fec16 Exploratory Data Analysis

Richard Robbins

March 19, 2022

Contents

1 Introduction

The w203 Unit 10 homework assignment revolves around some of the Federal Election Commission 2016 datasets available in the `fec16` R package. As I worked through that assignment I noticed several anomalies. In the wake of the assignment I decided to go back and take a much closer look at the data. This document summarizes what I found. This is a work in progress. I have identified what I believe to be the most meaningful issue in the data, however, as noted in Section 4.5, there are still issues to review.

2 Setup

This review looks primarily at two of the `fec16` datasets, `campaigns` and `results_house`. The working dataset for this exercise, `df`, is formed by an inner join of `campaigns` and `results_house`, after which candidates for whom votes were not recorded are removed. The working dataset is limited to columns of interest. The `candidates` dataset from the `fec2016` collection is also touched upon briefly in Section 3.3.

```
campaigns <- fec16::campaigns
candidates <- fec16::candidates
results_house <- fec16::results_house
```

```
df <- inner_join(campaigns, results_house) %>%
  drop_na(general_votes) %>%
  select (cand_id, cand_name, pty_cd, cand_pty_affiliation, party, incumbent,
          ttl_disb, general_votes, general_percent, won, state, district_id)
```

```
## Joining, by = "cand_id"
```

3 Troubling Signs

3.1 Distribution of General Election Voting Percentages

Figure 1 is a histogram of the general election voting percentages for candidates receiving votes as reflected in the `results_house` dataset. I was surprised to see the asymmetric mass on the left hand side of the histogram. I thought that perhaps seeing a larger group of candidates receiving low percentages might have reflected a third or fourth candidate in a contest between two other more dominant candidates. But I struggled to come up with a satisfactory explanation nonetheless.

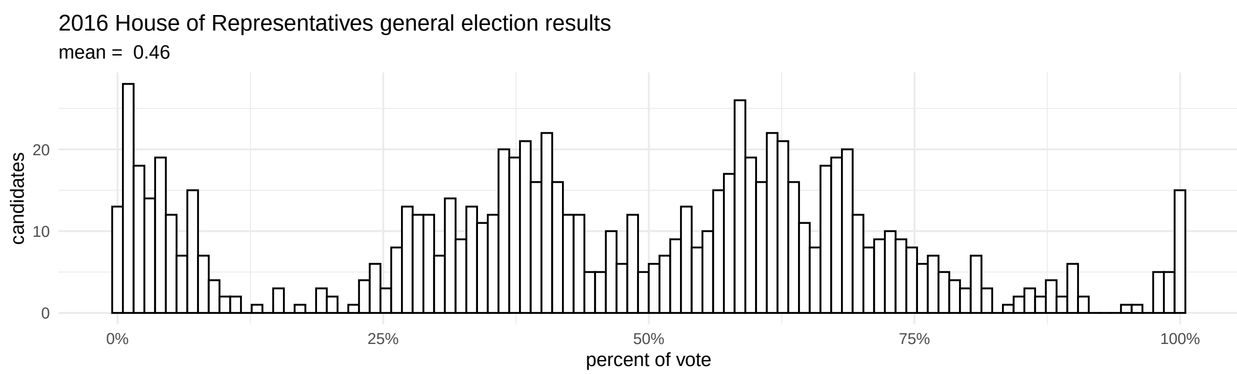

Figure 1: Voting Percentages

3.2 Party Affiliation

There are three ways to identify a candidate's party affiliation from the `campaigns` and `results_house` datasets: `results_house$party`, `campaigns$cand_pty_affiliation` and `campaigns$pty_cd`. Here are the unique values from each.

```
(unique (results_house$party))
```

```
##  [1] "REP"         "DEM"         "LIB"         "NAF"         "W"
##  [6] "IND"         "W(GRE)/GRE"  "W(LIB)"      "W(GRE)"      "W(D)"
## [11] "NOP"         "GRE"         "W(R)/R"      "PAF"         "W(NOP)"
## [16] "WF"          "IP"          "R/W"         "DCG"         "LBF"
## [21] "NPA"         "N"           "CON"         "W(R)"        "NNE"
## [26] "OTH"         "W(IND)"      "U"           "UST"         "W(D)/D"
## [31] "NLP"         "WC"          "DFL"         "IDP"         "LMN"
## [36] "REF"         "VPA"         "NPY"         "IAP"         "WDB"
## [41] "AO"          "MGW"         "RNN"         "PIP"         "FPR"
## [46] "EG"          "WUA"         "NSA"         "WOP"         "NBP"
## [51] "FI"          "LMP"         "TED"         "WTP"         "CRV"
## [56] "WEP"         "R/TRP"       "BLM"         "HBP"         "SID"
## [61] "TGP"         "PCC"         "UPJ"         "DNL"         "D/IP"
## [66] "W(IP)"       "R/IP"        "IP/R"        "PRO"         "D/PRO/WF/IP"
## [71] "R/CON"       "PG"          "W(D)/W"      "NPP"         "PPD"
## [76] "PRI"         "PPT"         "AM"          "UN"          "D/R"
## [81] "LBU"         "INP"         "WRN"         "TC"
```

```
(unique (campaigns$cand_pty_affiliation))
```

```
##  [1] "REP" "DEM" "IND" "GRE" "NNE" "UNK" "NPA" "LIB" "W"   "NON" "CON" "OTH"
## [13] "DFL" "IDP" "N/A" "UN"  "NPP" "PPT" "AMP" "CST" "GWP" "N"   "PBP" "PFD"
## [25] "PPY" "SEP"
```

```
(unique (campaigns$pty_cd))
```

```
## [1] 2 1 3
```

Party affiliation can be mapped as follows:

$$\text{for party or cand\_pty\_affiliation:} \begin{cases} \text{DEM} & \to \textit{Democrat} \\ \text{REP} & \to \textit{Republican} \\ \textit{else} & \to \textit{Other} \end{cases}$$

$$\text{for pty\_cd:} \begin{cases} 1 & \to \textit{Democrat} \\ 2 & \to \textit{Republican} \\ 3 & \to \textit{Other} \end{cases}$$

Which yields:

```
##                      Republicans Democrats Other
## party                        357       366   157
## cand_pty_affiliation         406       423    51
## pty_cd                       406       423    51
```

Using `party` yields 106 more "other party" candidates than the alternatives. The `cand_pty_affiliation` and `pty_cd` fields, as mapped, are equivalent for our purposes.

Let's review the winners (using `results_house$won`):

```
##                          Republicans Democrats Other
## party                            237       189    64
## cand_pty_affiliation             259       225     6
## pty_cd                           259       225     6
```

The numbers are troubling. The dataset indicates that 490 people were elected to the House of Representatives in the 2016 election. However, the number of voting representatives in the House of Representatives is fixed by law at no more than 435. Moreover, after the 2016 general election, all of the voting members of the House of Representatives were either Republicans or Democrats.

3.3 Multiple Observations For Some Candidates

While working with the data, I found instances of candidate IDs appearing more than once in `results_house` and, to a lesser degree, candidate names appearing more than once in `campaigns`.

To explore this, I wrote a pair of functions, one to count the number of times any candidate id (`cand_id`) appears more than once in a dataset, and another to count the number of times a candidate's name (`cand_name`) appears more than once in a dataset.

```r
redundant.names <- function (dataset){
  if ("cand_name" %in% colnames (dataset))
    {dataset %>% group_by (cand_name) %>% count() %>% filter (n>1)}
  else {NA}
}

redundant.ids <- function (dataset){
  if ("cand_id" %in% colnames (dataset))
    {dataset %>% group_by (cand_id) %>% count() %>% filter (n>1)}
  else {NA}
}
```

The table below shows the degree to which the various datasets include more than one observation for a single candidate id or name.

```
##         dataset redundant.names redundant.ids
## 1     campaigns              19             0
## 2    candidates              59             0
## 3 results_house              NA            97
## 4            df              52            52
```

Lets take a look at the data for a few candidates appearing more than once the working dataset.

```
## # A tibble: 10 x 7
##    cand_id   cand_name       pty_cd party ttl_disb general_votes general_percent
##    <chr>     <chr>            <dbl> <chr>    <dbl>         <dbl>           <dbl>
##  1 H0CT03072 DELAURO, ROSA L      1 DEM   1150879.        192274         0.621
##  2 H0CT03072 DELAURO, ROSA L      1 WF    1150879.         21298         0.0688
##  3 H0NY29054 REED, THOMAS W~      2 REP   3072934.        136964         0.490
##  4 H0NY29054 REED, THOMAS W~      2 CRV   3072934.         16420         0.0587
##  5 H0NY29054 REED, THOMAS W~      2 REF   3072934.           876         0.00313
##  6 H0NY29054 REED, THOMAS W~      2 IDP   3072934.          6790         0.0243
##  7 H2CT02112 COURTNEY, JOSE~      1 DEM   1154847.        186210         0.564
```

```
##  8 H2CT02112 COURTNEY, JOSE~       1 WF    1154847.          22608          0.0685
##  9 H2CT05131 ESTY, ELIZABETH       1 DEM   1447329.         163499          0.529
## 10 H2CT05131 ESTY, ELIZABETH       1 WF    1447329.          15753          0.0510
```

There is a pattern. The rows for the candidates differ in that it appears that a candidate's total vote and percent of vote received are splintered and allocated to several parties.

The campaign spend field (`ttl_disb`) field is consistent for a candidate. That is an artifact of how we joined `campaigns` and `results_house`. The `campaigns` dataset is the source of the campaign spending information and that dataset does not contain redundant campaign ID observations. When we perform the join, the spending information is replicated in each redundant observation in `results_house`. As a result, our working dataset not only has too many other party candidates, campaign spend is inflated.

Upon review of generally available election returns, for the candidates listed above, I confirmed that the total votes and correct percent of votes for each can be determined by adding the amounts shown in their respective redundant observations.

4 Mitigation

4.1 The Approach

Based on the foregoing, it seems reasonable to group the candidates by ID, name, party affiliation (using `pty_cd`) and, to be prudent, other indicator fields, such as campaign spend whether they were an incumbent and whether they won. Then we can derive the vote and percentage of votes for each group by taking the sum of those field for each row in the group.

In most cases where there are no duplicate entries, this step will result in the original row for the candidate being preserved. But for candidates with multiple entries based on the candidate ID field, this step will collapse those rows into a single row with the correct number of votes and percent of votes received.

The refined date frame can be derived from the original as follows:

```
df.deduped <- df %>%
  group_by(cand_id, cand_name, pty_cd, incumbent, won, ttl_disb, state) %>%
  summarise (general_votes = sum(general_votes),
             general_percent = sum(general_percent),
             .groups = "keep")
```

4.2 Assessing Mitigation Effectiveness

Let's review what the dataset looks like after the mitigation step described above.

```
(NROW(df))
```

```
## [1] 880
```

```
(NROW(df.deduped))
```

```
## [1] 792
```

```
(redundant.ids(df.deduped))
```

```
## # A tibble: 1 x 2
## # Groups:   cand_id [1]
##   cand_id       n
##   <chr>     <int>
## 1 H2NY03089     2
```

```
(redundant.names(df.deduped))
```

```
## # A tibble: 1 x 2
## # Groups:   cand_name [1]
##   cand_name               n
##   <chr>               <int>
## 1 KING, PETER T. HON.     2
```

1. The number of observations has been reduced from 880 to 792.
2. The number of redundant candidate IDs has been reduced to 1.
3. The number of redundant candidate names has been reduced to 1.

The same candidate is reflected in the second and third items above.

The breakdown by party is as follows:

```
##        Republicans Democrats Other
## pty_cd         366       376    50
```

And if we isolate on the winners:

```
##        Republicans Democrats Other
## pty_cd         236       194     6
```

4.3 Examining "Other Party" Winners

The six candidates identified as winning but not affiliated with a major party are:

```
## # A tibble: 6 x 9
## # Groups:   cand_id, cand_name, pty_cd, incumbent, won, ttl_disb, state [6]
##   cand_id   cand_name        pty_cd incumbent won   ttl_disb state general_votes
##   <chr>     <chr>             <dbl> <lgl>     <lgl>    <dbl> <chr>         <dbl>
## 1 H0MN04049 MCCOLLUM, BETTY       3 TRUE      TRUE   966856. MN           203299
## 2 H2IL13120 DAVIS, RODNEY L       3 TRUE      TRUE  2364757. IL           187583
## 3 H2MN08111 NOLAN, RICHARD ~      3 TRUE      TRUE  2892902. MN           179098
## 4 H4MO08162 SMITH, JASON T        3 TRUE      TRUE  1312777. MO           229792
## 5 H6PR00082 GONZALEZ COLON,~      3 FALSE     TRUE   917851. PR           718591
## 6 H8MP00041 SABLAN, GREGORI~      3 TRUE      TRUE    56475. MP            10605
## # ... with 1 more variable: general_percent <dbl>
```

- Betty McCollum and Richard Nolan are Democrats. The dataset is incorrect.
- Rodney David and Jason Smith are Republicans. The dataset is incorrect.

- Jennifer Gonzalez is neither a Republican nor a Democrat. She represents Puerto Rico as a delegate.
- Greogio Sablan is now a Republican. In 2016 he was neither a Republican nor a Democrat. He represents the Northern Mariana Islands as a delegate.

Because Puerto Rico and the Northern Mariana Islands are territories of the United States, and not states, their representatives in the House of Representatives are delegates with limited voting privileges. Delegates can currently vote in committee and in certain votes on the House floor, but not if their vote would be decisive. Delegates have a marginalized role in Congress and their constituents are not represented in Congress in the same manner as most citizens.

So, while there are 436 people identified as winners in the 2016 House of Representatives election, that include 2 delegates who are not considered voting members of the House. Accordingly, the data is now consistent with the limit on voting members in the House (435) and every voting member has been accounted for as a Republican or Democrat.

4.4 The Lone Duplicate Entry Remaining

That leaves one person with duplicate entries in the dataset. That distinction goes to Peter T. King. His record was not merged because when we formed the grouping, to be conservative, we added several indicators from the dataset to prevent merging records that warranted further examination. In this case, the dataset for Mr. King has one row indicating he is not an incumbent who lost and another indicating he is an incumbent who won. In fact, Mr. King was an incumbent and he won re-election in 2016 with 181,506 votes.

```
## # A tibble: 2 x 9
## # Groups:   cand_id, cand_name, pty_cd, incumbent, won, ttl_disb, state [2]
##   cand_id   cand_name        pty_cd incumbent won   ttl_disb state general_votes
##   <chr>     <chr>             <dbl> <lgl>     <lgl>    <dbl> <chr>         <dbl>
## 1 H2NY03089 KING, PETER T. ~      2 FALSE     FALSE 1310730. NY            23935
## 2 H2NY03089 KING, PETER T. ~      2 TRUE      TRUE  1310730. NY           157571
## # ... with 1 more variable: general_percent <dbl>
```

4.5 Revisiting that Histogram

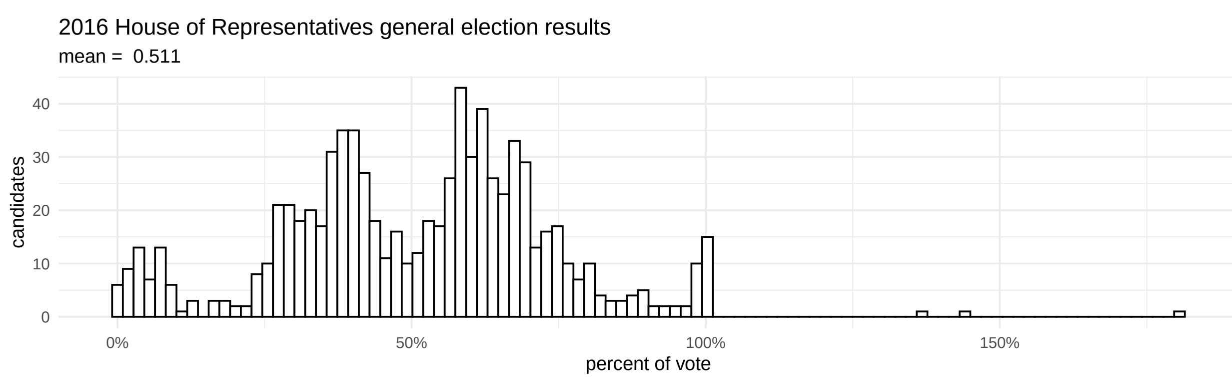

Figure 2: Revised Voting Percentages

Figure 2 reproduces the histogram from Figure 1 using the refined dataset. The asymmetric mass on the left hand side of the histogram is no longer present. The histogram appears to be much more symmetric.

However, we now appear to have a few candidates receiving more than 100% of the vote in their races. That suggests that there are cases where our mitigation approach is not warranted, *i.e.*, where the correct number of votes and percentages cannot be derived by adding the redundant observations.

5 Vote Totals, Write In Votes ansd Uncontested Elections

Figure 2 suggests we should take a closer look at the voting data contained in the `results_house` dataset. We should not see anyone with a vote percentage that exceeds 100%. In addition, it's reasonable to expect that in most cases, the sum of the votes received by candidates on the ballot will be less than total votes cast in order to account for write in candidates. Finally, we should think about uncontested elecetions. We explore each of these concepts below.

5.1 Vote Totals

We can derive the implied total number of votes cast from the number of votes received by a candidate and the percentage of votes received by that candidate, *i.e.,* $votes\_cast = \frac{general\_votes}{general\_percent}$. We would expect that number to be the same when calculuated across every observation for a particular race.

First, let's see votes cast for some observations.

```
results_house %>%
  drop_na(general_votes) %>%
  mutate (votes_cast = general_votes/general_percent) %>%
  select (state, district_id, cand_id, general_votes, general_percent, votes_cast)
```

```
## # A tibble: 1,291 x 6
##    state district_id cand_id   general_votes general_percent votes_cast
##    <chr> <chr>       <chr>             <dbl>           <dbl>      <dbl>
##  1 AL    01          H4AL01123        208083           0.964     215893
##  2 AL    02          H0AL02087        134886           0.488     276584
##  3 AL    02          H6AL02167        112089           0.405     276584
##  4 AL    03          H2AL03032        192164           0.669     287104
##  5 AL    03          H4AL03061         94549           0.329     287104
##  6 AL    04          H6AL04098        235925           0.985     239444
##  7 AL    05          H0AL05163        205647           0.667     308326
##  8 AL    05          H6AL05202        102234           0.332     308326
##  9 AL    06          H4AL06098        245313           0.745     329306
## 10 AL    06          H6AL06127         83709           0.254     329306
## # ... with 1,281 more rows
```

That looks good. The rows for a stete and district pair imply a consistent number of votes cast for the candidates running in that particular contest.

Next we calculate the number of unique votes cast totals for each contest.

```
results_house %>%
  drop_na(general_votes) %>%
  mutate (votes_cast = general_votes/general_percent) %>%
  group_by(state, district_id) %>%
  summarise(unique_votes_cast = length(unique(votes_cast)), .groups = "keep")
```

```
## # A tibble: 443 x 3
## # Groups:   state, district_id [443]
##    state district_id unique_votes_cast
##    <chr> <chr>                   <int>
##  1 AK    00                          1
##  2 AL    01                          1
##  3 AL    02                          1
##  4 AL    03                          1
##  5 AL    04                          1
##  6 AL    05                          1
##  7 AL    06                          1
##  8 AL    07                          1
##  9 AR    01                          1
## 10 AR    02                          1
## # ... with 433 more rows
```

That sample shows a consistent number for the contests shown. Let's look for exceptions.

```
results_house %>%
  drop_na(general_votes) %>%
  mutate (votes_cast = general_votes/general_percent) %>%
  group_by(state, district_id) %>%
  summarise(unique_votes_cast = length(unique(votes_cast)), .groups = "keep") %>%
  filter(unique_votes_cast > 1)
```

```
## # A tibble: 71 x 3
## # Groups:   state, district_id [71]
##    state district_id unique_votes_cast
##    <chr> <chr>                   <int>
##  1 AS    00                          2
##  2 CA    19                          2
##  3 CA    20                          2
##  4 CA    22                          2
##  5 CA    27                          2
##  6 CA    31                          2
##  7 CA    46                          2
##  8 CA    47                          2
##  9 CA    52                          2
## 10 CO    01                          2
## # ... with 61 more rows
```

That test shows more than a few contests where we appear to be getting inconsistent total votes cast numbers. Perhaps this is due to rounding errors. Let's round the votes cast number to the nearest integer.

```
results_house %>%
  drop_na(general_votes) %>%
  mutate (votes_cast = round(general_votes/general_percent, 0)) %>%
  group_by(state, district_id) %>%
  summarise(unique_votes_cast = length(unique(votes_cast)), .groups = "keep") %>%
  filter(unique_votes_cast > 1)
```

```
## # A tibble: 2 x 3
```

```
## # Groups:   state, district_id [2]
##   state district_id unique_votes_cast
##   <chr> <chr>                   <int>
## 1 KS    01                          2
## 2 PA    07                          2
```

We are down to just suspicious contests. Let's take a closer look.

```
results_house %>%
  drop_na(general_votes) %>%
  mutate (votes_cast = round(general_votes/general_percent, 0)) %>%
  group_by(state, district_id) %>%
  summarise(unique_votes_cast = length(unique(votes_cast)),
            unique_vote_list = paste(unique(votes_cast), collapse=", "),
            .groups = "keep") %>%
  filter(unique_votes_cast > 1)
```

```
## # A tibble: 2 x 4
## # Groups:   state, district_id [2]
##   state district_id unique_votes_cast unique_vote_list
##   <chr> <chr>                   <int> <chr>
## 1 KS    01                          2 257971, NA
## 2 PA    07                          2 NA, 379649
```

It seems that we were unable to calculate the total votes cast in some instances because the denominator, `general_percent` was missing. Let's see.

```
results_house %>%
  drop_na(general_votes) %>%
  filter(is.na (general_percent)) %>%
  select (state, district_id, cand_id, general_votes, general_percent)
```

```
## # A tibble: 2 x 5
##   state district_id cand_id   general_votes general_percent
##   <chr> <chr>       <chr>             <dbl>           <dbl>
## 1 KS    01          H6KS01146           874              NA
## 2 PA    07          H0PA07082        225678              NA
```

So we have two rows missing the percentage of votes received by the candidate.

5.2 Write In Votes

Now that we have confirmed that the data for the candidates participating in a contest implies a consistent total number of votes cast in the election, let's compare the total number of votes accounted for in each contest with the number of votes cast that we have derived. When the numbers match that means that there are no write in ballots included in the observations for that contest. When the total votes recorded is less than the total vote cast number we expect for that election the deficit represents either missing candidates or write in ballots. The total number of votes recorded should never exceed the votes cast number we recorded.

```r
vote_reconciliation <- results_house %>%
  drop_na(general_votes) %>%
  mutate (implied_votes_cast = round(general_votes/general_percent, 0)) %>%
  group_by(state, district_id) %>%
  summarise(implied_votes_cast = max(implied_votes_cast, na.rm=TRUE),
            votes_recorded = sum(general_votes),
            vote_deficit = implied_votes_cast - votes_recorded,
            .groups = "keep")

summary(vote_reconciliation$vote_deficit)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0    1047     182  209810
```

```r
sum(vote_reconciliation$vote_deficit == 0)
```

```
## [1] 299
```

```r
sum(between(vote_reconciliation$vote_deficit, 1, 1000))
```

```
## [1] 119
```

```r
sum(between(vote_reconciliation$vote_deficit, 1001, 10000))
```

```
## [1] 21
```

```r
sum(vote_reconciliation$vote_deficit > 10000)
```

```
## [1] 4
```

```r
vote_reconciliation %>% filter (vote_deficit > 10000)
```

```
## # A tibble: 4 x 5
## # Groups:   state, district_id [4]
##   state district_id         implied_votes_cast votes_recorded vote_deficit
##   <chr> <chr>                            <dbl>          <dbl>        <dbl>
## 1 AL    02                              276584         246975        29609
## 2 KY    01 - UNEXPIRED TERM             290623         209810        80813
## 3 KY    1 - UNEXPIRED TERM              290623          80813       209810
## 4 VA    11                              282322         247818        34504
```

We have non-conforming district identification numbers in the dataset. Let's take a look.

```r
results_house %>%
  drop_na(general_votes) %>%
  filter (!str_detect(district_id, "^[:digit:]+$")) %>%
  group_by (state, district_id) %>%
  summarise(count = n(), .groups="keep")
```

```
## # A tibble: 7 x 3
## # Groups:   state, district_id [7]
##   state district_id         count
##   <chr> <chr>               <int>
## 1 HI    01 - FULL TERM          4
## 2 HI    01 - UNEXPIRED TERM    10
## 3 KY    01 - FULL TERM          3
## 4 KY    01 - UNEXPIRED TERM     1
## 5 KY    1 - UNEXPIRED TERM      1
## 6 PA    02 - FULL TERM          2
## 7 PA    02 - UNEXPIRED TERM     2
```

Since we have been grouping by state and district id numbers, we should fix these labels.

```
vote_reconciliation <- results_house %>%
  drop_na(general_votes) %>%
  mutate (implied_votes_cast = round(general_votes/general_percent, 0)) %>%
  mutate (district_id = str_pad(str_extract(district_id, "[:digit:]{1,2}"),
                                2, pad="0")) %>%
  group_by(state, district_id) %>%
  summarise(implied_votes_cast = max(implied_votes_cast, na.rm=TRUE),
            votes_recorded = sum(general_votes),
            vote_deficit = implied_votes_cast - votes_recorded,
            .groups = "keep")

sum(vote_reconciliation$vote_deficit == 0)
```

```
## [1] 296
```

```
sum(between(vote_reconciliation$vote_deficit, 1, 1000))
```

```
## [1] 117
```

```
sum(between(vote_reconciliation$vote_deficit, 1001, 10000))
```

```
## [1] 21
```

```
sum(vote_reconciliation$vote_deficit > 10000)
```

```
## [1] 2
```

```
vote_reconciliation %>% filter (vote_deficit > 10000)
```

```
## # A tibble: 2 x 5
## # Groups:   state, district_id [2]
##   state district_id implied_votes_cast votes_recorded vote_deficit
##   <chr> <chr>                    <dbl>          <dbl>        <dbl>
## 1 AL    02                      276584         246975        29609
## 2 VA    11                      282322         247818        34504
```

Let's repeat out vote total integrity test.

```r
results_house %>%
  drop_na(general_votes) %>%
  mutate (votes_cast = round(general_votes/general_percent, 0)) %>%
  group_by(state, district_id) %>%
  summarise(unique_votes_cast = length(unique(votes_cast)), .groups = "keep") %>%
  filter(unique_votes_cast > 1)
```

```
## # A tibble: 2 x 3
## # Groups:   state, district_id [2]
##   state district_id unique_votes_cast
##   <chr> <chr>                   <int>
## 1 KS    01                          2
## 2 PA    07                          2
```

No changes there.

6 What About Homework 10?

A review of the model that I built for Unit 10 is beyond what I aim to achieve in this summary. However, I did redo much of my work using the refined dataset that reflects the changes described here.

Had I used this dataset I would have submitted a different model. Using the original dataset, I concluded that there was a statistically significant different between Republicans and Democrats when it came to the impact of campaign spending on votes received. In my model the constant associated with being a Democrat was higher than that of being a Republican, however, the effectiveness of campaign spend for Republicans outpaced that of Democrats. When I repeat my work using this data, that difference disappears. With this data, I found no difference between Republicans and Democrats. In both cases there was a significant difference between being affiliated with one of the major parties or not. With the cleaner data, the total residuals dropped and adjusted R^2 went from .258 to .426.