Final Project*

PSTAT 231

Villaseñor-Derbez J.C. | 8749749

Contents

1. What makes voter behavior prediction (and thus election forecasting) a hard problem?
2. What was unique to Nate Silver's approach in 2012 that allowed him to achieve good predictions?
3. What went wrong in 2016? What do you think should be done to make future predictions better?
4. Remove 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.

```
election_raw <- read.csv(here("data", "election", "election.csv")) %>%
  as_tibble()

census_meta <- read.csv(here("data", "census", "metadata.csv"), sep = ";") %>%
  as_tibble()

census <- read.csv(here("data", "census","census.csv")) %>%
  as_tibble() %>%
  mutate(CensusTract = as.factor(CensusTract))
```

```
election_federal <- election_raw %>%
  filter(fips == "US")

election_state <- election_raw %>%
  filter(state != "US", is.na(county))

election <- election_raw %>%
  filter(!is.na(county))
```

5. How many named presidential candidates were there in the 2016 election? Draw a bar chart of all votes received by each candidate

There were 31 explicitly mentioned presidential candidates, plus a category of `None of these acandidates`. Figure 1 shows the votes (on a log_{10}-scale) that each candidate received.

```
election_federal %>%
  group_by(candidate) %>%
  summarize(votes = sum(votes, na.rm = T)) %>%
  ungroup() %>%
  mutate(candidate = fct_reorder(.f = candidate, .x = votes)) %>%
  ggplot(aes(x = candidate, y = votes)) +
  geom_col() +
  coord_flip() +
```

*Code available on GitHUb at: https://github.com/jcvdav/PSTAT231/tree/master/final_project

```
  scale_y_continuous(trans = "log10") +
  labs(x = "Candidate", y = "Votes (log-10 Scale)")
```

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

```
county_winner <- election %>%
  group_by(fips) %>%
  mutate(total = sum(votes),
         pct = votes / total) %>%
  top_n(1) %>%
  ungroup() %>%
  select(fips, winner = candidate)
```

```
## Selecting by pct
```

```
state_winner <- election_state %>%
  group_by(state) %>%
  mutate(total = sum(votes),
         pct = votes / total) %>%
  top_n(1) %>%
  ungroup() %>%
  select(state, winner = candidate) %>%
  mutate(state = as.character(state))
```

```
## Selecting by pct
```

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

```
my_fun <- function(long, lat) {
  # long <- data$long
  # lat <- data$lat

  data <- cbind(long, lat) %>%
    rbind(cbind(long[1], lat[1]))
  st_sfc(st_polygon(list(as.matrix(data))))
}

state_dictionary <- tibble(abb = state.abb, state = tolower(state.name))

states <- map_data("state") %>%
  group_by(group, region) %>%
  summarize(geometry = my_fun(long, lat)) %>%
  ungroup() %>%
  st_sf(crs = 4326) %>%
  left_join(state_dictionary, by = c("region" = "state"))

counties <- map_data("county") %>%
  group_by(group, region, subregion) %>%
  summarize(geometry = my_fun(long, lat)) %>%
  ungroup() %>%
  st_sf(crs = 4326)
```

```
ggplot(data = counties,
       mapping = aes(fill = subregion)) +
```

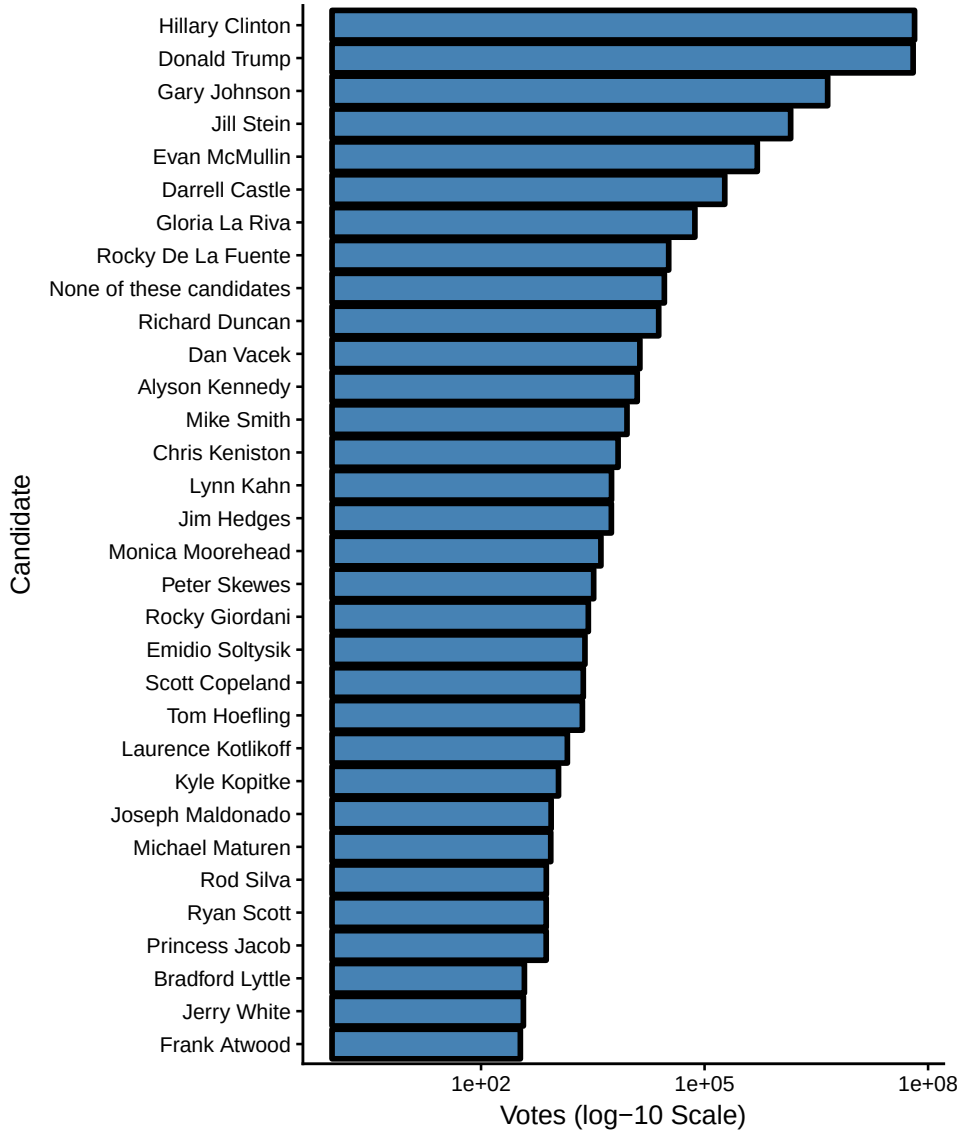
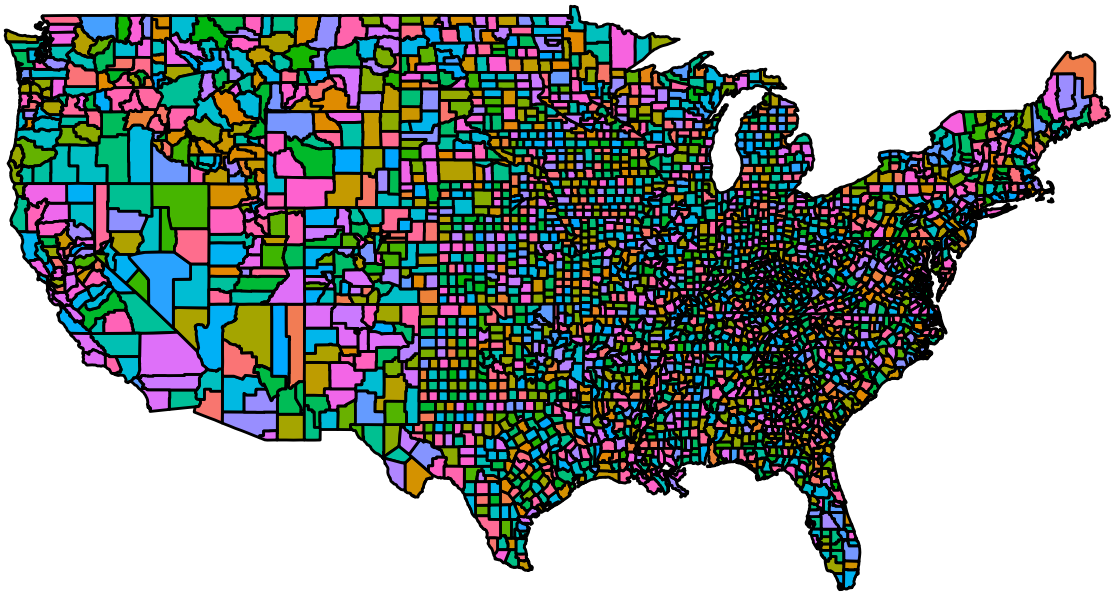


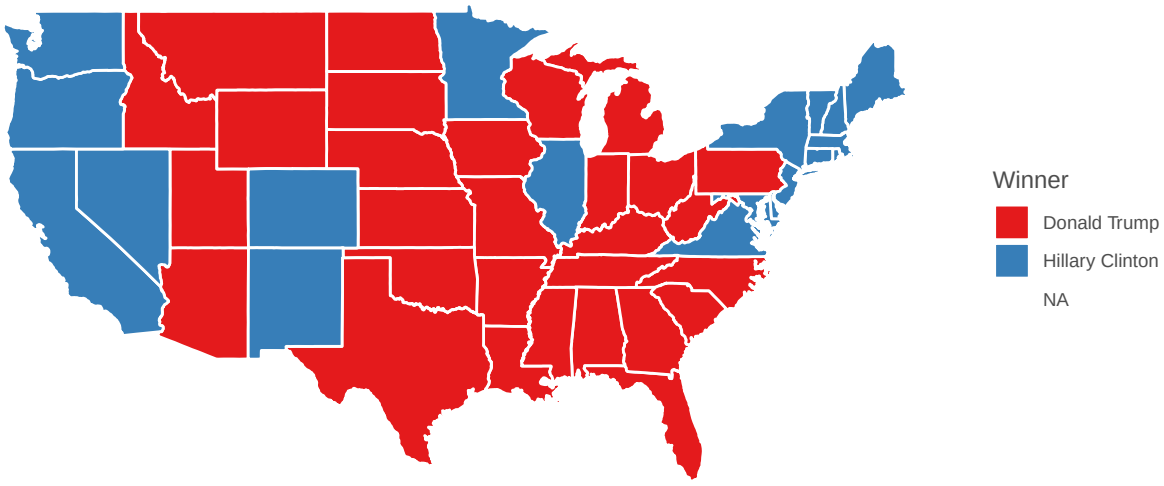
Figure 1: Number of votes that each presidential candidate received in the 2018 Presidential Elections.

```
  geom_sf(data = counties) +
  ggtheme_map() +
  theme(legend.position = "None")
```



Now color the map by the winning candidate for each state. First, combine states variable and `state_winner` we created earlier using `left_join()`. Note that `left_join()` needs to match up values of states to join the tables; however, they are in different formats: e.g. AZ vs. arizona. Before using `left_join()`, create a common column by creating a new column for states named `fips = state.abb[match(some_column, some_function(state.name))]`. Replace `some_column` and `some_function` to complete creation of this new column. Then `left_join()`. Your figure will look similar to `state_level` New York Times map.

```
states %>%
  left_join(state_winner, by = c("abb" = "state")) %>%
  ggplot(mapping = aes(fill = winner)) +
  geom_sf(color = "white") +
  ggtheme_map() +
  scale_fill_brewer(palette = "Set1") +
  guides(fill = guide_legend(title = "Winner"))
```


Winner
Donald Trump
Hillary Clinton
NA