# PSTAT131 Final Project

Yunning Chen (5272778), Anqi Jiang (5901582), Anne Lin (5116645)

## Instructions and Expectations

- You are allowed and encouraged to work with two partners on this project. Include your names, perm numbers, and whether you are taking the class for 131 or 231 credit.
- You are welcome to write up a project report in a research paper format – abstract, introduction, methods, results, discussion – as long as you address each of the prompts below. Alternatively, you can use the assignment handout as a template and address each prompt in sequence, much as you would for a homework assignment.
- There should be no raw R *output* in the body of your report! All of your results should be formatted in a professional and visually appealing manner. That means that visualizations should be polished – aesthetically clean, labeled clearly, and sized appropriately within the document you submit, tables should be nicely formatted (see `pander`, `xtable`, and `kable` packages). If you feel you must include raw R output, this should be included in an appendix, not the main body of the document you submit.
- There should be no R *codes* in the body of your report! Use the global chunk option `echo=FALSE` to exclude code from appearing in your document. If you feel it is important to include your codes, they can be put in an appendix.

## Background

The U.S. presidential election in 2012 did not come as a surprise. Some correctly predicted the outcome of the election correctly including Nate Silver, and many speculated about his approach.

Despite the success in 2012, the 2016 presidential election came as a big surprise to many, and it underscored that predicting voter behavior is complicated for many reasons despite the tremendous effort in collecting, analyzing, and understanding many available datasets.

Your final project will be to merge census data with 2016 voting data to analyze the election outcome.

To familiarize yourself with the general problem of predicting election outcomes, read the articles linked above and answer the following questions. Limit your responses to one paragraph for each.

1. What makes voter behavior prediction (and thus election forecasting) a hard problem?

**There are several factors to make the voter behavior prediction (and thus election forecasting) be a hard problem. Firstly, some people will lie or disguise their vote preference during the poll. For example, if someone wanted to vote for Trump in 2016, but he was afraid of being judged by his friends voting for Clinton, it is possible that he will say he doesn't have a preference during the poll. Secondly, polls may only reach certain groups and miss some groups of people. For example, if the poll was sent out online, it is possible that in 2016, most Trump voters didn't participated because they seldom use internet. If the poll was taken in big cities, it misses populations in rural areas, etc. Finally, people may change their decision over time. The outcome may interfered by some factors that can't be predicted quantitatively. For example, economic turbulence, one candidate's particularly successful campaign ad, one**

**candidate's attitude towards race and gender conflicts will all influence the election outcome, but they can't be measured quantitatively.**

2. What was unique to Nate Silver's approach in 2012 that allowed him to achieve good predictions?

**When predicting the probability, Silver looks at the full range of probabilities for predictions in one day, so for the following day he can use the model for the actual support to determine what is the probability that support has shifted to. Then he used hierarchical modeling. Incorporating the full range of probabilities, Silver calculates probability for Obama winning in each state, which will then tell if Obama can win the election. Also, when dealing with the pollster bias, Silver used previous election to estimate the possible bias.**

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

**In 2016, the polling errors are all in the same direction (over-estimate Clinton's winning and under-estimate Trump's winning), which leads to a large error in the prediction based on different polls. To make future predictions better, its better to build models that does not merely correlates voters' intention with the polling results. Incorporate additional kinds of available data such as voter enthusiasm measures, earned media tracking, Google search trends etc. which may also reveal voters' intention.**

# Data

The `project_data.RData` binary file contains three datasets: tract-level 2010 census data, stored as `census`; metadata `census_meta` with variable descriptions and types; and county-level vote tallies from the 2016 election, stored as `election_raw`.

## Election data

Some example rows of the election data are shown below:

| 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 |
| Cook County | 17031 | Hillary Clinton | IL | 1611946 |

The meaning of each column in `election_raw` is self-evident except `fips`. The accronym is short for Federal Information Processing Standard. In this dataset, `fips` values denote the area (nationwide, statewide, or countywide) that each row of data represent.

Nationwide and statewide tallies are included as rows in `election_raw` with `county` values of `NA`. There are two kinds of these summary rows:

- Federal-level summary rows have a `fips` value of `US`.
- State-level summary rows have the state name as the `fips` value.

4. Inspect rows with `fips=2000`. Provide a reason for excluding them. Drop these observations – please write over `election_raw` – and report the data dimensions after removal.

**We first inspect rows with `fips=2000`:**

| county | fips | candidate | state | votes |
|---|---|---|---|---|
| NA | 2000 | Donald Trump | AK | 163387 |

| county | fips | candidate | state | votes |
|---|---|---|---|---|
| NA | 2000 | Hillary Clinton | AK | 116454 |
| NA | 2000 | Gary Johnson | AK | 18725 |
| NA | 2000 | Jill Stein | AK | 5735 |
| NA | 2000 | Darrell Castle | AK | 3866 |
| NA | 2000 | Rocky De La Fuente | AK | 1240 |

**We can see that rows with "fips=2000" are votes in Alaska. The "county" column (which should be the county name) is NA, but, as we referred before, only nationwide and statewide tallies are included as rows in "county" values of NA. The reason for the null value in "county" is that the state Alaska doesn't have counties. Therefore, for "fips = 2000" and "fips = AK", we will have the same but duplicated votes. Hence, we need to exclude these observations to address the duplication which may affect our future prediction. Below is the table of the data dimensions after removal of observation with flip=2000:**

| | Row | Column |
|---|---|---|
| **number** | 18345 | 5 |

**After we removed the observations with flip=2000, the `election_raw` data has 18345 rows(observations) and 5 columns(variables).**

## Census data

The first few rows and columns of the `census` data are shown below.

| CensusTract | State | County | TotalPop | Men | Women |
|---|---|---|---|---|---|
| 1001020100 | Alabama | Autauga | 1948 | 940 | 1008 |
| 1001020200 | Alabama | Autauga | 2156 | 1059 | 1097 |
| 1001020300 | Alabama | Autauga | 2968 | 1364 | 1604 |
| 1001020400 | Alabama | Autauga | 4423 | 2172 | 2251 |
| 1001020500 | Alabama | Autauga | 10763 | 4922 | 5841 |
| 1001020600 | Alabama | Autauga | 3851 | 1787 | 2064 |

Variable descriptions are given in the `metadata` file. The variables shown above are:

| variable | description | type |
|---|---|---|
| CensusTract | Census tract ID | numeric |
| State | State, DC, or Puerto Rico | string |
| County | County or county equivalent | string |
| TotalPop | Total population | numeric |
| Men | Number of men | numeric |
| Women | Number of women | numeric |

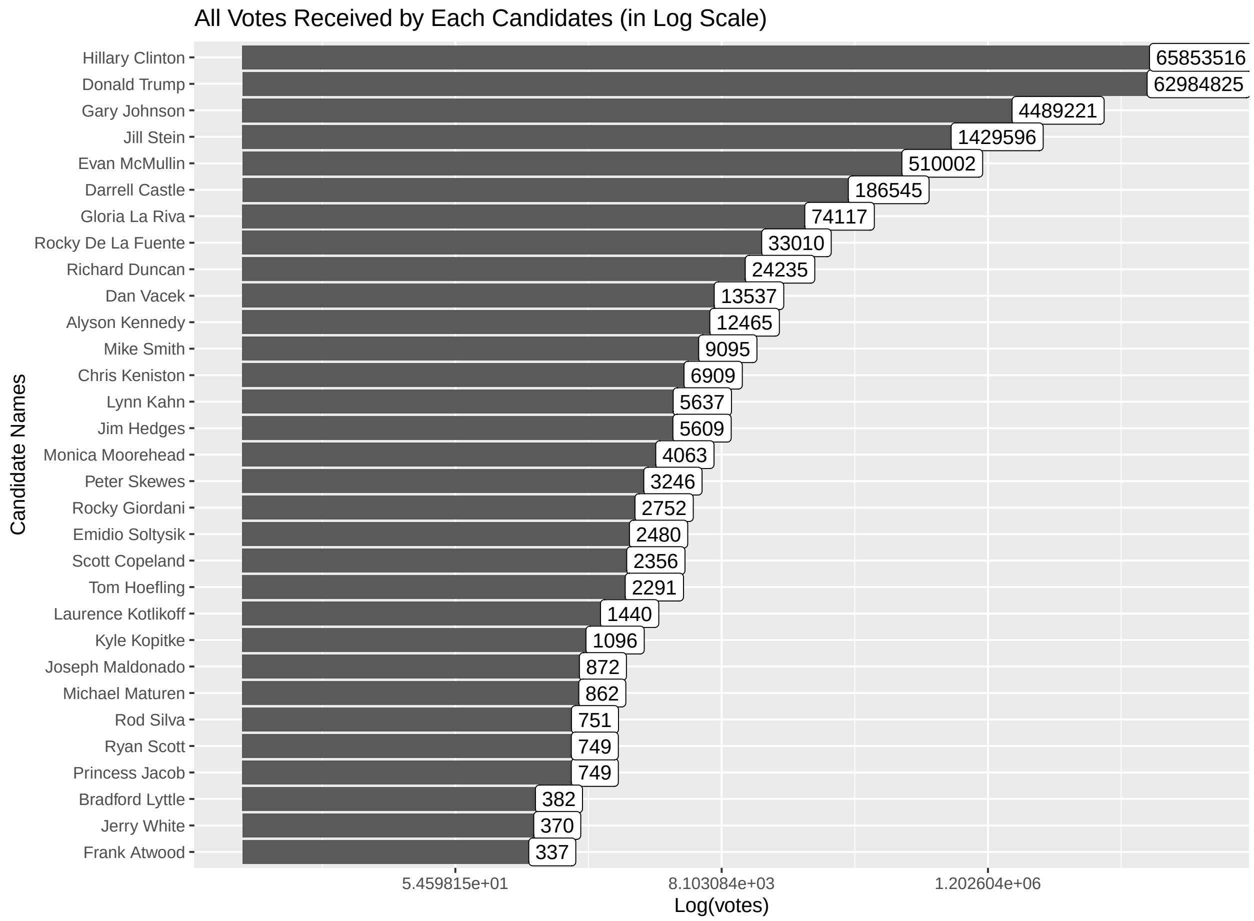
## Data preprocessing

5. Separate the rows of `election_raw` into separate federal-, state-, and county-level data frames:
    - Store federal-level tallies as `election_federal`.
    - Store state-level tallies as `election_state`.
    - Store county-level tallies as `election`. Coerce the `fips` variable to numeric.

**(For full coding detail, see Appendix)**

6. How many named presidential candidates were there in the 2016 election? Draw a bar graph of all votes received by each candidate, and order the candidate names by decreasing vote counts. (You may need to log-transform the vote axis.)

**Here is the bar graph of all votes received by each candidate:**



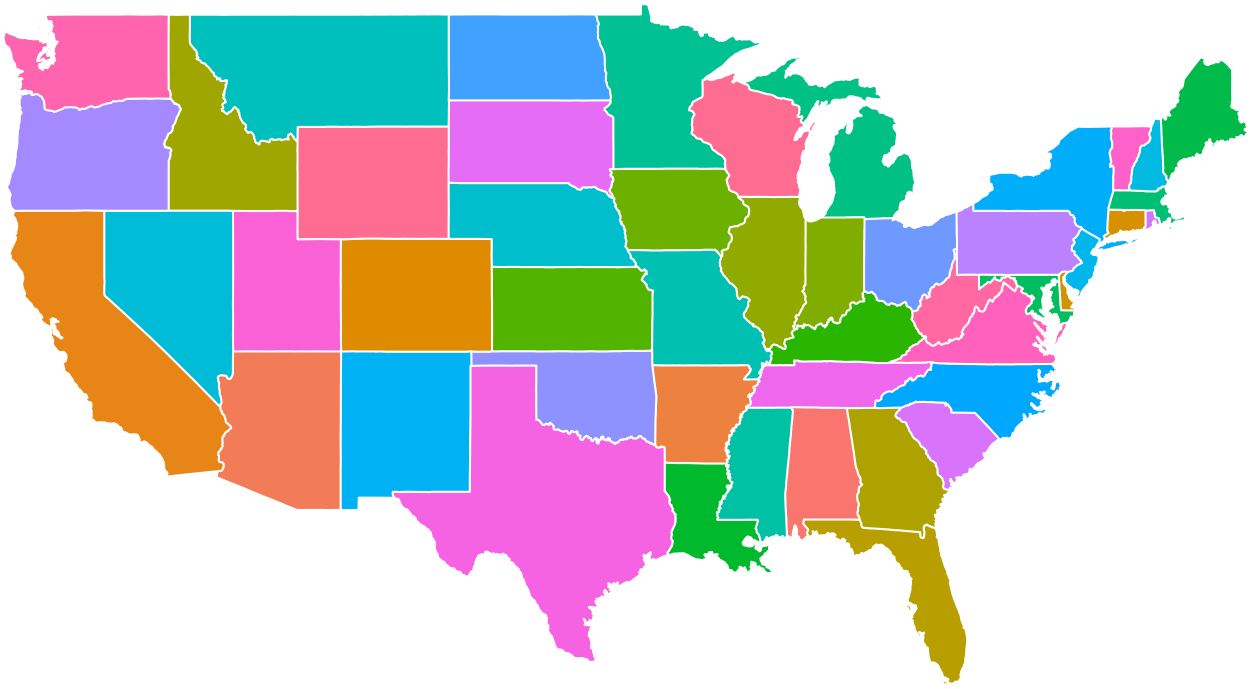
**There were 31 named presidential candidates in the 2016 election. The bar chart above displays all votes received by each candidate (including those not named) on a log scale with labels of the exact votes for each candidate.**

7. Create `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 `slice_max` (variable `state_winner` is similar).)

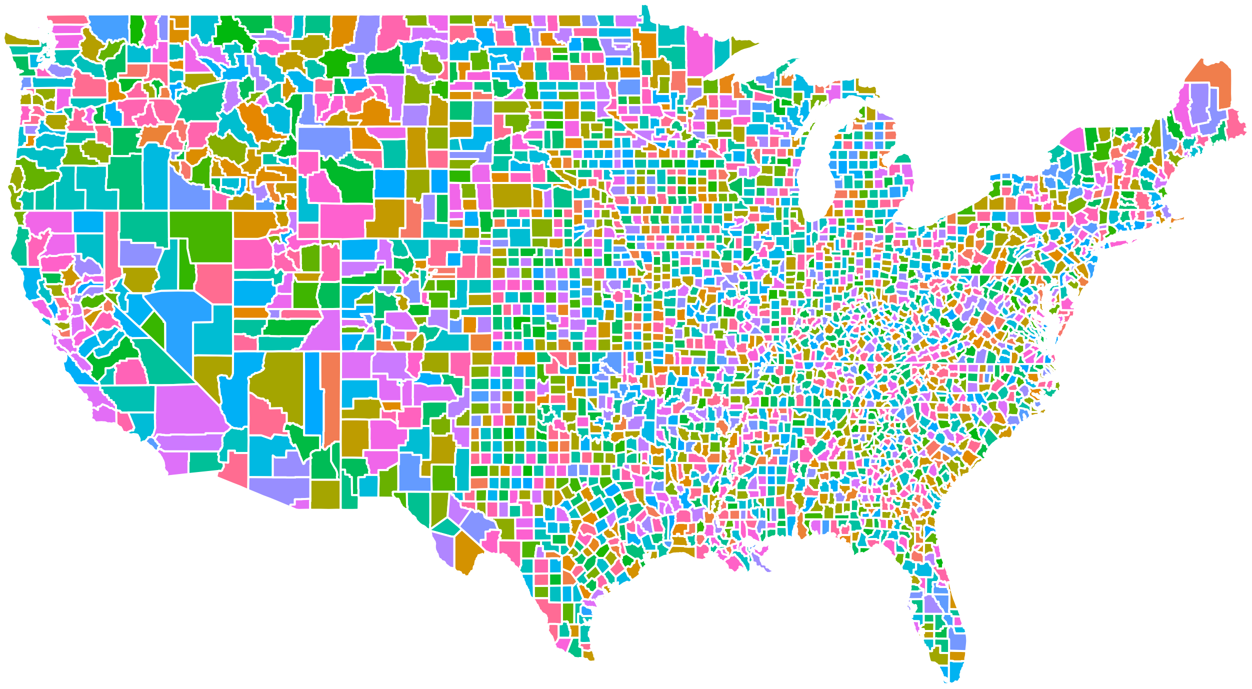
**(For full coding detail, see Appendix)**

## Visualization

Here you'll generate maps of the election data using `ggmap`. The .Rmd file for this document contains codes to generate the following map.



8. Draw a county-level map with `map_data("county")` and color by county.



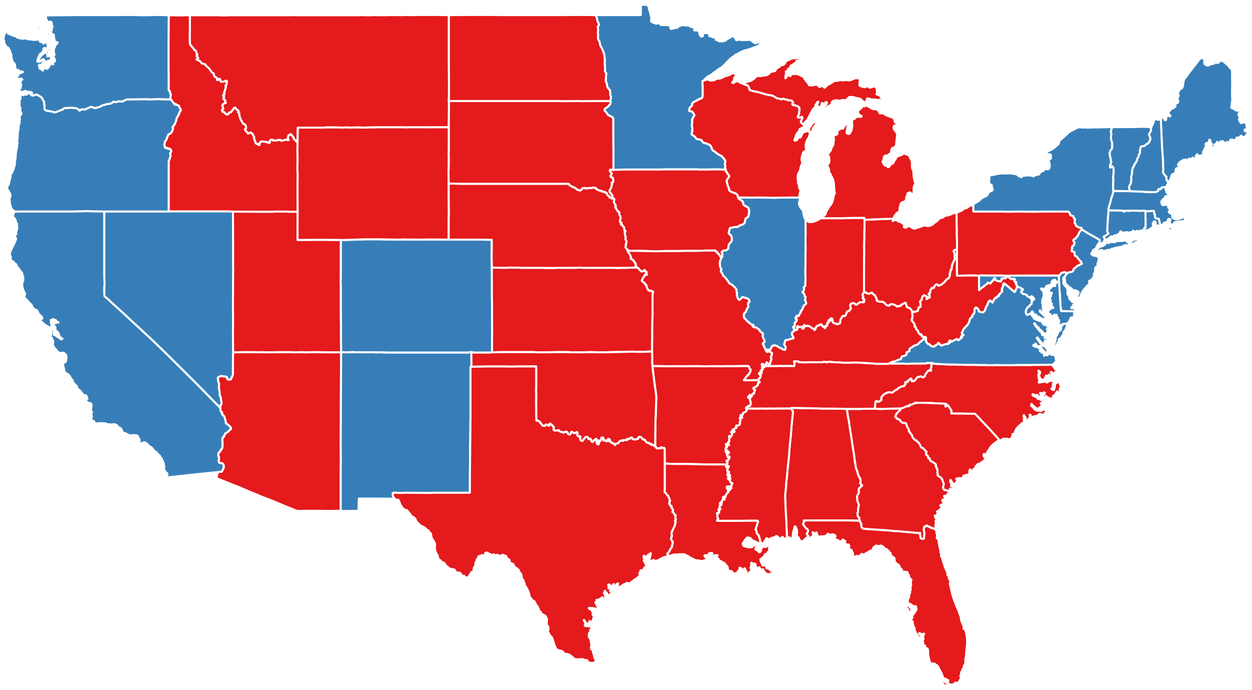
In order to map the winning candidate for each state, the map data (`states`) must be merged with with the election data (`state_winner`).

The function `left_join()` will do the trick, but needs to join the data frames on a variable with values that match. In this case, that variable is the state name, but abbreviations are used in one data frame and the full name is used in the other.

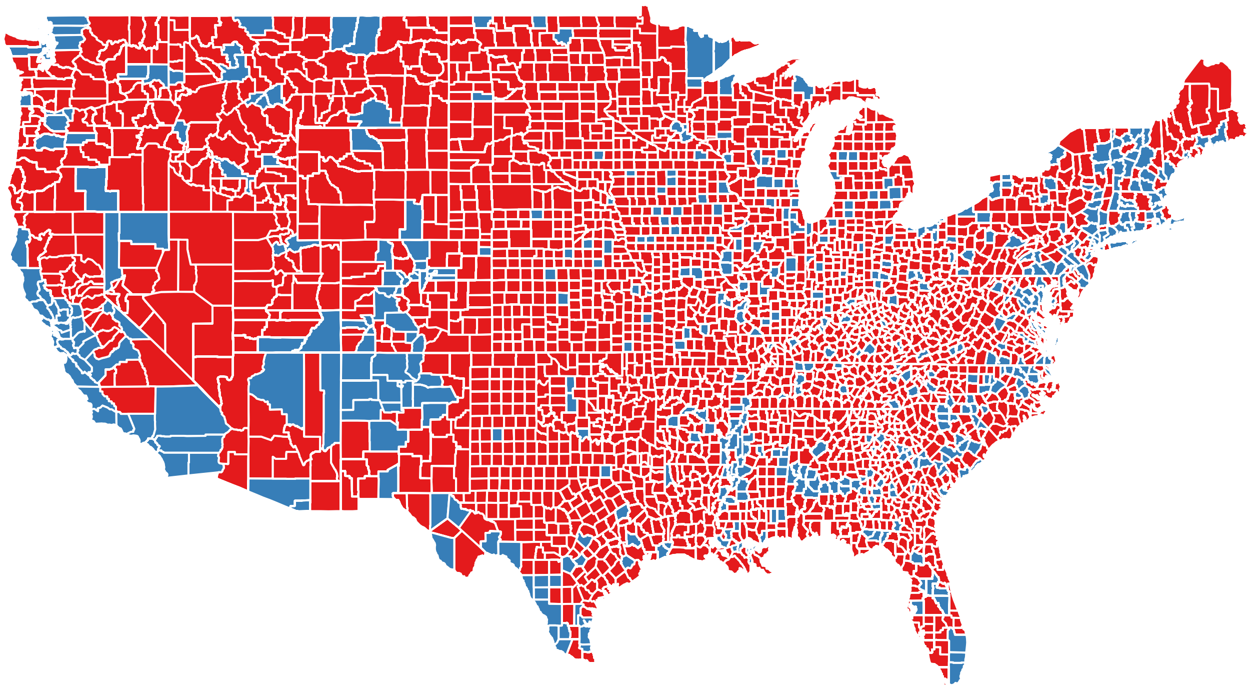
9. Use the following function to create a `fips` variable in the `states` data frame with values that match the `fips` variable in `election_federal`.

Now the data frames can be merged. `left_join(df1, df2)` takes all the rows from `df1` and looks for matches in `df2`. For each match, `left_join()` appends the data from the second table to the matching row in the first; if no matching value is found, it adds missing values.

10. Use `left_join` to merge the tables and use the result to create a map of the election results by state. Your figure will look similar to this state level New York Times map. (Hint: use `scale_fill_brewer(palette="Set1")` for a red-and-blue map.)

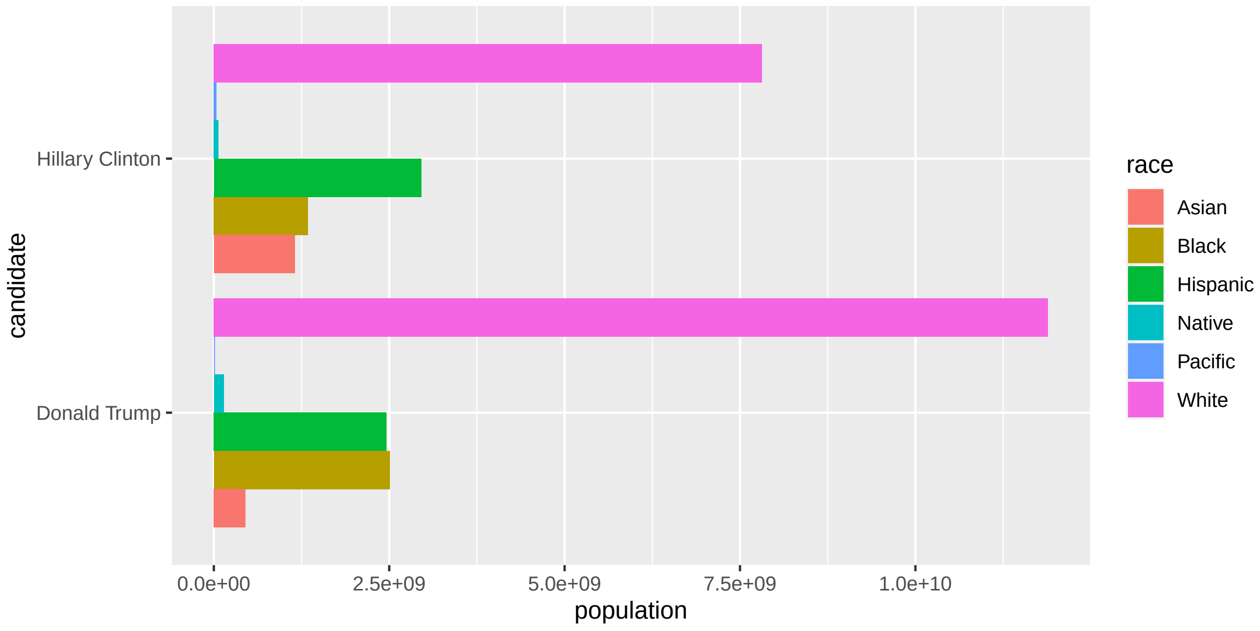


11. Now create a county-level map. The county-level map data does not have a `fips` value, so to create one, use information from `maps::county.fips`: split the `polyname` column to `region` and `subregion` using `tidyr::separate`, and use `left_join()` to combine `county.fips` with the county-level map data. Then construct the map. Your figure will look similar to county-level New York Times map.



12. Create a visualization of your choice using `census` data. Many exit polls noted that demographics played a big role in the election. If you need a starting point, use this Washington Post article and this R graph gallery for ideas and inspiration.

**Here we show a bar plot of total population of people with different races who voted Clinton and Trump:**



**In our graph, the x-axis denotes the total population with different races who voted Clinton and Trump. The y-axis denotes two main candidates: Hillary Clinton and Donald Trump.**

**According to the graph we draw, we can find that there are some differences in voting preferences between each race:**

**White people make up the most of total voting population and Pacific people occupy the least**

**of total voting population. Most strikingly, Donald Trump gains more votes on White people. He also gains more votes among Black, Native people than Hillary Clinton, but not much. Hillary Clinton gains more votes among Asian, Hispanic, and Pacific (slightly) people, most strikingly with Asian.**

**Therefore, we can suppose that the opinions of Donald Trump are more popular among Blacks, Natives, and Whites, and the opinions of Hillary Clinton are more attractive to Asian, Hispanic, and Pacifics.**

13. The `census` data contains high resolution information (more fine-grained than county-level). Aggregate the information into county-level data by computing population-weighted averages of each attribute for each county by carrying out the following steps:

- Clean census data, saving the result as `census_del`:
  - filter out any rows of `census` with missing values;
  - convert `Men`, `Employed`, and `Citizen` to percentages;
  - compute a `Minority` variable by combining `Hispanic`, `Black`, `Native`, `Asian`, `Pacific`, and remove these variables after creating `Minority`; and
  - remove `Walk`, `PublicWork`, and `Construction`.
- Create population weights for sub-county census data, saving the result as `census_subct`:
  - group `census_del` by `State` and `County`;
  - use `add_tally()` to compute `CountyPop`;
  - compute the population weight as `TotalPop/CountyTotal`;
  - adjust all quantitative variables by multiplying by the population weights.
- Aggregate census data to county level, `census_ct`: group the sub-county data `census_subct` by state and county and compute popluation-weighted averages of each variable by taking the sum (since the variables were already transformed by the population weights)
- Print the first few rows and columns of `census_ct`.

**(For full coding detail, see Appendix)**

**Here we print first 5 rows and 7 columns:**

| State | County | TotalPop | Men | White | Citizen | Income |
|---|---|---|---|---|---|---|
| Alabama | Autauga | 55221 | 0.4843 | 75.79 | 0.7375 | 51696 |
| Alabama | Baldwin | 195121 | 0.4885 | 83.1 | 0.7569 | 51074 |
| Alabama | Barbour | 26932 | 0.5383 | 46.23 | 0.7691 | 32959 |
| Alabama | Bibb | 22604 | 0.5341 | 74.5 | 0.774 | 38887 |
| Alabama | Blount | 57710 | 0.4941 | 87.85 | 0.7338 | 46238 |
| Alabama | Bullock | 10678 | 0.5301 | 22.2 | 0.7545 | 33293 |

14. If you were physically located in the United States on election day for the 2016 presidential election, what state and county were you in? Compare and contrast the results and demographic information for this county with the state it is located in. If you were not in the United States on election day, select any county. Do you find anything unusual or surprising? If so, explain; if not, explain why not.

**Here we select Santa Barbara County.**

**The election result for the county is:**

| candidate |
|---|
| Hillary Clinton |

**The election result for the state California is:**

| candidate |
|---|
| Hillary Clinton |

**We can see that the election result for Santa Barbara County corresponds to the result for the state California.**

**During the comparison between the demographic features of Santa Barbara County with the mean value on each feature of the California states, we realized that the measures for Santa Barbara County are all similar to the mean value with no large diverging values. This may explain the same voting outcome.**

**Taking a closer look, the `Citizen`, `poverty`, `IncomePerCapErr`, `Professional` and `Employed` measures of Santa Barbara County are especially very close to the mean, which may show that special demographics such as number of citizens, percentage under poverty level, percentage of people employed in management, business, science, and arts and percentage of employed people over 16 may determine someone's vote for Clinton.**

**To confirm our assumption, we take Kern County, which has a voting result for Trump in California for comparison. We can see that, indeed, Kern has large diverge numbers in `Citizen`, `poverty`, `IncomePerCapErr`, `Professional` and `Employed` features from the mean for California.**

**Therefore, we can see that, since Santa Barbara County has similar values with California on `Citizen`, `poverty`, `IncomePerCapErr`, `Professional`, and `Employed`, it has the same election result.**

Table 9: Table continues below

| State | County | Men | White | Citizen | Income | IncomeErr |
|---|---|---|---|---|---|---|
| California | NA | 0.5043 | 55.65 | 0.6823 | 59031 | 9828 |
| California | Santa Barbara | 0.4977 | 46.49 | 0.6334 | 66498 | 10789 |
| California | Kern | 0.5028 | 36.82 | 0.57 | 52382 | 8129 |

Table 10: Table continues below

| IncomePerCap | IncomePerCapErr | Poverty | ChildPoverty | Professional |
|---|---|---|---|---|
| 28014 | 3977 | 16.71 | 20.8 | 33.5 |
| 30753 | 4036 | 16.25 | 18.93 | 34.27 |
| 21038 | 2848 | 23.48 | 31.33 | 23.96 |

Table 11: Table continues below

| Service | Office | Production | Drive | Carpool | Transit | OtherTransp |
|---|---|---|---|---|---|---|
| 20.61 | 22.33 | 10.84 | 73.18 | 11.46 | 2.768 | 2.671 |
| 21.26 | 21.07 | 8.474 | 67.34 | 14.26 | 3.258 | 5.316 |
| 19.38 | 20.74 | 14.07 | 76.95 | 14.38 | 1.147 | 3.082 |

Table 12: Table continues below

| WorkAtHome | MeanCommute | Employed | PrivateWork | SelfEmployed | FamilyWork |
|---|---|---|---|---|---|
| 6.388 | 25.17 | 0.4161 | 70.23 | 9.499 | 0.4364 |
| 5.425 | 19.23 | 0.4705 | 75.4 | 7.748 | 0.2709 |
| 2.905 | 23.81 | 0.3865 | 76.73 | 6.228 | 0.2387 |

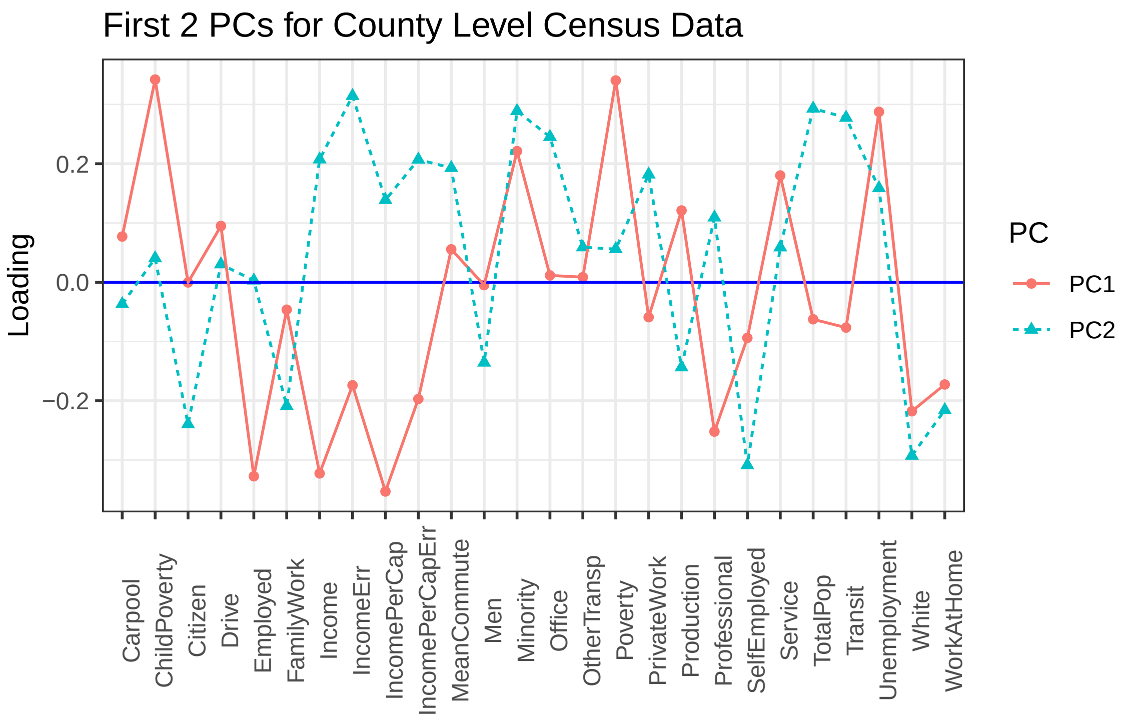
| Unemployment | Minority |
|---|---|
| 10.95 | 41.26 |
| 8.273 | 51.18 |
| 13.05 | 61.21 |

## Exploratory analysis

15. Carry out PCA for both county & sub-county level census data. Compute the first two principal components PC1 and PC2 for both county and sub-county respectively. Discuss whether you chose to center and scale the features and the reasons for your choice. Examine and interpret the loadings.

**We choose to center and scale the features. The reason for centering and scaling is that PCA with unscaled variables upweights variables with the highest variance, so if we have some high variance variables, the eights of other low variance variables will be close to 0, which makes the interpretation of PCs difficult. Considering that we have variables which may include 3, 4 or 5 digits numbers such as `TotalPop`, while we also have variables which only ranges from 0 to 1 such as `FamilyWork`, it's better for us to center and scale the features to avoid the uneven weight.**

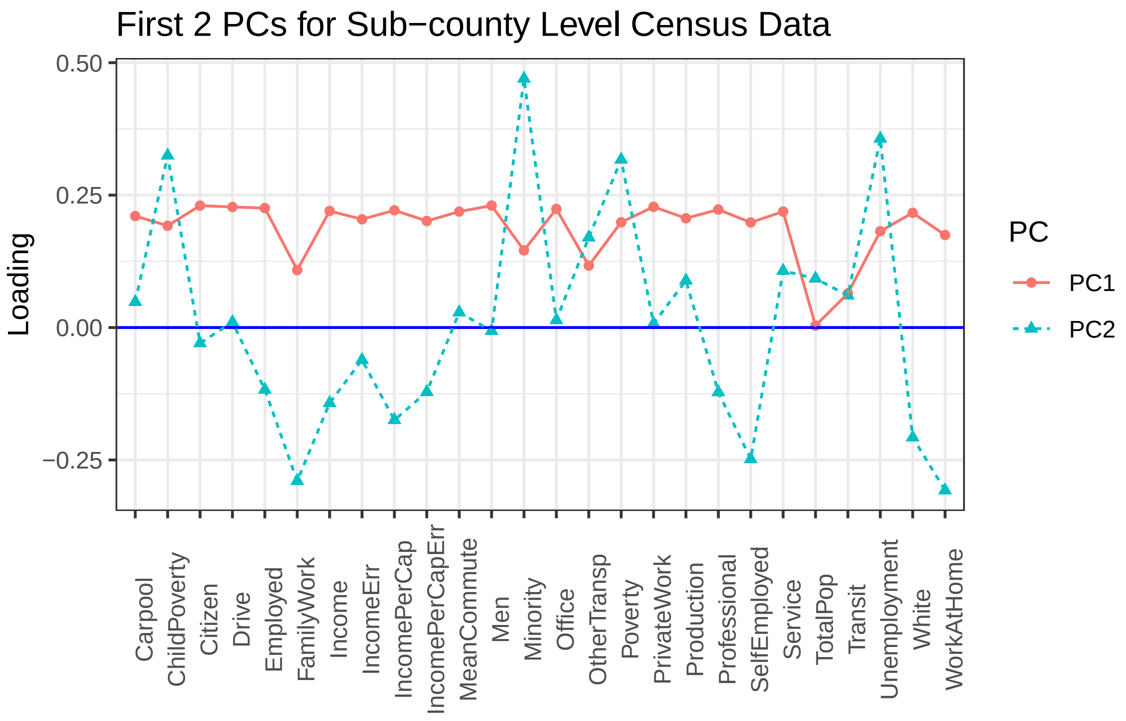
**This shows the first 2 PCs for county level census data:**



**From the Loading plot of county level census data, PC1 will be large and positive whenever**

**Employed, Income, IncomePerCap, Professional, White and Work at home are low while Child Poverty, Minority, Poverty, Production, Service, Unemployment are high. And the remaining variables are around average. This seems to describe counties that are poor, with a high unemployment rate and with more minorities. And PC2 will be large and positive whenever Citizen, FamilyWork, Men, Production, SelfEmployed, White and Work at home are low while Income, IncomeErr, IncomePerCap, IncomePerCapErr, MeanCommute, Minority, Office, Private Work, Professional, Total population, Transit and Unemployment are high. This seems to describe counties with good economic situation, high population, especially in minority and with more professional people. But the unemployment rate is high.**

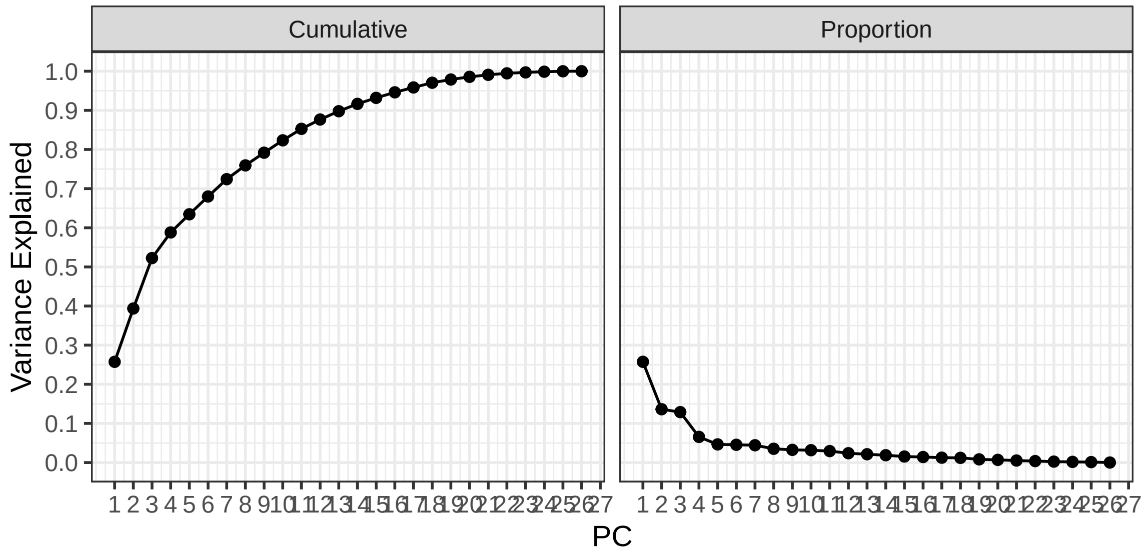
**This shows the first 2 PCs for sub-county level census data:**



**From the Loading plot of sub-county level census data, PC1 will be large and positive whenever Total Population is around average and the remaining variables are high. This seems to describe sub-counties with good economic situation and balanced race in population. And PC2 will be large and positive whenever FamilyWork, Income, IncomePerCap, Professional, Self Employed, White and Work at home are low while Child Poverty, Minority, Poverty, Unemployment are high. This seems to describe sub-counties with bad economic situation and high unemployment, and with high population proportion of minority.**

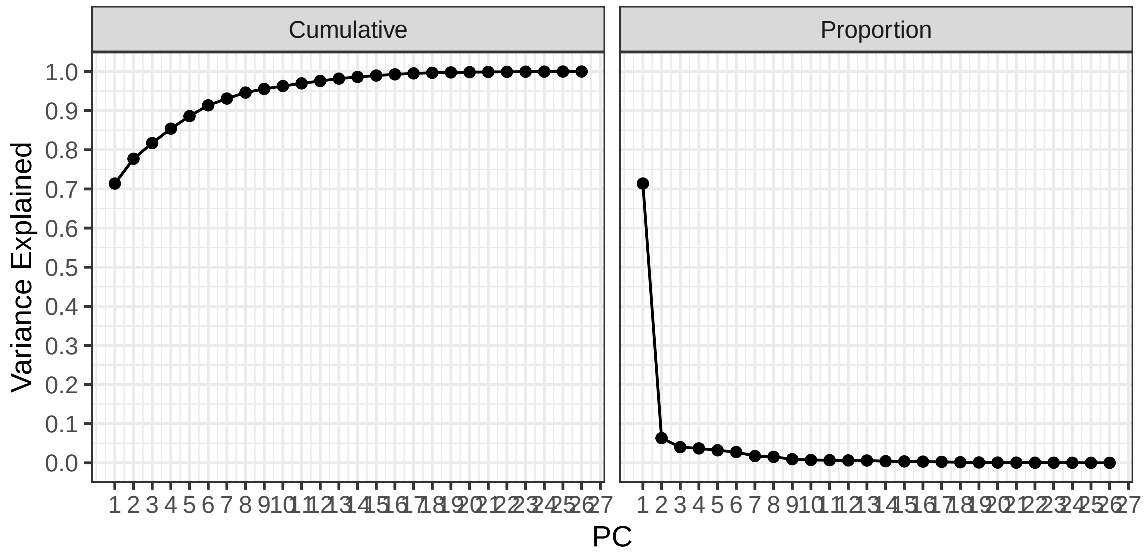
16. Determine the minimum number of PCs needed to capture 90% of the variance for both the county and sub-county analyses. Plot the proportion of variance explained and cumulative variance explained for both county and sub-county analyses.

**This shows the the proportion of variance explained and cumulative variance explained for county level census data:**



**From the cumulative variance plot, we can see that 90% of the total variance threshold is $q = 13$. From the screen plot, we can see that we can use first 13 PCs to explain 90% of the total variance.**

**This shows the the proportion of variance explained and cumulative variance explained for sub-county level census data:**



**From the cumulative variance plot, we can see that 90% of the total variance threshold is $q = 6$. From the scree plot, we can see that we can use first 6 PCs to explain 90% of the total variance.**

17. With `census_ct`, perform hierarchical 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 the county-level data as inputs instead of the original 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 cluster? Comment on what you observe and discuss possible explanations for these observations.

**This is the table showing the count number of data points per cluster in each data. "clusters" is in original features, and "clusters.2" is the data re-runed using the first 5 principal components**

**the county-level data as inputs:**

| clusters | n | clusters.2 | n |
|---|---|---|---|
| cluster 1 | 2957 | cluster 1 | 1919 |
| cluster 2 | 69 | cluster 2 | 905 |
| cluster 3 | 153 | cluster 3 | 163 |
| cluster 4 | 2 | cluster 4 | 86 |
| cluster 5 | 20 | cluster 5 | 8 |
| cluster 6 | 1 | cluster 6 | 27 |
| cluster 7 | 2 | cluster 7 | 47 |
| cluster 8 | 10 | cluster 8 | 8 |
| cluster 9 | 3 | cluster 9 | 18 |
| cluster 10 | 1 | cluster 10 | 37 |

**From the results shown on the table above, the hierarchical clustering with complete linkage on `census_ct` simply picks out only a few clusters with most of the data, then leaves the rest of the data as some outliers. While the hierarchical clustering using the first 5 principal components of the county-level data seperate the clusters more evenly.**

**For hierarchical clustering with complete linkage on `census_ct`, San Mateo County is in Cluster** *2* **.**

**For hierarchical clustering with complete linkage with first 5 PCs, San Mateo County is in Cluster** *7* **.**

**To comapre which approach seemed to put San Mateo County in a more appropriate cluster, we can compare the distance from San Mateo County data to the cluster data mean. If the distance is smaller, that means that the San Mateo County is more closer to the center of the cluster, so that approach seemed to put San Mateo County in a more appropriate cluster.**

**Now we calculate the mean values on each variables for counties in Cluster 2 and compare it with San Mateo County. (For simplicity here we only display first few columns).**

| | TotalPop | Men | White | Citizen | Income | IncomeErr |
|---|---|---|---|---|---|---|
| **cluster 2 mean** | 793141 | 0.4872 | 54.87 | 0.6872 | 67645 | 10288 |
| **San Mateo** | 746069 | 0.492 | 40.64 | 0.642 | 100370 | 16123 |

**The distance between the cluster mean and San Mateo for hierarchical clustering using `census_ct` is:** *59650*

**Then we calculate the mean values on each variables for counties in Cluster 7 and compare it with San Mateo County.**

| | PC1 | PC2 | PC3 | PC4 | PC5 |
|---|---|---|---|---|---|
| **cluster 7 mean** | -2.107 | 5.196 | -1.531 | 0.8336 | -0.5935 |
| **San Mateo** | -5.733 | 6.347 | -2.646 | 0.892 | 0.2025 |

**The distance between the cluster mean and San Mateo for the hierarchical clustering using the first 5 principal components of the county-level data is:** *4.044*

**We can see that the distance between the cluster mean and San Mateo for the hierarchical clustering using the first 5 principal components of the county-level data is much smaller**

**than the distance for hierarchical clustering using `census_ct`. Therefore, we can conclude that hierarchical clustering using the first 5 principal components of the county-level data put San Mateo County in a more appropriate cluster.**

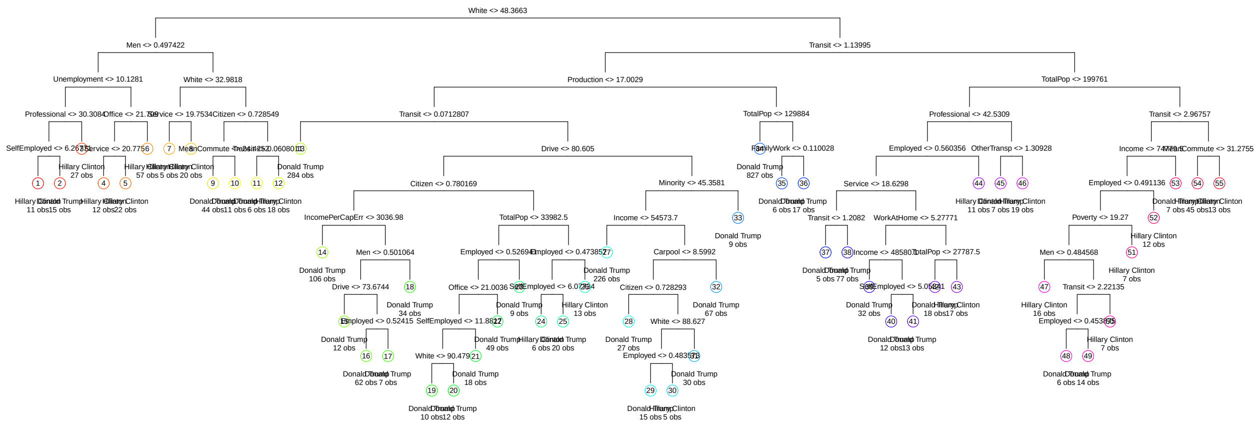
## Classification

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. Codes are provided in the .Rmd file that make the necessary changes to merge them into `election_cl` for classification.

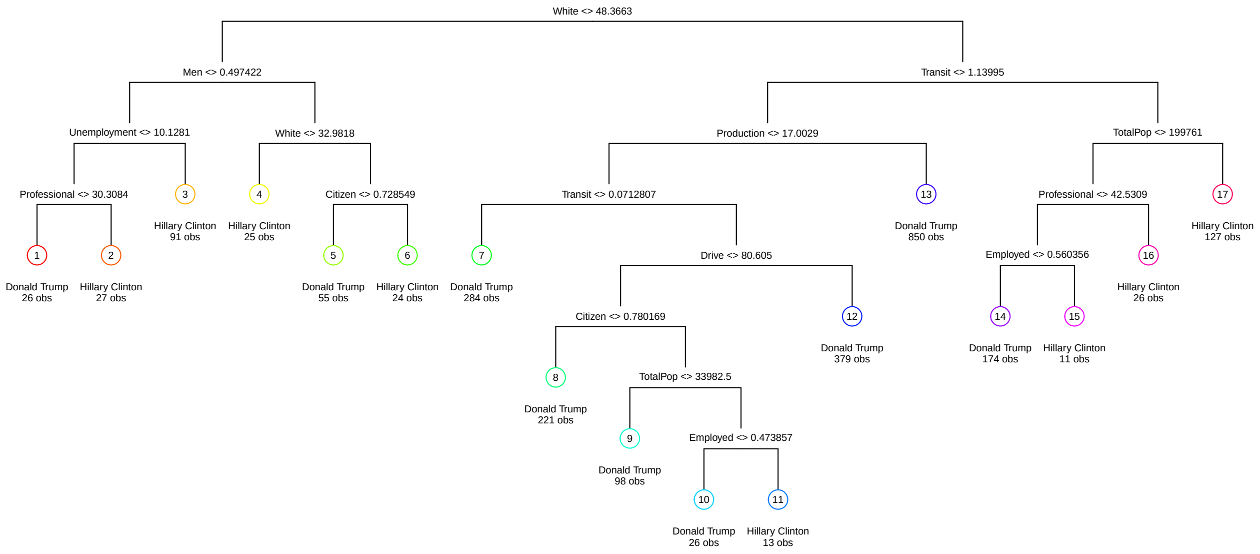
After merging the data, partition the result into 80% training and 20% testing partitions.

18. Decision tree: train a decision tree on the training partition, and apply cost-complexity pruning. Visualize the tree before and after pruning. Estimate the misclassification errors on the test partition, and intepret and discuss the results of the decision tree analysis. Use your plot to tell a story about voting behavior in the US (see this NYT infographic).

**This is the tree before pruning:**



**This is the tree after pruning:**



**This is the misclassification errors on the test partition:**

| Tree | Misclassification_Errors |
|---|---|
| Before pruning (t_0) | 0.08157 |
| After pruning (t_opt) | 0.07341 |

**Based on the tree after pruning, the root node is the percentage of white population (`White`). If the percentage of white population is greater than 48.3663% (the trees growing on the right), more prediction reaches "Donald Trump" as leaf nodes, and if the percentage of white population is less than 48.3663% (trees growing on the left), more prediction reaches "Hillary Clinton" as leaf nodes. Therefore, we can see Donald Trump wins the vast majority of counties with large white population.**

**For counties with larger white population, it seems that the percentage of people commuting on public transportation (`Transit`) and the percentage of people employed over 16 (`Employed`) splits the vote for Donald Trump and Hillary Clinton. If `Transit` is greater than 1.13995% (the trees growing on the right), unless the total population is less than 19976, `Profession` is less than 42.5309% and `Employed` is less than 0.560356%, all predictions reaches "Hillary Clinton". If `Transit` is less than 1.13995% (the trees growing on the left), unless the number of citizens is larger than 0.780169, the total population is larger than 33982.5 and `Employed` is greater than 0.473857%, all predictions reaches "Donald Trump".**

**Therefore, for counties with larger white population, Donald Trump wins the majority of counties with less percentage of people commuting on public transportation and less percentage of employed people, and Hillary Clinton wins the majority of counties with**

**For counties with smaller white population, it seems that percentage of people unemployed (`Unemployment`) and the percentage of people employed in management, business, science and art (`Professional`) splits the vote for Donald Trump and Hillary Clinton. If `Men` is less than 0.49742% (the trees growing on the left), unless percentage of people unemployed is less than 10.1281% and `Professional` is less than 30.3084%, all predictions reaches "Hillary Clinton".**

**Therefore, for counties with smaller white population, Hillary Clinton wins the majority of counties with higher percentage of people employed in management, business, science and art.**

**In conclusion, we can see Donald Trump wins the vast majority of counties with large white population, less people commuting on public transportation and less employed people. While Hillary Clinton wins the vast majority of counties with less white population, larger percentage of people unemployed and more people employed in management, business, science and art.**

19. Train a logistic regression model on the training partition to predict the winning candidate in each county and estimate errors on the test partition. What are the significant variables? Are these consistent with what you observed in the decision tree analysis? Interpret the meaning of one or two significant coefficients of your choice in terms of a unit change in the variables. Did the results in your particular county (from question 14) match the predicted results?

**This is the summary report of the logistic regression model in table form:**

| | Estimate | Std. Error | z value | Pr(>\|z\|) |
|---|---|---|---|---|
| **(Intercept)** | -19.65 | 9.499 | -2.068 | 0.03863 |
| **TotalPop** | 2.195e-07 | 3.735e-07 | 0.5878 | 0.5567 |
| **Men** | 5.225 | 5.045 | 1.036 | 0.3003 |
| **White** | -0.1756 | 0.06266 | -2.802 | 0.005079 |
| **Citizen** | 10.06 | 2.859 | 3.518 | 0.000435 |
| **Income** | -7.398e-05 | 2.668e-05 | -2.773 | 0.00556 |
| **IncomeErr** | -1.591e-05 | 6.639e-05 | -0.2396 | 0.8107 |
| **IncomePerCap** | 0.0002221 | 6.833e-05 | 3.251 | 0.001152 |

| | Estimate | Std. Error | z value | Pr(>\|z\|) |
|---|---|---|---|---|
| **IncomePerCapErr** | -0.0003854 | 0.0001879 | -2.051 | 0.04029 |
| **Poverty** | 0.02164 | 0.04076 | 0.5309 | 0.5955 |
| **ChildPoverty** | 0.002651 | 0.0256 | 0.1036 | 0.9175 |
| **Professional** | 0.2558 | 0.03909 | 6.543 | 6.014e-11 |
| **Service** | 0.3313 | 0.04943 | 6.702 | 2.051e-11 |
| **Office** | 0.0806 | 0.04563 | 1.766 | 0.07733 |
| **Production** | 0.1365 | 0.04179 | 3.265 | 0.001094 |
| **Drive** | -0.1725 | 0.0455 | -3.79 | 0.0001506 |
| **Carpool** | -0.164 | 0.06037 | -2.717 | 0.00658 |
| **Transit** | 0.09702 | 0.09716 | 0.9986 | 0.318 |
| **OtherTransp** | -0.06489 | 0.09461 | -0.6859 | 0.4928 |
| **WorkAtHome** | -0.09936 | 0.07703 | -1.29 | 0.1971 |
| **MeanCommute** | 0.04433 | 0.02428 | 1.826 | 0.06786 |
| **Employed** | 19.55 | 3.248 | 6.019 | 1.753e-09 |
| **PrivateWork** | 0.093 | 0.02177 | 4.273 | 1.933e-05 |
| **SelfEmployed** | -0.01513 | 0.05079 | -0.298 | 0.7657 |
| **FamilyWork** | -0.8388 | 0.3923 | -2.138 | 0.03252 |
| **Unemployment** | 0.2027 | 0.04026 | 5.036 | 4.748e-07 |
| **Minority** | -0.05047 | 0.06039 | -0.8358 | 0.4033 |

**And this is the estimate errors on the test partition:**

| Misclassification_Error |
|---|
| 0.06199 |

**Note that the Pr(>|z|) column shows how significant the variable is. If the value of Pr(>|z|) is very low, it means that the variable is important. Setting the Pr(>|z|)<0.05, we get the significant variables:** *Citizen*, *Professional*, *Service*, *Drive*, *Employed*, *PrivateWork* and *Unemployment* .

**In the decision tree analysis, our significant variable were `White`, `Employed`, `Transit`, `Unemployment` and `Professional`. Thus the significant variables we found here are mostly consistent with what we observed in the decision tree analysis.**

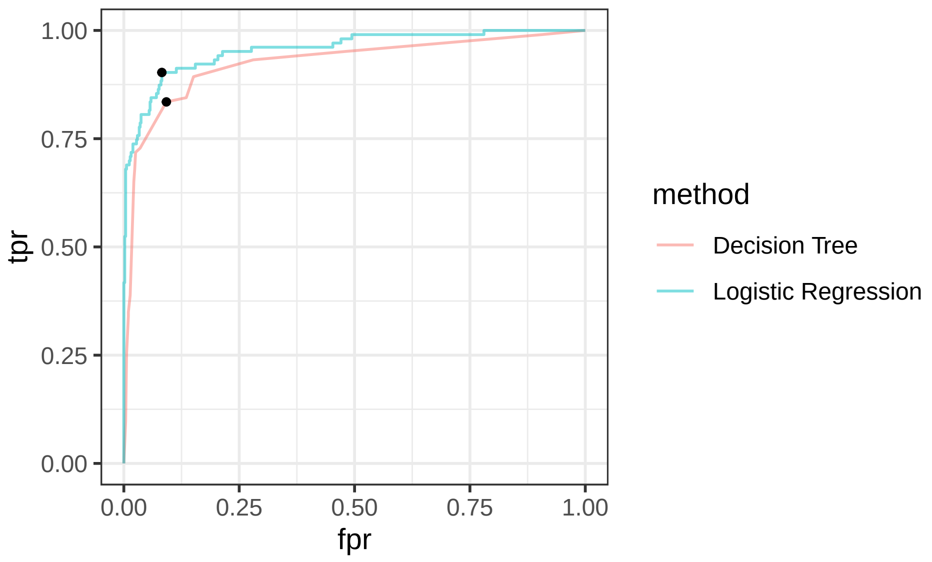
**For interpretation of the meaning of significant coefficients, take variable "Professional" as an example: a one-unit increase in Professional will change in odds by a factor of 0.2558, assuming other variables are held constant.**

**Then we used this model to predict the county winner in Santa Barbra County, below is the predicted result:**

| County | glm_prediction |
|---|---|
| Santa Barbra County | Hillary Clinton |

**Our predicted results in problem 14 was Hillary Clinton, which matches the result we have here.**

20. Compute ROC curves for the decision tree and logistic regression using predictions on the test data, and display them on the same plot. Based on your classification results, discuss the pros and cons of each method. Are the different classifiers more appropriate for answering different kinds of questions about the election?



**For Decision Trees:**

**Pros: it is applicable to both regression and classification settings and it doesn't have restrictions on variable types. Also, trees are intuitive and easy to visualize and interpret.**

**Cons: trees have high variance since they are sensitive to small changes in the data, as we can see that the missclassification errors before and after pruning are both higher than the misclassification error in logistic regression.**

**Since trees are intuitive and easy to visualize and interpret, it can be used to answer questions such as the demographic of voters for candidates (doesn't have to be exactly two candidates).**

**For Logistic Regression:**

**Pros: it works (better) in high dimensions and with noise variables, which is a good thing here since we have a lot variables in census and election data. It also works for categorical or continuous features, which is advantageous here because census and election data may include both categorical and continuous features.**

**Cons: it's difficult to generalize to multiple classes, which is not a big problem here since we only have two candidate for classification. However, if there're more than two major candidates in election, this will be a big problem.**

**Since from previous part, we can see that Logistic Regression performed a lower test misclassification error than Decision Trees, it can be used to predict the final result for the election between two candidates.**

## Taking it further

21. 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 or doesn't seem reasonable based on your understanding of these methods, propose possible directions (for example, collecting additional data or domain knowledge). In addition, propose and tackle *at least* one more interesting question. Creative and thoughtful analyses will be rewarded!

## Overall Insights

**This project helps us to analyze the 2016 election outcome by merging census data with the 2016 voting data. Instead of merely looking at polls of how may people voted each candidates, this projects shows that the demographics of the population (such as people's races, incomes, professions, etc.) are all important and they may be deciding factors for votes prediction. Extracting and interpreting the voting patterns for specific types of people to decide their final vote requires the process of machine learning.**

**Before starting the process of machine learning, data visualization and exploratory data analysis procedures are also important. These procedures transforms the numerical data into more understandable forms such as charts and graphs, which provide an easier way to understand the data. By firstly looking at the visualized data, we can get a general idea of the directions to proceed, providing useful insights for our following machine learning methods.**

**When proceeding with machine learning for prediction, choosing appropriate methods for specific questions become important. As we discussed in 20, different methods are suitable for different questions of intention.**

## Limitations

**One limit of this project is that the census and election data may be unmatchable. The census data was collected in 2020, but the election data was collected in 2016. We have already discussed that one of difficulties to make voter behavior prediction is people may change their mind over time. Therefore, there are some changes may happened from 2010 to 2016 which could affect our data sets, clusters, and prediction results.**

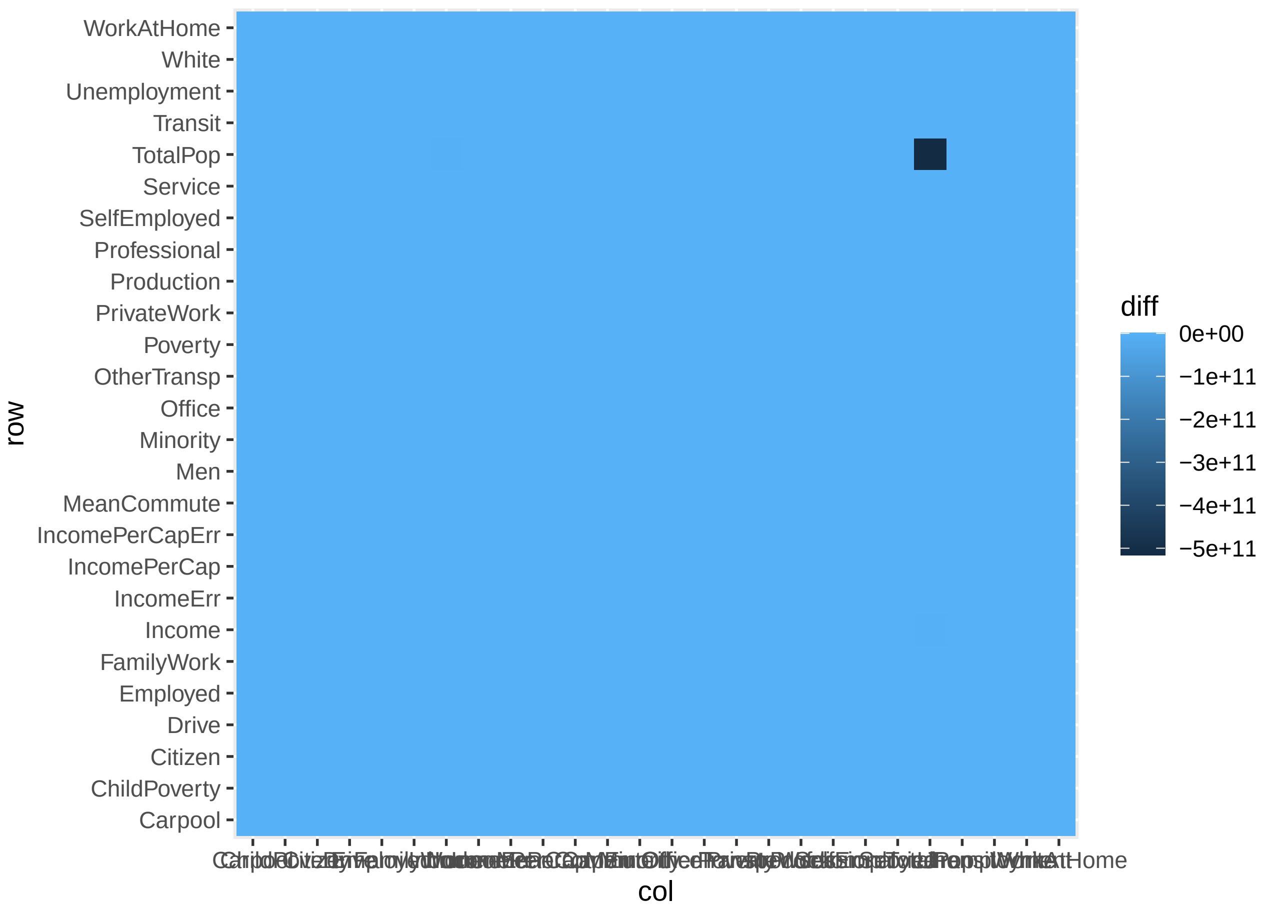
## Possible Further Direction

**In the future analysis, we can further our predictions by comparing the election data with previous election data. Collecting the previous data can help us know more about the historical trends and voters' intention. We also can add more classification variables in the demographic system since the voters' choices can be influenced by a lot of other variables such as the candidate's policies and the choices of people around. They may also determine the voters' choices.**

## Further Explorations

**One interesting question we want to explore is to see whether and how we can possibly improve the prediction accuracy. Here we will firstly see if discriminant analysis method can imrpove the prediction accuracy.**

**To perform discriminant analysis, we need to extract the feature covariance matrices for both candidates. Then construct a heatmap of the difference between the covariance matrices to compare these two matrices in order to choose the more appropriate discriminant analysis method (LDA or QDA). The heatmap is shown below.**



**The covariance matrices seem similar since most of the differences are 0. Thus, linear discriminant analysis (LDA) seems more appropriate. So we fit the LDA model using the training dataset, then compute the estimated classes and estimate misclassification errors for both training and testing data, which are shown in the following table:**

| | misclassification errors |
|---|---|
| **train** | 0.06919 |
| **test** | 0.06852 |

**Next, we used this model to predict the county winner in Santa Barbra County, below is the predicted result:**

| County | glm_prediction |
|---|---|
| Santa Barbra County | Hillary Clinton |

**Our predicted results is Hillary Clinton, which matches the result we got by using the logistic regression and the tree method.**

**However, here for LDA, test misclassification error is 0.06852, which is lower than the test misclassification error for the decision trees (0.07341) but is a little higher than the test misclassification error for Logistic regression (0.06199), so if we want to finally predict the winning candidate, Logistic regression is still a better method.**

**Now we want to explore on how to improve classification performance. A good way to consider is to use principal components to create new (and lower dimensional) sets of features with which to train a classification model.**

**From 16, we know the first 13 PCs to explain 90% of the total variance of the county-level census data. Therefore, here we will use PC 1-13.**

**After calculated the PC 1-13, partition the result into 80% training and 20% testing partitions. Since we previously found out that logistic regression has the highest accuracy among others on this question, we choose to fit a logistic regression model on training data to predict the winning candidate based on 13 PCs.**

**And this is the estimate errors on the test partition using PC 1-13:**

| Misclassification_Error |
|---|
| 0.0571 |

**Therefore, compared to the previous test errors, using principal components to as features for a classification model has the lowest misclassification error, and it was much lower than the misclassification errors for previous models. Therefore, we can reach an conclusion that using principal components for prediction improves the classification prediction.**

**Now we used this model to predict the county winner in Santa Barbra County, below is the predicted result:**

| County | glm_prediction |
|---|---|
| Santa Barbra County | Hillary Clinton |

**Still, our predicted results is Hillary Clinton, which matches the result we got by using all the previous method.**

**(For full detailed coding on how we performed LDA and PCA, see Appendix)**

## Codes

```
knitr::opts_chunk$set(echo = F,
                      cache = F,
                      results='markup',
                      message=F,
                      warning=F,
                      fig.align = 'center',
                      fig.height = 5,
                      fig.width = 5)

library(pander)
library(tidyverse)
library(ggmap)
library(maps)
library(modelr)
library(reshape2)
library(knitr)
library(ISLR)
library(ROCR)
library(class)
library(randomForest)
library(gbm)
library(tree)
library(maptree)
load('data/project_data.RData')
filter(election_raw, !is.na(county)) %>%
  head() %>%
  pander()
# inspect rows
election_raw%>%
  filter(fips==2000)%>%
  pander()
# drop these observations
election_raw <- election_raw %>%
  filter(fips != 2000)

# report data dimensions after removal
dim_data <- dim(election_raw)
rnames <- c("number")
cnames <- c("Row", "Column")
matrix(dim_data, nrow=1, ncol=2, byrow=TRUE, dimnames=list(rnames,cnames)) %>%
  pander()
census %>%
  select(1:6) %>%
  head() %>%
  pander(digits = 15)
census_meta %>% head() %>% pander()
# Separate the rows into separate federal-, state-, and county-level data frames
# federal level
election_federal <- election_raw%>%
  filter(fips %in% c("US"))
# state_level
election_state <- election_raw%>%
```

```r
  filter(!(fips %in% c("US")), fips==state)
# county-level
election <- election_raw%>%
  filter(!is.na(county)) %>%
  mutate(fips = as.numeric(fips))
# draw a bar graph of presidential candidates
election_federal %>%
  filter(candidate != ' None of these candidates') %>%
  ggplot(aes(x= reorder(candidate,votes), y = votes, label = votes)) +
  geom_bar(stat="identity")+
  scale_y_continuous(trans = "log")+
  ggtitle("All Votes Received by Each Candidates (in Log Scale)")+
  ylab("Log(votes)")+
  xlab("Candidate Names")+
  geom_label()+
  coord_flip()
# create `county_winner`
county_winner <- election%>%
  group_by(fips)%>%
  mutate(total = sum(votes), pct = votes/total)%>%
  slice_max(pct,n=1) %>%
  as.data.frame()

# `create state_winner`
state_winner <- election_state%>%
  group_by(state)%>%
  mutate(total=sum(votes), pct=votes/total)%>%
  slice_max(pct,n=1)%>%
  as.data.frame()
#install.packages("maps")

states <- map_data("state")

ggplot(states) +
  geom_polygon(aes(x = long,
                   y = lat,
                   fill = region,
                   group = group),
               color = "white") +
  coord_fixed(1.3) + # avoid stretching
  guides(fill=FALSE) + # no fill legend
  theme_nothing() # no axes
# Draw a county-level map
counties <- map_data("county")
ggplot(counties) +
  geom_polygon(aes(x = long,
                   y = lat,
                   fill = subregion,
                   group = group),
               color = "white") +
  coord_fixed(1.3) + # avoid stretching
  guides(fill=FALSE) + # no fill legend
  theme_nothing() # no axes
```

```r
name2abb <- function(statename){
  ix <- match(statename, tolower(state.name))
  out <- state.abb[ix]
  return(out)
}

states <- states %>% mutate(fips = name2abb(states$region))
# Transform format of state names in states
states_join <- left_join(state_winner, states, by=c("state" = "fips"))

ggplot(states_join) +
  geom_polygon(aes(x = long,
                   y = lat,
                   fill = candidate,
                   group = group),
               color = "white") +
  coord_fixed(1.3) + # avoid stretching
  guides(fill=FALSE)+  # no fill legend
  theme_nothing() + # no axes
  scale_fill_brewer(palette="Set1")
new_county.fip <- county.fips %>%
  separate(polyname, c("region","subregion"), sep=",")

new_counties <- left_join(new_county.fip, counties,
                          by=c("subregion" = "subregion"))

county_join <- left_join(county_winner, new_counties, by=c("fips" = "fips"))

ggplot(county_join) +
  geom_polygon(aes(x = long,
                   y = lat,
                   fill = candidate,
                   group = group),
               color = "white") +
  coord_fixed(1.3) + # avoid stretching
  guides(fill=FALSE)+  # no fill legend
  theme_nothing() + # no axes
  scale_fill_brewer(palette="Set1")
# we will try to present the graph visualized the state winner result by sex.
race<- census %>%
  mutate_all(~replace(., is.na(.), 0))%>%
  mutate(White = White*TotalPop,
         Black = Black*TotalPop,
         Hispanic = Hispanic*TotalPop,
         Native = Native*TotalPop,
         Asian = Asian*TotalPop,
         Pacific = Pacific*TotalPop) %>%
  group_by(State)%>%
    summarize(White = sum(White),
              Black = sum(Black),
              Hispanic = sum(Hispanic),
              Native = sum(Native),
              Asian = sum(Asian),
              Pacific = sum(Pacific))

race<-race%>%
  mutate(States=name2abb(tolower(race$State)))

# remove Purto Rico from state_sex since it is not included in state_winner
race <- race[-which(!(race$States %in% state_winner$state)),]

left_join(race,state_winner,by=c("States"="state")) %>%
  group_by(candidate) %>%
  summarize(White = sum(White),
              Black = sum(Black),
              Hispanic = sum(Hispanic),
              Native = sum(Native),
              Asian = sum(Asian),
              Pacific = sum(Pacific)) %>%
  gather(key = race, value = population,-candidate) %>%
  ggplot(aes(x = candidate, y = population, fill = race))+
  geom_bar(position="dodge", stat="identity")+
  coord_flip()
# census_del
census_del <- census %>%
  filter(complete.cases(census)) %>%
  select(-Women) %>%
  mutate(Men= Men/TotalPop,
         Employed = Employed/TotalPop,
         Citizen = Citizen/TotalPop) %>%
  mutate(Minority = Hispanic+Black+Native+Asian+Pacific) %>%
  select(-c(Hispanic, Black, Native, Asian, Pacific)) %>%
  select(-c(Walk, PublicWork, Construction))
# census_subct
census_subct  <- census_del %>%
  group_by(State, County) %>%
  add_tally(TotalPop, name = "CountyTotal") %>%
  mutate(population_weight = TotalPop/CountyTotal) %>%
  mutate_at(colnames(.)[5:29],
            ~.*population_weight)%>%
  as.data.frame()
# census_ct
census_ct <- census_subct %>%
  select(-CensusTract,-CountyTotal,-population_weight) %>%
  group_by(State,County) %>%
  summarise_all(sum) %>%
  as.data.frame()
head(census_ct)[,1:7] %>% pander()
county_winner %>%
  filter(county == "Santa Barbara County") %>%
  select(candidate) %>% pander()
state_winner %>%
  filter(fips == "CA") %>%
  select(candidate) %>% pander()
# demographic_compare
rbind(census_ct %>%
        group_by(State) %>%
        summarize_all(mean) %>%
        filter(State == "California"),
      census_ct%>%
        filter(State == "California",
               County == 'Santa Barbara'),
      census_ct%>%
        filter(State == "California",
               County == 'Kern')) %>%
  select(-TotalPop)%>% pander()
# county level census data
# center data
ct_mx <- census_ct[,3:28] %>%
  scale(center = T, scale = T)

# compute SVD
ct_svd <- svd(ct_mx)

# get loadings
ct_loadings <- ct_svd$v

# compute PCs
ct_pc <- ct_mx %*% ct_loadings %>% as.data.frame()
colnames(ct_pc) <- paste('PC', 1:2, sep = '')

# plot loadings
ct_loadings[,1:2] %>%
  as.data.frame() %>%
  rename(PC1 = V1, PC2 = V2) %>%
  mutate(variable = colnames(ct_mx)) %>%
  gather(key = 'PC', value = 'Loading', 1:2) %>%
  arrange(variable) %>%
  ggplot(aes(x = variable, y = Loading)) +
  geom_point(aes(shape = PC,color = PC)) +
  theme_bw() +
  geom_hline(yintercept = 0, color = 'blue') +
  geom_path(aes(linetype = PC, group = PC, color = PC)) +
  theme(axis.text.x = element_text(angle = 90)) +
  labs(x = '') +
  ggtitle("First 2 PCs for County Level Census Data")
#subcounty level census data
# center data
subct_mx <- census_subct[,4:29] %>%
  scale(center = T, scale = T)

# compute SVD
subct_svd <- svd(subct_mx)

# get loadings
subct_loadings <- subct_svd$v

# compute PCs
subct_pc <- subct_mx %*% subct_loadings %>% as.data.frame()
```

```r
colnames(subct_pc) <- paste('PC', 1:2, sep = '')

# plot loadings
subct_loadings[,1:2] %>%
  as.data.frame() %>%
  rename(PC1 = V1, PC2 = V2) %>%
  mutate(variable = colnames(subct_mx)) %>%
  gather(key = 'PC', value = 'Loading', 1:2) %>%
  arrange(variable) %>%
  ggplot(aes(x = variable, y = Loading)) +
  geom_point(aes(shape = PC,color = PC)) +
  theme_bw() +
  geom_hline(yintercept = 0, color = 'blue') +
  geom_path(aes(linetype = PC, group = PC, color = PC)) +
  theme(axis.text.x = element_text(angle = 90)) +
  labs(x = '') +
  ggtitle("First 2 PCs for Sub-county Level Census Data")
# compute PC variances for county level census data
ct_pc_vars <- ct_svd$d^2/(nrow(ct_mx) - 1)

# scree and cumulative variance plots for county level census data
tibble(PC = 1:min(dim(ct_mx)),
       Proportion = ct_pc_vars/sum(ct_pc_vars),
       Cumulative = cumsum(Proportion)) %>%
  gather(key = 'measure', value = 'Variance Explained', 2:3) %>%
  ggplot(aes(x = PC, y = `Variance Explained`)) +
  geom_point() +
  geom_path() +
  facet_wrap(~ measure) +
  theme_bw() +
  scale_x_continuous(breaks = 1:27, labels = as.character(1:27)) +
  scale_y_continuous(n.breaks = 10)
# compute PC variances for county level census data
subct_pc_vars <- subct_svd$d^2/(nrow(subct_mx) - 1)

# scree and cumulative variance plots for county level census data
tibble(PC = 1:min(dim(subct_mx)),
       Proportion = subct_pc_vars/sum(subct_pc_vars),
       Cumulative = cumsum(Proportion)) %>%
  gather(key = 'measure', value = 'Variance Explained', 2:3) %>%
  ggplot(aes(x = PC, y = `Variance Explained`)) +
  geom_point() +
  geom_path() +
  facet_wrap(~ measure) +
  theme_bw() +
  scale_x_continuous(breaks = 1:27, labels = as.character(1:27)) +
  scale_y_continuous(n.breaks = 10)
# hierarchical clustering with `census_ct`
# compute distances between points
d_mx <- dist(as.data.frame(census_ct), method = 'euclidean') %>%
  suppressWarnings()
# compute hierarchical clustering
hclust_out <- hclust(d_mx, method = 'complete')
```

```r
# cut at 10 clusters
clusters <- cutree(hclust_out, k = 10) %>%
  factor(labels = paste('cluster', 1:10))

# rerun with pc = 5
q <- 5
v_q <- ct_svd$v[, 1:q]
Z <- ct_mx %*% v_q
colnames(v_q) <- colnames(Z) <- paste('PC', 1:q, sep = '')
new_PC_data <- as.data.frame(Z)

# compute distances between points
d_mx.2 <- dist(new_PC_data, method = 'euclidean')

# compute hierarchical clustering
hclust_out.2 <- hclust(d_mx.2, method = 'complete')

# cut at 10 clusters
clusters.2 <- cutree(hclust_out.2, k = 10) %>%
  factor(labels = paste('cluster', 1:10))

# count number of data points per cluster in original data
cl1 <- tibble(clusters) %>% count(clusters)
# count number of data points per cluster in new PC data
cl2 <- tibble(clusters.2) %>% count(clusters.2)

cbind(cl1, cl2) %>% pander()
clusters[which(census_ct$County == 'San Mateo')] %>% pander()
clusters.2[which(census_ct$County == 'San Mateo')] %>% pander()
Cluster2 <- data.frame(
  rbind(colMeans(census_ct[clusters == "cluster 2",-1:-2]),
        census_ct[census_ct$County == 'San Mateo',-1:-2]),
  row.names = c("cluster 2 mean","San Mateo"))
Cluster2[,1:6] %>% pander()
dist(Cluster2)[1] %>% pander()
Cluster7 <- data.frame(
  rbind(colMeans(new_PC_data[clusters.2 == "cluster 7",]),
        new_PC_data[census_ct$County == 'San Mateo',]),
  row.names = c("cluster 7 mean","San Mateo"))
Cluster7 %>% pander()
dist(Cluster7)[1] %>% pander()
abb2name <- function(stateabb){
  ix <- match(stateabb, state.abb)
  out <- tolower(state.name[ix])
  return(out)
}

tmpwinner <- county_winner %>%
  ungroup %>%
  # coerce names to abbreviations
  mutate(state = abb2name(state)) %>%
  # everything lower case
  mutate(across(c(state, county), tolower)) %>%
  # remove county suffixes
  mutate(county = gsub(" county| columbia| city| parish",
                       "",
                       county))

tmpcensus <- census_ct %>%
  mutate(across(c(State, County), tolower))

election_county <- tmpwinner %>%
  left_join(tmpcensus,
            by = c("state"="State", "county"="County")) %>%
  na.omit()

## save meta information
election_meta <- election_county %>%
  select(c(county, fips, state, votes, pct, total))

## save predictors and class labels
election_county.2 <- election_county %>%
  select(-c(county, fips, state, votes, pct, total))
set.seed(20121)
election_ct_part <- resample_partition(election_county.2, p = c(test = 0.2, train = 0.8))
train <- as_tibble(election_ct_part$train)
test <- as_tibble(election_ct_part$test)
# train a decision tree on the training partition
nmin <- 20
tree_opts <- tree.control(nobs = nrow(train), minsize = nmin, mindev = exp(-6))
t_0 <- tree(as.factor(candidate) ~ ., data = train,
            control = tree_opts, split = 'deviance', na.action = na.pass)
draw.tree(t_0, cex = 0.75, size = 2.5, digits = 2)
# cost-complexity pruning
nfolds <- 10
cv_out <- cv.tree(t_0, K = nfolds)
# convert to tibble
cv_df <- tibble(alpha = cv_out$k,
                impurity = cv_out$dev,
                size = cv_out$size)
# choose optimal alpha
best_alpha <- slice_min(cv_df, impurity) %>% slice_min(size)
# select final tree
t_opt <- prune.tree(t_0, k = best_alpha$alpha)
# plot the resulting tree
draw.tree(t_opt, cex = 0.6, size = 2.5, digits = 2)
# predict
# predict on test data
classes_test <- test %>% pull(candidate)
preds_test_t0 <- predict(t_0, test, type = 'class')
preds_test_topt <- predict(t_opt, test, type = 'class')

# misclassification errors
mis_err_t0 <- mean(classes_test != preds_test_t0)
mis_err_topt <- mean(classes_test != preds_test_topt)
data.frame(Tree = c("Before pruning (t_0)", "After pruning (t_opt)"),
           Misclassification_Errors = c(mis_err_t0, mis_err_topt)) %>% pander()
# train logistic model
fit_glm <- glm(as.factor(candidate) ~ ., family = 'binomial', data = train) %>%
  suppressWarnings()
summary(fit_glm)$coeff %>% pander()
# compute estimated probabilities
p_hat_test <- predict(fit_glm, test, type = 'response')
# bayes classifier
y_hat_test <- factor(p_hat_test > 0.5, labels = c("Donald Trump", "Hillary Clinton"))

# misclassification errors
glm_err <- mean(classes_test != y_hat_test)
data.frame(Misclassification_Error = glm_err) %>% pander()
names(summary(fit_glm)$coeff[-1,4])[summary(fit_glm)$coeff[-1,4] < 0.001] %>% pander()
pred_SB <- y_hat_test[which(election_ct_part$test$idx ==
                              which(election_county$county == "santa barbara"))]
data.frame(County = "Santa Barbra County",
           glm_prediction = as.character(pred_SB)) %>% pander()
# decision tree
probs_test <- predict(t_opt, newdata = test, type = 'vector')
topt_prediction <- prediction(predictions = probs_test[, 2],
                              labels = classes_test)
topt_perf <- performance(topt_prediction, 'tpr', 'fpr')
rate_df <- tibble(tpr = slot(topt_perf, 'y.values'),
                  fpr = slot(topt_perf, 'x.values'),
                  thresh = slot(topt_perf, 'alpha.values')) %>%
  unnest(everything()) %>%
  mutate(youden = tpr - fpr)

# logistic regression
# create prediction object
prediction_glm <- prediction(predictions = p_hat_test,
                             labels = test$candidate)
# compute error rates as a function of probability threshhold
perf_glm <- performance(prediction.obj = prediction_glm, 'tpr', 'fpr')
rates_glm <- tibble(fpr = perf_glm@x.values[[1]],
                    tpr = perf_glm@y.values[[1]],
                    thresh = perf_glm@alpha.values[[1]])
rates_glm <- rates_glm %>% mutate(youden = tpr - fpr)

# plot ROC
new_rate_df <- rate_df %>% mutate(method = "Decision Tree")
new_rates_glm <- rates_glm %>% mutate(method = "Logistic Regression")
both_rate <- rbind(new_rate_df, new_rates_glm) %>% as.data.frame()

both_rate %>%
  ggplot(aes(x = fpr, y = tpr)) +
  geom_path(aes(color = method), alpha=0.5) +
  geom_point(data = slice_max(rates_glm, youden), colour="black", size = 1) +
  geom_point(data = slice_max(rate_df, youden), colour="black", size = 1) +
  theme_bw()
# feature covariance matrices for Donald Trump as candidate
DT_feature <- election_county.2 %>%
  filter(candidate == "Donald Trump") %>%
  select(-candidate) %>%
  cov(.,.,use = "everything",
    method = "pearson")

# feature covariance matrices for Hillary Clinton as candidate
HC_feature <- election_county.2 %>%
  filter(candidate == "Hillary Clinton") %>%
  select(-candidate) %>%
  cov(.,.,use = "everything",
    method = "pearson")

# heatmap of the difference between the covariance matrices
library(reshape2)
library(ggplot2)
diff_matrix <- as_tibble(DT_feature - HC_feature)
diff_matrix <- diff_matrix %>% bind_cols(row=names(diff_matrix))
diff_matrix <- gather(diff_matrix, key='col', value='diff', -27)
diff_matrix %>%
  ggplot(aes(y=row, x=col)) +
  geom_raster(aes(fill=diff))
# prediction using LDA
# fit LDA model
lda_fit_train <- MASS::lda(as.factor(candidate) ~., method = 'mle', data=train)
# compute estimated classes for both train and test data
lda_preds_train <- predict(lda_fit_train, train)
lda_preds_test <- predict(lda_fit_train, test)

# misclassification errors on train and test data
train_error <- mean(train$candidate != lda_preds_train$class)
test_error <- mean(test$candidate != lda_preds_test$class)

cnames.lda <- c("misclassification errors")
rnames.lda <- c("train", "test")
lda_err_val <- c(train_error, test_error)
matrix(lda_err_val, nrow=2, ncol=1, byrow=TRUE,
       dimnames=list(rnames.lda, cnames.lda)) %>%
  pander()
pred_SB_lda <- lda_preds_test$class[which(election_ct_part$test$idx ==
                              which(election_county$county == "santa barbara"))]
data.frame(County = "Santa Barbra County",
           glm_prediction = as.character(pred_SB_lda)) %>% pander()
# prediction using PCA
# get outcome
outcome <- election_county.2$candidate

# fix q
q <- 13

# compute pcs
pc_mx <- election_county.2[,-1] %>%
  scale(center = T, scale = T)
pc_svd <-svd(pc_mx)
```

```r
# get loadings
pc_loadings <- pc_svd$v[,1:q]

# compute PCs
pc <- pc_mx %*% pc_loadings %>% as.data.frame()
colnames(pc_loadings) <- colnames(pc)<- paste('PC',1:q,sep = '')

# append as inputs to class labels
pc_data<- tibble(candidate=factor(outcome)) %>%
  bind_cols(as_data_frame(pc))

# train test separate
set.seed(11111)
pc_data_part <- resample_partition(pc_data, p = c(test = 0.2, train = 0.8))
train_pc <- as_tibble(pc_data_part$train)
test_pc <- as_tibble(pc_data_part$test)

# logistic regression predict
fit_glm_pc <- glm(candidate ~ ., family = 'binomial', data = train_pc)
# compute estimated probabilities
p_hat_test_pc <- predict(fit_glm_pc, test_pc, type = 'response')
# bayes classifier
y_hat_test_pc <- factor(p_hat_test_pc > 0.5, labels = c("Donald Trump", "Hillary Clinton"))

# misclassification errors
glm_err_pc <- mean(test_pc$candidate != y_hat_test_pc)
data.frame(Misclassification_Error = glm_err_pc) %>% pander()
# predict SB vote using PCA
pred_SB_pc <- y_hat_test_pc[which(election_ct_part$test$idx ==
                              which(election_county$county == "santa barbara"))]
data.frame(County = "Santa Barbra County",
           glm_prediction = as.character(pred_SB_lda)) %>% pander()
```