as a group, the influence a variable has on the result could be the inverse of the relationship if it was observed individually or what we would believe to be true based off of our political knowledge.

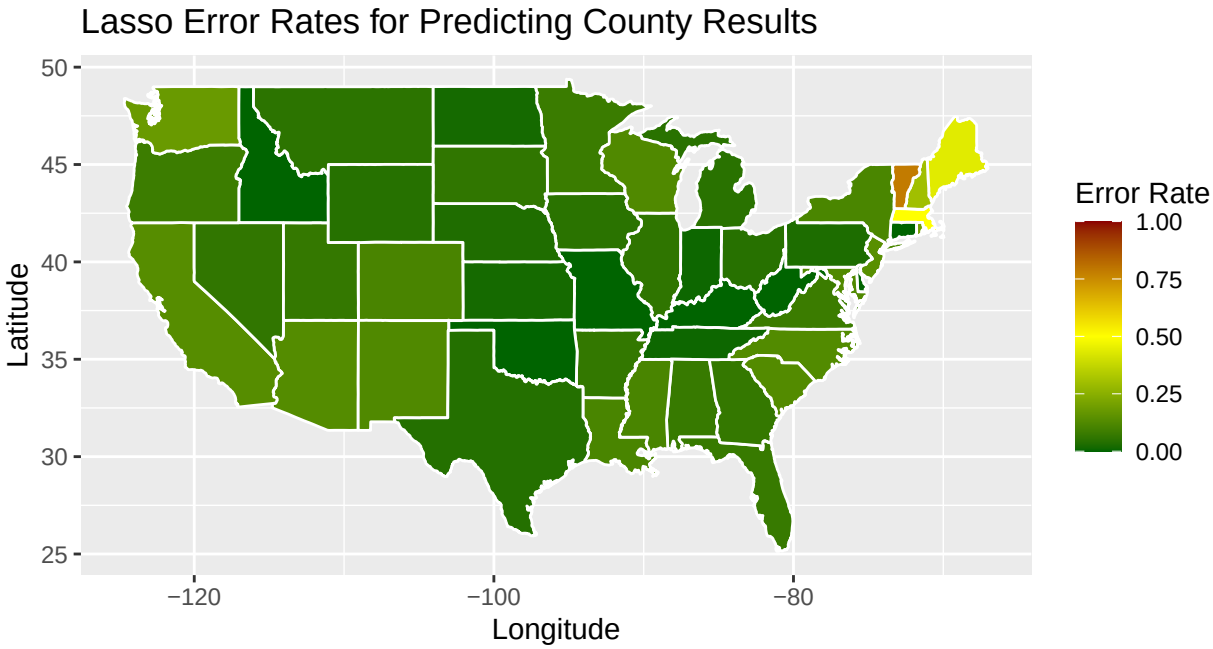
Map of Errors

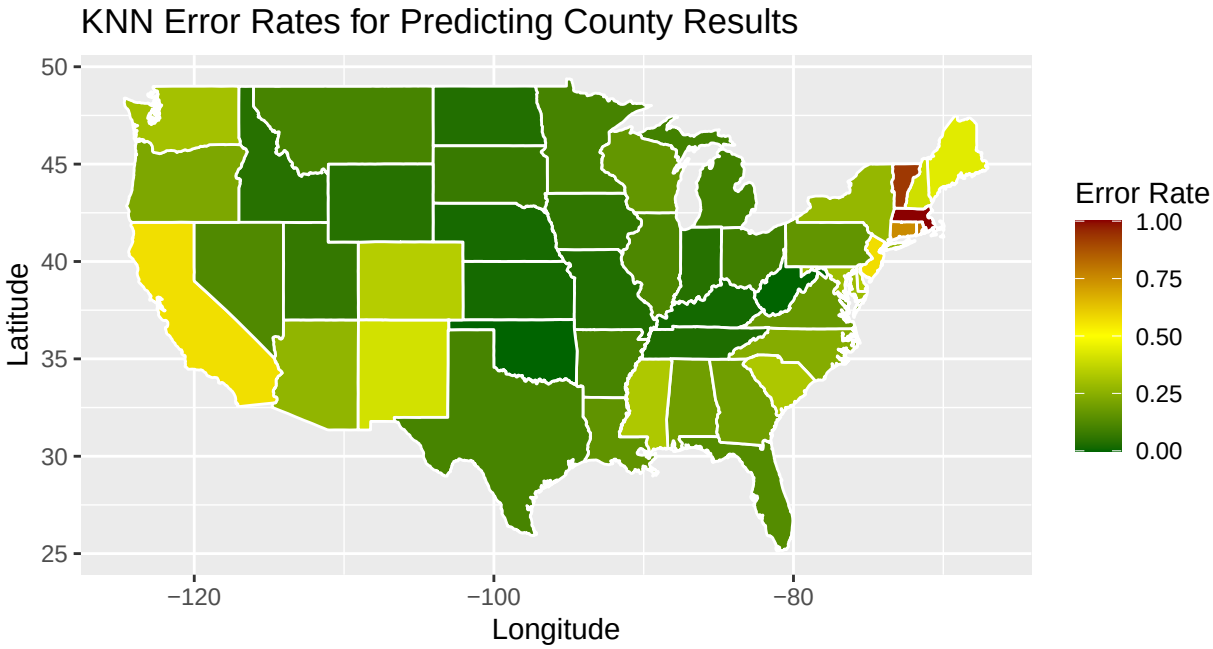
In this section, the maps show each classification model's error rate for the state based off of county winner predictions (i.e. If the state had 6/10 counties predicted correctly, the error rate for that state would be 0.4). We can see similar results to what we saw with the train and test errors found in the `records` table, which is that the decision tree has the worst errors out of the original three classification models (decision tree, logistic regression, and lasso). This also matches our findings in the ROC curves graph. We also created a map for our KNN model error rates.

The maps below highlight a notable trend through all three models that the states with the higher error rates and number of misclassified counties are those on the west coast and the far northeast section of the map. We used the 3 models (decision tree, logistic regression, and lasso) to predict the winner of each county. Then all of the predictions errors were averaged for each county in order to get a state average.


Decision Tree Error Rates for Predicting County Results
Latitude
Longitude
50
45
40
35
30
25
−120
−100
−80
Error Rate
1.00
0.75
0.50
0.25
0.00

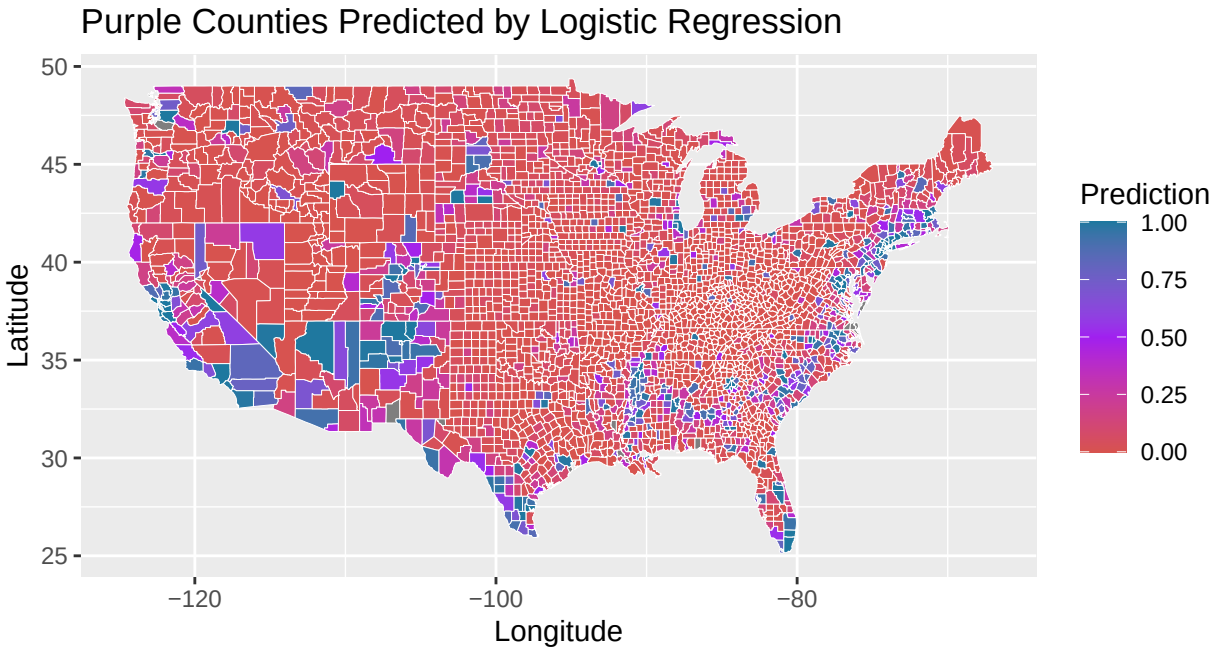

Logisitc Regression Error Rates for Predicting County Results
Latitude
Longitude
50
45
40
35
30
25
−120
−100
−80
Error Rate
1.00
0.75
0.50
0.25
0.00





The Decision Tree and KNN error map looks more yellow because they have the highest error rates. Out of the original 3 models, the decision tree was the worst performing. The smaller states in the northeast have higher error rates, probably due to the fact that they have less counties than the other states so one false prediction for a county is going to affect those counties more than it would affect a big state with many counties. California stands out on the west coast as being pretty inaccurate for KNN model and slightly for the decision tree model as well.

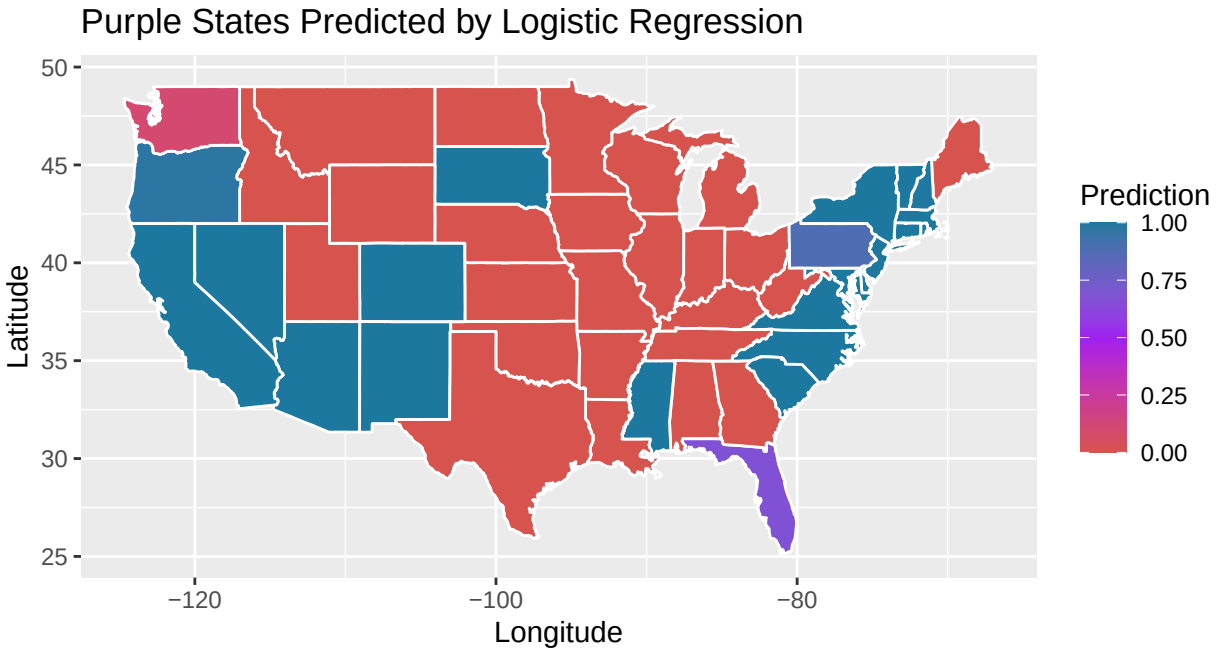
"Purple" Counties

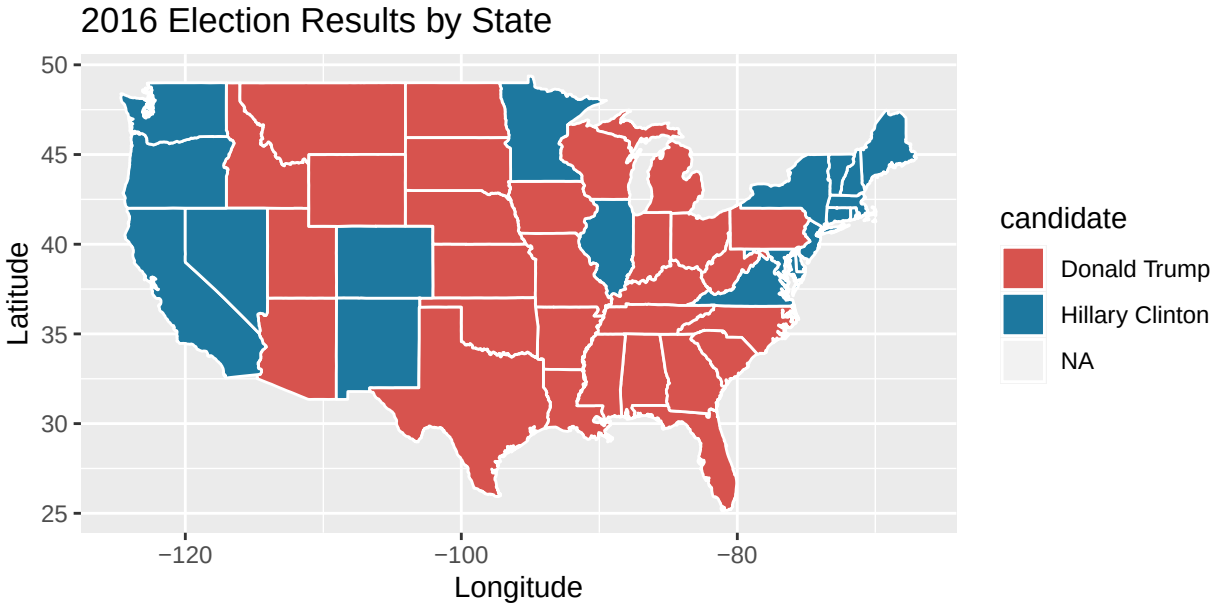


In this map, we have colored the counties in the US to correspond with the prediction from the logistic regression model we created in question 17. The "purple" counties are the counties between "red" and "blue" meaning that the model predicted that Clinton (blue) and Trump (red) had very close probabilities of winning. One notable thing about this map is that the center of the US has a large concentration of red counties, meaning that the model predicted the probability of Trump winning to be very high in these counties. On the other hand, a lot of the blue and purple counties seem to be on the outer edges of the country. Most of the purple counties also seem to be next to or very close to a blue county. Our model did a good job in predicting the blue counties in comparison to the map of the actual results from the election by county (see question 10 map). According to the results of the election, most of the purple counties seemed to have ended up voting blue.

An important factor to remember when looking at a graph like this is that while it looks like there is a lot of red in terms of area, the way the election works is through population of each state. So even though there is a lot of red in the middle section of the US, those counties might have overall little sway in the vote because of their small populations compared to those with large populations, like those in California.

"Purple" States





We also chose to use our logistic regression model to predict the "purple" outcomes of the states. The most significant part of this graph is the purple states of Pennsylvania and Florida and all of the other states had predictions from the model where it was not very close between Clinton and Trump. Interestingly, the model was somewhat accurate in determining which states would be blue, but according to the map of the 2016 election results, the model incorrectly predicted Arizona, South Dakota, Mississippi, South Carolina, Maine, Illinois and Minnesota. Although the model was incorrect for 2016, when looking at the results of the 2020 election, the model was actually correct to predict Arizona blue, Pennsylvania blue and Florida purple. In the 2020, Arizona and Pennsylvania flipped to blue and Florida was an incredibly close race, but ultimately Trump did win.

It also seems that our model predicted a high number of blue states. This would yield similar results to what the 2016 models were predicting when most were predicting a Hillary Clinton victory, which would end in a

surprise vistory by Trump. While these models might have been off, as we can see with our linear regression model and the 2020 election results, election predicting models are very hard to make accurate but could be helpful predicting reuslts for future years.

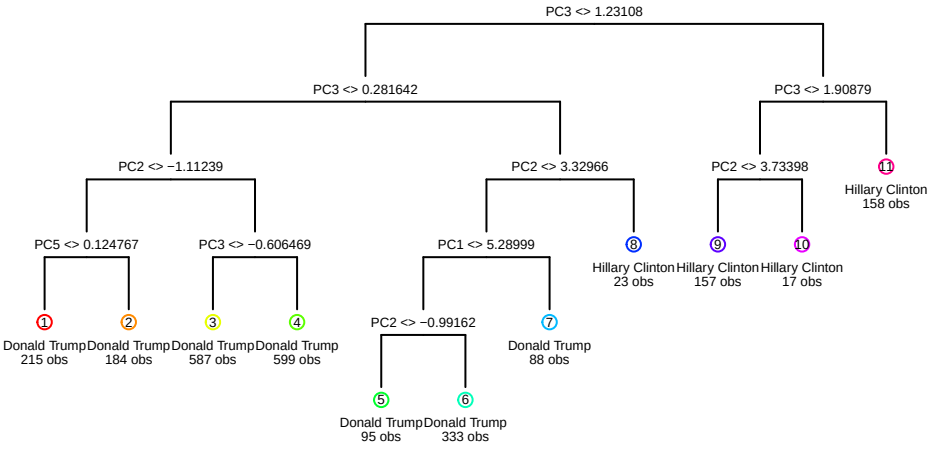
Possible Improvements/Reduction of Errors

The first thought to improve the error rates of these models and the accuracy of the predictions would be to gather more data, and also to gather more accurate data. In the articles from the beginning of the project, it is clear that there was a lot of error with the responses to the surveys about who people were voting for and even if they voted at all. Maybe changing the bias of the question posed in these surveys and making sure the person in question feels that they will not be judged for their answer would help the accuracy of the data. Also, using more advanced statistical tools like Nate Silver did to predict the 2012 election would help. For example using Bayes' Theorem to use conditional probability based on the previous data point would help improve the accuracy of the predictions.

Another observation made during the process of this project was that the classification models were trained on the original data instead of the dimension reduced principal components. A possible way to improve the results would be to use the first 5 principle components instead of the original data to train these 3 models again, just as we did in part 15 with the heirarchical clustering. Therefore we will take our original models further by training them on the first 5 principal components instead of the original data.

Classification with the first 5 Principal Components

Decision Tree



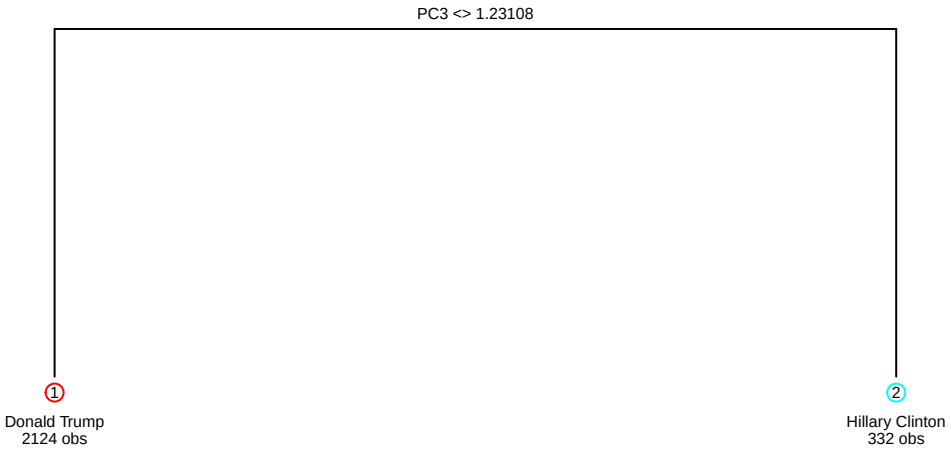


Table 5: Records

	train.error	test.error
tree	0.1013844	0.0764228
logistic	NA	NA
lasso	NA	NA

Already, the decision trees for this model are much more simple than the decision trees produced with the original data. But, the test and train errors for the decision tree trained on the first 5 PCs is higher than the decision tree modeled from the original data.

Logistic Regression (unpenalized)

Once again, the test and train errors for the unpenalized logistic regression model trained on the first 5 PCs is higher than the logistic regression model trained on the original data.

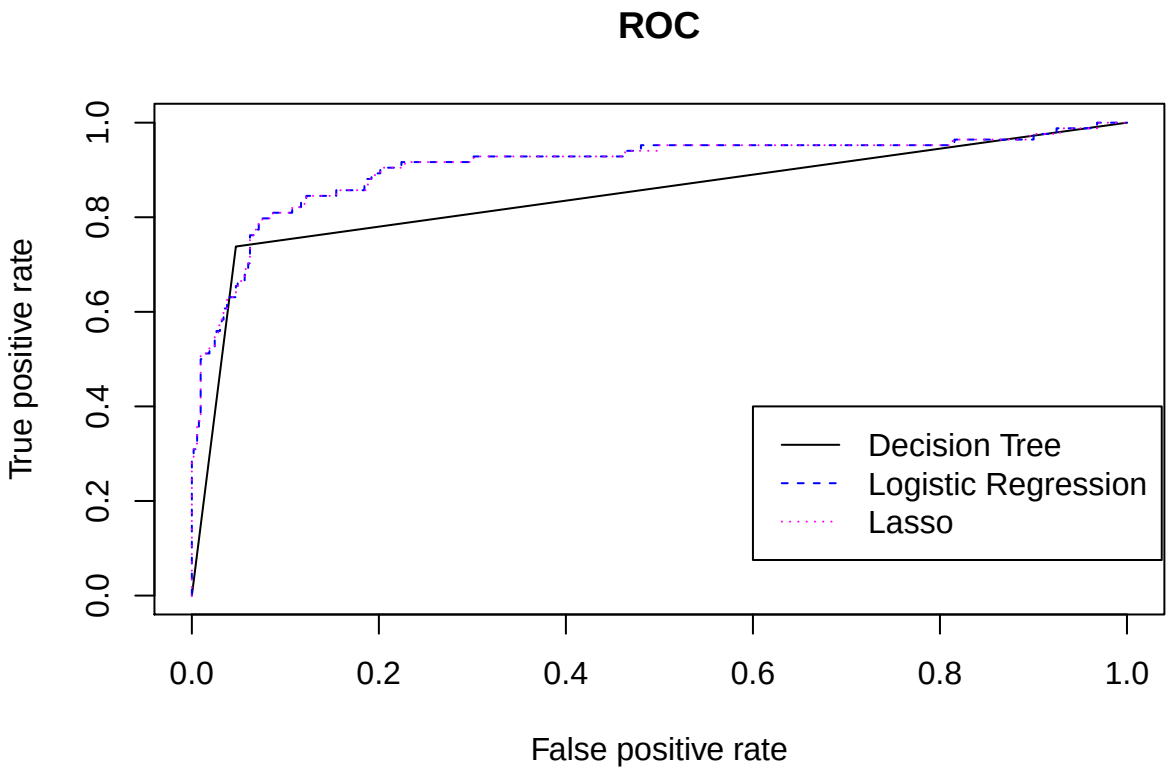
LASSO Logistic Regression (penalized)

Table 6: Records

	train.error	test.error
tree	0.1013844	0.0764228
logistic	0.1021987	0.0845528
lasso	0.1026059	0.0845528

The optimal lambda value is 0.001 in cross validation. The test and train errors for this model are still higher than the errors for the model trained on the original data.

ROC



What is different about the models trained on the first 5 PC, is that the decision tree is now the model with the lowest errors instead of the logistic regression models. In conclusion sticking with the original data will yield less train and test error and will produce a better fit model.