Data_Wrangling_Project

Chuqiao Liu

5/1/2020

Abstract

2020 is an election year. However, predicting voter behavior is complicated for many reasons despite the tremendous effort in collecting, analyzing, and understanding many available datasets. For my final project, I analyzed the 2016 presidential election dataset, visualized winning candidates by state level, and found connections between election and census data. I explore the questions including: relationships between voting results and unemployment rate relationships between voting results and state population

Data

In this project, I worked on three datasets election data, census data and column metadata. First, I imported data from the local file.

Election data

Following is the first few rows of the 'election.raw' data:

county	fips	candidate	state	votes
NA	US	Donald Trump	US	62984825
NA	US	Hillary Clinton	US	65853516
NA	US	Gary Johnson	US	4489221
NA	US	Jill Stein	US	1429596
NA	US	Evan McMullin	US	510002

county	fips	candidate	state	votes
NA	US	Darrell Castle	US	186545

```
## [1] 18351     5
```

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. some rows in `election.raw` are summary rows. These rows have `county` value of `NA`. There are two kinds of summary rows:

- Federal-level summary rows have `fips` value of `US`.
- State-level summary rows have names of each states as `fips` value.

Census data

Following is the first few rows of the `census` data:

CensusTract	State	County	TotalPop	Men	Women	Hispanic	White	Black	Native	Asia
1001020100	Alabama	Autauga	1948	940	1008	0.9	87.4	7.7	0.3	0.
1001020200	Alabama	Autauga	2156	1059	1097	0.8	40.4	53.3	0.0	2.
1001020300	Alabama	Autauga	2968	1364	1604	0.0	74.5	18.6	0.5	1.
1001020400	Alabama	Autauga	4423	2172	2251	10.5	82.8	3.7	1.6	0.
1001020500	Alabama	Autauga	10763	4922	5841	0.7	68.5	24.8	0.0	3.
1001020600	Alabama	Autauga	3851	1787	2064	13.1	72.9	11.9	0.0	0.

```
## [1] 74001    37
```

`census` is a large dataset containing 36 variables and 74001 data points.
Census data: column metadata

Column information is given in `metadata`. Following is the first few rows of the `census` data:

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
Hispanic	% of population that is Hispanic/Latino	numeric
White	% of population that is white	numeric

CensusTract	Census.tract.ID	numeric
Black	% of population that is black	numeric
Native	% of population that is Native American or Native Alaskan	numeric
Asian	% of population that is Asian	numeric
Pacific	% of population that is Native Hawaiian or Pacific Islander	numeric
Citizen	Number of citizens	numeric
Income	Median household income ($)	numeric
IncomeErr	Median household income error ($)	numeric
IncomePerCap	Income per capita ($)	numeric
IncomePerCapErr	Income per capita error ($)	numeric
Poverty	% under poverty level	numeric
ChildPoverty	% of children under poverty level	numeric
Professional	% employed in management, business, science, and arts	numeric
Service	% employed in service jobs	numeric
Office	% employed in sales and office jobs	numeric
Construction	% employed in natural resources, construction, and maintenance	numeric
Production	% employed in production, transportation, and material movement	numeric
Drive	% commuting alone in a car, van, or truck	numeric
Carpool	% carpooling in a car, van, or truck	numeric
Transit	% commuting on public transportation	numeric
Walk	% walking to work	numeric
OtherTransp	% commuting via other means	numeric
WorkAtHome	% working at home	numeric
MeanCommute	Mean commute time (minutes)	numeric
Employed	% employed (16+)	numeric
PrivateWork	% employed in private industry	numeric
PublicWork	% employed in public jobs	numeric
SelfEmployed	% self-employed	numeric
FamilyWork	% in unpaid family work	numeric
Unemployment	% unemployed	numeric

There are some interesting varaibles in the `census` data. For example, commuting vehicles and different working status.

Data wrangling

I removed summary rows from `election.raw` data:

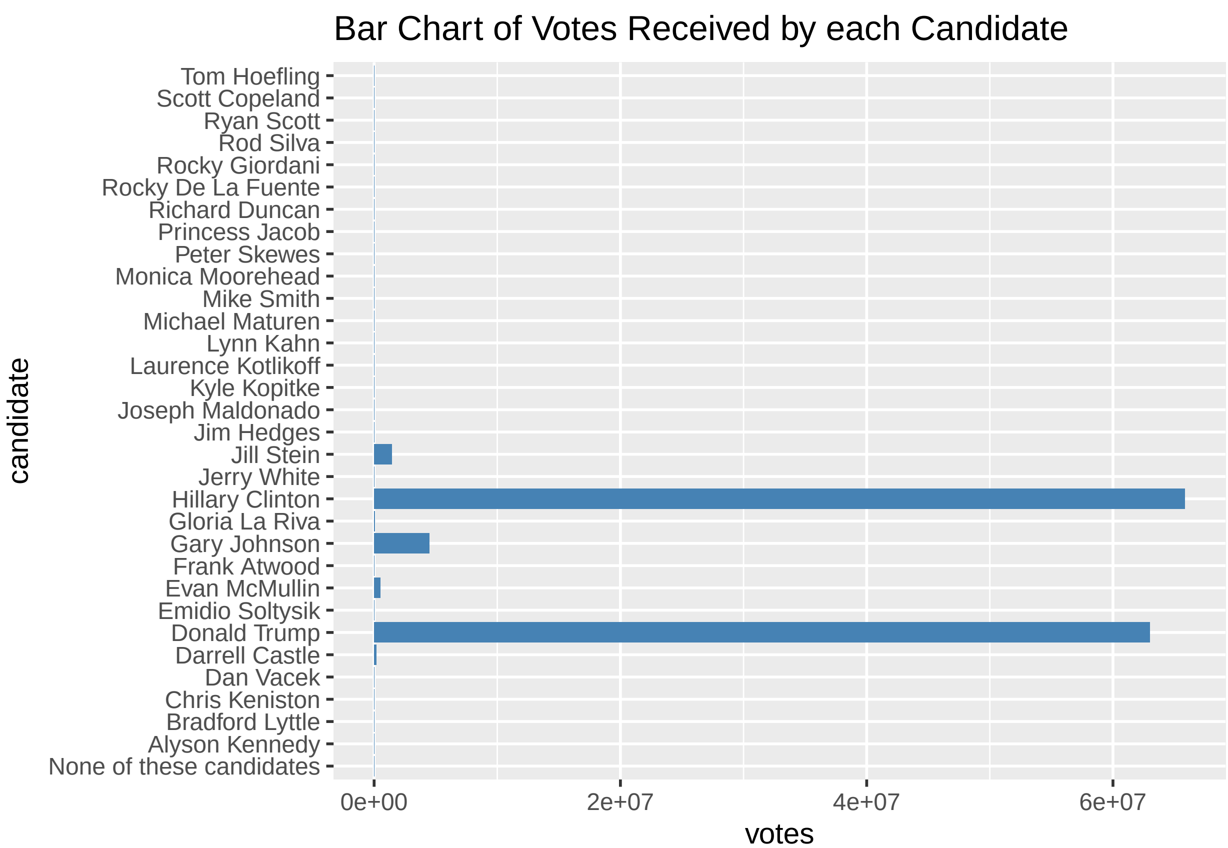
```
* Federal-level summary into a `election_federal`.
```

* State-level summary into a `election_state`.

* Only county-level data is to be in `election`.

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

Based on the election data set, there were 32 named presidential candidates in the 2016 election. And we draw bar chart of all votes reveived by each candidate.



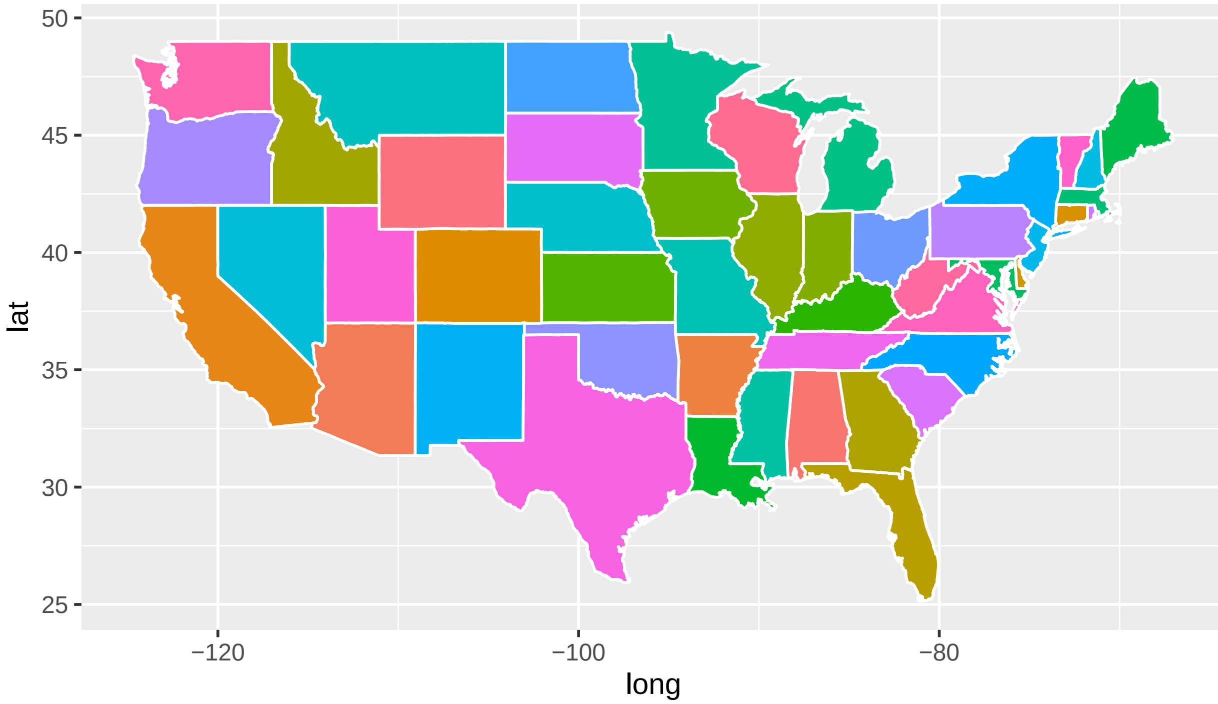
We can clearly see from the bar chart that Hillary Clinton and Donald Trump won substantially more votes than other candidates.

Next, I created new variables `county_winner` and `state_winner` by taking the candidate with the highest proportion of votes.

Visualization

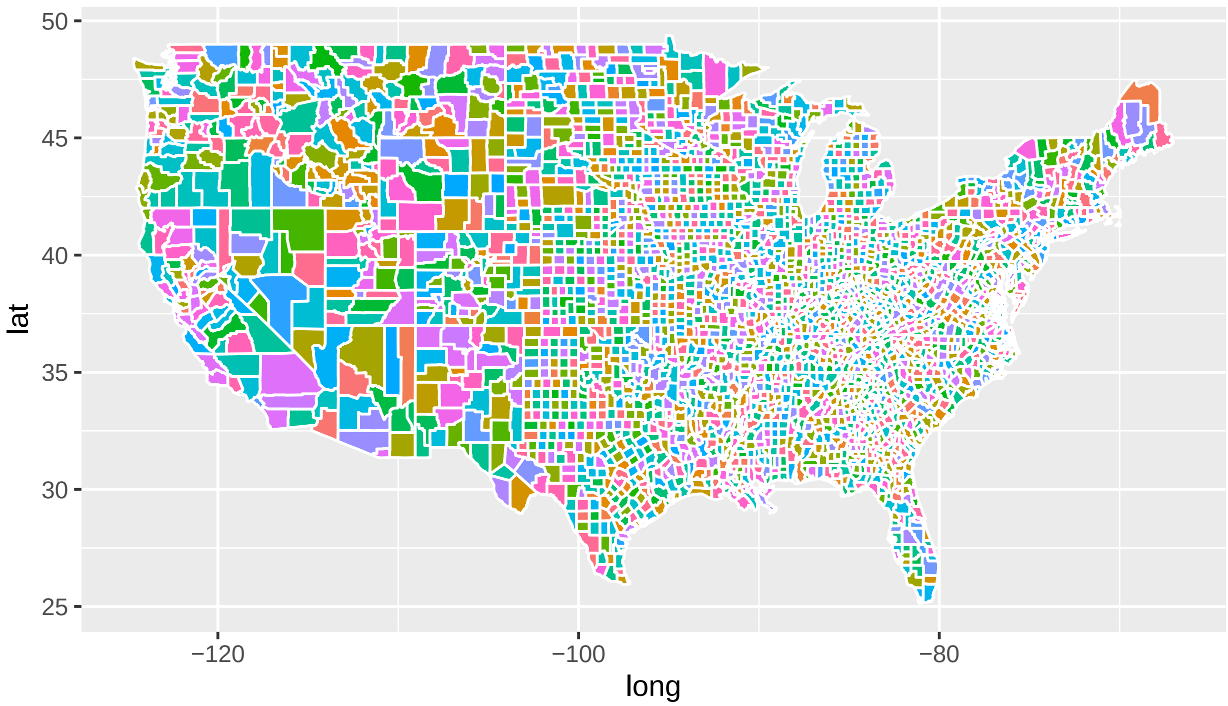
Visualization is crucial for gaining insight and intuition during data wrangling. I mapped data onto maps.

The R package `ggplot2` can be used to draw maps.



The variable `states` contain information to draw white polygons, and fill-colors are determined by `region`.

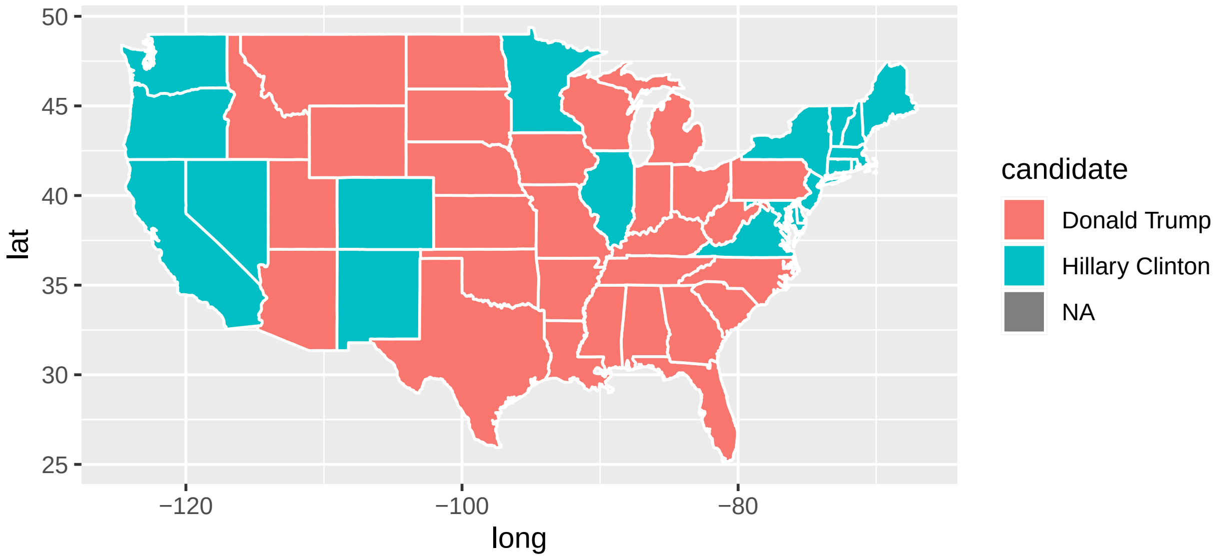
Then, I draw county-level map by creating `counties = map_data("county")`. Color by county



Next, I colored the map by the winning candidate for each state.
First, I combined `states` variable and `state_winner` I created earlier using `left_join()`. Note that `left_join()` needs to match up values of states to join the tables; however, they are in different formats: e.g. `AZ` vs. `arizona`.
Before using `left_join()`, I created a common column by creating a new column for `states` named `fips = state.abb[match(some_column, some_function(state.name))]`.
I replaced `some_column` and `some_function` to complete creation of this new column. Then `left_join()`.
The figure that I had looks similar to state_level [New York Times map] (https://www.nytimes.com/elections/results/president).

```
states = states %>%
  mutate(fips = state.abb[match(region, tolower(state.name))])
states_win = left_join(states, state_winner, by="fips")

ggplot(data = states_win) +
  geom_polygon(aes(x = long, y = lat, fill = candidate, group = group), color = "white")
  coord_fixed(1.3)
```




```
#  guides(fill=FALSE)
```

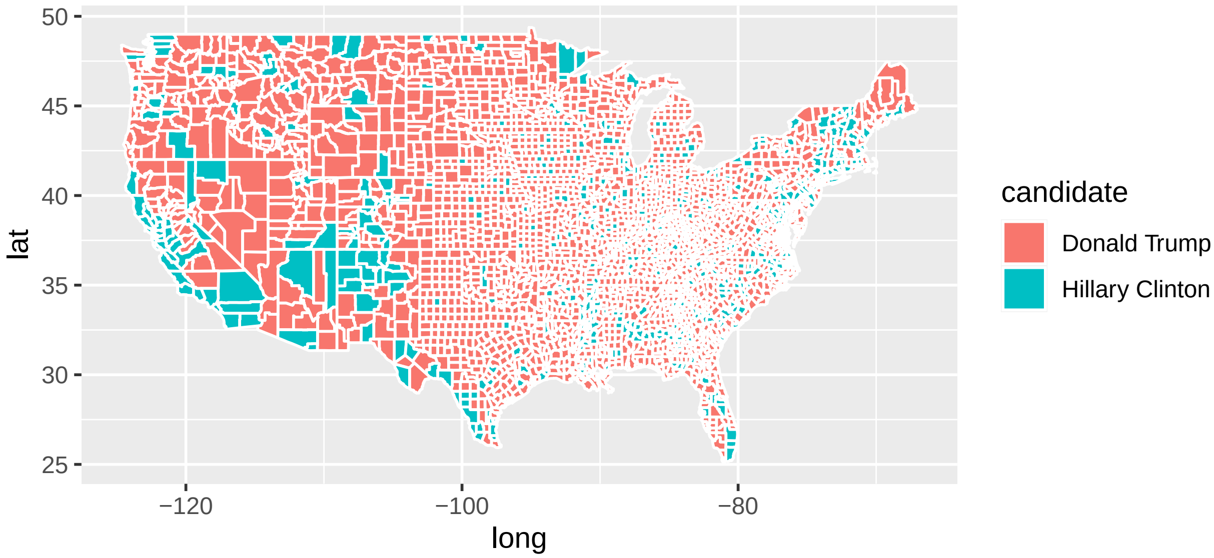
The “NA” in the map was caused by the region of District of Columbia, which was not included in any states.

The variable `county` does not have `fips` column. So I created 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`. The figure that I had looks similar to county-level New York Times map.

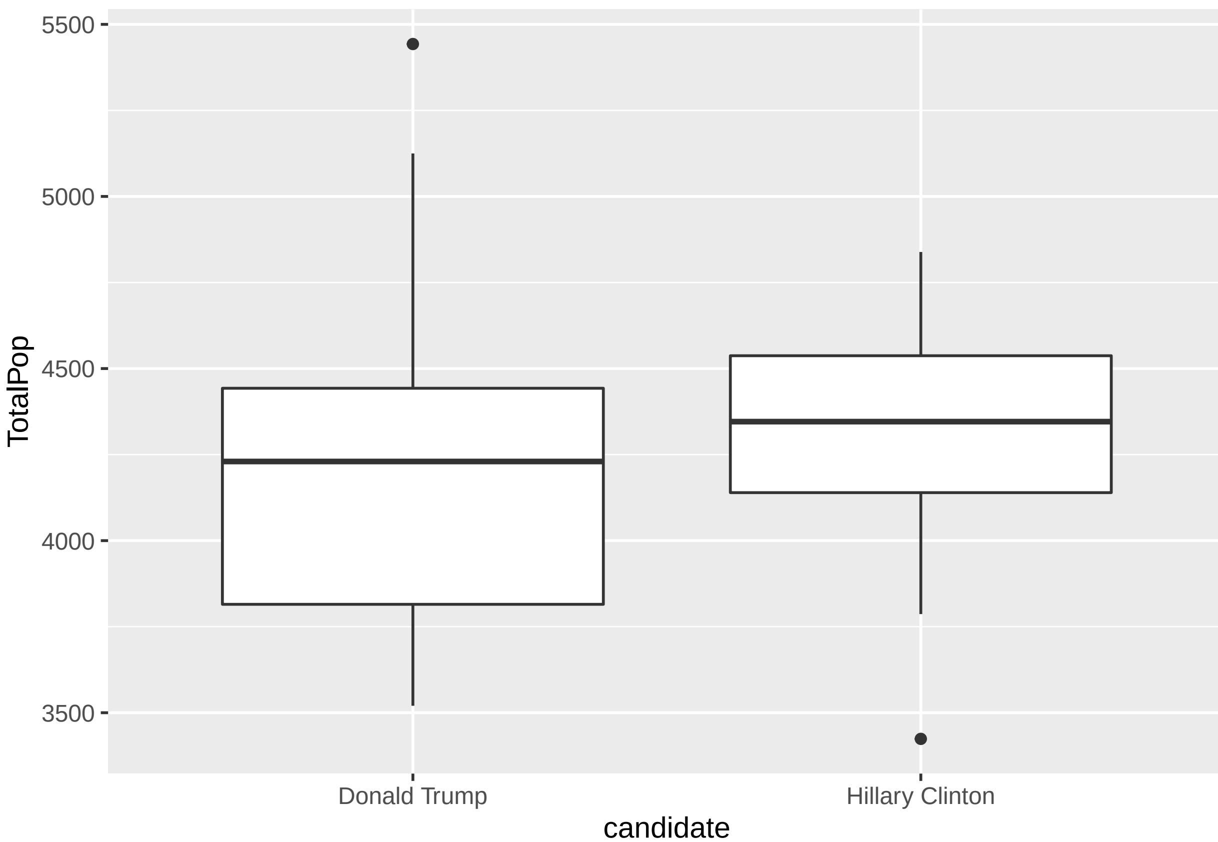
```
county.fips = as.data.frame(maps::county.fips)
county.fips = county.fips %>%
  separate(polyname, c("region","subregion"), sep=",") %>%
  separate(subregion, c("subregion", "part"), sep=":")
## change the county name of shannon to oglala lakota ##
## with corresponding fips (updated in May, 2015) ##
county.fips[county.fips$fips == "46113",]$subregion = "oglala lakota"
county.fips[county.fips$fips == "46113",]$fips = "46102"

county = left_join(counties, county.fips, by="subregion")
county$fips = as.factor(county$fips)
county_win = left_join(county, county_winner, by="fips")

ggplot(data = county_win) +
  geom_polygon(aes(x = long, y = lat, fill = candidate, group = group), color = "white")
  coord_fixed(1.3)
```



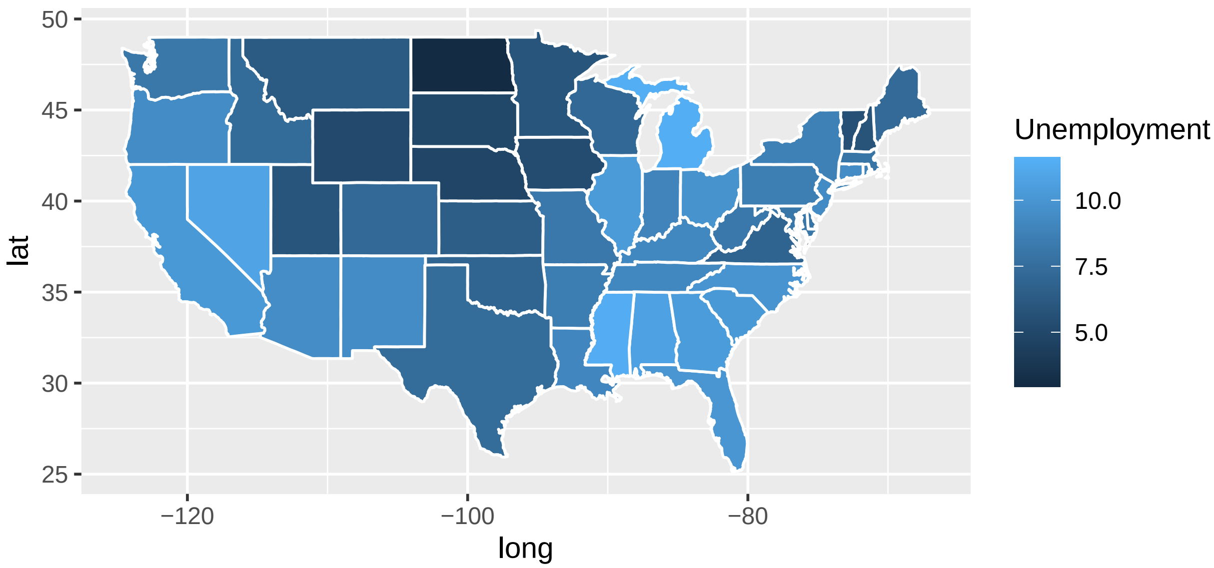
Many exit polls noted that demographics played a big role in the election. I Used this Washington Post article and this R graph gallery for ideas and inspiration.



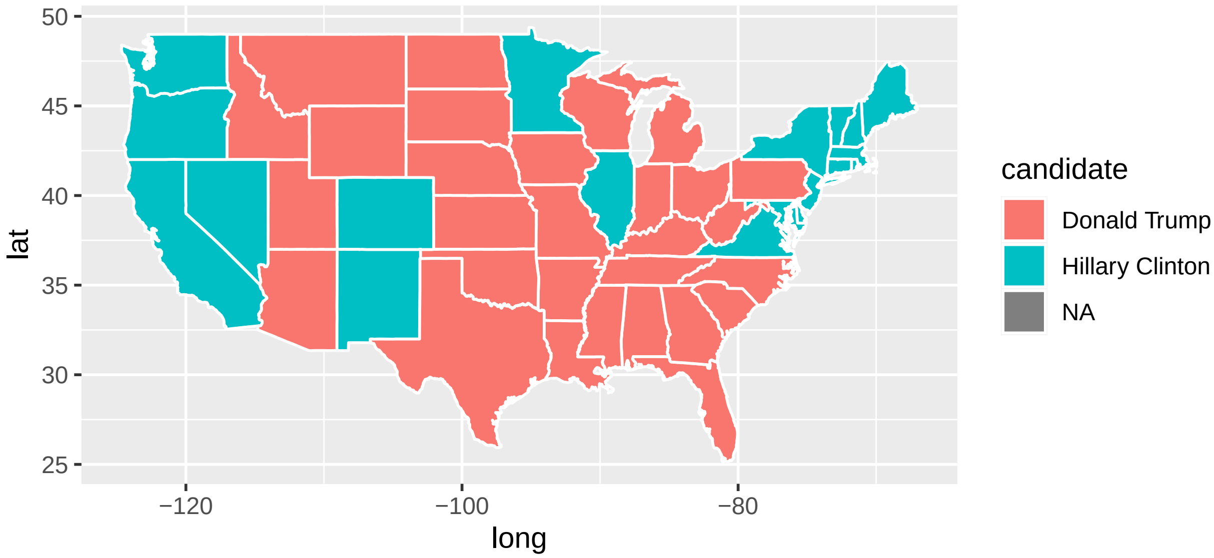
I make this boxplot to compare the total population of the state voting for the different candidate. We can see that Hillary Clinton get the votes from the large population state while the small population state perfer the Donald Trump.

```
Unemploy = census %>%
  group_by(State) %>%
  summarise_at("Unemployment", funs(mean(., na.rm=TRUE)))
Unemploy$region = tolower(Unemploy$State)
states = map_data("state")
states$region = as.factor(states$region)
states_unemploy = left_join(states, Unemploy, by="region")

ggplot(data = states_unemploy) +
  geom_polygon(aes(x = long, y = lat, fill = Unemployment, group = group), color = "whit
  coord_fixed(1.3)
```




```
# Compare to the state-level winning candidate map
ggplot(data = states_win) +
  geom_polygon(aes(x = long, y = lat, fill = candidate, group = group), color = "white")
  coord_fixed(1.3)
```



By comparing the state-level unemployment map with state-level winning candidate map, I found that for map area between longitude -125 and -90, states with higher unemployment rate are more likely to have Hillary Clinton as the state winner;
but for map area between longitude -90 and -65, states with higher unemployment rate are more likely to have Donald Trump as the state winner. There should be other important factors that outweight the effect of unemployment rate on predicting the state-level winner.

The `census` data contains high resolution information (more fine-grained than county-level). I aggregated the information into county-level data by computing `TotalPop`-weighted average of each attributes for each county. I also created the following variables:

- *Clean census data* ***`census.del`***: start with `census`, I filtered out any rows with missing values and I converted {`Men`, `Employed`, `Citizen`} attributes to a percentages (meta data seems to be inaccurate).
  Then, I computed `Minority` attribute by combining {Hispanic, Black, Native, Asian, Pacific}, 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`}, I used `add_tally()` to compute `CountyTotal`. Also, I computed the weight by `TotalPop/CountyTotal`.

- *County census data,* ***`census.ct`***: start with `census.subct`, I used `summarize_at()` to compute weighted sum

Here are few rows of `census.ct_`:

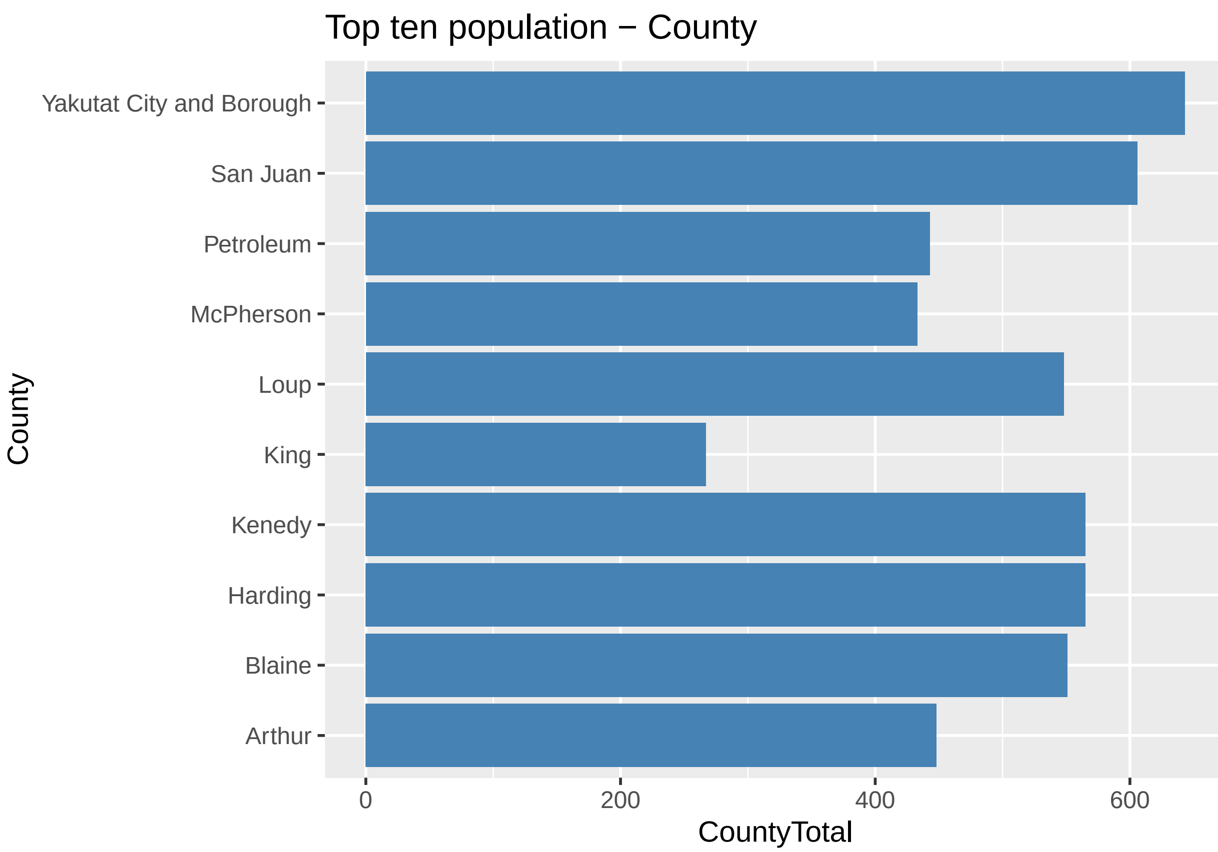
State	County	Men	White	Citizen	Income	IncomeErr	IncomePerCap	IncomePe
Alabama	Autauga	48.43266	75.78823	73.74912	51696.29	7771.009	24974.50	3

State	County	Men	White	Citizen	Income	IncomeErr	IncomePerCap	IncomePe
Alabama	Baldwin	48.84866	83.10262	75.69406	51074.36	8745.050	27316.84	3
Alabama	Barbour	53.82816	46.23159	76.91222	32959.30	6031.065	16824.22	2
Alabama	Bibb	53.41090	74.49989	77.39781	38886.63	5662.358	18430.99	3
Alabama	Blount	49.40565	87.85385	73.37550	46237.97	8695.786	20532.27	2
Alabama	Bullock	53.00618	22.19918	75.45420	33292.69	9000.345	17579.57	3

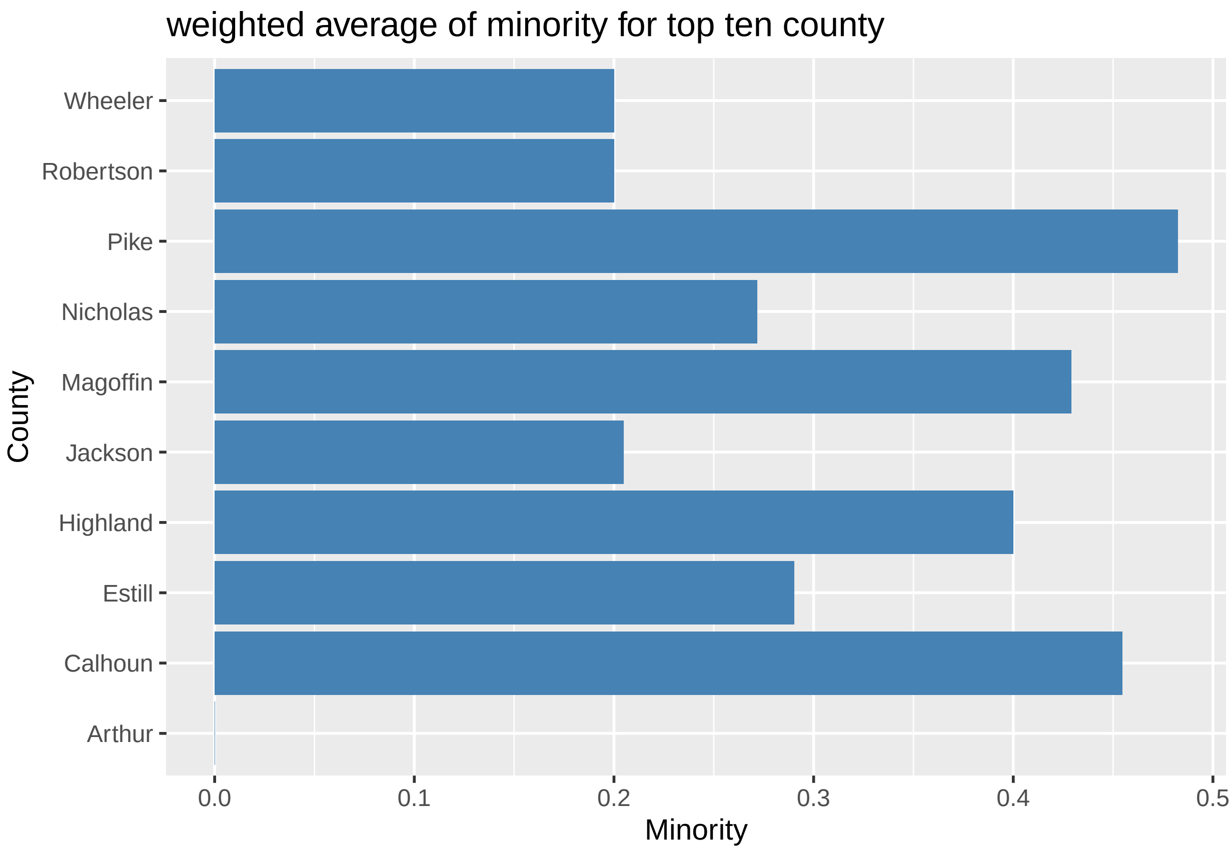
```
## [1] 3218   28
```

Here I draw two graphs to visualize more details of variables that I created.

```
## Adding missing grouping variables: `State`
```




```
## Adding missing grouping variables: `State`
```



`census.ct` can be very useful if we want to manipulate census data at a county level.

Conclusion

The election data containing several levels including federal, state, county, and sub-county level. The census dataset also contains data from different region at different levels. Difficulties of this project includes understanding variables, cleaning dataset, aggregate data into different levels.

Future Work

More work can be done to construct more complicated political relevant questions and visualization. This can be done by creating a user interface using the Rshiny package.