2020 US Presidential Elections: The Swing States

'CYO' project, edX HarvardX PH125.9x 'Data Science: Capstone'

jrollb

6 Jan, 2020

Contents

Introduction

Executive summary

By now, the 2020 US presidential elections have been decided. The electoral college has elected Joe Biden as the 46th president of the United States. Prior to that, the US Supreme Court has rejected all law suits that claimed that the elections were irregular. The 45th president Donald Trump still insists that the election was fraudulent. However, no evidence could be produced. On the contrary, vote recounts confirmed the result.

Can an analysis of the published election data give some insight here? In this assignment, I will ask the question whether the swing states moved significantly against the trend predicted from other states. Why? Had the swing states flipped due to fraud, a significant deviation from the nationwide trends in favor of Joe Biden should be visible. Of course, the reverse conclusion is not possible, i.e., significant deviations do not necessarily signal fraud.

More explicitly, I will hold out ten 'battleground' states (i.e. those where close results were expected), including the swing states, but also others such that amount of electoral votes for both parties is equal. Based on the data of the remaining states I fit a model to the changes between 2016 and 2020 and 'predict' the outcome of the states held out for validation. It is important to stress that this is not a prediction of the result of the 2020 election before it took place. Instead, it is an attempt to capture the nationwide trends in a model in order to formulate an expectation of the outcome in the held-out states.

The analysis presented here shows a different result: Most of the counties show no signals of a large deviation from the general character of the election, i.e., an increased overall voter turnout, but with a general trend favoring Biden. The by far largest deviation is in favor of the Republican candidate: Miama-Dade in Florida. It can be linked to the high number of Latinos voting for Trump in this election. A by far smaller deviation was the Atlanta (GA) area. This is within the expected model error. Besides, the votes have been recounted and the strong support for Biden can be explained. The analysis did not produce any signal of irregularities.

There are limitations to this analysis, so it would be inappropriate to oversell the results as hard evidence of any kind. I will describe the data used and will discuss the limitations in the summary section at the end. A major difficulty is related to prevalence. There is a large number of rural counties with very low population (typically favoring Republicans) and a small number of urban counties with very high population (typically favoring Democrats). This made modeling very challenging and required several attempts.

I used local regression ('loess') with a single feature to capture the general trends. This way I tried to reduce the prevalence problem, as 'loess' performs separate regressions for the high and low population areas. Then I applied linear regression, CART, random forest, extreme gradient boosting, and k-nearest neighbors. I picked the random forest model as final model. Finally, I subtracted the model predictions from the actual vote differences between the two candidates and analyzed the deviations.

Data

After starting with one data set, data quality issues made it necessary to merge three data sets into one. The three sets are:

- Data set "Election, COVID, and Demographic Data by County"
  - Contains results from the US presidential elections 2016 and 2020 by county
  - Source: Kaggle, link https://www.kaggle.com/etsc9287/2020-general-election-polls
  - Copyright label: The data set is labeled "CC0: Public Domain". As a sense check, the data contains public voting outcomes, public corona case numbers, and public census data.
- Data set "US Election 2020"
  - Contains results from the US presidential elections 2020 by county
  - Source: Kaggle, link https://www.kaggle.com/unanimad/us-election-2020
  - Copyright label: "CC0: Public Domain"
- Data set "County Presidential Election Returns 2000-2016"
  - Contains results from the US presidential elections 2000-2016 by county

  - Source: MIT Election Data + Science Lab (https://electionlab.mit.edu/data), link: https://doi.org/10.7910/DVN/VOQCHQ
  - Copyright label: CC0 - "Public Domain Dedication"

Goals

US presidential election are a huge topic, such that a reasonably limited goal has to be set. The specific question I want to investigate here is whether the results in the swing states can be explained by the average features of all the other states. If fraud had been present, this should be visible as deviations from such a model. The absence of inexplicable deviations would make fraud of the scale claimed unlikely.

The goal is to derive a model describing the general trends of the 2020 US presidential elections and use it to detect deviations. For the latter, I will try to decide whether they can be explained or not.

Key steps

The key steps are

- Put together a data set with good data quality
- Select a few 'battleground states' (i.e., states that had the potential to flip the result of the previous election) as hold-out sample
- Fit various models to the spread changes and compare their performance
  - As a first step fit the average changes by local regression
  - As a second step explain the residuals in terms of demographic and geographic features
- Make a prediction for the outcomes in counties within the hold-out sample of 'battleground states'
- Analyze the largest deviations of the actual outcome from the model prediction
- Interpret the result

Overarching assumptions

For this project I determine a hold-out sample of 'battleground states', i.e. those states where a close result could be expected. The selection is not random and so there is some risk to introduce a bias. However, my assumption (null hypothesis, if you will) is that these states follow the general "trend" and I am looking for signs against it. The selection includes five 'swing states', and I added Republican dominated states to match the number of electoral votes. I think that brings in a fair amount of ballance. The selection is

```
# Define hold out sample of states
states_validation <- c("MI","WI","GA","PA","AZ","NC","IA","OH","FL","WV")
```

These states should have a total of 73 electoral votes for each the Democrats and the Republicans.

Methods

This section covers:

- Data pre-processing
- Data exploration and visualization
- Insights gained
- Modeling approach
- Final model training and prediction

Data pre-processing

I started out with the data set "Election, COVID, and Demographic Data by County", but soon noticed that the data quality was not sufficient. Thus, I joined the two other data sets listed above. This involved a considerable amount of data wrangling and data cleansing, which is provided in the accompanying R-file. The result of this is provided as RDS file.

I load this data set:

```
# Read the merged data set
Election.US.16.20 <- readRDS("RDS/Election.US.16.20.rds")
```

For some states the election data for 2016 are on county level and for 2020 on township level or other administrative district levels. The ones I could not repair I will now identify and exclude:

```
# Identify gap states
gap_states <- Election.US.16.20 %>%
  filter(is.na(TotalPop) | is.na(votes.TOT.16) | is.na(votes.TOT.20)) %>%
  pull(state) %>% unique() %>% sort()
gap_states
```

```
## [1] "AK" "CT" "MA" "ME" "NH" "RI" "VT"
```

```
# Exclude them
Election.US.16.20 <- Election.US.16.20 %>%
  filter(!(state %in% gap_states))
```

For later use I define the relative change of vote numbers and the spreads and spread changes

```
# Add some columns needed later
Election.US.16.20 <- Election.US.16.20 %>%
  mutate(spread20 = pct.DEM.20 - pct.REP.20,
         spread16 = pct.DEM.16 - pct.REP.16,
         delta.spread = spread20 - spread16,
         delta.pct.total_votes = votes.TOT.20/votes.TOT.16 - 1,
         pct.Employed = Employed/TotalPop,
         pct.cases = cases/TotalPop)
```

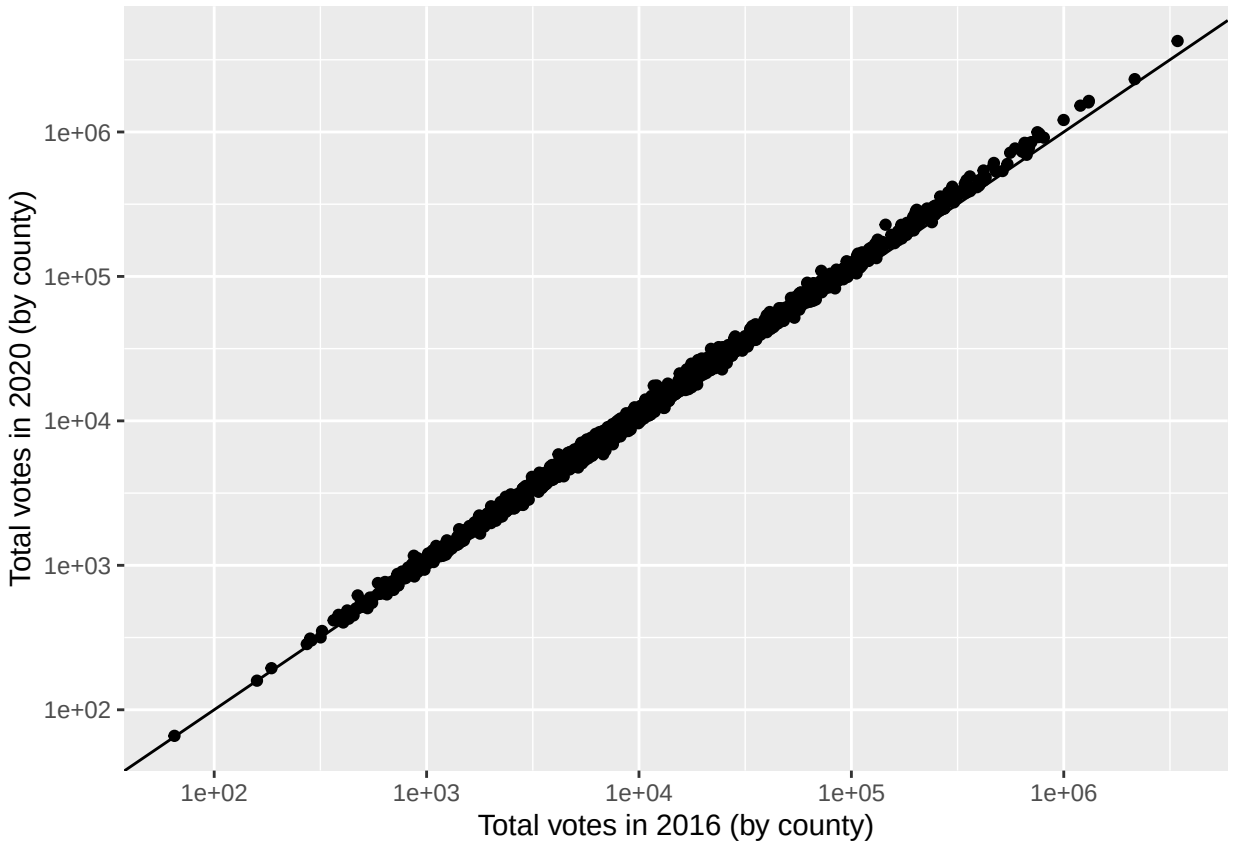
As indicated above, I split out a 'validation' set (the hold-out sample) and call the remaining data 'election'

```
# Split out the hold-out sample into 'validation' and the rest into 'election'
validation <- Election.US.16.20 %>%
  filter(state %in% states_validation)

election <- Election.US.16.20 %>%
  filter(!(state %in% states_validation))
```

The election data set is for data exploration and model fitting. Once a final model is selected, it will be applied to the validation set.

Data exploration and visualization

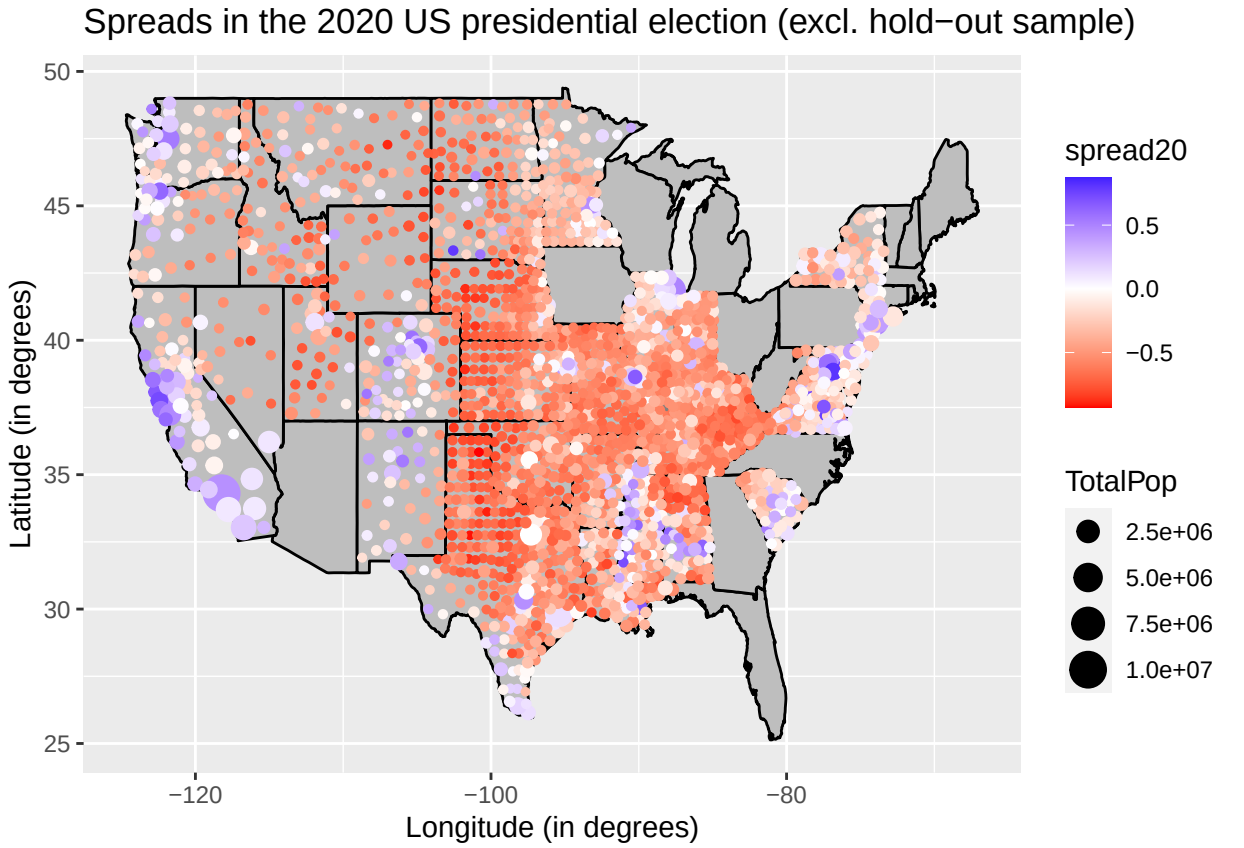
Understand the data quality In order to get an impression of the completeness of the data, I plot the total votes for the two election years



This plot looked much different before more recent updates of the vote numbers were used. It had many outliers, mostly 'raining' down from the straight line above. This was due to unfinished vote counts. With the updates and after filtering out states with county/township matching issues, the graph indicates a reliable relationship between past votes and votes in 2020. A slight slope increase can already be observed, i.e., the voter turnouts increased compared to 2016.

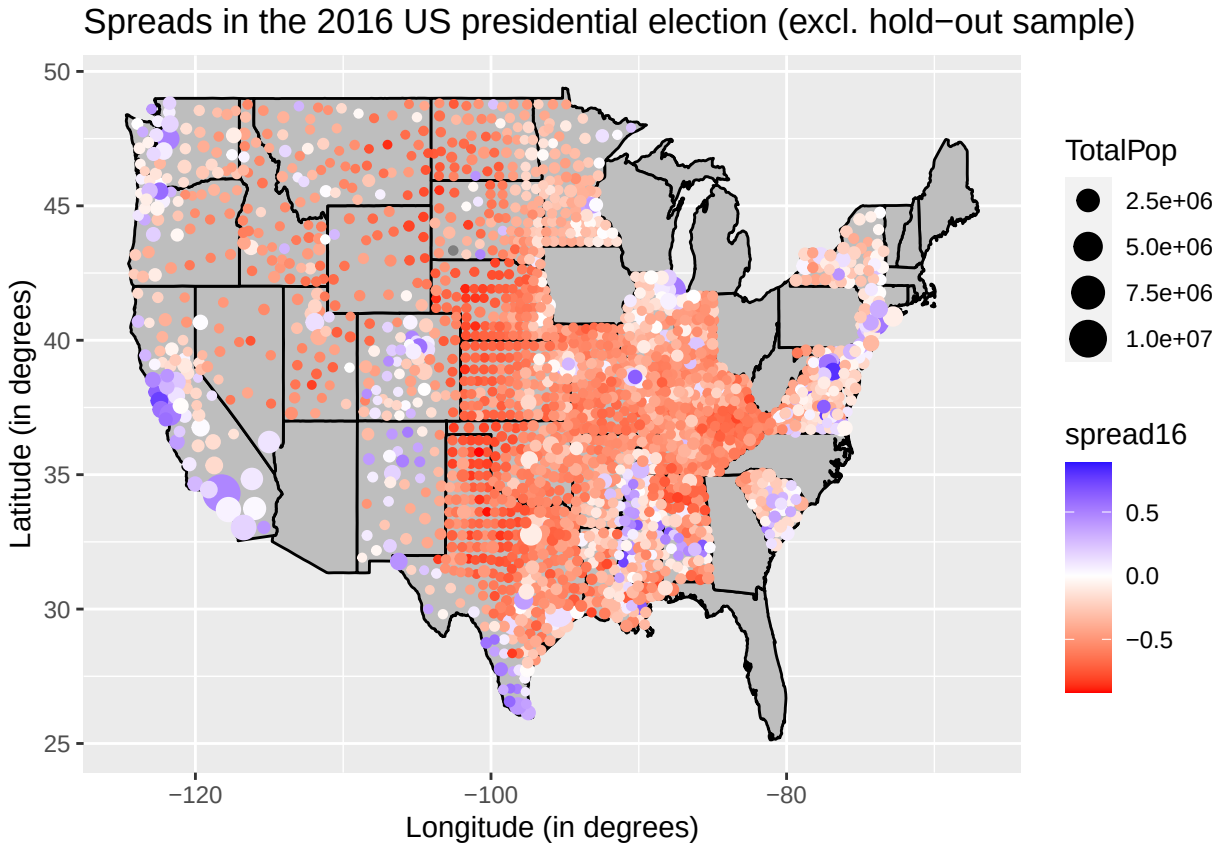
Visualization of the main features Apart from the voter turnout increase, the distribution of spreads over the country are interesting. Let us first illustrate the spread for the 2020 election, i.e., the difference of the percentage of votes for Joe Biden and those for Donald Trump. Note that the states held out for validation are empty to avoid 'snooping'. The color coding indicates the party dominance, i.e., blue counties voted predominantly for Biden, red ones for Trump. The size of points are chosen to be the population of the respective county. The plot only shows the contingent states (excluding Hawaii and Alaska).

```
election %>%
  filter(!(state %in% c("AK","HI")) & long != 0 & lat != 0) %>%
  ggplot() +
  geom_polygon(data = state_map,aes(long,lat,group = group),fill="grey",color="black") +
  geom_point(aes(long,lat,color=spread20,size=TotalPop)) +
  scale_color_gradient2(low="red",midpoint = 0.0,high = "blue") + # coord_fixed() +
  ggtitle("Spreads in the 2020 US presidential election (excl. hold-out sample)") +
  xlab("Longitude (in degrees)") +
  ylab("Latitude (in degrees)")
```



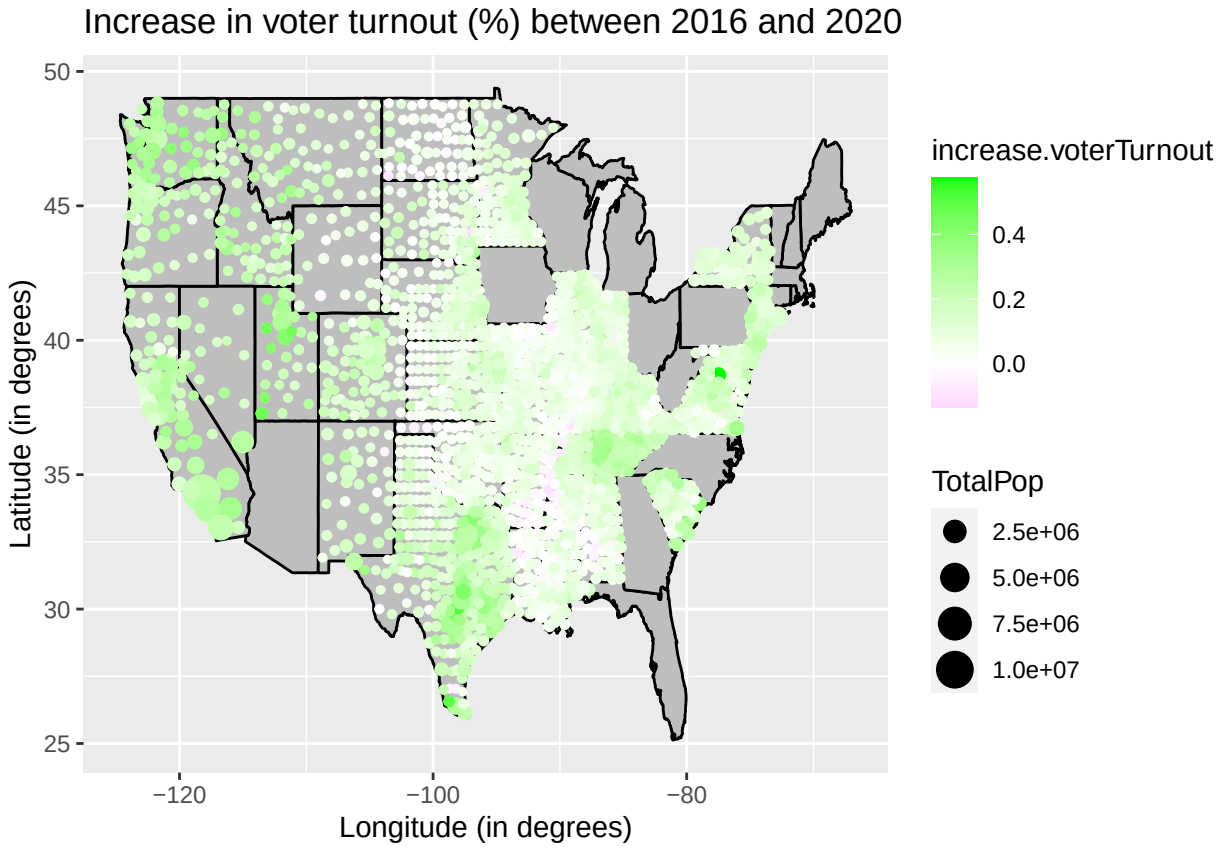
In many cases, the highly populated urban areas (major cities) are blue and the rural areas red. An example is Nevada, where most counties have a Republican majority, but the two big cities Las Vegas and Reno are lightly blue. But there are exceptions to this. For example, the Texan counties near the border to Mexico. Below I will look into that further.

The same plot for 2016 is nearly indistinguishable. I think it is instructive to show the plot to illustrate how similar the voting behavior is.



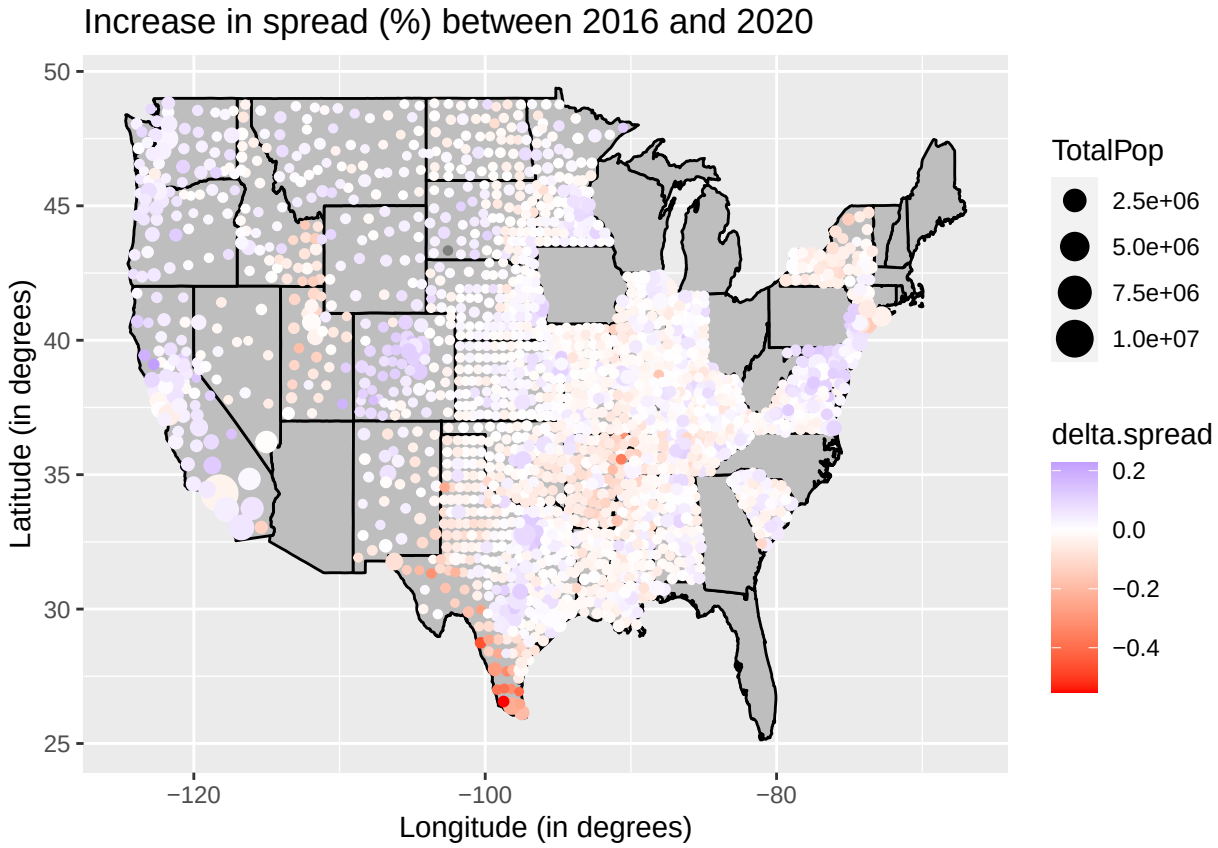
The only difference clearly visible is the southern border of Texas. Otherwise it is hardly possible to detect differences with the naked eye.

However, we can look at the differences. We can consider two effects. Firstly, one special feature of these elections was the high voter turnout due to voting by mail. I plot the relative change of the number of total votes, highlighted green for increase and magenta for decrease.



The voter turnout increased in many regions across the states with some areas staying at the same level. Counties with vote number decrease are the exception. Note that even with all spreads constant, the local voter turnout can shift the outcome from one to the other candidate. Suppose that (hypothetically) in a state the percentage of votes for Republicans and Democrats stay the same. If the total voter turnout in an urban area with Democrats dominating increases, it will shift the total votes of the state towards Democrats. And vice versa.

Secondly, the change in spread between the two elections, i.e., the difference of spreads 2020 and 2016, reflects the earlier observation that spreads only slightly changed. Zero change is shown in white color. There seems a widespread trend towards the Democratic side (blue), however, with a few regions moving toward Republican side (red).



The color coding in the graphs above indicates that nuances decide. People vote similar in each election and slight shifts in voter turnout and candidate preferences decide over the outcome.

There are several geographical regions where the change typically goes into one direction. This information is contained in the longitude and latitude (i.e., the variables 'lat' and 'long').

In the spread change graph above some counties in the southwest of Texas show changes above average. Let's do a sense check by looking at the ten largest changes:

state	county	delta.spread	votes.DEM.16	votes.REP.16	votes.DEM.20	votes.REP.20
TX	Starr	-0.5510832	9289	2224	9123	8247
TX	Maverick	-0.4638387	10397	2816	8332	6881
TX	Kenedy	-0.4002328	99	84	65	127
TX	Jim Hogg	-0.3887235	1635	430	1197	833
TX	Zapata	-0.3828382	2063	1029	1826	2033
AR	Poinsett	-0.3634074	1880	5502	1424	5918
TX	Duval	-0.3267884	2783	1316	2575	2443
TX	Brooks	-0.3199993	1937	613	1470	998
TX	Reeves	-0.3081527	1659	1417	1395	2254
AR	Clay	-0.3067493	1199	3781	962	4086

For example, in the county Starr in Texas a majority voted for the Democratic candidate both in 2016 and 2020. But there was a considerable increase in votes for the Republican candidate from 2016 to 2020. Could that be a data quality issue? The Office of Secretary of State in Texas provides election information at https://www.sos.state.tx.us/elections/index.shtml. For county Starr, https://elections.sos.state.tx.us/elchist319_county214.htm, there were 2,224 votes for the Republican candidate and 9,289 votes for the Democratic candidate. So the summary shows that the large difference of votes in 2016 can be confirmed in this case.

For 2020, the page https://results.texas-election.com/county shows 8,247 Republican and 9,123 Democrat votes at the time of writing, which is shown as final result. The data set used for this project is probably still not be final, so a few votes might still be inaccurate. But it shows that the relatively large changes between 2016 and 2020 can be confirmed.

Then the question is what the main features are. Below, we apply various algorithms to find them. The danger of searching them directly is the following. For example, for the counties in Texas, there is a predominant feature, a very high percentage of Hispanics, see the right column (numbers in %) in the following table:

state	county	delta.spread	Hispanic
TX	Starr	-0.5510832	99.2
TX	Maverick	-0.4638387	95.3
TX	Kenedy	-0.4002328	88.5
TX	Jim Hogg	-0.3887235	93.8
TX	Zapata	-0.3828382	94.1
AR	Poinsett	-0.3634074	2.8
TX	Duval	-0.3267884	89.3
TX	Brooks	-0.3199993	94.1
TX	Reeves	-0.3081527	75.0
AR	Clay	-0.3067493	1.9

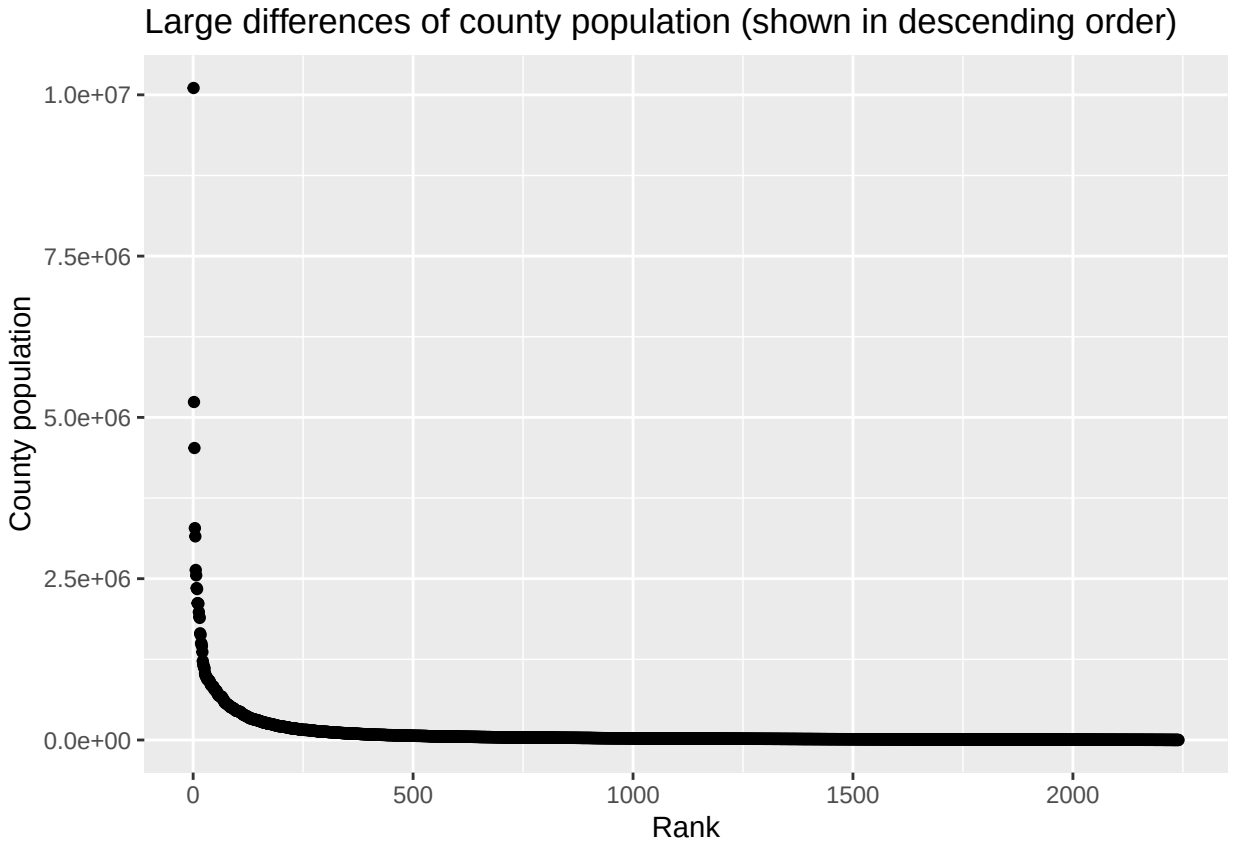
So, one could think that the Hispanics community voted more for Trump. However, while this is probably true for some groups, it is not valid for Hispanics in general. Looking at the highly populated areas, the behavior is opposite:

state	county	delta.spread	TotalPop	Hispanic
TX	Harris	0.0058982	4525519	42.2
TX	Dallas	0.0538718	2552213	39.6
TX	Tarrant	0.0888616	1983675	28.2
TX	Bexar	0.0462937	1892004	59.8
TX	Travis	0.0612323	1176584	33.9
TX	Collin	0.1239950	914075	15.1

In these counties, there is also a high percentage of Hispanics, but the trend was toward Democrats. So selecting from the most extreme cases might not be reliable.

Strategy to extract features In order to continue a bit more systematic, let us pick up the theme from above and assume that the spread is essentially the same as in the last election. This is best seen by plotting the spread 2020 against the spread 2016.

Before doing so, I would like to comment on the issue of prevalence. The population numbers of the counties vary over a very large range. There are many rural counties with a low population number and a much smaller number of urban counties with a very high population number.



This leads to the problem that unweighted averages will be highly weighted by the small rural counties due to their number. On the other hand, weighted averages will give emphasis to the urban counties.

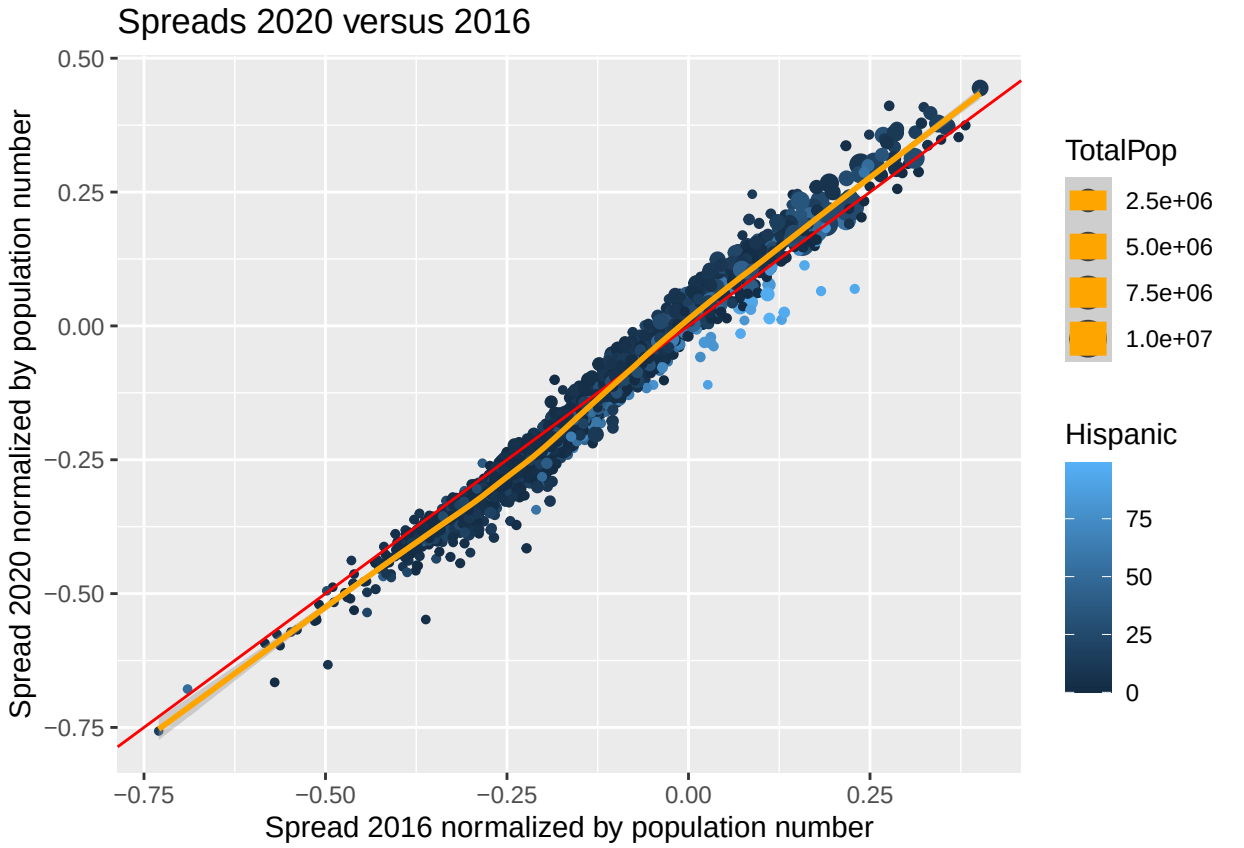
I will try the following approach:

- Properties scaling with the population number will be normalized with the total population of the county
- To capture the general nationwide features I will use local regression ('loess') in order to average only over counties with similar size. Note that there is a strong dependency of the spread on the population that allows for this.
- As many of the already scaled properties depend on population number, I assume that the machine learning algorithms will help separating the highly and weakly populated areas

The general nationwide features Let's plot the spread of the 2020 election against the spread of the 2016 election. As I will try to base the swing state prediction on the 2016 voting results, I will not use the common spread definition of Democrat votes minus Repuclican votes over the total number of votes. Instead I will divide by the population number, see "Delta.20.rel" and "Delta.16.rel" in the following code section.

```
election <- election %>%
  mutate(Delta.20.rel = (votes.DEM.20 - votes.REP.20)/TotalPop,
         Delta.16.rel = (votes.DEM.16 - votes.REP.16)/TotalPop)
```

The plot shows that the spreads in 2016 and 2020 are very similar

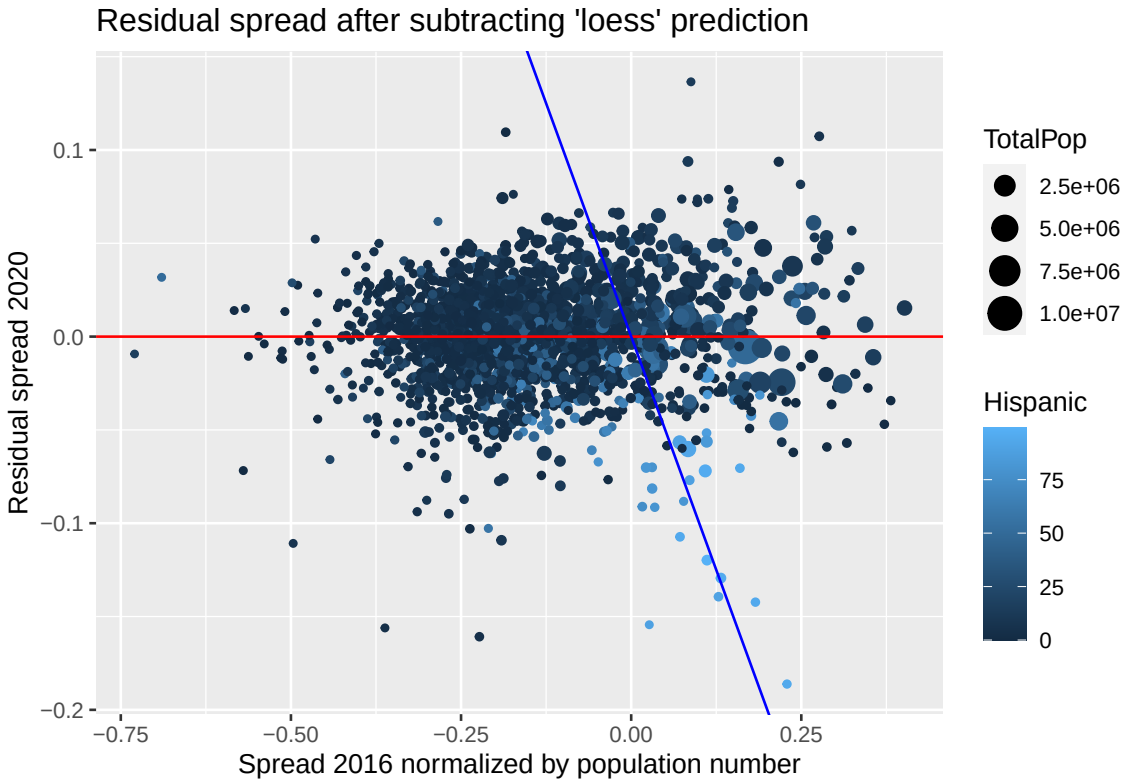


The red straight line marks where the spreads are equal. The orange 'loess' regression line shows that for Democrat voters the spread on average became more positive and for Republican voters the spread became more negative. This can be interpreted as **increasing polarization**. However, it is interesting to note that the lines cross at slightly negative spread values. This means that in counties with close results there was a tendency towards the Democrat side. These observations more or less fit the common narratives.

The second step consists of explaining the deviations from the average behavior (orange line). The large changes observed earlier can be made visible by coloring the counties with high Hispanic population, which contain the counties near the southern border. This is one example of deviations from the average behavior.

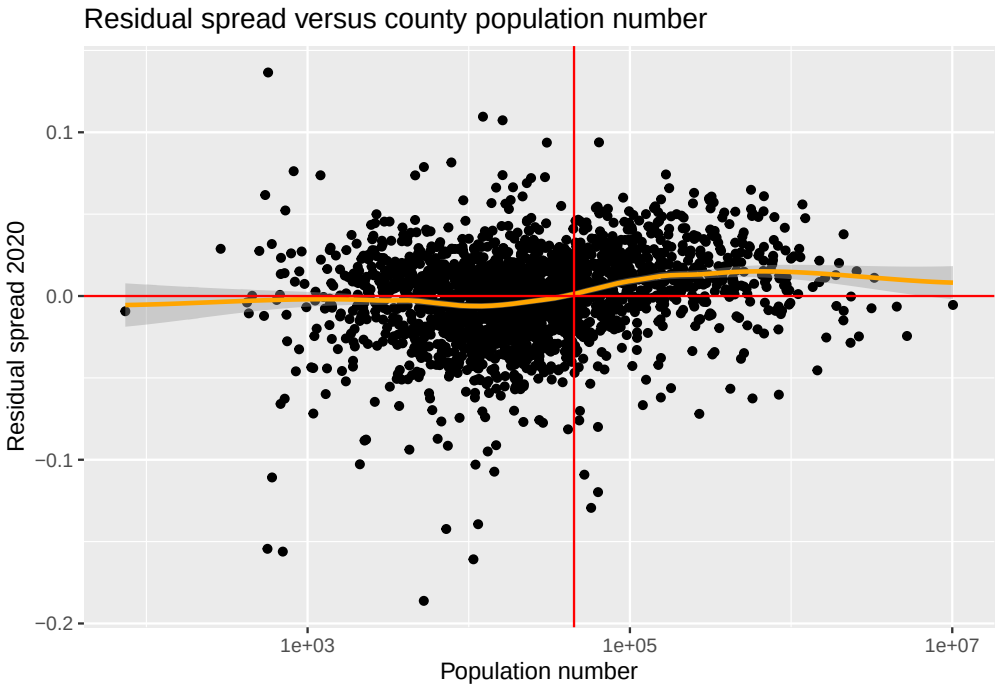
For that purpose I now subtract the 'loess' predictions:

```
fitLoessBaseline <- loess(formula = Delta.20.rel ~ Delta.16.rel,
                          data = election, span = 0.3, method.args = list(degree=1))

election <- election %>%
  mutate(Delta.20.rel.baseline = predict(fitLoessBaseline),
         residual.16.20 = Delta.20.rel - Delta.20.rel.baseline)
```

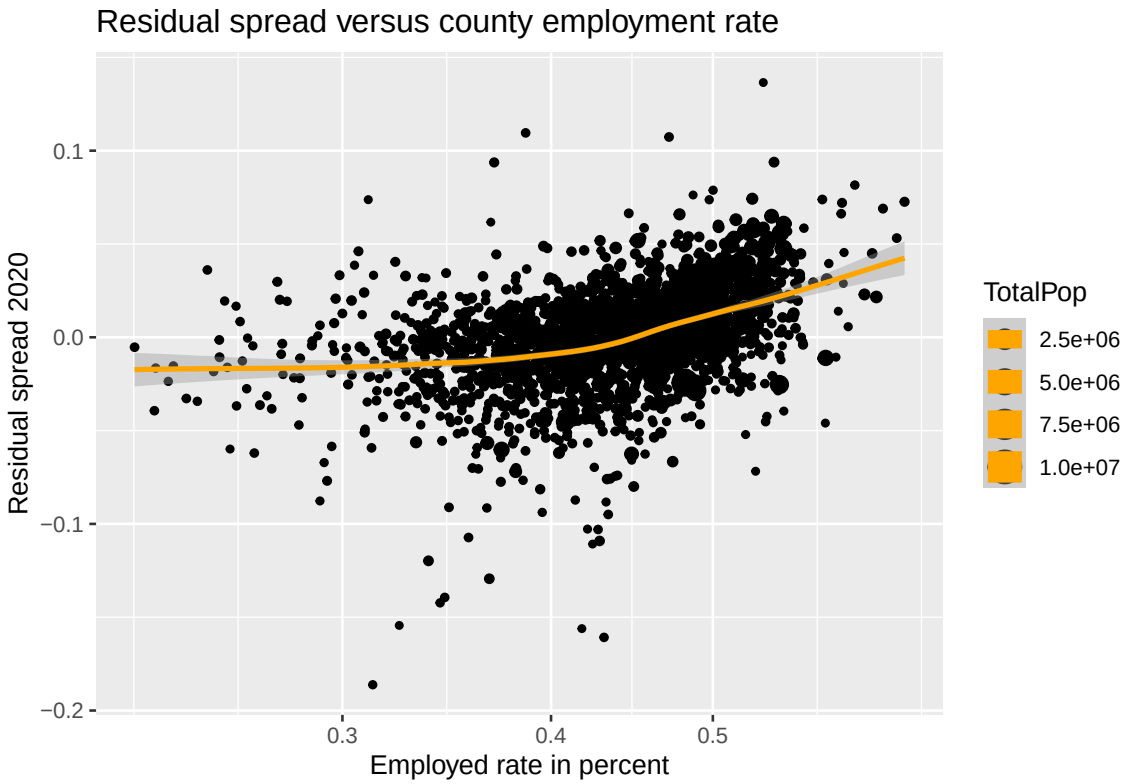


The red line marks the zero residual with respect to the "baseline" model (using "loess"). The dots close to the blue line (slope -1) are those where a sign switch of the 2020 spread in the respective county is possible. The cloud of blue dots on the lower right has already been identified as candidate for a feature. But what else can we find?

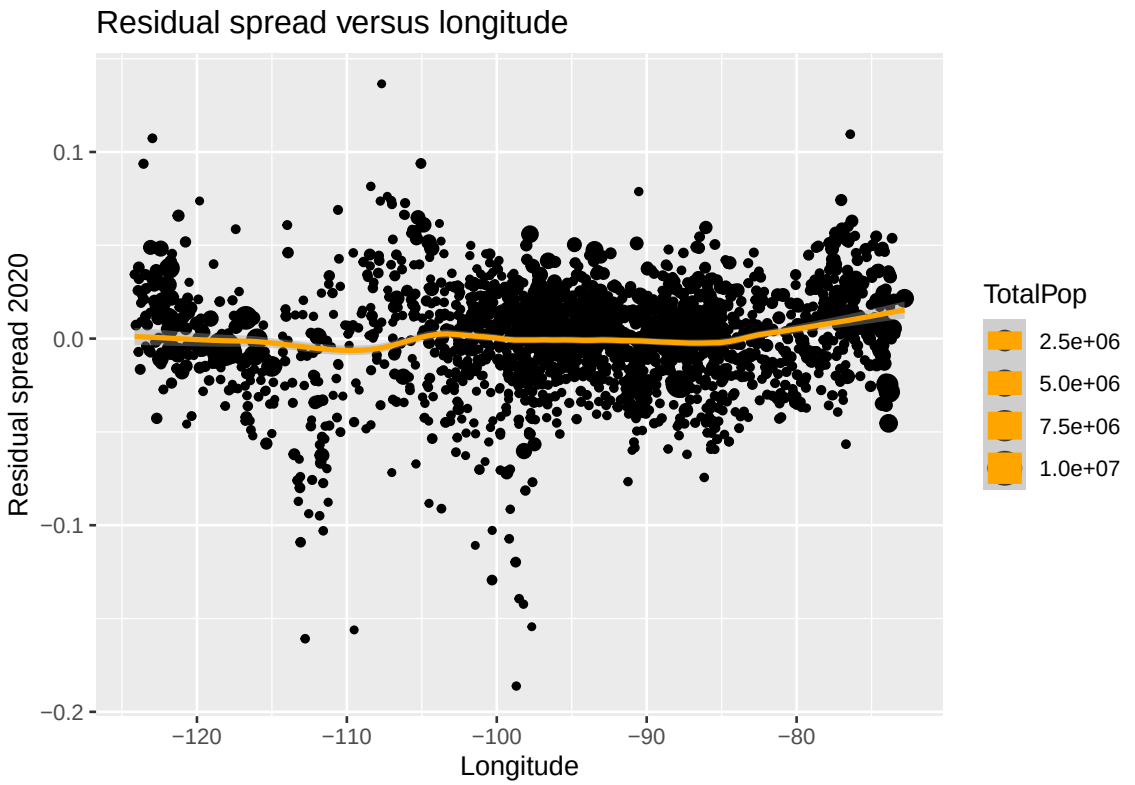
Let's first look at the dependency between the residuals (over the baseline) and the population, plotted with logarithmic abscissa:



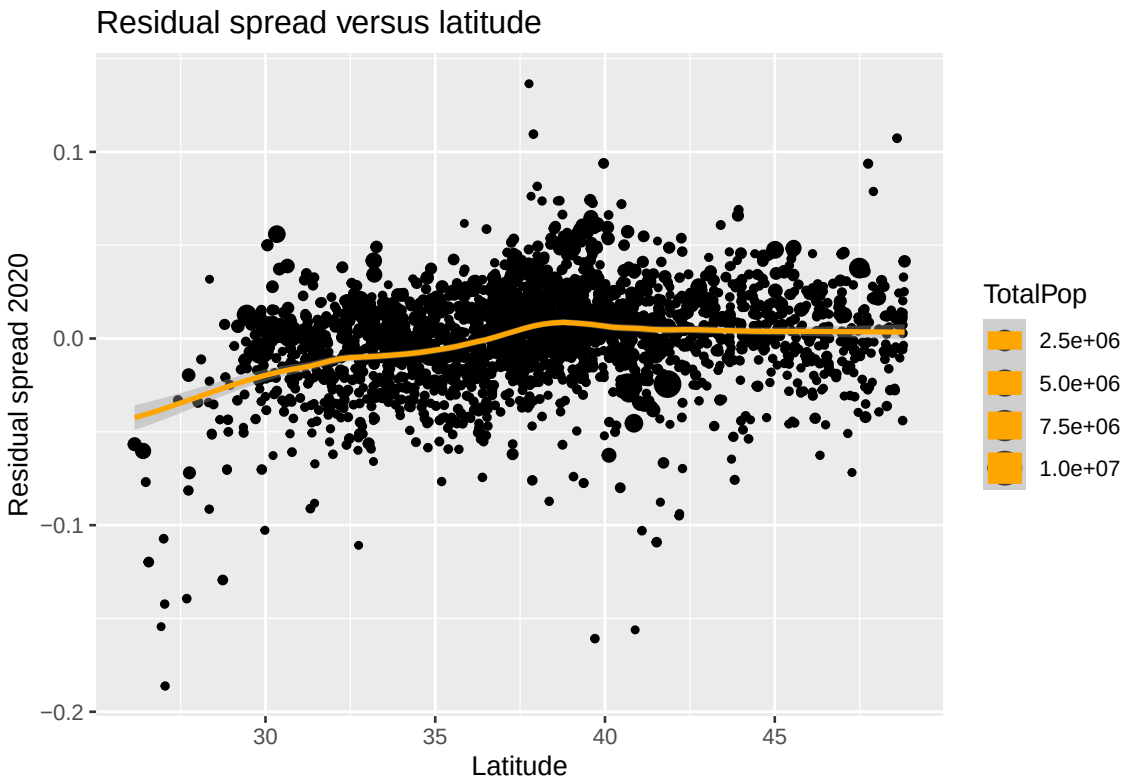
There is additional spread increase in the highly populated counties. It looks like the separation is at population about 45'000. (This was put in manually, not by numerical methods.)



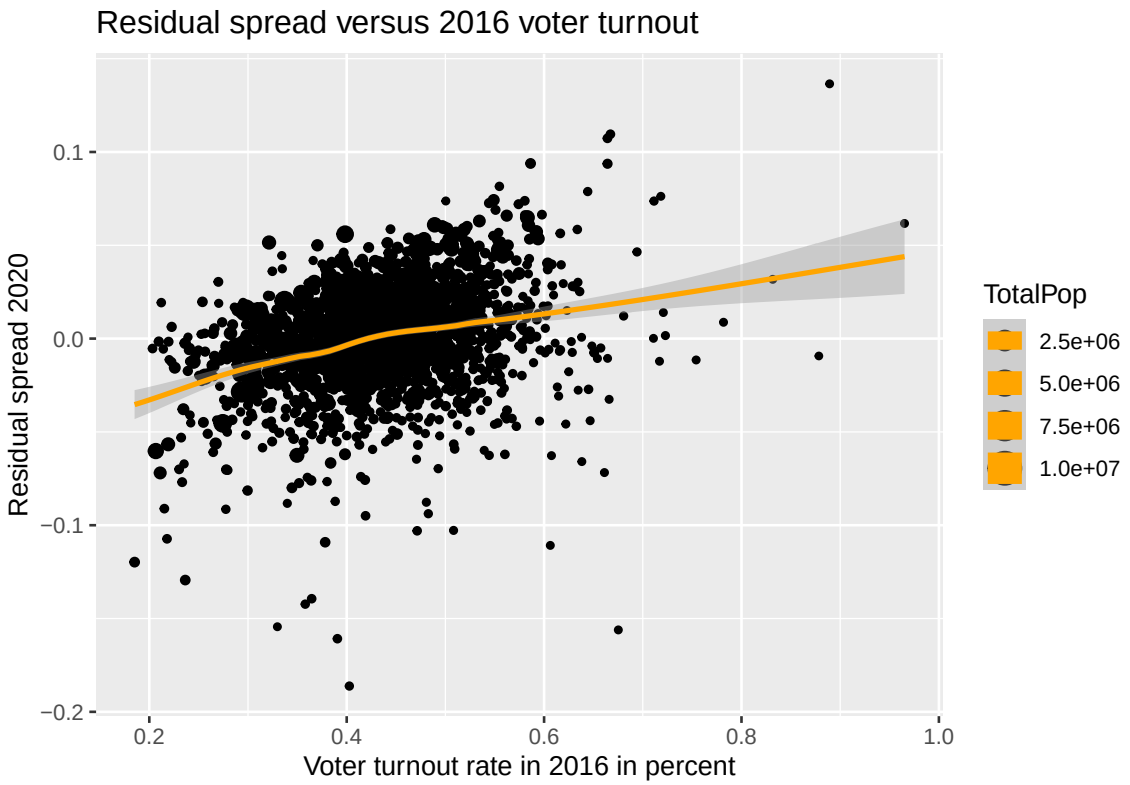
But earlier the graphs show also a strong east-west trend. I exclude Hawaii, as its longitudes are outliers. I will exclude both Hawaii and Alaska in the following.



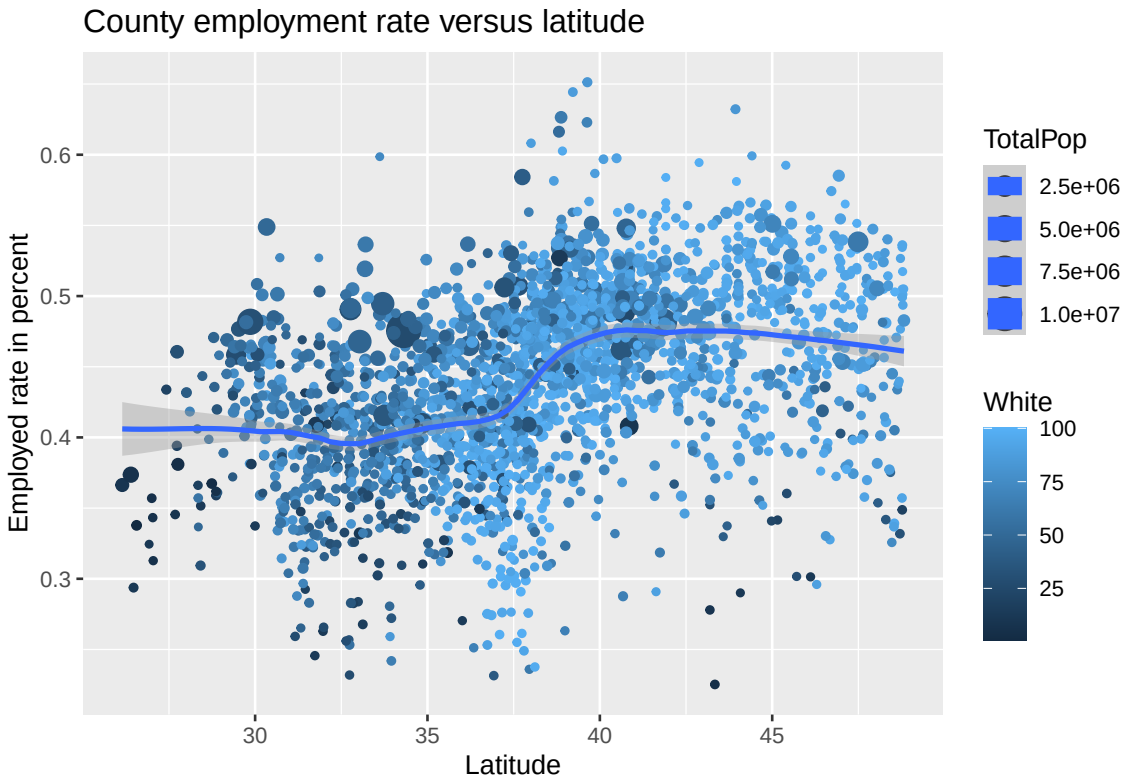
It seems that in the East the spreads go up more than the nationwide average. Similar we can look at the gradient north to south:



In the very South there seems to be a trend towards the Republican candidate. Similarly, the residuals seem to depend on the relative voter turnout rate (total votes in 2016 over population):



Overall, there are hidden correlations between the features. E.g., the spatial differences of employment rate are striking:



(There is also a dependency between employment and longitude and between employment and total population, but it is not shown here.)

Other features like current daily increase in Corona infections ("cases") relative to total population does not show a significant effect. (Plot not shown here.) Note that without normalizing new Corona infections to population, a spurious effect arises, which can be traced back to the fact that Corona cases scale with population.

Insights gained

The variables lat (latitude), long (longitude), and TotalPop (county population) are most important variables. Employment rates show clear features, but there is correlation with the population number Other variables may be important, too.

A simple model from the above would be (e.g.) to

- Take means of spread changes conditional on population being above or below 45'000 and predicting these
- Apply linear regression to the relative change of population number

A slight modification would be to replace the linear regression in the second point by a CART model with features population and latitude. The advantage would be that the high voter turnout increase observed in the western states would not influence predictions for the eastern states.

However, despite the simplicity and transparency of such simple models, the plots show that longitude and latitude reveal several features. In order to cover these, these variables should both be included in the models.

The issue about the counties in southern Texas observed above might be relevant for Florida. The media reported that a part of the Hispanic voters have been convinced Trump supporters. As Florida is in the hold-out sample, the algorithms cannot learn this behavior (unless, e.g., a link to the Texan counties near the border could be made). Thus, it is to be expected that Florida will not be very well explained.

Summary:

- There is an overall increase of votes, but the increase grows with population size and (due to correlation) Democratic dominance.
- There are geographic regions that show either increase or decrease of spread (difference in percentages of the candidate) over the whole region. The variables 'lat' and 'long' should be taken into account.
- The spread change is small in small counties, but higher in those with high population numbers
- All properties scaling with population have to be normalized
- Many variables correlate with population, even after normalizing with population. It seems that bucketing of population size (e.g.) into <5'000, 5'000 to 200'000, and >200'000, could help separating this out. Thus, a tree or random forest model might work here.
- Although gender might play a considerable role, it is difficult to prove this with the data set. This is due to the fact that the percentage of one sex varies in a very small band around 50%. I will keep the variable nevertheless.
- Employment and income show some features.
- Corona cases do not seem to be highly explaining. However, note that these are the current daily increases close to election day, not the cumulative cases. (It is conceivable that a high amount of cumulative cases is necessary (but not sufficient!) to increase the explanatory power.) Note that it might be difficult to separate local experience from nationwide news coverage about Corona. For this analysis I use the working assumption that the main effect of Covid-19 is the overall high voter turnout due to voting by mail, i.e., I will not include the Corona case numbers.

Modeling approach

Defining an appropriate loss function As stated above, the changes of both voter turnout and spread are driven mostly by population. Population, in turn, contains demographic information, because it distinguishes between rural and urban areas. The variables identified above are

- Geographic coordinates (lat, long)
- Population number (TotalPop)
- Statics from the last election (TurnoutRate.16, Delta.16.rel)
- Race (Black.pct, Hispanic.pct, Asian.pct, Native.pct, Pacific.pct, White.pct)
- Employment rates (Employment.pct)
- I also include the percentage of women and the percentage of construction workers

The dependent variable will be what I will call relative spread:

$$\frac{votes.DEM.20 - votes.REP.20}{TotalPop}$$

The numerator, the difference between votes for the Democrat candidate and the votes for the Republican candidate, I will be call absolute spread. Note that spread is usually defined with the total numbers of votes in the denominator. However, this requires knowledge of the total vote number, which itself would be a dependent variable. In contrast, the population number in the given data set is a constant.

For the loss function I use the RMSE:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_i (y_i - \hat{y}_i)^2}$$

where i is the county index and N is the total number of counties. In R,

```
RMSE <- function(yPred,yAct){
  sqrt(mean((yPred - yAct)^2))
}
```

From my first approaches, I will also show a second loss function, a weighted RMSE, defined as

$$\text{RMSE}_\text{w} = \sqrt{\frac{\sum_i w_i (y_i - \hat{y}_i)^2}{\sum_i w_i}}$$

where the sum runs over all counties i in the test set and the weights are the total populations of the counties. Further, y represents the dependent variable of concern, which is the difference of votes for Deomcrats and the votes for Republicans. In R, the function can be defined as

```
RMSE.weighted <- function(yPred,yAct,weight){
  sqrt(weighted.mean((yPred - yAct)^2,w = weight))
}
```

The weight is the total population.

For application of the various machine learning algorithms, I prepare a data frame "change.16.20", which only contains the dependent variable and the features I want to include in the machine learning algorithms. Note that, as before, I exclude Hawaii due to the extreme geographic coordinates. Each data set contains the dependent variables "y" and various dependent variables:

```
change.16.20 <- election %>%
  filter(!is.na(lat) & !is.na(long) & !(state == "HI")) %>%
  mutate(y = residual.16.20,
         White.pct = White/100,
```

```
        Black.pct = Black/100,
        Hispanic.pct = Hispanic/100,
        Asian.pct = Asian/100,
        Native.pct = Native/100,
        Pacific.pct = Pacific/100,
        Women.pct = Women/TotalPop,
        Employment.pct = Employed/TotalPop,
        Construction.pct = Construction/TotalPop,
        TurnoutRate.16 = votes.TOT.16/TotalPop) %>%
  select(countyID,
        county,
        state,
        y,
        lat,
        long,
        TotalPop,
        TurnoutRate.16,
        Delta.16.rel,
        White.pct,
        Black.pct,
        Hispanic.pct,
        Asian.pct,
        Native.pct,
        Pacific.pct,
        Women.pct,
        Employment.pct,
        Construction.pct
  )
```

Divide into a training and test set:

```
# Set seed (for reproducibility)
set.seed(123)
# Partition for total vote number percentage change
index_test <- createDataPartition(change.16.20$y,times = 1,p = 0.3,list = FALSE)
change.16.20.train <- change.16.20[-index_test,]
change.16.20.test <- change.16.20[index_test,]
```

Here, I have chosen a test set of 30% of all observation. This seems a good choice as the training set is still large enough to expect good fits, while the test set is large enough to reduce the risk of overfitting.

Model overview The models I will fit are the following:

- Baseline model
  - Above, I have subtract the average voting behavior in terms of a local regression
  - The residual (after subtracting the local regression line) then has a mean close to zero
  - The baseline model is the average of the residual
- Linear regression
- CART model
- Random forest
- Extreme gradient boosting
- k-nearest neighbors (KNN)

Model training

Baseline model

As a baseline model I use the average changes

```
mu <- mean(change.16.20.train$y)
# print(paste0("Average residual spread: ",round(mu*100,1),"%"))
```

The resulting average residual precentage spread is 0%. As expected it is very small, because we have already subtracted the baseline fit earlier.

model	RMSE	RMSE.weighted
baseline	0.0271421	0.0243186

This is the benchmark other methods will be measured against.

Linear regression

Does linear regression improve the result significantly? During development, I started with a large feature list and reduced the number of variables in different ways. I also played with using population number as weights in the regression. Assuming that the baseline model has removed the main part dependent on population, I have decided to fit without weighting. This is also because some of the machine learning algorithms below do not allow for weighting.

I selected three feature lists to demonstrate one possible way to start with many feature variables and reduce them stepwise to get similar results with a simpler model. The output below shows the feature lists and the summary of the linear regression with the R base method "lm". Remember that 'y' is the residual spread after subtracting the local regression ('loess') fit representing the general features across all states in the training set.

```
## [1] "####################################################################"
## [1] "1.) Model with feature list 1"
## [1] "####################################################################"
##  [1] "y"                "lat"              "long"             "TotalPop"
##  [5] "TurnoutRate.16"   "Delta.16.rel"     "Black.pct"        "Hispanic.pct"
##  [9] "Asian.pct"        "Native.pct"       "Pacific.pct"      "Women.pct"
## [13] "Employment.pct"   "Construction.pct"
## [1] "- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -"
## [1] "Summary:"
##
## Call:
## lm(formula = y ~ ., data = tmpData)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.161369 -0.012031  0.001852  0.014165  0.100974
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -2.972e-02  1.653e-02  -1.798 0.072318 .
## lat              -6.640e-04  1.933e-04  -3.435 0.000609 ***
## long              1.559e-04  6.599e-05   2.363 0.018260 *
## TotalPop          1.325e-09  2.414e-09   0.549 0.583083
## TurnoutRate.16    8.080e-02  9.704e-03   8.326  < 2e-16 ***
## Delta.16.rel      1.847e-02  7.140e-03   2.587 0.009780 **
```

```
## Black.pct        -3.186e-02  6.820e-03  -4.671 3.25e-06 ***
## Hispanic.pct     -3.314e-02  6.361e-03  -5.210 2.14e-07 ***
## Asian.pct         1.121e-01  3.266e-02   3.434 0.000611 ***
## Native.pct        4.455e-02  9.009e-03   4.946 8.41e-07 ***
## Pacific.pct       7.329e-01  3.083e-01   2.377 0.017565 *
## Women.pct        -3.380e-02  2.653e-02  -1.274 0.202936
## Employment.pct    1.330e-01  1.193e-02  11.155  < 2e-16 ***
## Construction.pct -5.472e-01  1.336e-01  -4.096 4.41e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02309 on 1550 degrees of freedom
## Multiple R-squared:  0.2742, Adjusted R-squared:  0.2681
## F-statistic: 45.05 on 13 and 1550 DF,  p-value: < 2.2e-16
##
## [1] "################################################################"
## [1] "2.) Model with feature list 2"
## [1] "################################################################"
## [1] "y"                "lat"              "long"             "TotalPop"
## [5] "Black.pct"        "Hispanic.pct"     "Employment.pct"   "Construction.pct"
## [9] "TurnoutRate.16"
## [1] "- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -"
## [1] "Summary:"
##
## Call:
## lm(formula = y ~ ., data = tmpData)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.158094 -0.012648  0.001534  0.014250  0.110240
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -6.479e-02  8.256e-03  -7.848 7.79e-15 ***
## lat              -1.813e-04  1.756e-04  -1.032 0.302105
## long              7.666e-05  6.382e-05   1.201 0.229872
## TotalPop          7.673e-09  2.144e-09   3.579 0.000356 ***
## Black.pct        -1.930e-02  5.030e-03  -3.837 0.000129 ***
## Hispanic.pct     -2.703e-02  5.405e-03  -5.001 6.35e-07 ***
## Employment.pct    1.297e-01  1.147e-02  11.303  < 2e-16 ***
## Construction.pct -6.614e-01  1.241e-01  -5.329 1.13e-07 ***
## TurnoutRate.16    6.164e-02  9.459e-03   6.516 9.74e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02361 on 1555 degrees of freedom
## Multiple R-squared:  0.2389, Adjusted R-squared:  0.235
## F-statistic: 61.01 on 8 and 1555 DF,  p-value: < 2.2e-16
##
## [1] "################################################################"
## [1] "3.) Model with feature list 3"
## [1] "################################################################"
## [1] "y"                "TotalPop"         "Black.pct"        "Hispanic.pct"
## [5] "Employment.pct"   "Construction.pct" "TurnoutRate.16"
```

```
## [1] "- - - - - - - - - - - - - - - - - - - - - - - - - - - - - -"
## [1] "Summary:"
##
## Call:
## lm(formula = y ~ ., data = tmpData)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.159426 -0.012441  0.001618  0.014263  0.111569
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -7.715e-02  5.328e-03 -14.480  < 2e-16 ***
## TotalPop          7.486e-09  2.143e-09   3.493 0.000490 ***
## Black.pct        -1.499e-02  4.481e-03  -3.344 0.000845 ***
## Hispanic.pct     -2.724e-02  4.637e-03  -5.874 5.19e-09 ***
## Employment.pct    1.283e-01  1.086e-02  11.820  < 2e-16 ***
## Construction.pct -6.682e-01  1.236e-01  -5.408 7.37e-08 ***
## TurnoutRate.16    5.807e-02  9.288e-03   6.252 5.21e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02363 on 1557 degrees of freedom
## Multiple R-squared:  0.237,  Adjusted R-squared:  0.2341
## F-statistic: 80.61 on 6 and 1557 DF,  p-value: < 2.2e-16
```

```
##      model       RMSE RMSE.weighted
## 1 baseline 0.02714213    0.02431861
## 2 lm.ftl.1 0.02291207    0.02443503
## 3 lm.ftl.2 0.02325137    0.03384320
## 4 lm.ftl.3 0.02324589    0.03375699
```

The first feature list is the one I started from for all the algorithms used here. It gives an idea about which features could be relevant. However, it can be expected that these features have considerable correlations between them. This is the reason it was cut down to lists two and three. I select the third feature list as it yields comparable results with a smaller number of features. In the third list, all variables are marked statistically relevant; however, the R squared is rather small.

Although it will not make a difference for "lm", I nevertheless use cross validation to be consistent with the other algorithms. In general, cross validation can help to get a robust choice of the fit parameters. For "lm" it will yield the same result as before with the same feature list, but I use it to select the feature list that I deem most appropriate.

I append the results to the list of RMSEs:

```
kable(Results)
```

model	RMSE	RMSE.weighted
baseline	0.0271421	0.0243186
lm.ftl.1	0.0229121	0.0244350
lm.ftl.2	0.0232514	0.0338432
lm.ftl.3	0.0232459	0.0337570
lm.cv	0.0232459	0.0337570

Of course, much more time and effort could be spend for finding a robust model. However, the plots shown

earlier indicate that linear regression will likely not reflect the complicated structure of the data. For that reason I leave the analysis at this rather crude level.

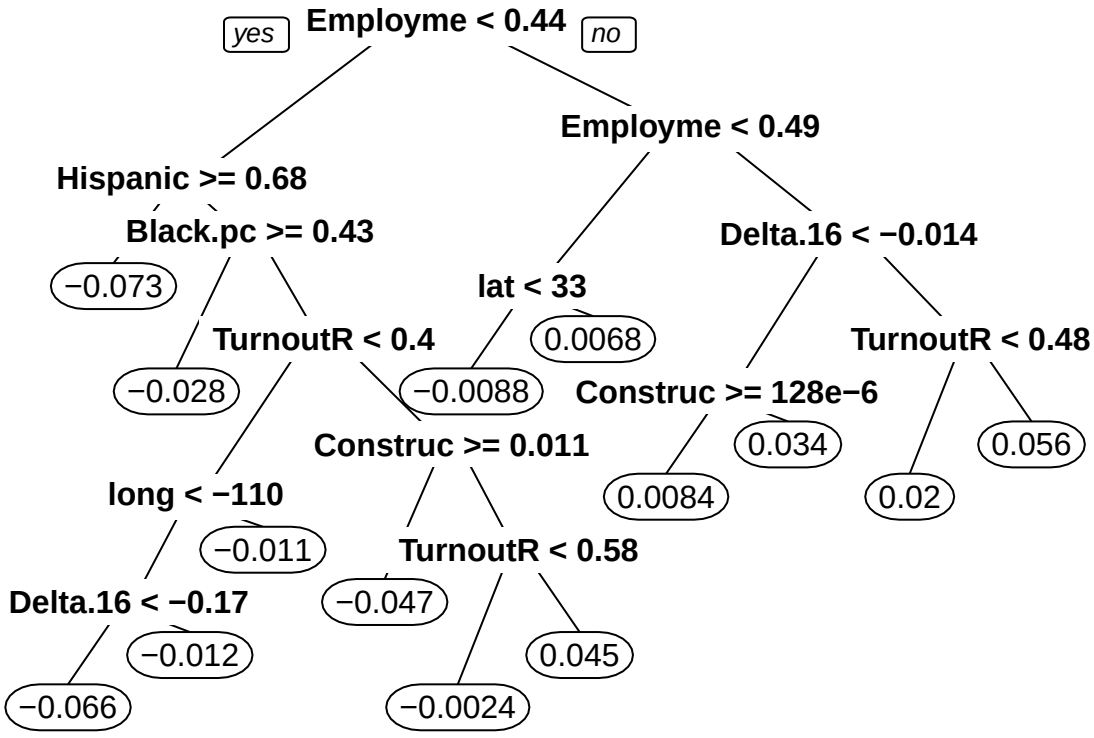
CART model (here: regression tree)

I first do an example fit and below repeat the fit with cross validation. Here and below I will always start with the following set of features:

```
# Define the set of features to be investigated
featureList <- c("y","lat","long", "TotalPop", "Delta.16.rel","TurnoutRate.16",
                 "Black.pct","Hispanic.pct","Asian.pct","Native.pct","Pacific.pct",
                 "Women.pct","Employment.pct","Construction.pct")
```

Then I fit a regression tree using 'rpart' from the package with the same name. The plot is made with help of the package 'rpart.plot' (function 'prp').

```
# Select these features
tmpData <- change.16.20.train %>%
  select(starts_with(featureList))

# Fit the model
# fit.weighted <- rpart(y ~ .,data = tmpData,weights = TotalPop)
fit <- rpart(y ~ .,data = tmpData)
prp(fit)
```



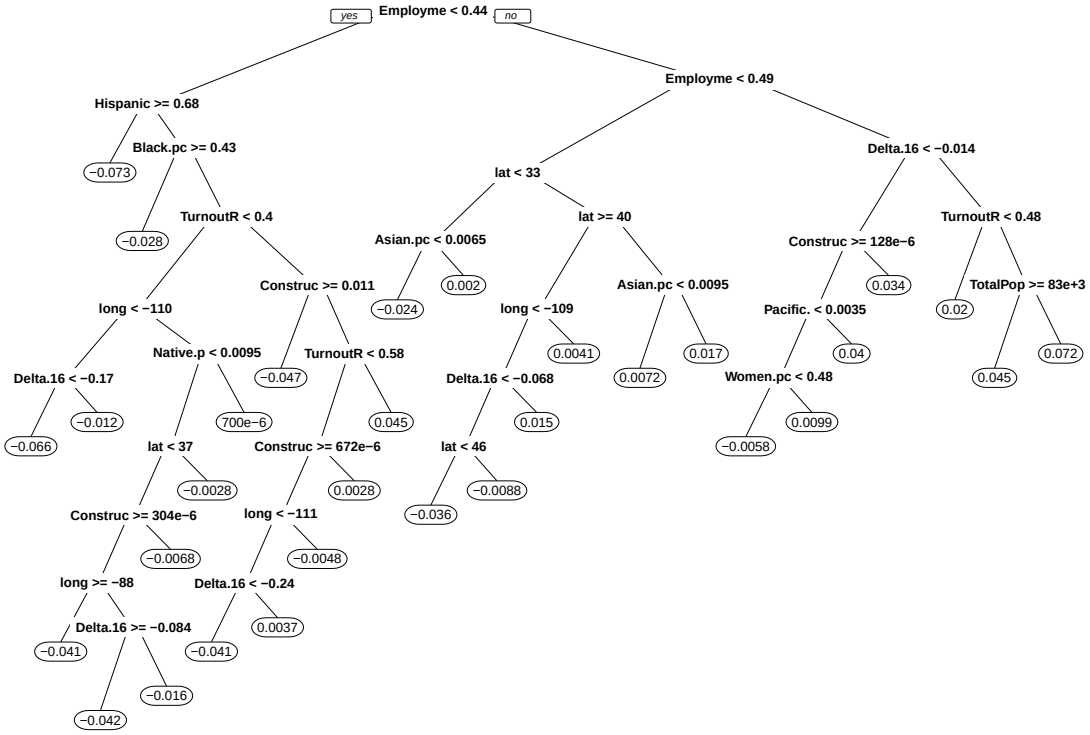

```
# Predict
change.16.20.test <- change.16.20.test %>%
  mutate(y_hat_rpart = predict(fit, newdata = change.16.20.test))

## Calculate RMSEs
thisRMSE <- with(change.16.20.test, RMSE(yPred = y_hat_rpart,yAct = y))
thisRMSE.weighted <- with(change.16.20.test,
                          RMSE.weighted(yPred = y_hat_rpart,yAct = y,weight = TotalPop))
```

```
## Results
Results <- Results %>% bind_rows(data.frame(model = c("rpart"),
                        RMSE = thisRMSE,
                        RMSE.weighted = thisRMSE.weighted))
```

Note that the values are the differences of means from the baseline model. A negative value can, but does not necessarily have to mean that those groups voted for Trump; it can be that in regions with low employment rate, the groups did not follow the nationwide trend as much.

In order to get a more robust result, I apply cross validation. I use the 'train' method from the 'caret' package.

After cross validation the best regression tree can be displayed as follows:



This tree essentially splits counties with low employment rates from those with high employment rates. In counties with low employment rates Hispanic and Black voters did not follow the nationwide trends. For counties with high employment, it looks like the preferences from 2016 were further amplified.

The results for the CART model, appended to the list, are the last two items of the following.

model	RMSE	RMSE.weighted
baseline	0.0271421	0.0243186
lm.ftl.1	0.0229121	0.0244350
lm.ftl.2	0.0232514	0.0338432
lm.ftl.3	0.0232459	0.0337570
lm.cv	0.0232459	0.0337570
rpart	0.0226266	0.0214106
rpart.cv	0.0216363	0.0223476

Random forest

It is conceivable that due to the complexity of geographic and demographic features random forests work well. I use the 'randomForest' method from the package with the same name. Again, I start with the features shown above and first do a single fit to understand the important variables and then proceed to cross validation afterwards.

The variable importance for the single fit turns out as

Variable	Overall
Employment.pct	0.1521649
TurnoutRate.16	0.1257806
lat	0.1250786
long	0.1151468
Hispanic.pct	0.1150624
Delta.16.rel	0.0986849
Construction.pct	0.0770503
TotalPop	0.0694289
Asian.pct	0.0591017
Black.pct	0.0572748
Women.pct	0.0511550
Native.pct	0.0380815
Pacific.pct	0.0125422

The single fit puts emphasis on the geographical distribution, but keeps many variables. Possibly these are too many variables and they can be reduced by cross validation.

For cross validation, I use five-fold cross validation in order to limit the run time. I fix the tree number to 150. The number of randomly sampled variables is to be optimized by the algorithm.

With the final model from cross validation, the variable importance is

Variable	Overall
Employment.pct	0.1836754
Hispanic.pct	0.1315941
TurnoutRate.16	0.1278276
long	0.1195170
lat	0.1183231
Delta.16.rel	0.1041862
Construction.pct	0.0754067
TotalPop	0.0568971
Black.pct	0.0562958
Asian.pct	0.0510573
Women.pct	0.0475963
Native.pct	0.0354620
Pacific.pct	0.0111200

and the RMSE table now looks as follows:

model	RMSE	RMSE.weighted
baseline	0.0271421	0.0243186
lm.ftl.1	0.0229121	0.0244350
lm.ftl.2	0.0232514	0.0338432

model	RMSE	RMSE.weighted
lm.ftl.3	0.0232459	0.0337570
lm.cv	0.0232459	0.0337570
rpart	0.0226266	0.0214106
rpart.cv	0.0216363	0.0223476
rf	0.0176672	0.0156721
rf.cv	0.0174968	0.0157709

The parameter 'mtry' selected by cross validation is

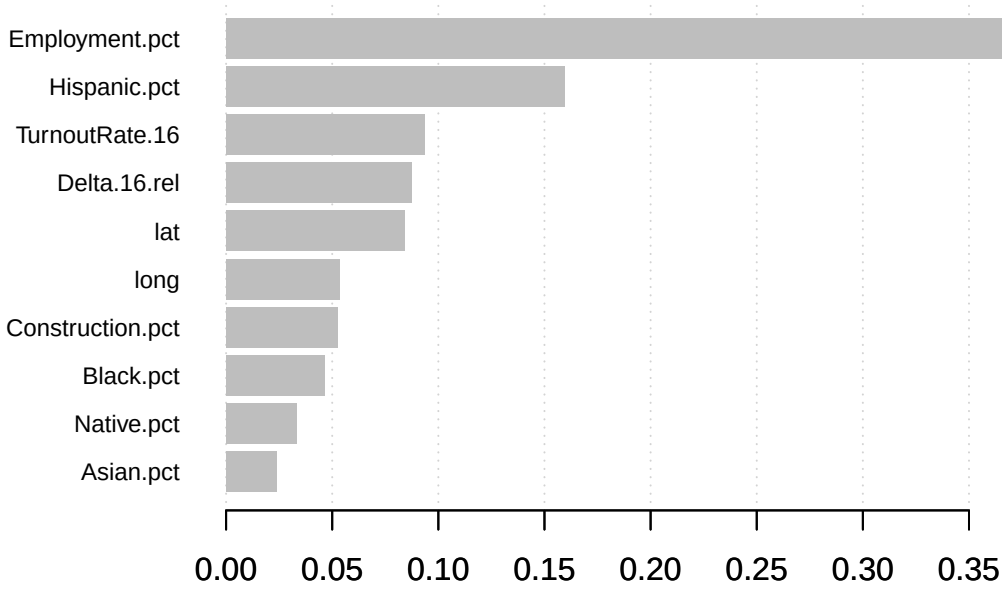
mtry
8

Indeed, the number of features has been reduced, but not by much. Possibly some overfitting effect remains.

Extreme gradient boosting

A machine learning algorithm often mentioned recently is extreme gradient boosting. Let's see how it performs compared to the other algorithms. I use the R package 'xgboost'. A tutorial can be found under the link https://xgboost.readthedocs.io/en/latest/R-package/xgboostPresentation.html. As often advertised, it indeed runs quickly. I directly jump to the cross validation training method provided in the package.

The 'xgboost' package provides a plot for variable importance:



The most important variable here is the employment rate, which was also detected earlier in the explorative analysis and by the other algorithms.

The results updated with the xgboost outcome are

model	RMSE	RMSE.weighted
baseline	0.0271421	0.0243186
lm.ftl.1	0.0229121	0.0244350
lm.ftl.2	0.0232514	0.0338432

model	RMSE	RMSE.weighted
lm.ftl.3	0.0232459	0.0337570
lm.cv	0.0232459	0.0337570
rpart	0.0226266	0.0214106
rpart.cv	0.0216363	0.0223476
rf	0.0176672	0.0156721
rf.cv	0.0174968	0.0157709
xgb.cv	0.0216435	0.0211175

Most probably the parameters used in the algorithm can be tuned to improve results.

KNN

In order to cover a broad set of machine learning techniques, I also apply the k-nearest neighbors algorithm. Again, I first apply a fit once (using "knnreg") and the use cross validation to get a more robust result.

The RMSE results look as follows (please see last two rows):

model	RMSE	RMSE.weighted
baseline	0.0271421	0.0243186
lm.ftl.1	0.0229121	0.0244350
lm.ftl.2	0.0232514	0.0338432
lm.ftl.3	0.0232459	0.0337570
lm.cv	0.0232459	0.0337570
rpart	0.0226266	0.0214106
rpart.cv	0.0216363	0.0223476
rf	0.0176672	0.0156721
rf.cv	0.0174968	0.0157709
xgb.cv	0.0216435	0.0211175
knnreg	0.0279832	0.0249716
knn.cv	0.0263400	0.0257673

The 'loess' algorithm

The 'loess' algorithm is restricted to 1-4 variables. Given the complexity of the inputs indicates that this is not the best model for this purpose. I leave it out here. I have used it in the baseline model, though.

Final model choice and prediction

The results so far are

model	RMSE	RMSE.weighted
baseline	0.0271421	0.0243186
lm.ftl.1	0.0229121	0.0244350
lm.ftl.2	0.0232514	0.0338432
lm.ftl.3	0.0232459	0.0337570
lm.cv	0.0232459	0.0337570
rpart	0.0226266	0.0214106
rpart.cv	0.0216363	0.0223476
rf	0.0176672	0.0156721
rf.cv	0.0174968	0.0157709
xgb.cv	0.0216435	0.0211175

model	RMSE	RMSE.weighted
knnreg	0.0279832	0.0249716
knn.cv	0.0263400	0.0257673

Among these, random forests work best. (But xgb got a very good result in a fraction of the time.)

I fit the model now on the whole election data set (still without the validation swing states sample). For reference I make a prediction on the whole 'training' set, i.e., on the whole set that went into the model training. This is just to make it comparable to the result in the validation set. It is expected to be too favorable.

The variable importance is

Variable	Overall
Employment.pct	0.2617004
lat	0.1967925
Hispanic.pct	0.1878332
long	0.1805217
TurnoutRate.16	0.1555432
Delta.16.rel	0.1531146
Construction.pct	0.1033572
Black.pct	0.0806056
Asian.pct	0.0796985
TotalPop	0.0754094
Women.pct	0.0605368
Native.pct	0.0492911
Pacific.pct	0.0118897

and the result is

model	RMSE	RMSE.weighted
baseline	0.0271421	0.0243186
lm.ftl.1	0.0229121	0.0244350
lm.ftl.2	0.0232514	0.0338432
lm.ftl.3	0.0232459	0.0337570
lm.cv	0.0232459	0.0337570
rpart	0.0226266	0.0214106
rpart.cv	0.0216363	0.0223476
rf	0.0176672	0.0156721
rf.cv	0.0174968	0.0157709
xgb.cv	0.0216435	0.0211175
knnreg	0.0279832	0.0249716
knn.cv	0.0263400	0.0257673
final.rf.cv (in sample)	0.0073922	0.0062570

The in-sample fit is too optimistic and indicates to some overfitting. It is crucial to consider the out-of-sample validation instead.

Prediction for the validation set

In order to make predictions for the validation set, the derived columns needed are added to the validation set. Then I use the trained model to do the predictions per county. Finally, I evaluate the RMSEs to compare the out-of-sample performance with the in-sample performance.

model	RMSE	RMSE.weighted
baseline	0.0271421	0.0243186
lm.ftl.1	0.0229121	0.0244350
lm.ftl.2	0.0232514	0.0338432
lm.ftl.3	0.0232459	0.0337570
lm.cv	0.0232459	0.0337570
rpart	0.0226266	0.0214106
rpart.cv	0.0216363	0.0223476
rf	0.0176672	0.0156721
rf.cv	0.0174968	0.0157709
xgb.cv	0.0216435	0.0211175
knnreg	0.0279832	0.0249716
knn.cv	0.0263400	0.0257673
final.rf.cv (in sample)	0.0073922	0.0062570
final (out of sample)	0.0199906	0.0250891

The RMSE of the final model applied to the validation set is comparable to the training outcomes for the random forest model. It shows that despite some indication for overfitting in-sample, it performs as expected on the validation set.

Now we know how the model performs on the county level. In order to determine the winner predicted by the model for each state, I aggregate the absolute spread (the absolute difference in votes) per state.

Evaluation on the state level One big question is how the county predictions work when summed up for each state.

```
StateResults <- validation.extended %>%
  group_by(state) %>%
  summarize(pred.abs.spread.20 = sum(pred.Delta.20.abs),
            actual.abs.spread.20 = sum(votes.DEM.20 - votes.REP.20),
            pred.DEM.won = pred.abs.spread.20 > 0,
            actual.DEM.won = actual.abs.spread.20 > 0)
```

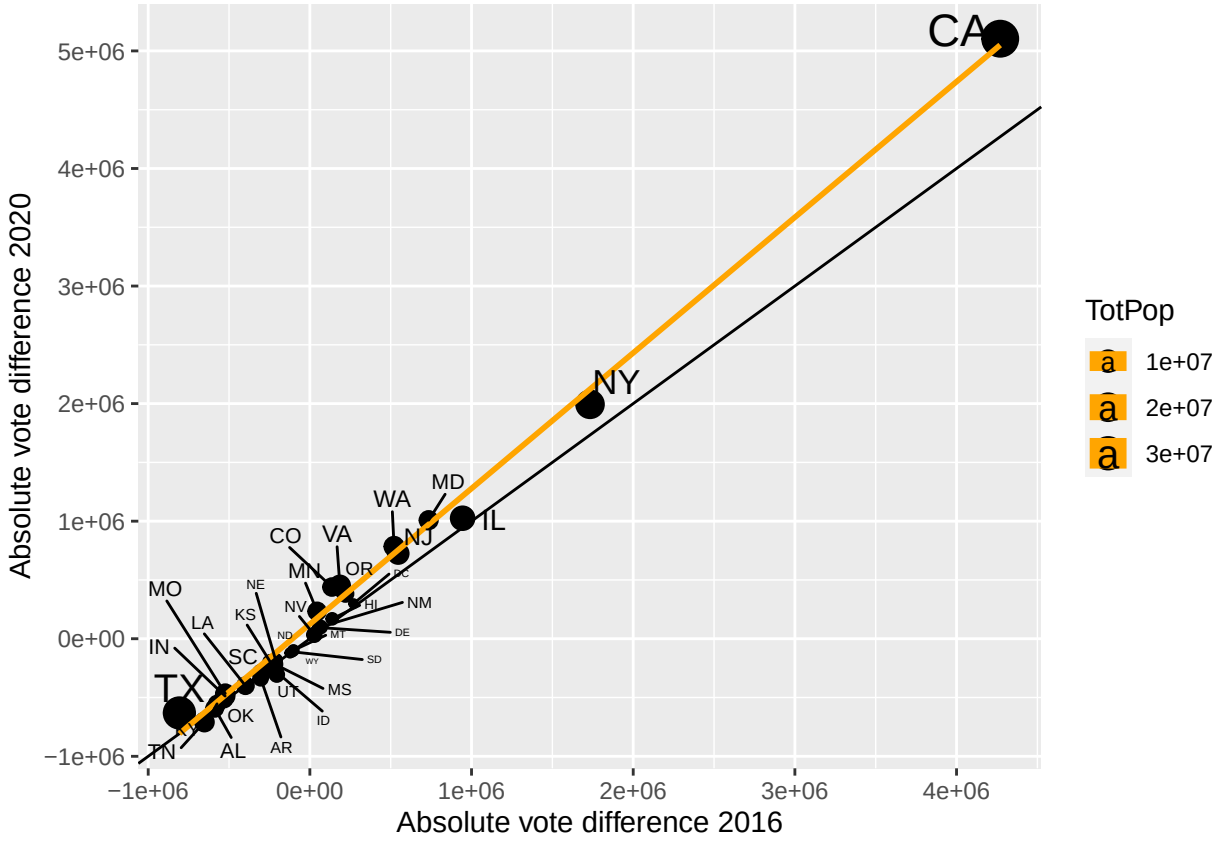
The result is

state	pred.abs.spread.20	actual.abs.spread.20	pred.DEM.won	actual.DEM.won
AZ	-29398.6931	10457	FALSE	TRUE
FL	104822.7382	-371686	TRUE	FALSE
GA	-128297.3638	11779	FALSE	TRUE
IA	-78011.9486	-138611	FALSE	FALSE
MI	207544.1091	154188	TRUE	TRUE
NC	211.2833	-74481	TRUE	FALSE
OH	-227141.2805	-475669	FALSE	FALSE
PA	275290.6529	81660	TRUE	TRUE
WI	142890.7296	20608	TRUE	TRUE
WV	-329376.5800	-309398	FALSE	FALSE

For seven out of ten states the prediction matches the actual outcome. For three states the model predicts the wrong winner. This can be taken as the starting point for further investigation. Before we do so, it should be noted that it is not the ultimate goal to predict the correct winner per state. This could have been achieved in a much simpler way, namely by performing the analysis on the state level. I will demonstrate this in the next section. The goal is to find deviations at the most granular available level (county granularity) in order to detect outliers. This will follow further below.

State aggregated regression

If the focus is not outlier detection and the explanation by demographic or geographic details, a regression of the state results can be done. This is illustrated by the following plot:



As this is only an auxiliary analysis, I will just do a linear regression on "election" and predict for "validation". This corresponds to the orange line in the graph above.

The result yield a similar result as the county analysis

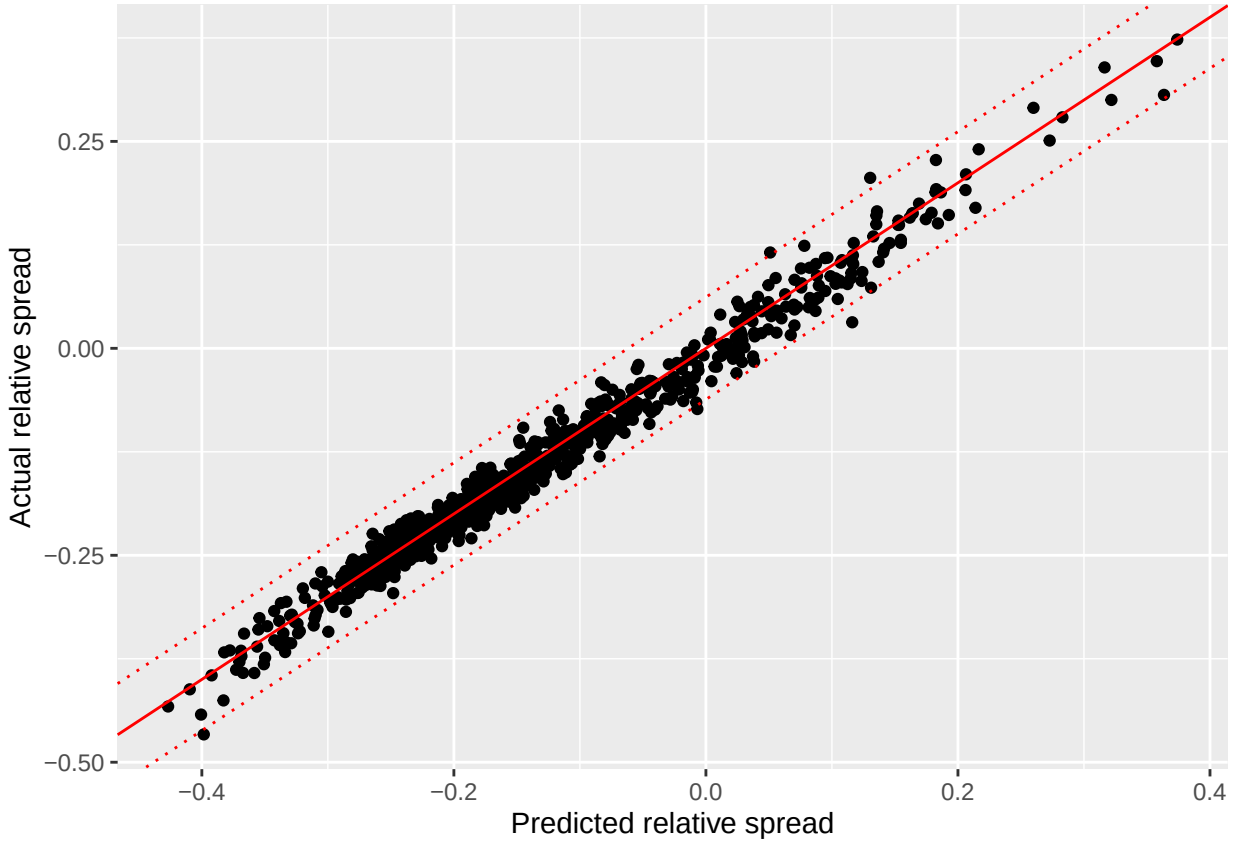
state	pred.diff.20	act.diff.20	pred.DEM.won	actual.DEM.won
AZ	19204.122	10457	TRUE	TRUE
FL	-5800.645	-371686	FALSE	FALSE
GA	-119110.528	11779	FALSE	TRUE
IA	-45485.058	-138611	FALSE	FALSE
MI	112096.771	154188	TRUE	TRUE
NC	-75477.629	-74481	FALSE	FALSE
OH	-390994.264	-475669	FALSE	FALSE
PA	73352.473	81660	TRUE	TRUE
WI	98354.934	20608	TRUE	TRUE

state	pred.diff.20	act.diff.20	pred.DEM.won	actual.DEM.won
WV	-222276.390	-309398	FALSE	FALSE

It even gives the correct winner for the states of Arizona and Florida. It is not completely surprising that the aggregated fit even comes out better: By aggregation, the prevalence problem due to extremely different population numbers is largely removed. This improves the fit result. However, the aggregated approach does not allow for a drill-down into county features. As such the use for the present goals is very limited.

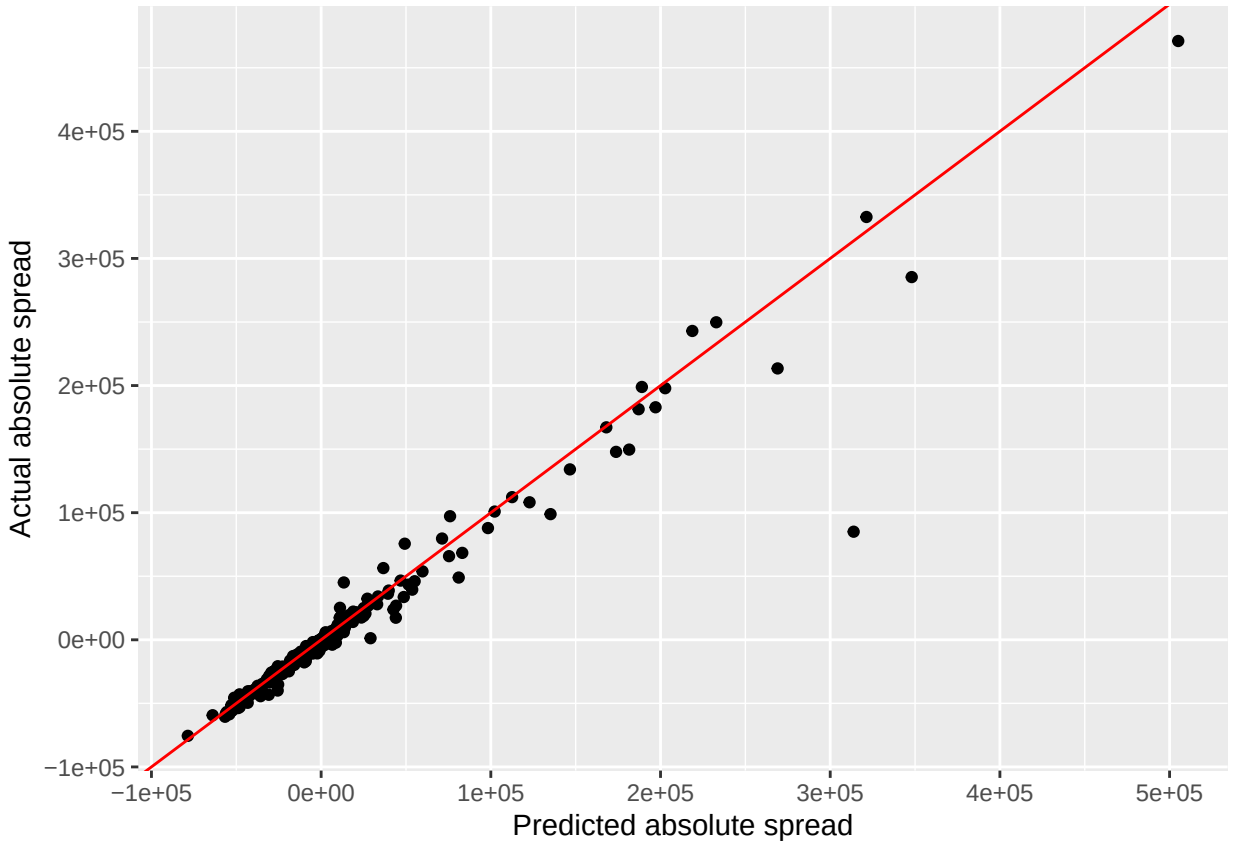
Analysis of result

I now return to the analysis of the final model. The relative spreads on the validation set look fairly reasonable:



There are 805 observations, so I included dotted lines at about three times (the quantile qnorm(0.001) = -3.0902323 times) the RMSE of 0.0199906 and should have very roughly speaking only single outliers on each side. This is at least roughly the case, maybe exceed somewhat. It becomes interesting once we multiply with the population size in the next step.

In absolute terms, when everything is scaled up by the population numbers, there is one large outlier:



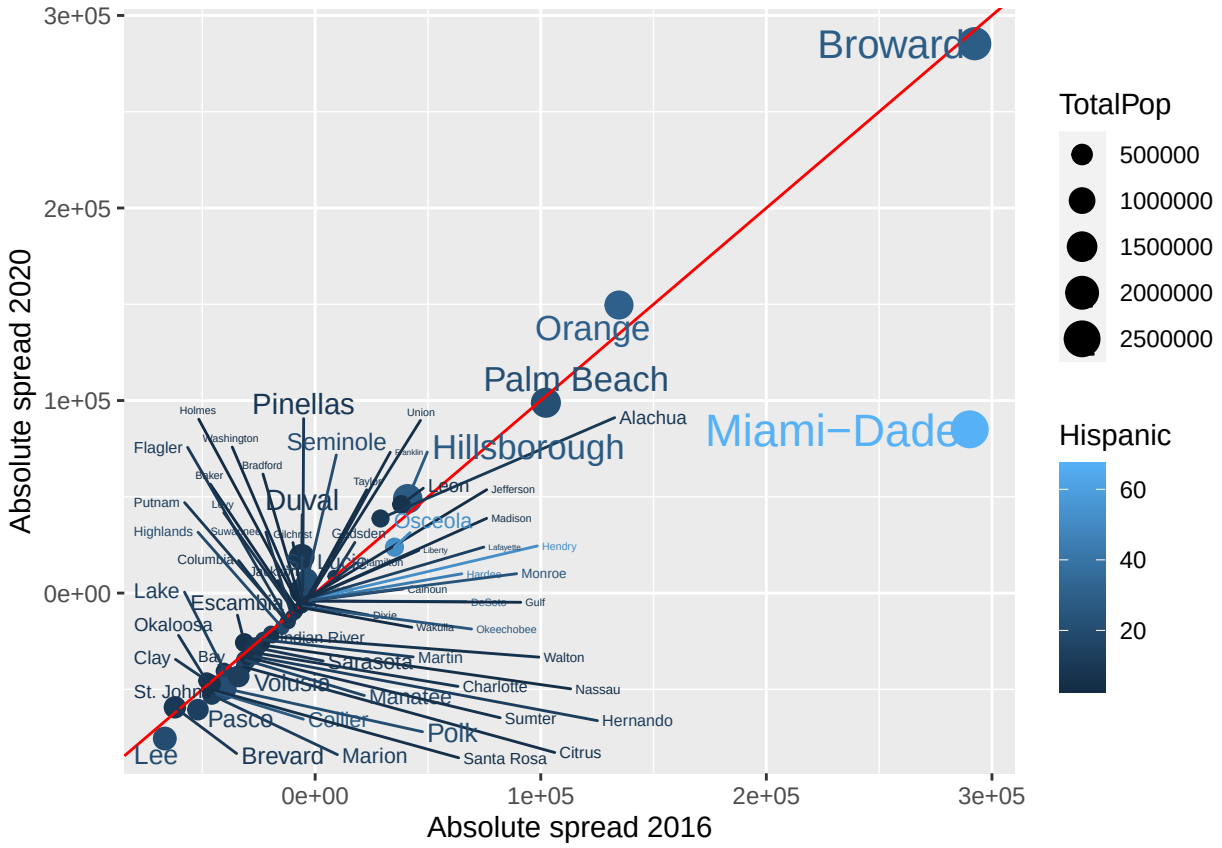
This is Miami-Dade, for which the outcome for the numbers of votes is as follows:

state	county	TotalPop	DEM.16	REP.16	DEM.20	REP.20	actual.Delta	predicted.Delta
FL	Miami-Dade	2702602	624146	333999	617864	532833	85031	313816

The population is 2.7 million, multiplied with the RMSE is about 50 thousand for a rough order of magnitude of the standard error. The model does not capture the very large deviation of about 200 thousand from the 2016 spread. This means that the biggest outlier, where voting results do not match the 'prediction' from other states, is one in favor of Trump. There are no outliers as strong as this in favor of the Democratic candidate.

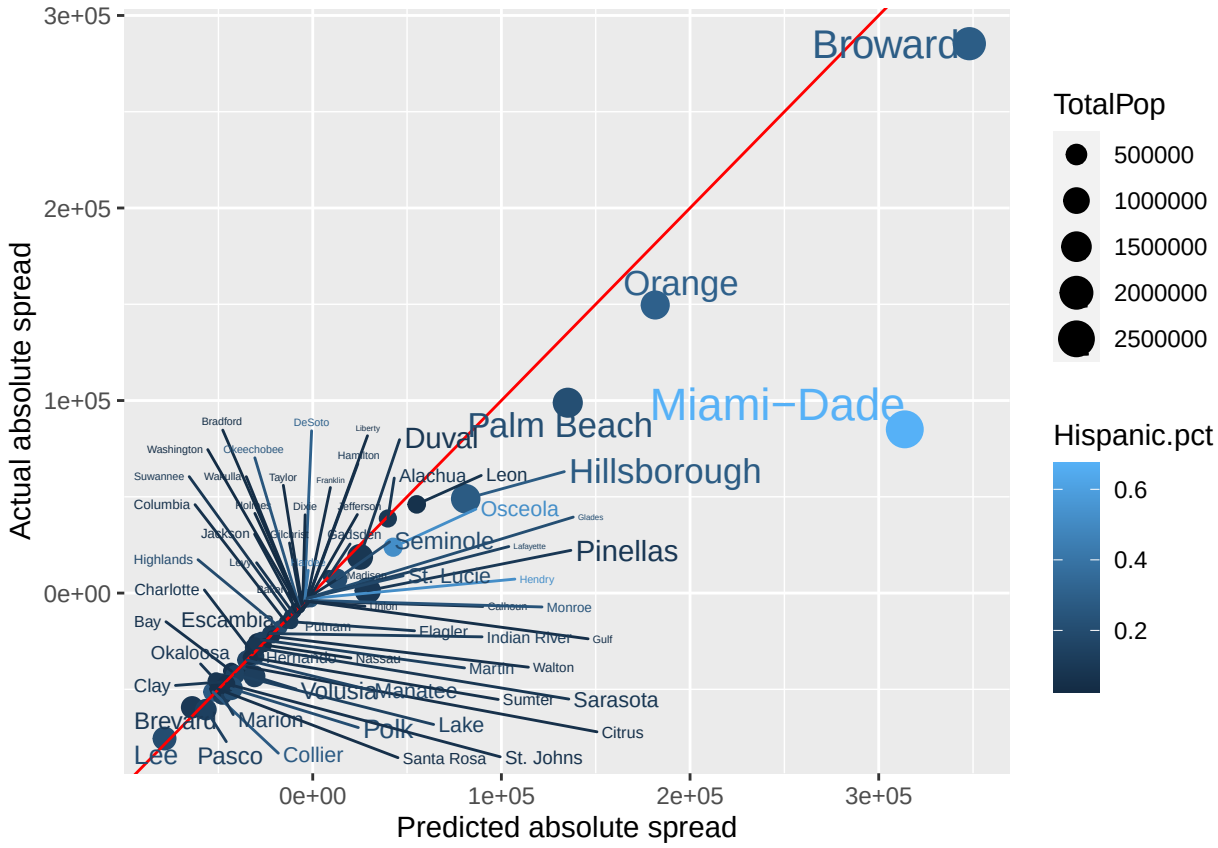
What is the outlier? Let's take a look at individual states.

Florida As reported by the media, in Florida many Latinos turned into Trump supporters between 2016 and 2020, which had a strong influence on the Florida results. What the data contains can be illustrated as follows:



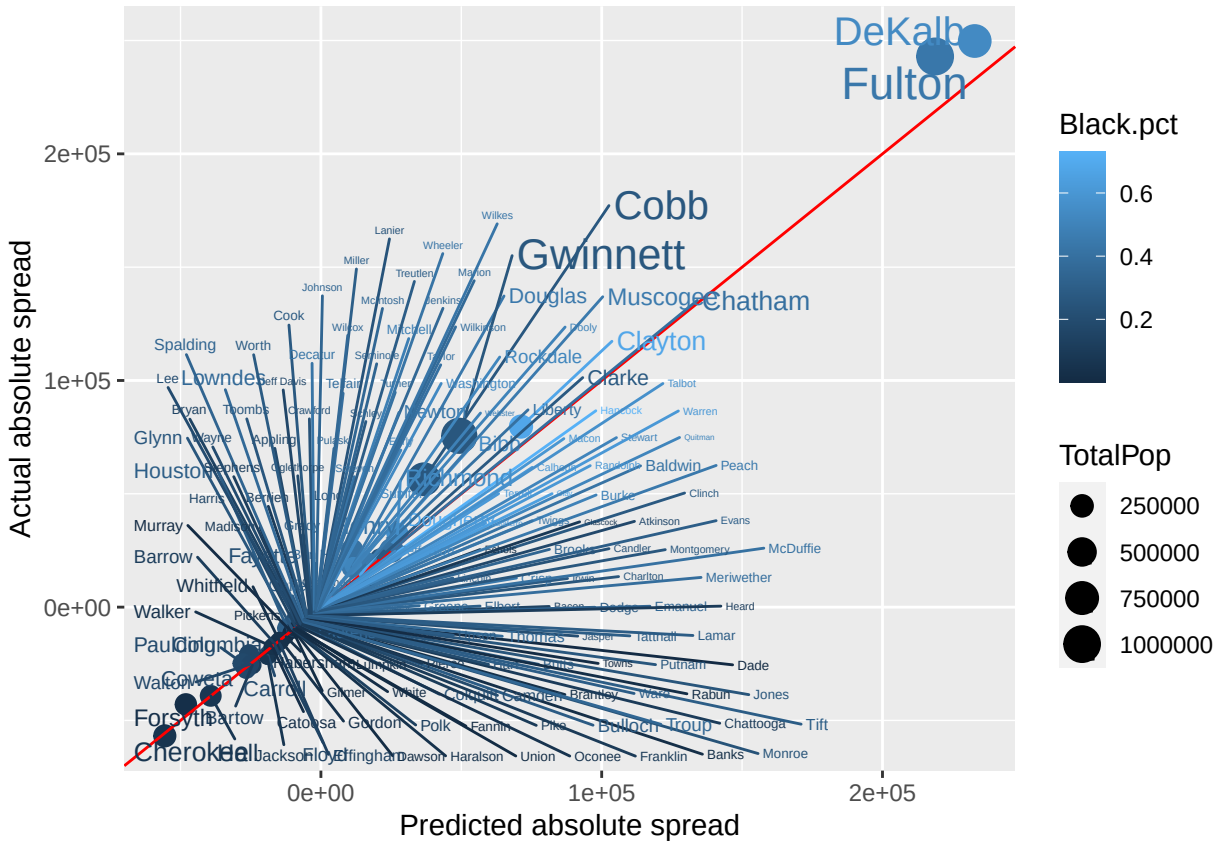
This confirms that the Miami-Dade data point shows a high percentage of Hispanic population. While the data does not allow to make the link with certainty, it seems reasonable to explain the outlier with the increased support for Trump among Latinos.

In order to understand what the model did not capture, the actual versus predicted numbers can be shown.



It seems that the model predicted an overall increase of Democrat votes, which did not happen in Florida. The previous graph shows that the highly populated counties except Miami-Dade voted as in the previous election.

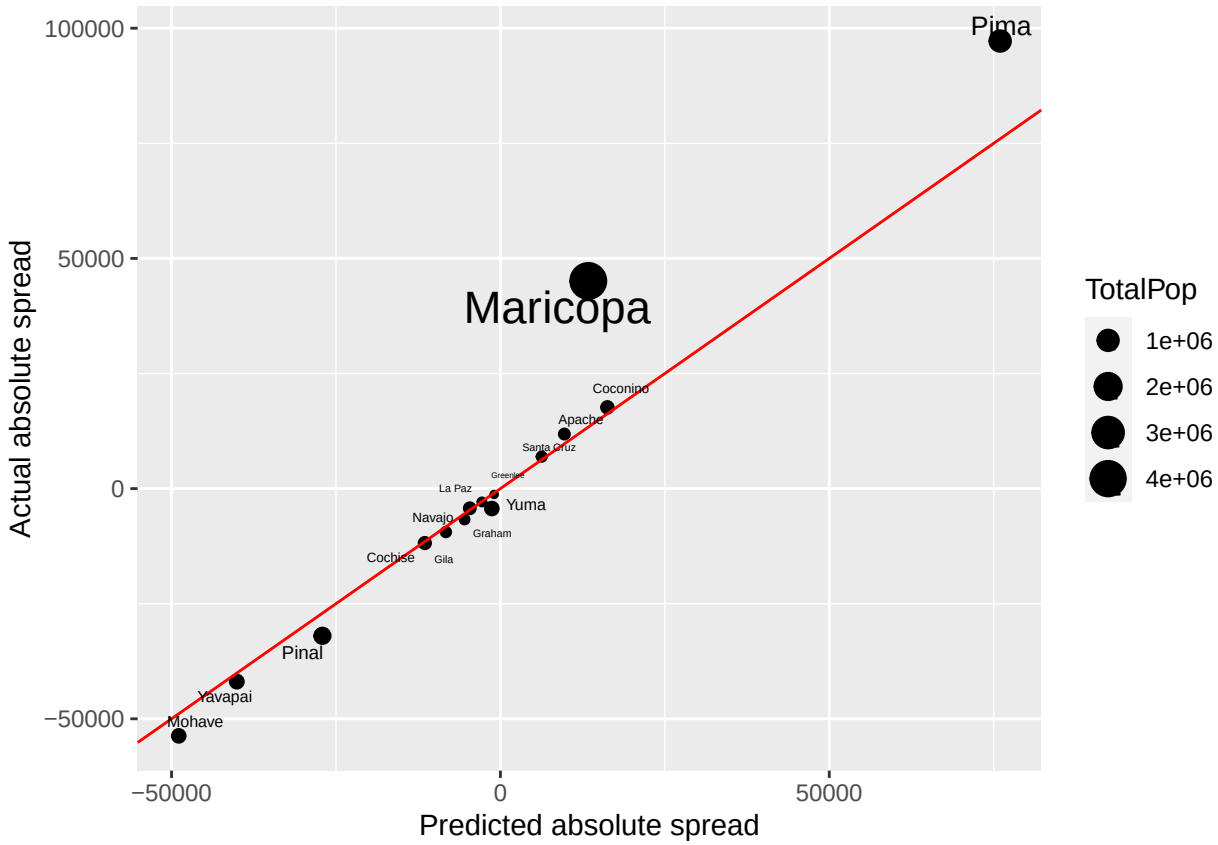
Georgia Why was Georgia not predicted correctly?



Especially in the largest counties of the Atlanta areas, DeKalb, Fulton, Gwinnet and Cobb, the model did not capture the very large increase of votes for Joe Biden over the trend derived from other states. The population numbers of these counties are between 700 thousand and a million. Multiplied with the RMSE this is 15 to 20 thousand. So the actual deviations are well within 2 standard errors (in this simplification), which makes them hardly outliers.

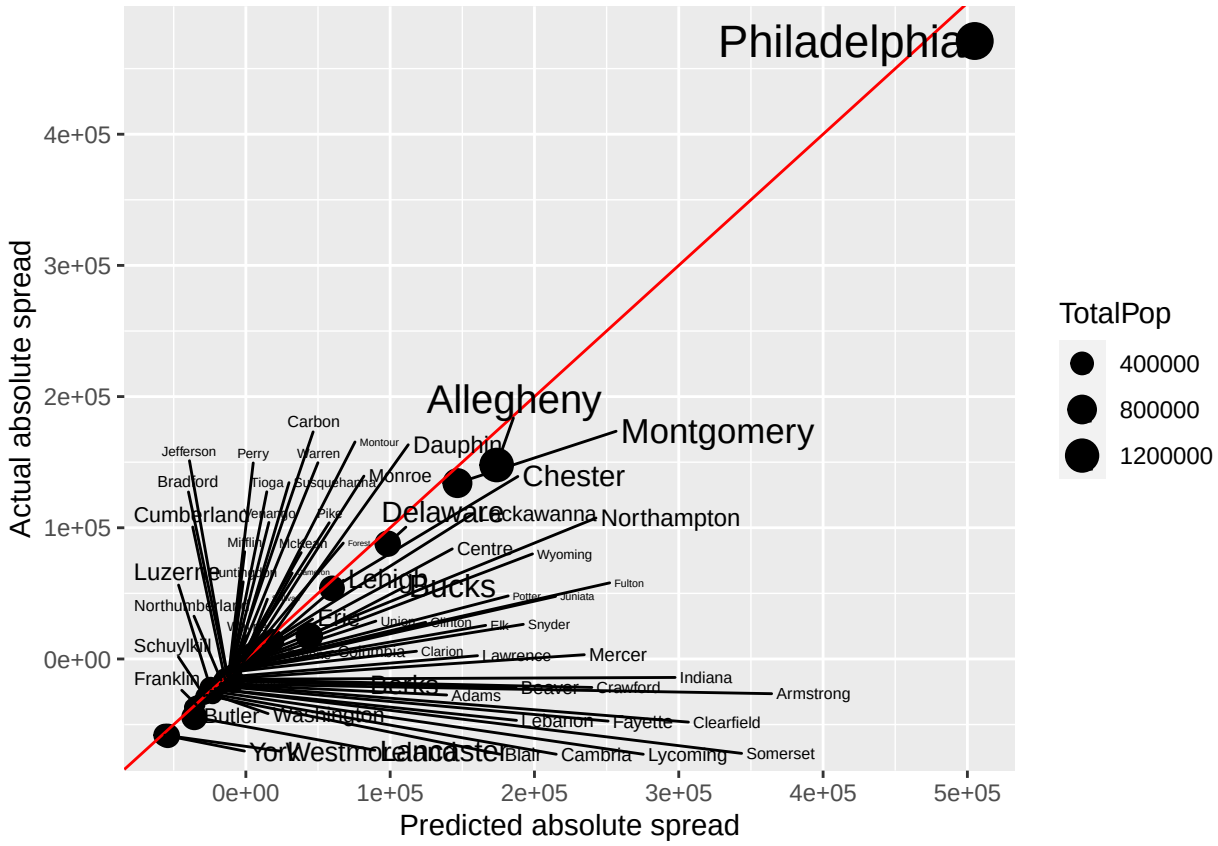
Although the data does not allow for a hard proof, the plot above indicates that there is a large percentage of Black population. Given the protests against systemic racism in Georgia and other places in the US in 2020, it seems likely that Black voters made a difference here. Again, the data cannot proof the last conclusion, but it is consistent with that explanation.

Arizona Finally, in the third state with wrong prediction, Arizona, the race was very close. It is not a real surprise or mistake that the model did not predict the sign correctly, because when close to zero the prediction is almost random. It turns out that the counties Maricopa (i.e., the Phoenix area) and Pima (i.e. the Tucson area) voted more strongly for Biden than the nationwide trend (captured by the model) would indicate.



The population of Maricopa is more than 4 million, Pima has population of about a million. Multiplied with the RMSE the errors would be 80 thousand or 20 thousand, respectively. In that sense neither of them is an outlier, but well within the expected range.

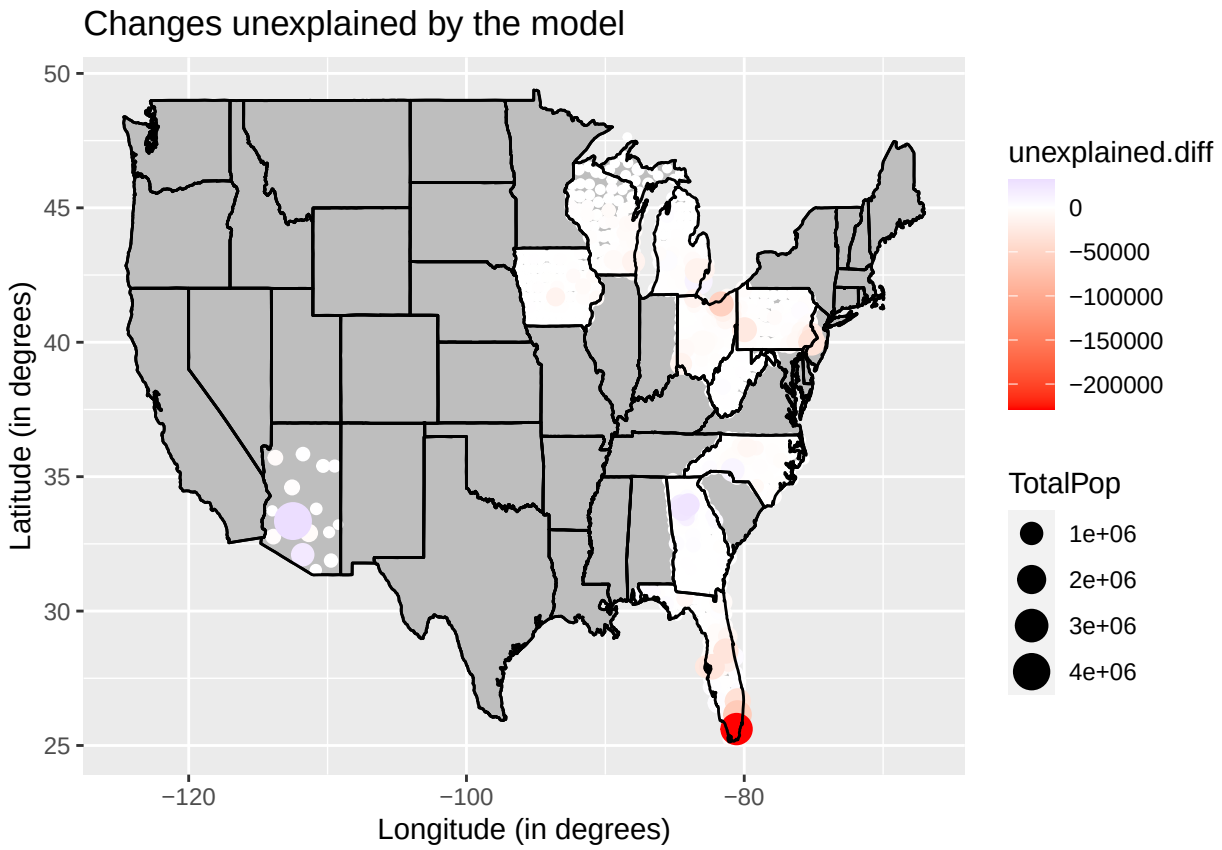
Pennsylvania As Pennsylvania played a crucial role, it seems important to analyze it:



It seems that in this state, which played a pivotal role in the election, the only material deviation from the 'trend' (represented by the fitted model) is a slight under-prediction of Biden votes in the largest counties. This means that the fact that Pennsylvania was won by Biden can be explained from the nationwide trend. Even if not a proof, it lacks of any support for fraud claims.

I close with a map of the held-out battleground states, showing the deviations from the predictions.

```
validation.extended %>%
  mutate(unexplained.diff = actual.Delta.20.abs - pred.Delta.20.abs) %>%
  arrange(abs(unexplained.diff)) %>%
  filter(!(state %in% c("AK","HI")) & long != 0 & lat != 0) %>%
  ggplot() +
  geom_polygon(data = state_map,aes(long,lat,group = group),fill="grey",color="black") +
  geom_point(aes(long,lat,color=unexplained.diff,size=TotalPop)) +
  geom_polygon(data = state_map,aes(long,lat,group = group),alpha = 0, color="black") +
  scale_color_gradient2(low="red",midpoint = 0.0,high = "blue") + # coord_fixed() +
  ggtitle("Changes unexplained by the model") +
  xlab("Longitude (in degrees)") +
  ylab("Latitude (in degrees)")
```



This demonstrates that the largest 'signal' of deviation from the general trend is one in Trump's favor. (These are the unexpectedly high number of Latino votes in Florida.) Much smaller is the excess number in Atlanta (GA). The votes in Georgia have been confirmed by recounts. As discussed, they are also reasonable given the developments in 2020. There are smaller deviations in both directions, including Phoenix, which was discussed above.

Results

Performance

The results for the application of the machine learning algorithms are

```
kable(Results)
```

model	RMSE	RMSE.weighted
baseline	0.0271421	0.0243186
lm.ftl.1	0.0229121	0.0244350
lm.ftl.2	0.0232514	0.0338432
lm.ftl.3	0.0232459	0.0337570
lm.cv	0.0232459	0.0337570
rpart	0.0226266	0.0214106
rpart.cv	0.0216363	0.0223476
rf	0.0176672	0.0156721
rf.cv	0.0174968	0.0157709
xgb.cv	0.0216435	0.0211175
knnreg	0.0279832	0.0249716
knn.cv	0.0263400	0.0257673
final.rf.cv (in sample)	0.0073922	0.0062570
final (out of sample)	0.0199906	0.0250891

They show that by fitting various models, the performance in terms of RMSE could be improved compared to the baseline model. The final model shows that a considerable RMSE reduction could be achieved and that this was confirmed on the validation set. When applied to absolute numbers per state, it gave a reasonable prediction of the state outcomes:

```
kable(StateResults)
```

state	pred.abs.spread.20	actual.abs.spread.20	pred.DEM.won	actual.DEM.won
AZ	-29398.6931	10457	FALSE	TRUE
FL	104822.7382	-371686	TRUE	FALSE
GA	-128297.3638	11779	FALSE	TRUE
IA	-78011.9486	-138611	FALSE	FALSE
MI	207544.1091	154188	TRUE	TRUE
NC	211.2833	-74481	TRUE	FALSE
OH	-227141.2805	-475669	FALSE	FALSE
PA	275290.6529	81660	TRUE	TRUE
WI	142890.7296	20608	TRUE	TRUE
WV	-329376.5800	-309398	FALSE	FALSE

The wrongly predicted states show effects "not explained" by the model in the sense that the predicted expectation value was off. These were further investigated above. As discussed there, deviations from the expectation value within the bounds given by the model RMSE should actually not be considered outliers.

Discussion of the deviations

As the goal was to look for deviations from the model covering the nationwide trends, the three wrongly predicted states are most interesting. The observations are:

- The only strong outlier was the county Miami-Dade, which pushed the result in favor for Trump. It was several times the RMSE away from the prediction and much larger in absolute vote numbers (roughly

around 200k votes) than the deviations in Georgia and Arizona (a few ten thousands each). But it can be explained by a very large number of Latinos voted for Trump in Florida, which has been extensively covered in the press.

- In Georgia, urban counties (Atlanta area) with a large Black community voted more strongly for Biden than the model predicted. But the deviation is still in a range, which can be expected from the RMSE value for the model. Given the 2020 protests against systemic racism in Georgia and overall the US, it seems very plausible that a few ten thousand votes can be explained by these events. With several vote recounts under a respected Republican Secretary of State, fraud can be excluded from the calculation.
- In Arizona the race was so close that a wrong prediction by the model is not surprising. The largest deviations are well within the expected range derived from the RMSE.
- Other cases of deviations were mostly in favor of the Republican candidate and were not further considered here.

Conclusion

Summary

I have tried to predict swing state results by nationwide trends that can be fitted from the data. I held out ten 'battleground' states and used the model to 'predict' the outcome in these states. I wrote 'predict' in quotes, because it is not a prediction of the result made before the election. Instead, it is a prediction based on the results of other states.

Summarizing, the election outcome in a selected sample of battleground states can be reasonably well explained from a model fitting the voting behavior in the remaining states. Deviations from the model were discussed and did not give hints for fraud. The greatest outlier to the model is one in favor for Trump, which is likely reflecting the many Latino votes for Trump. The second largest, in favor for Biden, can probably be explained a high voter turnout among Black voters of the Atlanta area. Although the data does not contain information necessary to link this to the events and protests in 2020, it would be surprising if this did not have influence on the election results.

If the general trend explains the results well in a large number of counties, the claim that the election was fraudulent becomes a hard sell. Because the fraud, like inserting votes, would then have to be orchestrated across the nation, which seems highly implausible.

Potential impact

If the analysis would be improved and combined with other sources of information (polls, media reports, etc.) it could have the potential to put unproven claims to the test.

Limitations

- Data quality: When I started the project the counting was still being finalized. The original dataset I used was preliminary and incomplete. So I had to combine it with additional datasets. An unresolved problem was that the data for some states were given by county in one election and township (or other types of administrative districts) in the other. While for some states the mapping can be found on the internet, it would require extra effort to include these states into the analysis.
- Prevalence: There are enormous differences in county population, which was an issue throughout the analysis. By subtracting off a local regression fit using a variable which scales with population, I hoped to reduce this issue. Some of the used implementations of the algorithms do not allow for weighting, which then could not be used.
- Explanation and interpretation based on statistics: It has to be kept in mind that the data does not contain the information who elected how. The only information is whether certain voter groups have a larger weight in one or the other county. This implies that only hints can be obtained from the data, but never certainty.

- Error analysis: I included a rough error analysis, but it was not expanded in full detail. Although I think it sufficed to separate outliers from expected fluctuations, it could be put on a sounder footing.

Future work

A major requirement for further work would be a sound database. I had to exclude states from the analysis, where vote counts of different elections were given in different granularity (e.g., counties versus townships). The demographic data was taken as is from the original dataset. It does not reflect changes in demographics between elections. Fixing these and other issues would be a necessary first step for further work.

The objectivity of the evaluation and interpretation could be improved by determining error bars more thoroughly. First of all, the residual error of the model could be quantified better, taking into account the different population sizes. Secondly, it would be interesting to add the 'sampling' error of a binomial (or other appropriate) distribution given a fixed spread as a reference. Thirdly, if the analysis is repeated for several election periods it might be possible to determine typical fluctuations between elections.

The analysis left out the expectations from polling data completely. This was not needed for the goals set. For further work this and other inputs could be merged. Further, I have left out the question about influencing of the public opinion by mass media, social media, or other groups. This would be an interesting project, but would probably require a lot of additional data and different techniques.