to the electoral college. By realizing that many of the errors in surveys were correlated due to underlying causes, Nate Silver was able to make use of hierarchical models to reduce uncertainty in the census and make accurate forecasts.

**3. What went wrong in 2016? What do you think should be done to make future predictions better?**

There are numerous theories regarding why the election of 2016 was surprising. They all center around a systemic polling anomaly of around 3-6% in the polls. These shadow voters either lied in the polls or did not respond, while voting for Trump. Then, turning out on election day, they delivered to Trump, Florida, Pennsylvania, Michigan, and Wisconsin, all states expected to go to Hillary. Trump won by rightly pointing out how Clinton's support of NAFTA resulted in the deterioration of the industrial economies of the Midwest. Furthermore, although Silver compensated for Hillary's slightly worse odds odds of winning the electoral college, he did not take into account her widespread unpopularity, with upwards of 70% of voters distrusting her.

In the future, we have to take into account a greater degree of polling uncertainty than previously thought. 2016 showed that not only were our elections suceptible to meddling from foreign powers, but also demonstated how volatile they already were. It is clear that polls can be misleading and a good model needs to be more resilient to systemic uncertainty.

# Election data

The meaning of each column in election.raw is clear except fips. The accronym is short for Federal Information Processing Standard.

In our dataset, fips values denote the area (US, state, or county) that each row of data represent. For example, fips value of 6037 denotes Los Angeles County.

**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 dimension of the election dataframe is 18345 rows by 5 columns. The state of Alaska has a Federal Information Processing Standard value of 2000. The state-level summary rows of Alaska are already available when we read the data, so it makes no sense to have duplicate records.

| county | fips | candidate | state | votes |
|---|---|---|---|---|
| Los Angeles County | 6037 | Hillary Clinton | CA | 2464364 |
| Los Angeles County | 6037 | Donald Trump | CA | 769743 |
| Los Angeles County | 6037 | Gary Johnson | CA | 88968 |
| Los Angeles County | 6037 | Jill Stein | CA | 76465 |
| Los Angeles County | 6037 | Gloria La Riva | CA | 21993 |

```
## # A tibble: 18,345 x 5
##    county fips  candidate                state    votes
##    <chr>  <chr> <fct>                    <chr>    <dbl>
##  1 <NA>   US    Donald Trump             US    62984825
##  2 <NA>   US    Hillary Clinton          US    65853516
##  3 <NA>   US    Gary Johnson             US     4489221
##  4 <NA>   US    Jill Stein               US     1429596
##  5 <NA>   US    Evan McMullin            US      510002
```

```
##  6 <NA>   US    Darrell Castle              US      186545
##  7 <NA>   US    Gloria La Riva              US       74117
##  8 <NA>   US    Rocky De La Fuente          US       33010
##  9 <NA>   US    " None of these candidates" US       28863
## 10 <NA>   US    Richard Duncan              US       24235
## # ... with 18,335 more rows
```

```
## [1] 18345     5
```

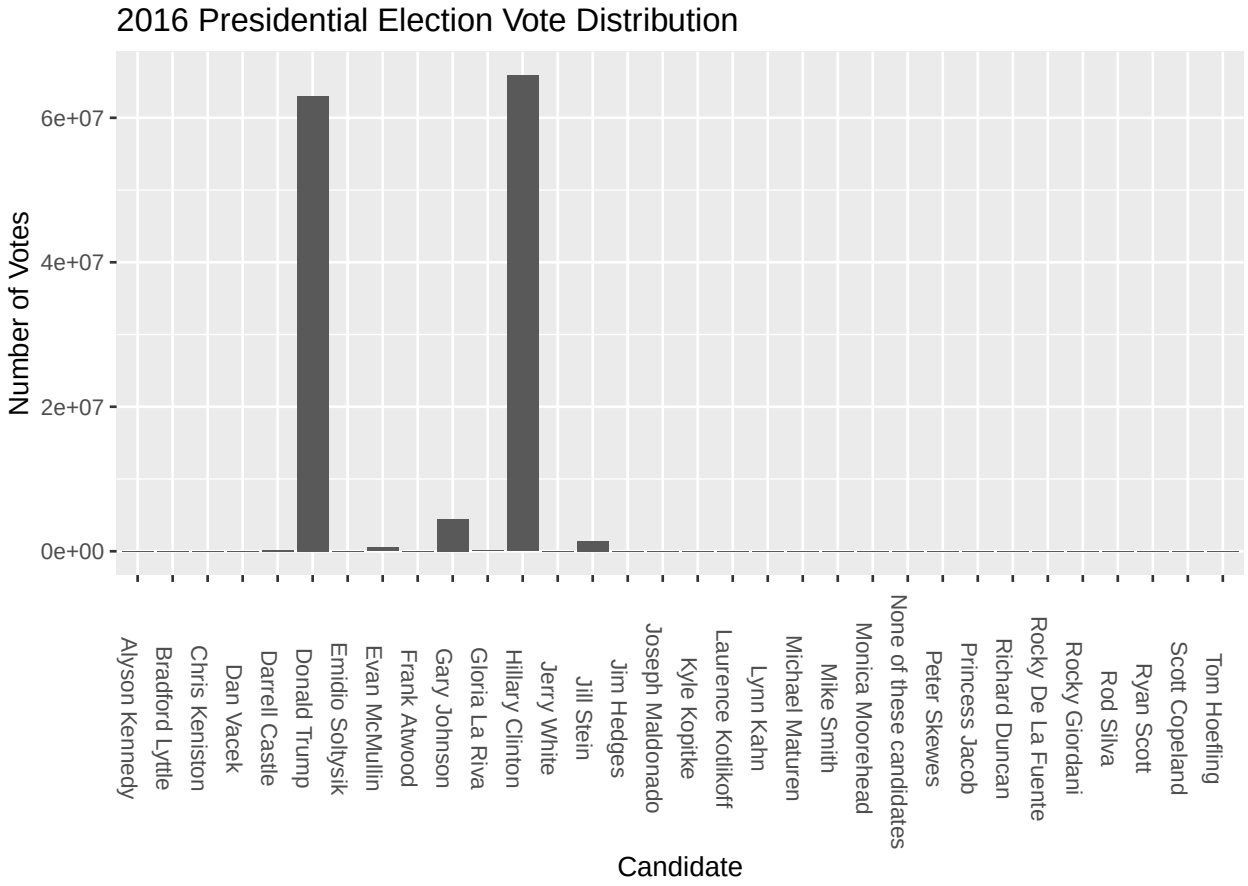
# Data Wrangling

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

```
## [1] TRUE
```

**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. Either way, the results should be clear and legible.**

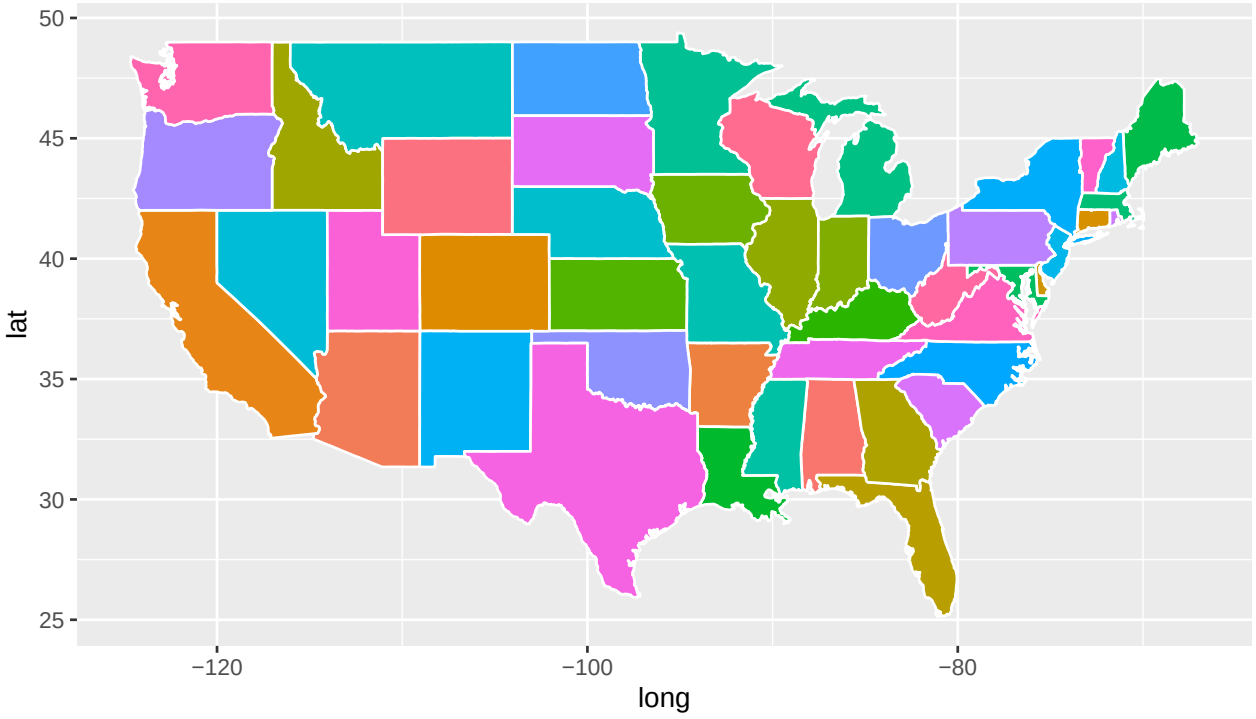
```
## [1] 32
```



**7. Create variables county_winner and state_winner by taking the candidate with the highest proportion of votes. Hint: to create county_winner, start with election, group by fips, compute total votes, and pct=votes/total. Then choose the highest row using top_n (variable state_winner is similar).**

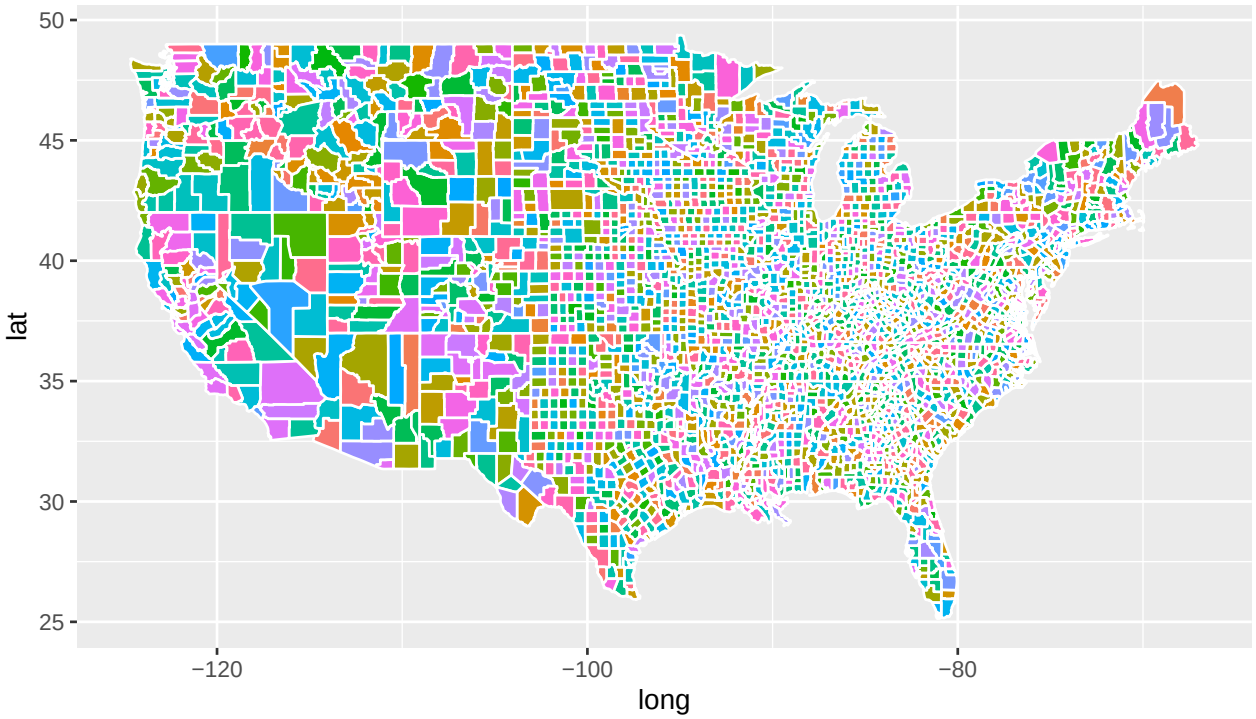
## Visualization

Visualization is crucial for gaining insight and intuition during data mining. We will map out data onto maps. The R package ggplot2 can be used to draw maps, consider the following code.



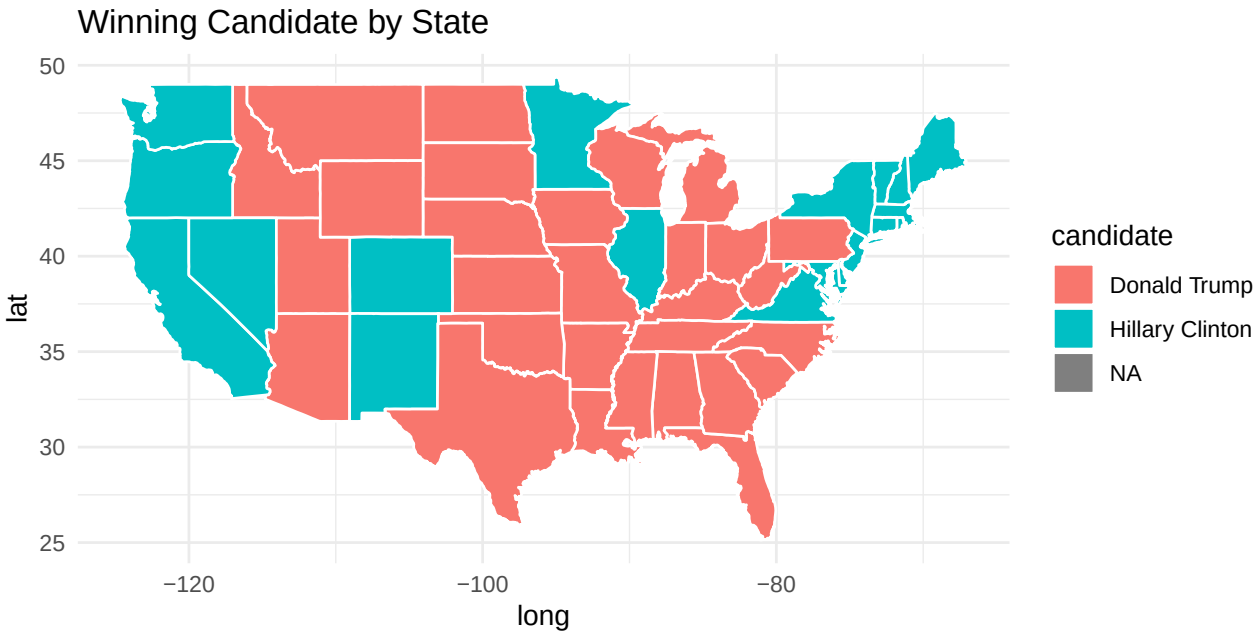

The variable 'states' contains information to draw white polygons, and fill-colors determined by 'region'

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




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

First, combine states variable and state_winner we created earlier using left_join(). Note that left_join() needs to match up values of states to join the tables. A call to left_join() takes all the values from the first table and looks for matches in the second table. If it finds a match, it adds the data from the second table; if not, it adds missing values.

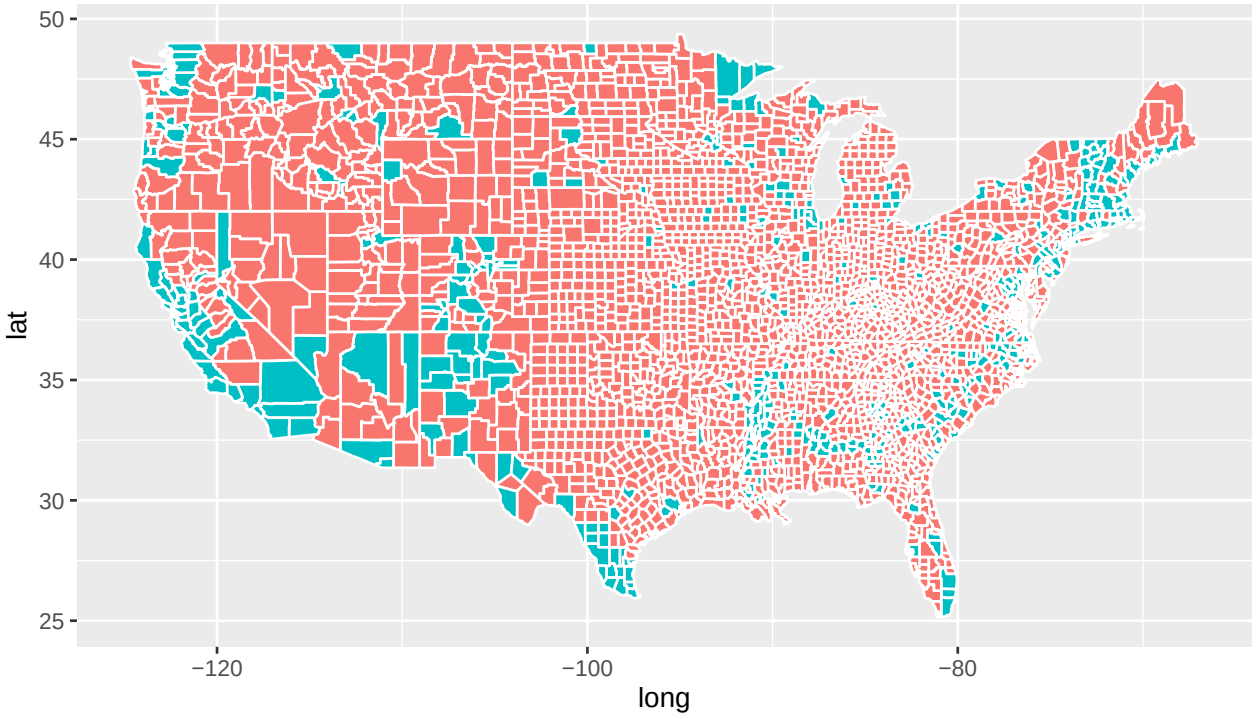



A county-level map of the United States was created so that each county is colored by the winner of that

county. Donald Trump is represented by the red counties and Hillary Clinton is represented by the blue counties.

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

Split the 'polyname' column to 'region' and 'subregion'. Use 'left_join()' combine 'county.fips' into 'county'. Also, 'left_join()' previously created variable 'county_winner'. Your figure will look similar to county-level NY Times map.

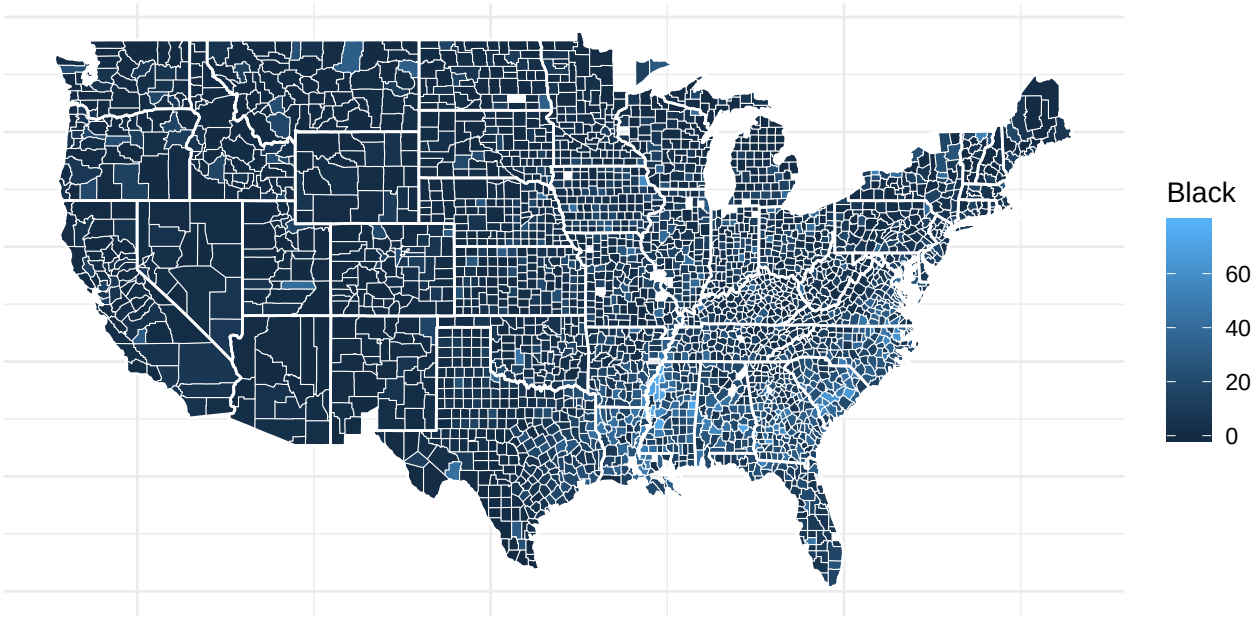



**11. Create a visualization of your choice using census data.**

Many exit polls noted that demographics played a big role in the election. Use the washington post article and R graph gallery for ideas and inspiration.

Here in our visualization, we have a depiction of the variation in ethnicity, which is relevant for election prediction since African American voters overwhelmingly tend to vote Democratic.

Percentage Black by County




**12. The 'census' data contains high resolution information (more fine-grained than county-level). In this problem, we aggregate the information into county-level data by computing 'TotalPop'-weighted average of each attribte for each county. Create the following variables:**

Clean census data census.del: start with census, filter out any rows with missing values, convert {Men, Employed, Citizen} attributes to percentages (meta data seems to be inaccurate), compute Minority attribute by combining {Hispanic, Black, Native, Asian, Pacific}, remove these variables after creating Minority, remove {Walk, PublicWork, Construction}. Many columns seem to be related, and, if a set that adds up to 100%, one column will be deleted.

Sub-county census data, census.subct: start with census.del from above, group_by() two attributes {State, County}, use add_tally() to compute CountyTotal. Also, compute the weight by TotalPop/CountyTotal.

County census data, census.ct: start with census.subct, use summarize_at() to compute weighted sum

Print few rows of census.ct:

| State | County | Men | Citizen | Income | IncomeErr | IncomePerCap | IncomePerCapErr | Poverty | C |
|---|---|---|---|---|---|---|---|---|---|
| Alabama | Autauga | 48.43266 | 73.74912 | 51696.29 | 7771.009 | 24974.50 | 3433.674 | 12.91231 | |
| Alabama | Baldwin | 48.84866 | 75.69406 | 51074.36 | 8745.050 | 27316.84 | 3803.718 | 13.42423 | |
| Alabama | Barbour | 53.82816 | 76.91222 | 32959.30 | 6031.065 | 16824.22 | 2430.189 | 26.50563 | |
| Alabama | Bibb | 53.41090 | 77.39781 | 38886.63 | 5662.358 | 18430.99 | 3073.599 | 16.60375 | |
| Alabama | Blount | 49.40565 | 73.37550 | 46237.97 | 8695.786 | 20532.27 | 2052.055 | 16.72152 | |
| Alabama | Bullock | 53.00618 | 75.45420 | 33292.69 | 9000.345 | 17579.57 | 3110.645 | 24.50260 | |

## Dimensionality Reduction

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

Save the first two principle components PC1 and PC2 into a two-column data frame, call it ct.pc and subct.pc, respectively. Discuss whether you chose to center and scale the features before running PCA and the reasons for your choice. What are the three features with the largest absolute values of the first principal component? Which features have opposite signs and what does that mean about the correaltion between these features?

Our model uses Principal Component analysis for county and sub-county level data. Before running PCA, we normalize and clean the data. It is important to clean the data because otherwise, PCA can lead us towards erroneous conclusions by exacerbating the anomalies in the covariance matrix of the original variables. Therefore it is neccesary to center and scale the features before running PCA, since otherwise features with larger magnitudes will dominate. This enables us to make sure we have more accurate values for the variances.

The first principal component is IncomePerCap (Income per Capita) whereas the second is SelfEmployed in county-level data. In sub county-level data, PC1 is IncomePerCap and PC2 is Drive.

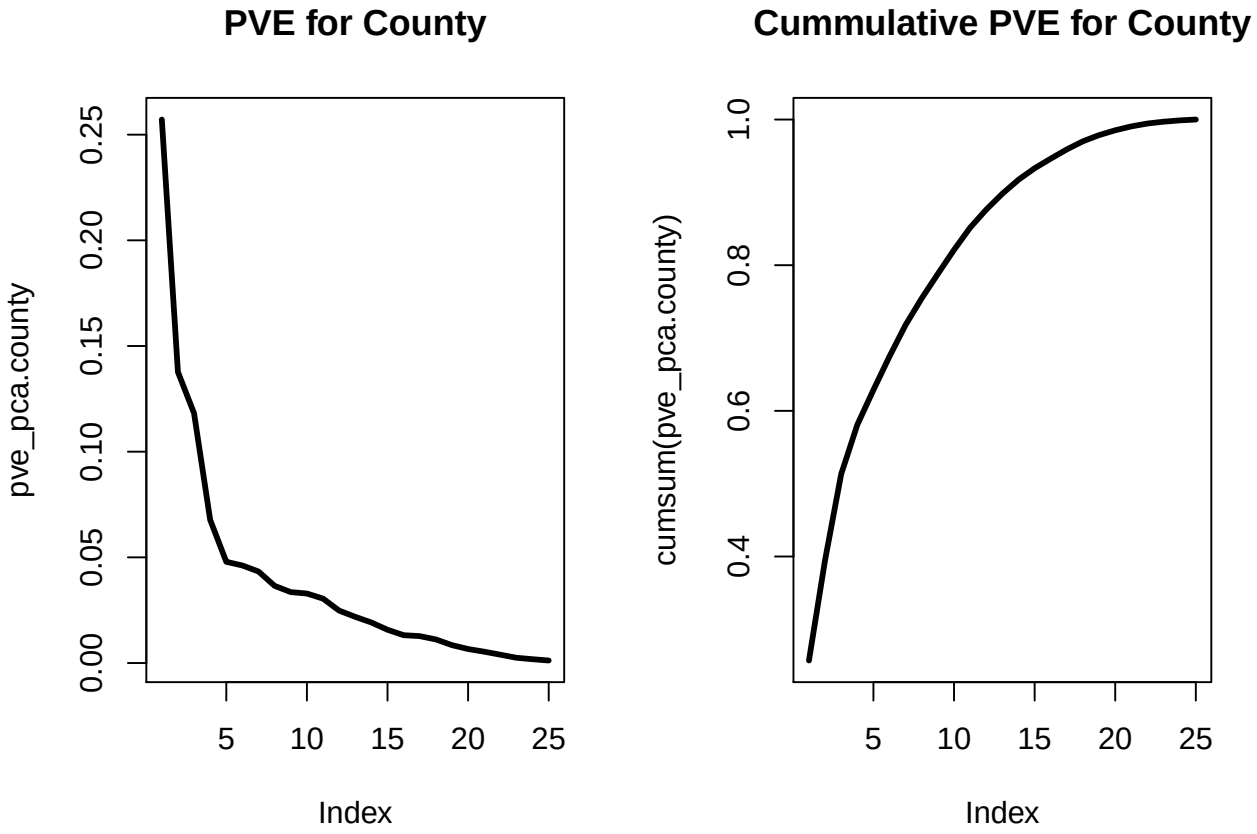
The largest absolute values for the first principal component occur in IncomePerCap, Income, and ChildPoverty.

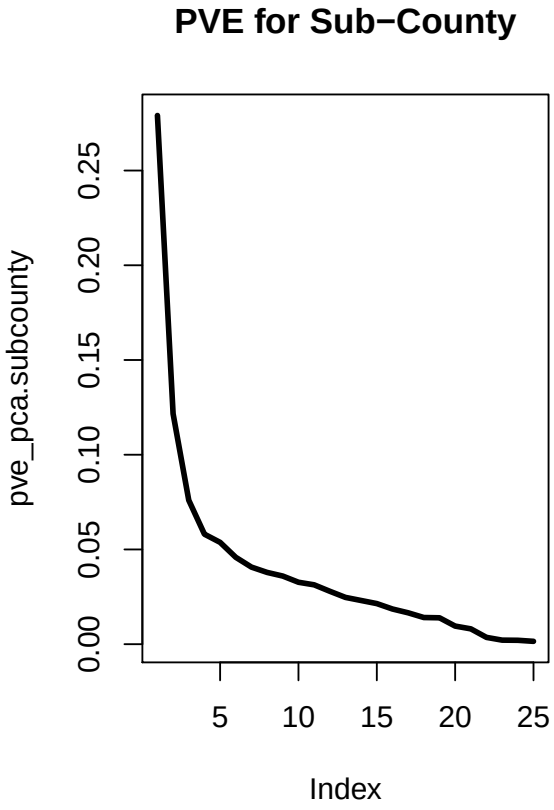
Although principal components are meant to be orthogonal, if we look at 2 different features that have opposite signs this means that the components are negatively correlated. For example, 'Drive' and 'Work From Home' have opposite signs as do 'Employed' and 'Poverty'. This makes sense since people who drive do not work from home. Similar reasoning applies to Employed/Poverty and other components with opposite signs. We discovered these relationships from biplots, however they have not been included as the large size of the dataset causes problems with knitting. However the code has been left commented out in the attached R markdown file

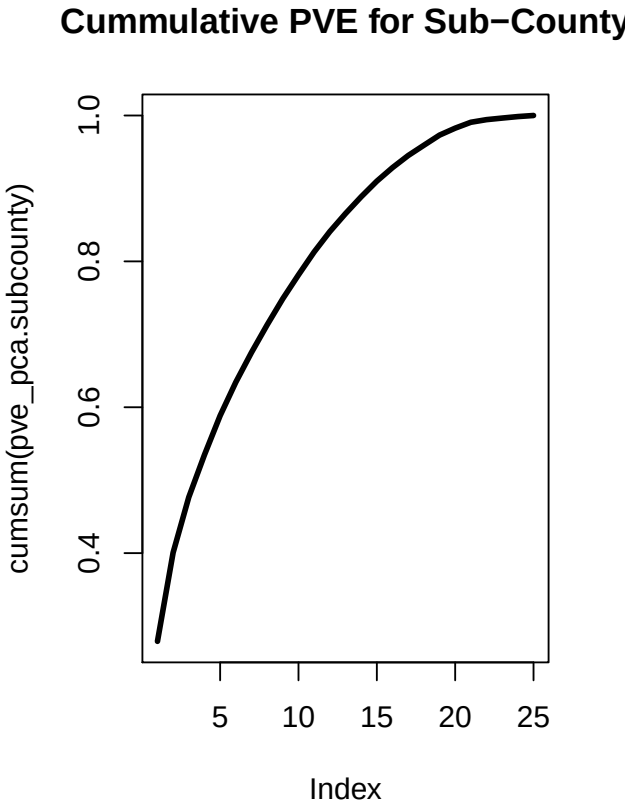
**14. Determine the minimum number of PCs needed to capture 90% of the variance for both the county and sub-county analyses.**

Plot proportion of variance explained (PVE) and cumulative PVE for both county and sub-county analyses.

PVE and Cumulative PVE plots attached for PCA. Application of the PCA algorithm reveals that 90% of the variance in the data for the counties can be explained by 14 principal components whereas for the sub-county level data, 90% of the variance can be explained by means of 15 principal components.






```
## [1] 15
```

```
## [1] 14
```

# Clustering

**15. With 'census.ct' perform hierachical clustering with complete linkage.**

Cut the tree to partition the observations into 10 clusters. Re-run the hierarchical clustering algorithm using the first 5 principal components of ct.pc as inputs instead of the originald features. Compare and contrast the results. For both approaches investigate the cluster that contains San Mateo County. Which approach seemed to put San Mateo County in a more appropriate clusters? Comment on what you observe and discuss possible explanations for these observations.

```
## clust.county
##    1    2    3    4    5    6    7    8    9   10
## 2398  739    6    7    5   45    1   10    3    4
```

```
## clust15_county
##    1    2    3    4    5    6    7    8    9   10
##  621  154 1208  167  867  106   35   38   13    9
```

```
##          pca
## original    1    2    3    4    5    6    7    8    9   10
##        1   350  154 1195    8  583   75   33    0    0    0
##        2   251    0   11  153  281   20    0    5   11    7
##        3     0    0    0    2    2    2    0    0    0    0
##        4     0    0    0    0    0    6    1    0    0    0
##        5     5    0    0    0    0    0    0    0    0    0
```

```
##        6    12     0     1     0     0     0     0    31     1     0
##        7     1     0     0     0     0     0     0     0     0     0
##        8     0     0     0     4     0     3     1     0     0     2
##        9     0     0     0     0     1     0     0     1     1     0
##        10    2     0     1     0     0     0     0     1     0     0
```

| State | County | Men | Citizen | Income | IncomeErr | IncomePerCap | IncomePerCapErr | Pov |
|---|---|---|---|---|---|---|---|---|
| California | Alameda | 49.00514 | 64.73888 | 83129.49 | 12634.54 | 37299.07 | 4705.082 | 12.714 |
| California | Contra Costa | 48.83284 | 65.64452 | 89623.17 | 13784.88 | 39265.13 | 4964.571 | 10.884 |
| California | Marin | 48.27963 | 70.02243 | 98924.65 | 17537.96 | 60992.69 | 9349.889 | 8.330 |
| California | San Francisco | 50.89265 | 73.58313 | 85425.17 | 14863.15 | 52230.87 | 7837.429 | 13.351 |
| California | San Mateo | 49.19773 | 64.20050 | 100369.92 | 16123.02 | 47881.29 | 6115.552 | 8.011 |
| California | Santa Clara | 50.26387 | 60.56144 | 100743.85 | 15214.63 | 43879.60 | 5480.027 | 9.747 |

| State | County | Men | Citizen | Income | IncomeErr | IncomePerCap | IncomePerCapErr | Pov |
|---|---|---|---|---|---|---|---|---|
| California | Marin | 48.27963 | 70.02243 | 98924.65 | 17537.96 | 60992.69 | 9349.889 | 8.330 |
| California | San Francisco | 50.89265 | 73.58313 | 85425.17 | 14863.15 | 52230.87 | 7837.429 | 13.351 |
| California | San Mateo | 49.19773 | 64.20050 | 100369.92 | 16123.02 | 47881.29 | 6115.552 | 8.011 |
| California | Santa Clara | 50.26387 | 60.56144 | 100743.85 | 15214.63 | 43879.60 | 5480.027 | 9.747 |
| Colorado | Douglas | 49.62603 | 68.20415 | 107492.55 | 12492.38 | 45499.77 | 5241.097 | 3.972 |
| Colorado | Pitkin | 53.01378 | 75.98163 | 72835.73 | 19605.62 | 55518.70 | 15743.782 | 9.853 |

The first approach worked better and seemed to put San Mateo County in the more appropriate cluster. When using the original features, the results showed that the observation that contained San Mateo was surrounded by other counties in California, which is also visually represented on the New York Times map. The second approach still grouped San Mateo with other counties in California, but those specific counties were not as close in location to San Mateo, and did not include all of the California counties either.

From my personal belief the reason to why clustering worked so well with the original feature rather than the first five PCA. Due to the fact that PC transforms the variables so that they are linearly uncorrelated, in doing so makes the data harder to cluster. On the other hand, working with the original features we have accounted for possibly correlated variables. This will allow us to have better clustering results.

# Classification

In order to train classification models, we need to combine county_winner and census.ct data. This seemingly straightforward task is harder than it sounds. Following code makes necessary changes to merge them into election.cl for classification.

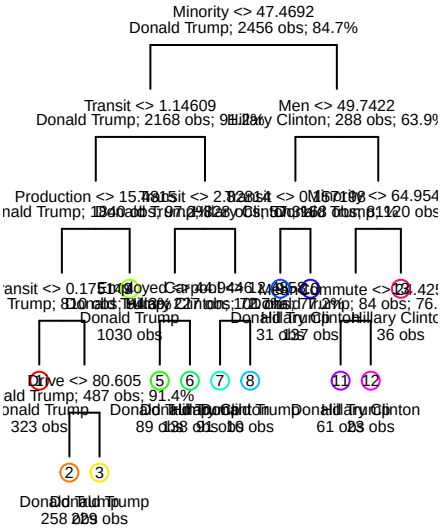
Using the following code, partition data into 80% training and 20% testing:

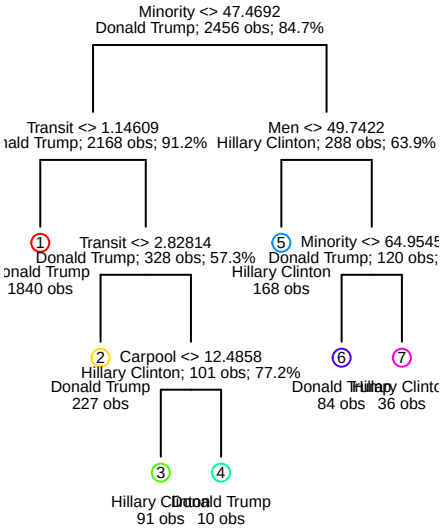
Using the following code, define 10 cross-validation folds:

Using the following error rate function:

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

Prune tree to minimize misclassification error. Be sure to use the folds from above for cross-validation. Visualize the trees before and after pruning. Save training and test errors to records variable. Intepret and discuss the results of the decision tree analysis. Use this plot to tell a story about voting behavior in the US.




The first split along our greedy tree algorithm is for Transit. Although it is not easy to why 'Transit' should be such an important split, it ultimately does not matter as secondary factors along both branches of the decision tree ultimately end up decisively deciding how voters will vote. Furthermore, although we are not certain why Transit is an important predictor, we are not unusually surprised by this result since 'Drive' was the second principal component for sub-county level data.

It is likely that how people commute to work is correlated with a number of other factors. Whether a voter is in a minority demographic, their Employment status, or Income. We can see this very clearly in the decision tree. Where minority voters will not tend to vote for Trump, but very rich voters will. In the right and left branches of the decision tree, we see that Minority voters tend not to prefer Trump, whereas poorer or unemployed voters do.

This clearly shows that Trump was able to successfully capitalize on the political situation in America, exploiting the plight and joblessness of millions and the hope they had for the return of manufacturing jobs. A key benefit of decision trees is the advantage in interpretability, as we have shown by outlining how income and ethnicity demographic have a clear predictive efficacy with regards to voter preference.

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

Save training and test errors to records variable. What are the significant variables? Are the consistent with what you saw in decision tree analysis? Interpret the meaning of a couple of the significant coefficients in terms of a unit change in the variables.

```
##       FamilyWork          Service     Professional          Carpool
##     7.073250e-01     3.277930e-01     2.384782e-01     2.297207e-01
##          Transit         Employed            Drive       WorkAtHome
##     2.102035e-01     2.052921e-01     2.006382e-01     1.845355e-01
##     Unemployment       Production         Minority          Citizen
##     1.760323e-01     1.492162e-01     1.253507e-01     1.005932e-01
##      OtherTransp      PrivateWork           Office      MeanCommute
##     8.672473e-02     7.326522e-02     7.101689e-02     4.480878e-02
##              Men     SelfEmployed          Poverty     ChildPoverty
##     3.583783e-02     2.933280e-02     2.906057e-02     1.190457e-02
##     IncomePerCap IncomePerCapErr        IncomeErr           Income
##     1.715070e-04     1.532909e-04     8.219987e-05     4.448899e-05
##      CountyTotal
```

```
##    3.282166e-07
```

```
##         Service    Professional         Transit        Employed
##    3.277930e-01    2.384782e-01    2.102035e-01    2.052921e-01
##    Unemployment      Production        Minority         Citizen
##    1.760323e-01    1.492162e-01    1.253507e-01    1.005932e-01
##     PrivateWork          Office     MeanCommute             Men
##    7.326522e-02    7.101689e-02    4.480878e-02    3.583783e-02
##    SelfEmployed         Poverty    ChildPoverty    IncomePerCap
##    2.933280e-02    2.906057e-02    1.190457e-02    1.715070e-04
##     CountyTotal          Income       IncomeErr IncomePerCapErr
##    3.282166e-07   -4.448899e-05   -8.219987e-05   -1.532909e-04
##     OtherTransp      WorkAtHome           Drive         Carpool
##   -8.672473e-02   -1.845355e-01   -2.006382e-01   -2.297207e-01
##      FamilyWork
##   -7.073250e-01
```

```
##  [1] "Citizen"      "IncomePerCap" "Professional" "Service"
##  [5] "Production"   "Drive"        "Carpool"      "Transit"
##  [9] "WorkAtHome"   "Employed"     "PrivateWork"  "Unemployment"
## [13] "Minority"
```

```
## [1] NA
```

```
## [1] NA
```

The significant variables are the following ::

Men,Citizen,Income,IncomePerCap,Professional,Service,Production,Drive,Carpool,WorkAtHome,Employed,PrivateWork,Famil
and they were found by selecting variables with a p-value less than 0.05

Our calculated training error for the logistic regression model was .07043974 and the test error was 0.06829268. We see that the training error increased and the test error decreased by about 25%, when compared against decision tree errors, showing that the logistic regression model performed better.

**18. You may notice that you get a warning 'glm.fit: fitted probabilities numerically 0 or 1 occured'**

As we discussed in class, this is an indication that we have perfect separation (some linear combination of variables perfectly predicts the winner). This is usually a sign that we are overfitting. One way to control overfitting in logistic regression is through regularization. Use the cv.glmnet function from the glmnet library to run K-fold cross validation and select the best regularization parameter for the logistic regression with LASSO penalty. Reminder: set alpha=1 to run LASSO regression, set lambda = c(1, 5, 10, 50) * 1e-4 in cv.glmnet() function to set pre-defined candidate values for the tuning parameter $\lambda$. This is because the default candidate values of $\lambda$ in cv.glmnet() is relatively too large for our dataset thus we use pre-defined candidate values. What is the optimal value of $\lambda$ in cross validation? What are the non-zero coefficients in the LASSO regression for the optimal value of $\lambda$? How do they compare to the unpenalized logistic regression? Save training and test errors to the records variable.

```
## [1] 0.001
```

```
## 26 x 1 sparse Matrix of class "dgCMatrix"
##                               1
## (Intercept)     -3.006975e+01
## Men              .
## Citizen          1.160501e-01
## Income          -5.486996e-06
## IncomeErr       -7.460712e-05
```

```
## IncomePerCap     7.771447e-05
## IncomePerCapErr -1.826500e-05
## Poverty          3.961651e-02
## ChildPoverty     8.961000e-03
## Professional     1.892848e-01
## Service          2.716122e-01
## Office           2.179355e-02
## Production       9.666756e-02
## Drive           -1.452137e-01
## Carpool         -1.686381e-01
## Transit          2.197083e-01
## OtherTransp     -1.817940e-02
## WorkAtHome      -9.321582e-02
## MeanCommute      1.767254e-02
## Employed         1.845617e-01
## PrivateWork      6.797187e-02
## SelfEmployed     .
## FamilyWork      -5.432461e-01
## Unemployment     1.592338e-01
## Minority         1.142745e-01
## CountyTotal      3.784059e-07
```

Lasso regression returns 24 non-zero coefficients, compared to the 25 non-zero coefficients returned by regular logistic regression because lasso regression dropped 'Child Poverty'. Lasso also shrunk the coefficients so that they are always closer to zero because of the applied tuning parameter, $\lambda$.

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

Display them on the same plot. Based on your classification results, discuss the pros and cons of the various methods. Are the different classifiers more appropriate for answering different kinds of questions about the election?

*Note : Although the code runs correctly by itself, I am having a compatibility issue between my operating system and ROCR leading to problems with knitting. I have simply added the image produced by my own computer on the following page and set echo=FALSE for the accomanying segment of code.*

After computing the test error for the decision trees, logistic regression, and lasso regression, we see that logistic regression returns the lowest test error. Making it the best classifier for accurately predicting election winner results. Although this was the best model for classification, there are other pros and cons of using other models. The decision tree model's results were easily readable, but the non-parametric method predicted a decision boundary that was too overfitting, preventing the model from fitting the data well due to high variance. Whereas, Lasso logistic regression was beneficial in reducing the variance of the model's coefficients while increasing bias, and shrunk them to values that closely modeled the linear regression.

# Taking it further

**20. This is an open question. Interpret and discuss any overall insights gained in this analysis and possible explanations.**

Use any tools at your disposal to make your case: visualize errors on the map, discuss what does/doesn't seems reasonable based on your understanding of these methods, propose possible directions (collecting additional data, domain knowledge, etc). In addition, propose and tackle at least one more interesting

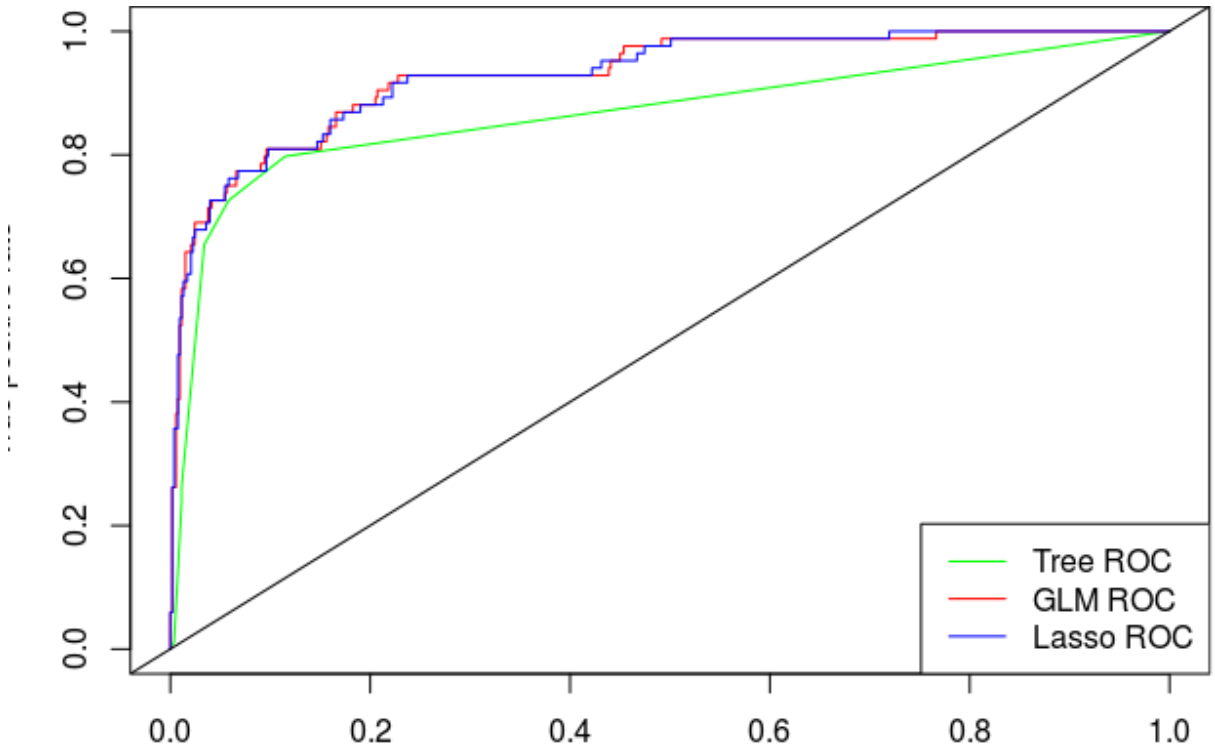



Figure 1: ROC curves

question. Creative and thoughtful analyses will be rewarded! This part will be worth up to a 20% of your final project grade!

*Data preprocessing: we aggregated sub-county level data before performing classification. Would classification at the sub-county level before determining the winner perform better? What implicit assumptions are we making?*

Classification at the sub-county level, if done correctly would improve our model. However, we have to acknowledge 2 different assumptions. First, we have the underlying IID assumption which means that the different voters vote according to the same underlying distribution (even though this distribution can be particularly complex). This sounds like a reasonable assumption, as we can consider all voters to be American citizens, however, different states are dominated by widely diverse idealogies so it could equally easily be violated. The second assumption is that the data is symmetric, namely, that it does not matter which order we classify the data. I.e. we can classify according to sub-county then state, or vice versa. This can often be the case if the two categories are independent, however it is not clear that this is the always the case in the data for the US election.

*Exploring additional classification methods: KNN, LDA, QDA, SVM, random forest, boosting etc. (You may research and use methods beyond those covered in this course). How do these compare to logistic regression and the tree method?*

Although Random Forest comes close when we look at the ROC curve. Logistic regression wins out with the highest AUC. This is not unusually surprising, since we have to classify voters into one of just 2 categories in the general election, so the simplicity and fit of logistic regression to the problem makes it the superiour algorithm. However, methods such as SVM and regression may be better if we were attempting to classify a continuous degree of support.

*Conduct an exploratory analysis of the "purple" counties– the counties which the models predict Clinton and Trump were roughly equally likely to win. What is it about these counties that make them hard to predict?*

These counties are hard to predict because they could equally as easily go either way. While states such as California (except with Reagan!) and New York are always blue, other states such as Iowa and Pennsylvania often go either way, historically speaking. Since so many factors can affect what decides an election, and the number of voters is distributed roughly equally amongst the two parties, we have high uncertainty regarding the outcome of the election with respect to these states.

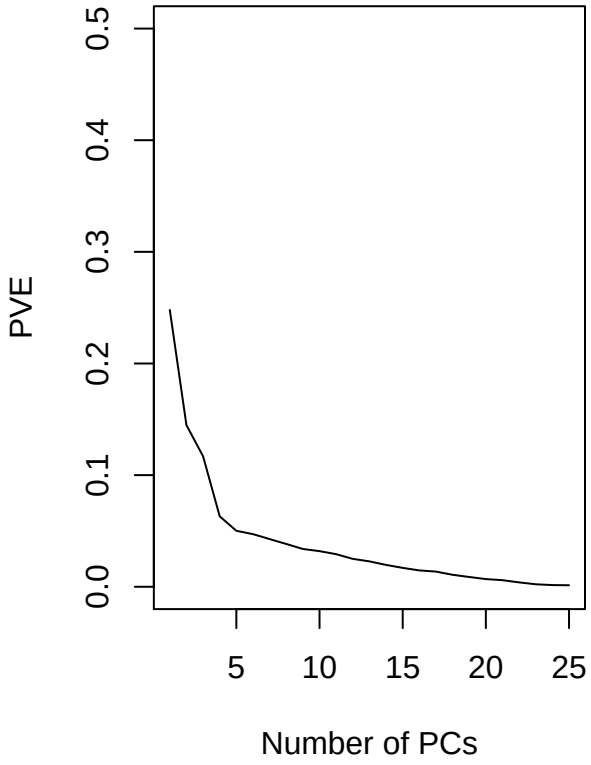
Philosophically speaking, there is a 100% chance that the state will go to one candidate or another, determined by the laws of causality and physics. However, we assign probabilities to these events, because we are attempting to not quantify reality, but in fact, our own state of knowledge. Therefore, when we say that the probability is 50%, we are in effect admitting our own ignorance about all the complexities and parameters that come into play in these states.

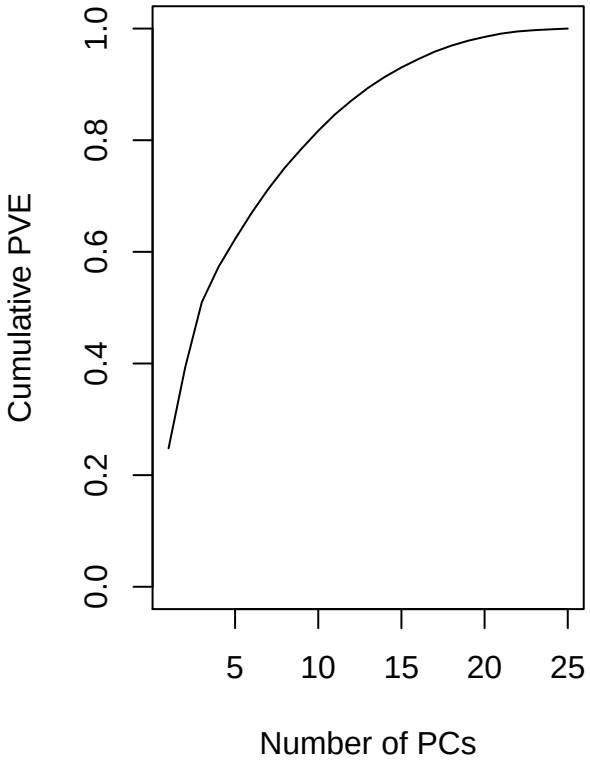
*Instead of using the native attributes (the original features), we can use principal components to create new (and lower dimensional) set of features with which to train a classification model. This sometimes improves classification performance. Compare classifiers trained on the original features with those trained on PCA features.*

We were able to successfully transform the Principal components in order to train a superior model. Although we were able to improve the properties of the classifier according to metrics such as AUC, ultimately, the classifiers still predicted a Clinton win, although the probability was lower than with the naive classifier. We can look at this and conclude that our models did not accurately take into account systemic uncertainty. In reality, Trump voters are not as likely to honestly answer in the polls, therefore, this demographic was systemically underrepresented in polling data, leading to erroneous conclusions in our statistical models.

Table 1: PCA-Transformed Training Set

| candidate | PC1 | PC2 | PC3 | PC4 | PC5 |
|---|---|---|---|---|---|
| Hillary Clinton | 0.6554947 | 2.9094841 | -0.5329601 | -0.1447754 | 0.5653384 |
| Donald Trump | 0.6961327 | 0.6253840 | 3.3493435 | 0.9537902 | 1.0491547 |
| Donald Trump | -1.1732301 | -3.2491812 | 1.2714535 | 0.6971366 | 0.6695406 |
| Donald Trump | 1.9074743 | 0.3555093 | -0.6925285 | -0.0190179 | -0.0698617 |
| Hillary Clinton | -1.9854498 | 1.6625141 | -0.7910009 | -0.9516777 | 0.7964415 |
| Donald Trump | -1.3296210 | 0.6247439 | 0.9118829 | 1.2221642 | -0.5435842 |




| | train.error | test.error |
|---|---|---|
| tree | 0.1062704 | 0.1073171 |
| logistic | 0.1530945 | 0.1447154 |
| lasso | 0.0997557 | 0.0943089 |