Predictr Code

Kiernan Nicholls

Spring, 2019

```
library(devtools) # installing functions
install_cran("here") # for local storage
install_cran("tidyverse") # for data manipulation
install_cran("verification") # for forecast analysis
install_github("hrbrmstr/wayback") # for internet archives
```

```
library(readr)     # reading data
library(dplyr)     # wrangling data
library(tidyr)     # tidying data
library(stringr)   # character strings
library(wayback)   # reading archives
library(ggplot2)   # plotting data
library(magrittr)  # piping data
library(lubridate) # dates strings
```

Read Input Data

Input data has be manually archived on the The Wayback Machine is a digital archive of the World Wide Web run by the Internet Archive, a nonprofit organization. Using the `wayback` package, "memento" files can be retrieved from the internet and scraped by the `readr` package into tibble data frames.

Read market data

First, we will read prediction market data courtesy of PredictIt, an exchange owned and operated by the Victoria University of Wellington. As part of their operating agreement with the Commodity Futures Trading Commission, PredicIt provides market history data for free to academic researchers.

The data was provided via email as a tab-separated file and can be loaded with `readr`. Two separate files were sent with the data on the Maine 2nd and New York 27th congressional districts, which were accidentally left out from the the main file. All data can be found in the `/data` folder.

```
DailyMarketData <-
  here::here("data", "DailyMarketData.csv") %>%
  read_delim(delim = "|",
             na = "n/a",
             col_types = cols(
               MarketId = col_character(),
               ContractName = col_character(),
               ContractSymbol = col_character(),
               Date = col_date(format = "")))

Market_ME02 <-
  here::here("data", "Market_ME02.csv") %>%
  read_csv(col_types = cols(ContractID = col_character(),
                            Date = col_date(format = "%m/%d/%Y")))

Contract_NY27 <-
```

```
  here::here("data" , "Contract_NY27.csv") %>%
  read_csv(na = c("n/a", "NA"),
           skip = 156,
           col_types = cols(ContractID = col_character(),
                            Date = col_date(format = "%m/%d/%Y")))

DailyMarketData
```

```
## # A tibble: 44,711 x 11
##    MarketId MarketName MarketSymbol ContractName ContractSymbol Date
##    <chr>    <chr>      <chr>        <chr>        <chr>          <date>
##  1 2918     Will Eliz~ WARREN.MASE~ <NA>         <NA>           2017-01-27
##  2 2918     Will Eliz~ WARREN.MASE~ <NA>         <NA>           2017-01-28
##  3 2918     Will Eliz~ WARREN.MASE~ <NA>         <NA>           2017-01-29
##  4 2918     Will Eliz~ WARREN.MASE~ <NA>         <NA>           2017-01-30
##  5 2918     Will Eliz~ WARREN.MASE~ <NA>         <NA>           2017-01-31
##  6 2918     Will Eliz~ WARREN.MASE~ <NA>         <NA>           2017-02-01
##  7 2918     Will Eliz~ WARREN.MASE~ <NA>         <NA>           2017-02-02
##  8 2918     Will Eliz~ WARREN.MASE~ <NA>         <NA>           2017-02-03
##  9 2918     Will Eliz~ WARREN.MASE~ <NA>         <NA>           2017-02-04
## 10 2918     Will Eliz~ WARREN.MASE~ <NA>         <NA>           2017-02-05
## # ... with 44,701 more rows, and 5 more variables: OpenPrice <dbl>,
## #   LowPrice <dbl>, HighPrice <dbl>, ClosePrice <dbl>, Volume <dbl>
```

Read member data

Congressional member data is used to provide party information as well as ideology and leadership scores. The data comes from the [the @unitedstates project]05 and GovTrack.

```
## Current members of the 115th
## Archived: 2018-10-22 at 18:11
legislators_current <-
  "https://theunitedstates.io/congress-legislators/legislators-current.csv" %>%
  read_memento(timestamp = "2018-10-22", as = "raw") %>%
  read_csv(col_types = cols(govtrack_id = col_character()))

# The ideology and leadership scores of the 115th
# Calculated with cosponsorship analysis
# Archived 2019-01-21 17:13:08
sponsorshipanalysis_h <-
  str_c("https://www.govtrack.us/",
        "data/analysis/by-congress/115/sponsorshipanalysis_h.txt") %>%
  read_memento(timestamp = "2019-03-23", as = "raw") %>%
  read_csv(col_types = cols(ID = col_character()))

sponsorshipanalysis_s <-
  str_c("https://www.govtrack.us/",
        "data/analysis/by-congress/115/sponsorshipanalysis_s.txt") %>%
  read_memento(timestamp = "2019-03-23", as = "raw") %>%
  read_csv(col_types = cols(ID = col_character()))

legislators_current
```

```
## # A tibble: 534 x 34
##    last_name first_name middle_name suffix nickname full_name birthday
```

```
##    <chr>     <chr>      <chr>       <chr>  <chr>    <chr>     <date>
##  1 Brown     Sherrod    <NA>        <NA>   <NA>     Sherrod ~ 1952-11-09
##  2 Cantwell  Maria      <NA>        <NA>   <NA>     Maria Ca~ 1958-10-13
##  3 Cardin    Benjamin   L.          <NA>   <NA>     Benjamin~ 1943-10-05
##  4 Carper    Thomas     Richard     <NA>   <NA>     Thomas R~ 1947-01-23
##  5 Casey     Robert     P.          Jr.    Bob      Robert P~ 1960-04-13
##  6 Corker    Bob        <NA>        <NA>   <NA>     Bob Cork~ 1952-08-24
##  7 Feinstein Dianne     <NA>        <NA>   <NA>     Dianne F~ 1933-06-22
##  8 Hatch     Orrin      G.          <NA>   <NA>     Orrin G.~ 1934-03-22
##  9 Klobuchar Amy        Jean        <NA>   <NA>     Amy Klob~ 1960-05-25
## 10 McCaskill Claire     <NA>        <NA>   <NA>     Claire M~ 1953-07-24
## # ... with 524 more rows, and 27 more variables: gender <chr>, type <chr>,
## #   state <chr>, district <dbl>, senate_class <dbl>, party <chr>,
## #   url <chr>, address <chr>, phone <chr>, contact_form <chr>,
## #   rss_url <chr>, twitter <chr>, facebook <chr>, youtube <chr>,
## #   youtube_id <chr>, bioguide_id <chr>, thomas_id <chr>,
## #   opensecrets_id <chr>, lis_id <chr>, fec_ids <chr>, cspan_id <dbl>,
## #   govtrack_id <chr>, votesmart_id <dbl>, ballotpedia_id <chr>,
## #   washington_post_id <lgl>, icpsr_id <dbl>, wikipedia_id <chr>
```

Read model data

Forecasting model data is courtesy of FiveThirtyEight, who provides the top-level output of their proprietary model for free to the public.

```
## District level 538 House model history
## Updated:  2018-11-06 at 01:56
## Archived: 2018-11-06 at 12:06
house_district_forecast <-
  str_c(site = "https://projects.fivethirtyeight.com/",
        file = "congress-model-2018/house_district_forecast.csv") %>%
  read_memento(timestamp = "2018-11-06", as = "raw") %>%
  read_csv()

# Seat level 538 Senate model history
# Updated:  2018-11-06 at 11:06
# Archived: 2018-11-06 at 21:00
senate_seat_forecast <-
  str_c(site = "https://projects.fivethirtyeight.com/",
        file = "congress-model-2018/senate_seat_forecast.csv") %>%
  read_memento(timestamp = "2018-11-06", as = "raw") %>%
  read_csv()

house_district_forecast
```

```
## # A tibble: 299,760 x 12
##    forecastdate state district special candidate party incumbent model
##    <date>       <chr>    <dbl> <lgl>   <chr>     <chr> <lgl>     <chr>
##  1 2018-08-01   AK           1 NA      Don Young R     TRUE      clas~
##  2 2018-08-01   AK           1 NA      Alyse S.~ D     FALSE     clas~
##  3 2018-08-01   AK           1 NA      Others    <NA>  FALSE     clas~
##  4 2018-08-01   AL           1 NA      Bradley ~ R     TRUE      clas~
##  5 2018-08-01   AL           1 NA      Robert K~ D     FALSE     clas~
##  6 2018-08-01   AL           2 NA      Martha R~ R     TRUE      clas~
##  7 2018-08-01   AL           2 NA      Tabitha ~ D     FALSE     clas~
```

```
##  8 2018-08-01   AL           3 NA      Mike Rog~ R     TRUE      clas~
##  9 2018-08-01   AL           3 NA      Mallory ~ D     FALSE     clas~
## 10 2018-08-01   AL           4 NA      Robert B~ R     TRUE      clas~
## # ... with 299,750 more rows, and 4 more variables: win_probability <dbl>,
## #   voteshare <dbl>, p10_voteshare <dbl>, p90_voteshare <dbl>
```

Read election results data

Election results data is courtesy of FiveThirtyEight and their parent company ABC News, whose Decision Desk called outcomes of races on election night.

This data is used to assess the accuracy of each predictive method.

```
# Midterm election results via ABC and 538
# Used in https://53eig.ht/2PiFb0f
# Published: 2018-12-04 at 17:56
# Archived:  2018-04-04 at 16:08
forecast_results_2018 <-
  str_c(site = "https://raw.githubusercontent.com/",
        fold = "fivethirtyeight/data/master/forecast-review/",
        file = "forecast_results_2018.csv") %>%
  read_memento(timestamp = "2019-04-04", as = "raw") %>%
  read_csv(col_types  = cols(
    Democrat_Won = col_logical(),
    Republican_Won = col_logical(),
    uncalled = col_logical(),
    forecastdate = col_date(format = "%m/%d/%y"),
    category = col_factor(ordered = TRUE,
                          levels = c("Solid D",
                                     "Likely D",
                                     "Lean D",
                                     "Tossup (Tilt D)",
                                     "Tossup (Tilt R)",
                                     "Lean R",
                                     "Likely R",
                                     "Safe R"))))

forecast_results_2018
```

```
## # A tibble: 1,518 x 11
##    cycle branch race  forecastdate version Democrat_WinPro~
##    <dbl> <chr>  <chr> <date>       <chr>              <dbl>
##  1  2018 Gover~ AK-G1 2018-11-06   classic         0.311
##  2  2018 Gover~ AL-G1 2018-11-06   classic         0.0169
##  3  2018 Gover~ AR-G1 2018-11-06   classic         0.000620
##  4  2018 Gover~ AZ-G1 2018-11-06   classic         0.0128
##  5  2018 Gover~ CA-G1 2018-11-06   classic         0.988
##  6  2018 Gover~ CO-G1 2018-11-06   classic         0.950
##  7  2018 Gover~ CT-G1 2018-11-06   classic         0.790
##  8  2018 Gover~ FL-G1 2018-11-06   classic         0.772
##  9  2018 Gover~ GA-G1 2018-11-06   classic         0.322
## 10  2018 Gover~ HI-G1 2018-11-06   classic         0.999
## # ... with 1,508 more rows, and 5 more variables:
## #   Republican_WinProbability <dbl>, category <ord>, Democrat_Won <lgl>,
## #   Republican_Won <lgl>, uncalled <lgl>
```

Format Data for Comparison

Once data is collected from the Internet Archive, each tibble will need to be formatted in a similar style. This will be done using `tidyverse` data manipulation tools.

Ultimately, each tibble will need similar `date` and `race` variables, which together can be used to perform relational joins for comparison. Using all 4 primary data sets, we can create a tibble for each predictive method with all the data needed for comparison.

Format member data

```
members <- legislators_current %>%
  unite(first_name, last_name,
        col = name,
        sep = " ") %>%
  rename(gid     = govtrack_id,
         chamber = type,
         class   = senate_class,
         birth   = birthday) %>%
  select(name, gid, birth, state, district, class, party, gender, chamber) %>%
  arrange(chamber)

members$name     %<>% iconv(to = "ASCII//TRANSLIT")
members$name     %<>% str_replace_all("Robert Menendez", "Bob Menendez")
members$name     %<>% str_replace_all("Robert Casey",    "Bob Casey")
members$name     %<>% str_replace_all("Bernard Sanders", "Bernie Sanders")
members$chamber  %<>% recode("rep" = "house", "sen" = "senate")
members$district %<>% str_pad(width = 2, pad = "0")
members$class    %<>% str_pad(width = 2, pad = "S")
members$party    %<>% recode("Democrat"    = "D",
                             "Independent" = "D",
                             "Republican"  = "R")

members$district <- if_else(condition = is.na(members$district),
                            true = members$class,
                            false = members$district)

# Create district code as relational key
members %<>%
  unite(col = race,
        state, district,
        sep = "-",
        remove = TRUE) %>%
  select(-class) %>%
  arrange(name)

# Format member stats for join
members_stats <-
  bind_rows(sponsorshipanalysis_h, sponsorshipanalysis_s,
            .id = "chamber") %>%
  select(ID, chamber, party, ideology, leadership) %>%
  rename(gid = ID)
members_stats$chamber %<>% recode("1" = "house", "2" = "senate")
members_stats$party %<>% recode("Democrat"    = "D",
```

```
                                "Independent" = "D",
                                "Republican"  = "R")
members_stats$gid %<>% as.character()
# Add stats to frame by GovTrack ID
members %<>% inner_join(members_stats, by = c("gid", "party", "chamber"))

members
```

```
## # A tibble: 534 x 9
##    name     gid   birth      race  party gender chamber ideology leadership
##    <chr>    <chr> <date>     <chr> <chr> <chr>  <chr>      <dbl>      <dbl>
##  1 A. Ferg~ 4127~ 1967-11-15 GA-03 R     M      house      0.672      0.280
##  2 A. McEa~ 4127~ 1961-10-10 VA-04 D     M      house      0.351      0.342
##  3 Adam Ki~ 4124~ 1978-02-27 IL-16 R     M      house      0.724      0.734
##  4 Adam Sc~ 4003~ 1960-06-22 CA-28 D     M      house      0.275      0.529
##  5 Adam Sm~ 4003~ 1965-06-15 WA-09 D     M      house      0.239      0.473
##  6 Adrian ~ 4122~ 1970-12-19 NE-03 R     M      house      0.749      0.627
##  7 Adriano~ 4127~ 1954-09-27 NY-13 D     M      house      0.284      0.351
##  8 Al Green 4006~ 1947-09-01 TX-09 D     M      house      0.258      0.591
##  9 Al Laws~ 4126~ 1948-09-21 FL-05 D     M      house      0.380      0.290
## 10 Alan Lo~ 4125~ 1941-03-08 CA-47 D     M      house      0.199      0.564
## # ... with 524 more rows
```

Format market data

```
markets <- DailyMarketData %>%
  rename(mid      = MarketId,
         name     = MarketName,
         symbol   = MarketSymbol,
         party    = ContractName,
         open     = OpenPrice,
         close    = ClosePrice,
         high     = HighPrice,
         low      = LowPrice,
         volume   = Volume,
         date     = Date) %>%
  select(date, everything()) %>%
  select(-ContractSymbol)

# Get candidate names from full market question
markets$name[str_which(markets$name, "Which party will")] <- NA
markets$name %<>% word(start = 2, end = 3)

# Recode party variables
markets$party %<>% recode("Democratic or DFL" = "D",
                          "Democratic"        = "D",
                          "Republican"        = "R")

# Remove year information from symbol strings
markets$symbol %<>% str_remove(".2018")
markets$symbol %<>% str_remove(".18")

# Divide the market symbol into the name and race code
```

```r
markets %<>%
  separate(col = symbol,
           into = c("symbol", "race"),
           sep = "\\.",
           extra = "drop",
           fill = "left") %>%
  select(-symbol)

# Recode the original contract strings for race variables
markets$race %<>% str_replace("SENATE", "S1")
markets$race %<>% str_replace("SEN",    "S1")
markets$race %<>% str_replace("SE",     "S1")
markets$race %<>% str_replace("AL",     "01")   # at large
markets$race %<>% str_replace("OH12G",  "OH12") # not sure
markets$race %<>% str_replace("MN99",   "MNS2") # special election
markets$race[markets$name == "SPEC"] <- "MSS2"  # special election
markets$race[markets$mid  == "3857"] <- "CAS1"  # market name mustyped
markets$name[markets$name == "PARTY"] <- NA     # no name
markets$name[markets$name == "SPEC"]  <- NA     # no name

markets$race <- paste(str_sub(markets$race, 1, 2), # state abbreviation
                      sep = "-",                   # put hyphen in middle
                      str_sub(markets$race, 3, 4)) # market number)

# Remove markets incorectly repeated
# Some not running for re-election
markets %<>% filter(mid != "3455", # Paul Ryan
                    mid != "3507", # Jeff Flake
                    mid != "3539", # Shea-Porter
                    mid != "3521", # Darrell Issa
                    mid != "3522", # Repeat of 4825
                    mid != "4177", # Repeat of 4232
                    mid != "4824") # Repeat of 4776

# Divide the data based on market question syntax
# Market questions provided name or party, never both
markets_with_name <- markets %>%
  filter(is.na(party)) %>%
  select(-party)

markets_with_party <- markets %>%
  filter(is.na(name)) %>%
  select(-name)

# Join with members key to add party, then back with rest of market
markets <- markets_with_name %>%
  inner_join(members, by = c("name", "race")) %>%
  select(date, mid, race, party, open, low, high, close, volume) %>%
  bind_rows(markets_with_party)

# Add in ME-02 and NY-27 which were left out of initial data
ny_27 <- Contract_NY27 %>%
  rename_all(tolower) %>%
```

```
  slice(6:154) %>%
  mutate(mid = "4729",
         race = "NY-27",
         party = "R") %>%
  select(-average)

me_02 <- Market_ME02 %>%
  rename_all(tolower) %>%
  rename(party = longname) %>%
  filter(date != "2018-10-10") %>%
  mutate(mid = "4945",
         race = "ME-02")

markets_extra <-
  bind_rows(ny_27, me_02) %>%
  select(date, mid, race, party, open, low, high, close, volume)

markets_extra$party[str_which(markets_extra$party, "GOP")] <- "R"
markets_extra$party[str_which(markets_extra$party, "Dem")] <- "D"

# Bind with ME-02 and NY-27
markets %<>%  bind_rows(markets_extra)

markets
```

```
## # A tibble: 41,933 x 9
##    date       mid   race  party  open   low  high close volume
##    <date>     <chr> <chr> <chr> <dbl> <dbl> <dbl> <dbl>  <dbl>
##  1 2017-01-27 2918  MA-S1 D      0.79  0.71  0.86  0.74   1102
##  2 2017-01-28 2918  MA-S1 D      0.74  0.74  0.78  0.78   1010
##  3 2017-01-29 2918  MA-S1 D      0.78  0.76  0.78  0.77    581
##  4 2017-01-30 2918  MA-S1 D      0.77  0.76  0.78  0.78    631
##  5 2017-01-31 2918  MA-S1 D      0.78  0.77  0.81  0.81   1378
##  6 2017-02-01 2918  MA-S1 D      0.81  0.79  0.82  0.8     768
##  7 2017-02-02 2918  MA-S1 D      0.8   0.79  0.8   0.79     50
##  8 2017-02-03 2918  MA-S1 D      0.79  0.78  0.8   0.78    592
##  9 2017-02-04 2918  MA-S1 D      0.78  0.78  0.79  0.79     10
## 10 2017-02-05 2918  MA-S1 D      0.79  0.79  0.8   0.8       6
## # ... with 41,923 more rows
```

Format model data

```
# Format district for race variable
model_district <- house_district_forecast %>%
  mutate(district = str_pad(string = district,
                            width = 2,
                            side = "left",
                            pad = "0"))

# Format class for race variable
model_seat <- senate_seat_forecast %>%
  rename(district = class) %>%
  mutate(district = str_pad(string = district,
```

```
                            width = 2,
                            side = "left",
                            pad = "S"))

model_combined <-
  bind_rows(model_district, model_seat, .id = "chamber") %>%
  # Create race variable for relational join
  unite(col = race,
        state, district,
        sep = "-",
        remove = TRUE) %>%
  rename(name = candidate,
         date = forecastdate,
         prob = win_probability,
         min_share = p10_voteshare,
         max_share = p90_voteshare) %>%
  filter(name != "Others") %>%
  select(date, race, name, party, chamber, everything()) %>%
  arrange(date, name)

# Recode identifying variable for clarification
model_combined$chamber %<>% recode("1" = "house",
                                   "2" = "senate")

# Only special elections are for senate.
model_combined$special[is.na(model_combined$special)] <- FALSE

# Convert percent vote share values to decimal
model_combined[, 10:12] <- model_combined[, 10:12] * 0.01

# Recode incumbent Independent senators for relational joins with Markets
# Both caucus with Democrats and were endoresed by Democratic party
model_combined$party[model_combined$name == "Bernard Sanders"]   <- "D"
model_combined$party[model_combined$name == "Angus S. King Jr."] <- "D"
model_combined %<>% filter(name != "Zak Ringelstein")

# Seperate model data by model format
# According to 538, the "classic" model can be used as a default
model <- model_combined %>%
  filter(model == "classic") %>%
  select(-model)

model_lite <- model_combined %>%
  filter(model == "lite") %>%
  select(-model)

model_deluxe <- model_combined %>%
  filter(model == "deluxe") %>%
  select(-model)

model
```

```
## # A tibble: 101,543 x 11
##    date       race  name  party chamber special incumbent     prob
```

```
##    <date>     <chr> <chr> <chr> <chr>   <lgl>   <lgl>        <dbl>
##  1 2018-08-01 VA-04 A. D~ D     house   FALSE   TRUE       0.999
##  2 2018-08-01 GA-03 A. D~ R     house   FALSE   TRUE       1.000
##  3 2018-08-01 LA-03 Aaro~ LIB   house   FALSE   FALSE      0.000300
##  4 2018-08-01 ID-02 Aaro~ D     house   FALSE   FALSE      0.0036
##  5 2018-08-01 IA-01 Abby~ D     house   FALSE   FALSE      0.880
##  6 2018-08-01 VA-07 Abig~ D     house   FALSE   FALSE      0.334
##  7 2018-08-01 IL-16 Adam~ R     house   FALSE   TRUE       0.977
##  8 2018-08-01 CA-28 Adam~ D     house   FALSE   TRUE       1
##  9 2018-08-01 WA-09 Adam~ D     house   FALSE   TRUE       0.983
## 10 2018-08-01 NE-03 Adri~ R     house   FALSE   TRUE       1
## # ... with 101,533 more rows, and 3 more variables: voteshare <dbl>,
## #   min_share <dbl>, max_share <dbl>
```

Format election results

```
results <- forecast_results_2018 %>%
  filter(branch  != "Governor",
         version == "classic") %>%
  separate(col    = race,
           into   = c("state", "district"),
           sep    = "-") %>%
  rename(winner   = Democrat_Won) %>%
  mutate(district = str_pad(district, width = 2,  pad   = "0")) %>%
  unite(state, district,
        col = race,
        sep = "-") %>%
  select(race, winner) %>%
  filter(race != "NC-09") # Harris fraud charges

results
```

```
## # A tibble: 469 x 2
##    race  winner
##    <chr> <lgl>
##  1 AK-01 FALSE
##  2 AL-01 FALSE
##  3 AL-02 FALSE
##  4 AL-03 FALSE
##  5 AL-04 FALSE
##  6 AL-05 FALSE
##  7 AL-06 FALSE
##  8 AL-07 TRUE
##  9 AR-01 FALSE
## 10 AR-02 FALSE
## # ... with 459 more rows
```

Compare Predictive Methods

Once each data frame has been properly formatted, they can be filtered to remove redundant predictions. Each row in both sets will contain the day's probability of a Democratic party candidate winning.

```
# Take the complimentary probability if only GOP data
# Find race codes for markets with data on only one candidate
```

```r
single_party_markets <- markets %>%
  group_by(date, race) %>%
  summarise(n = n()) %>%
  filter(n == 1) %>%
  ungroup() %>%
  pull(race) %>%
  unique()

# Invert the GOP prices for markets with only GOP candidates
invert <- function(x) 1 - x

invert_gop <- markets %>%
  filter(race %in% single_party_markets,
         party == "R") %>%
  mutate(close = invert(close),
         party = "D")

# Take all but the only GOP markets
original_dem <- markets %>%
  filter(!race %in% invert_gop$race,
         party == "D")

# Combined both back together
markets2 <-
  bind_rows(original_dem, invert_gop) %>%
  select(date, race, close) %>%
  arrange(date, race)

# Create model data with only dem party info
model2 <- model %>%
  group_by(date, race, party) %>%
  summarise(prob = sum(prob)) %>%
  ungroup() %>%
  filter(party == "D") %>%
  select(-party)

# Join democratic predictions from both markets and models for comparison
# Keep market and model data in seperate columns
messy <-
  inner_join(markets2, model2,
             by  = c("date", "race")) %>%
  filter(date  >= "2018-08-01",
         date  <= "2018-11-05") %>%
  rename(model  = prob,
         market = close)

messy
```

```
## # A tibble: 8,847 x 4
##    date       race  market  model
##    <date>     <chr>  <dbl>  <dbl>
##  1 2018-08-01 AZ-S1   0.66 0.738
##  2 2018-08-01 CA-12   0.91 1
##  3 2018-08-01 CA-22   0.3  0.0493
```

```
##  4 2018-08-01 CA-25   0.61 0.745
##  5 2018-08-01 CA-39   0.61 0.377
##  6 2018-08-01 CA-48   0.72 0.666
##  7 2018-08-01 CA-49   0.74 0.795
##  8 2018-08-01 CA-S1   0.94 1
##  9 2018-08-01 CO-05   0.06 0.0273
## 10 2018-08-01 CO-06   0.58 0.648
## # ... with 8,837 more rows
```

```
# Make the data tidy with each prediction as an observation
tidy <- messy %>%
  gather(model, market,
         key = method,
         value = prob) %>%
  arrange(date, race, method)

tidy
```

```
## # A tibble: 17,694 x 4
##    date       race  method   prob
##    <date>     <chr> <chr>   <dbl>
##  1 2018-08-01 AZ-S1 market 0.66
##  2 2018-08-01 AZ-S1 model  0.738
##  3 2018-08-01 CA-12 market 0.91
##  4 2018-08-01 CA-12 model  1
##  5 2018-08-01 CA-22 market 0.3
##  6 2018-08-01 CA-22 model  0.0493
##  7 2018-08-01 CA-25 market 0.61
##  8 2018-08-01 CA-25 model  0.745
##  9 2018-08-01 CA-39 market 0.61
## 10 2018-08-01 CA-39 model  0.377
## # ... with 17,684 more rows
```

```
# Add in results to determine binary hits/misses
hits <- tidy %>%
  mutate(pred = prob > 0.5) %>%
  inner_join(results, by = "race") %>%
  mutate(hit = pred == winner) %>%
  select(date, race, method, prob, pred, winner, hit)

hits
```

```
## # A tibble: 17,500 x 7
##    date       race  method   prob pred  winner hit
##    <date>     <chr> <chr>   <dbl> <lgl> <lgl>  <lgl>
##  1 2018-08-01 AZ-S1 market 0.66   TRUE  TRUE   TRUE
##  2 2018-08-01 AZ-S1 model  0.738  TRUE  TRUE   TRUE
##  3 2018-08-01 CA-12 market 0.91   TRUE  TRUE   TRUE
##  4 2018-08-01 CA-12 model  1      TRUE  TRUE   TRUE
##  5 2018-08-01 CA-22 market 0.3    FALSE FALSE  TRUE
##  6 2018-08-01 CA-22 model  0.0493 FALSE FALSE  TRUE
##  7 2018-08-01 CA-25 market 0.61   TRUE  TRUE   TRUE
##  8 2018-08-01 CA-25 model  0.745  TRUE  TRUE   TRUE
##  9 2018-08-01 CA-39 market 0.61   TRUE  TRUE   TRUE
## 10 2018-08-01 CA-39 model  0.377  FALSE TRUE   FALSE
## # ... with 17,490 more rows
```

```r
# Run a welch two sample t-test?
hits %$%
  t.test(formula = hit ~ method,
         alternative = "greater")
```

```
##
##  Welch Two Sample t-test
##
## data:  hit by method
## t = 4.1209, df = 17433, p-value = 1.895e-05
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.01338999        Inf
## sample estimates:
## mean in group market  mean in group model
##            0.8603429            0.8380571
```

```r
# Run a 2-sample test for equality of proportions?
hits %>%
  select(date, race, method, hit) %>%
  spread(key = method,
         value = hit) %>%
  select(market, model) %>%
  colSums() %>%
  prop.test(n = nrow(hits)/2 %>% rep(2))
```

```
##
##  2-sample test for equality of proportions with continuity
##  correction
##
## data:  . out of nrow(hits)/2 %>% rep(2)
## X-squared = 16.794, df = 1, p-value = 4.166e-05
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.01157269 0.03299874
## sample estimates:
##    prop 1    prop 2
## 0.8603429 0.8380571
```

```r
hits %>%
  group_by(pred, winner, method) %>%
  summarise(prob = mean(prob),
            n = n()) %>%
  arrange(pred, winner)
```

```
## # A tibble: 8 x 5
## # Groups:   pred, winner [4]
##   pred  winner method  prob     n
##   <lgl> <lgl>  <chr>  <dbl> <int>
## 1 FALSE FALSE  market 0.230  3003
## 2 FALSE FALSE  model  0.168  2808
## 3 FALSE TRUE   market 0.406   847
## 4 FALSE TRUE   model  0.365   847
## 5 TRUE  FALSE  market 0.593   375
## 6 TRUE  FALSE  model  0.637   570
```

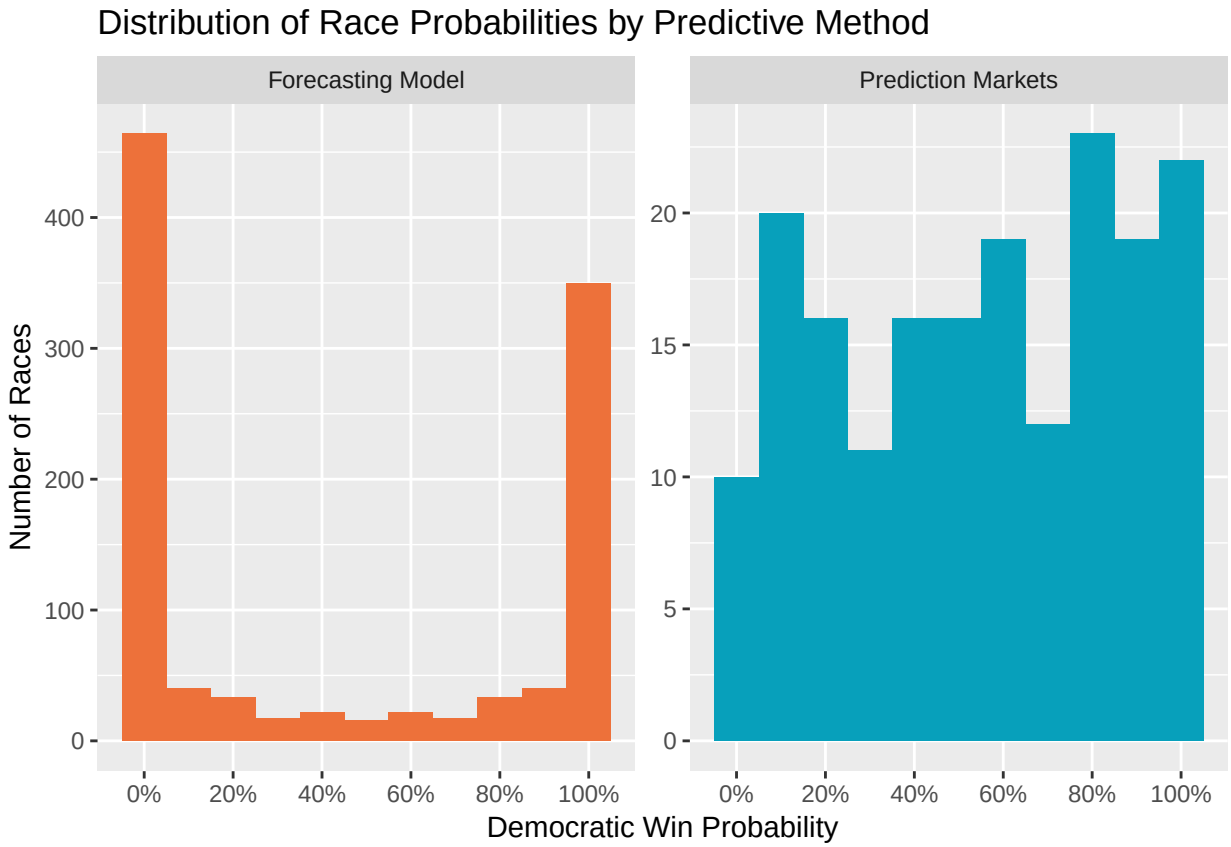
```
## 7 TRUE  TRUE   market 0.795  4525
## 8 TRUE  TRUE   model  0.845  4525
```

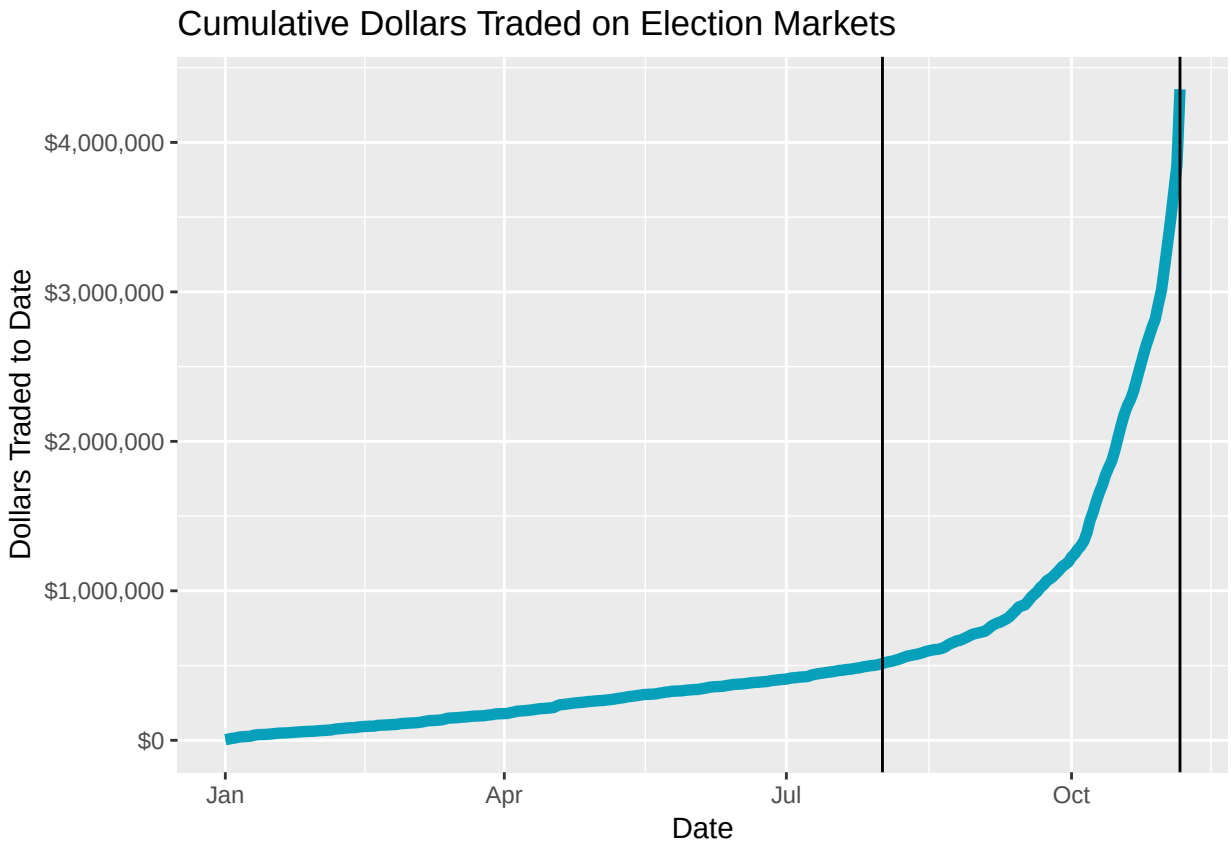
```
hits %>%
  mutate(brier_score = (winner - prob)^2) %$%
  t.test(formula = brier_score ~ method)
```

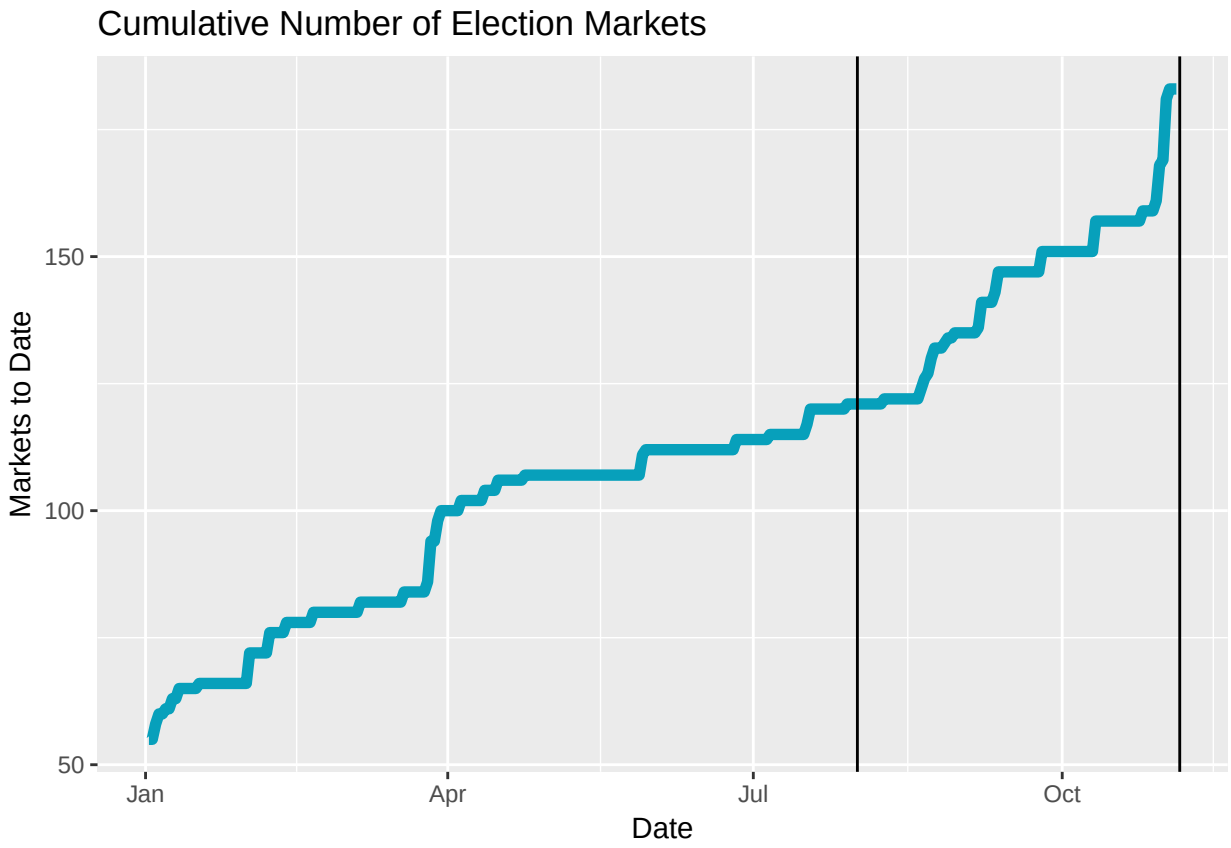
```
##
##  Welch Two Sample t-test
##
## data:  brier_score by method
## t = -0.33902, df = 16943, p-value = 0.7346
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.005016567  0.003537138
## sample estimates:
## mean in group market  mean in group model
##            0.1083634            0.1091031
```

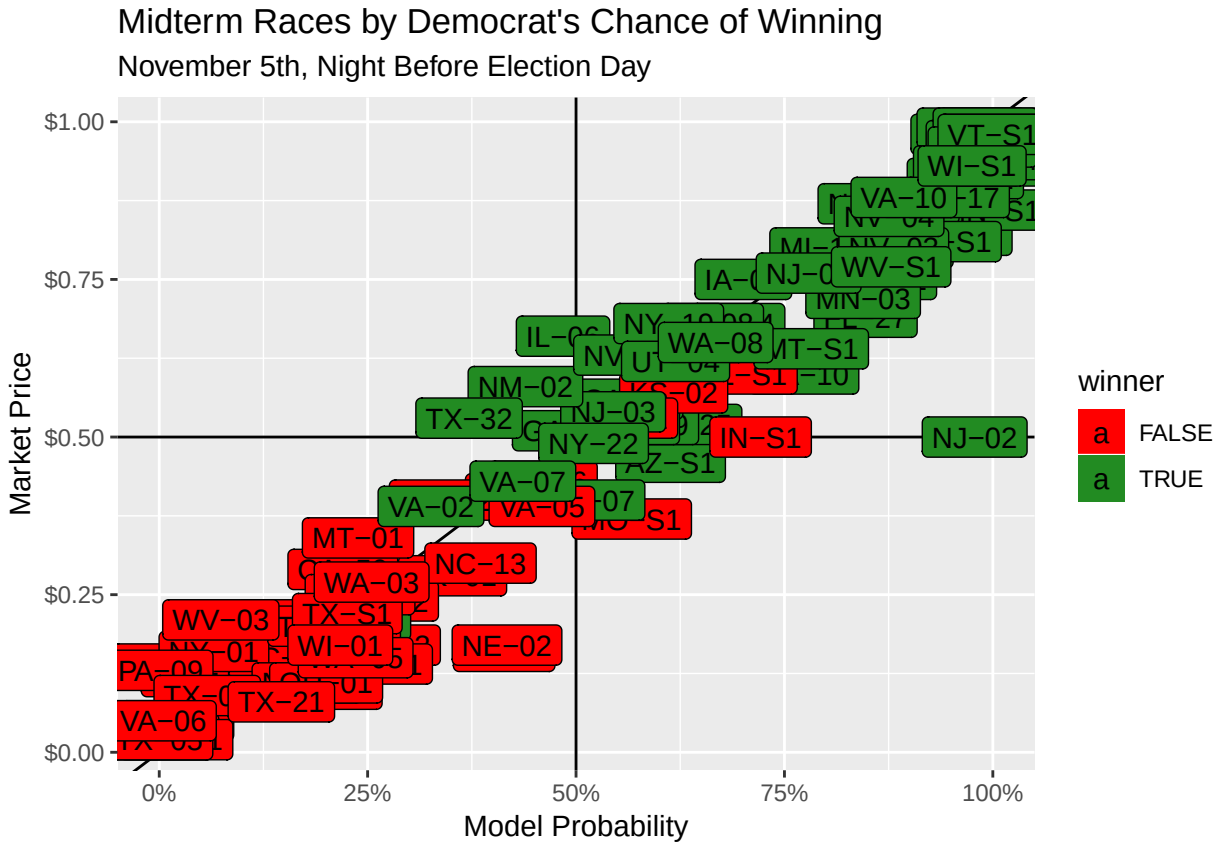
```
hits_model  <- hits %>% filter(method == "model")
hits_market <- hits %>% filter(method == "market")

brier_model <- verification::brier(
  obs = hits_model$winner,
  pred = hits_model$prob,
  baseline = rep(0.5, nrow(hits_model)),
  bins = TRUE)

brier_market <- verification::brier(
  obs = hits_market$winner,
  pred = hits_market$prob,
  baseline = rep(0.5, nrow(hits_market)),
  bins = TRUE)
```

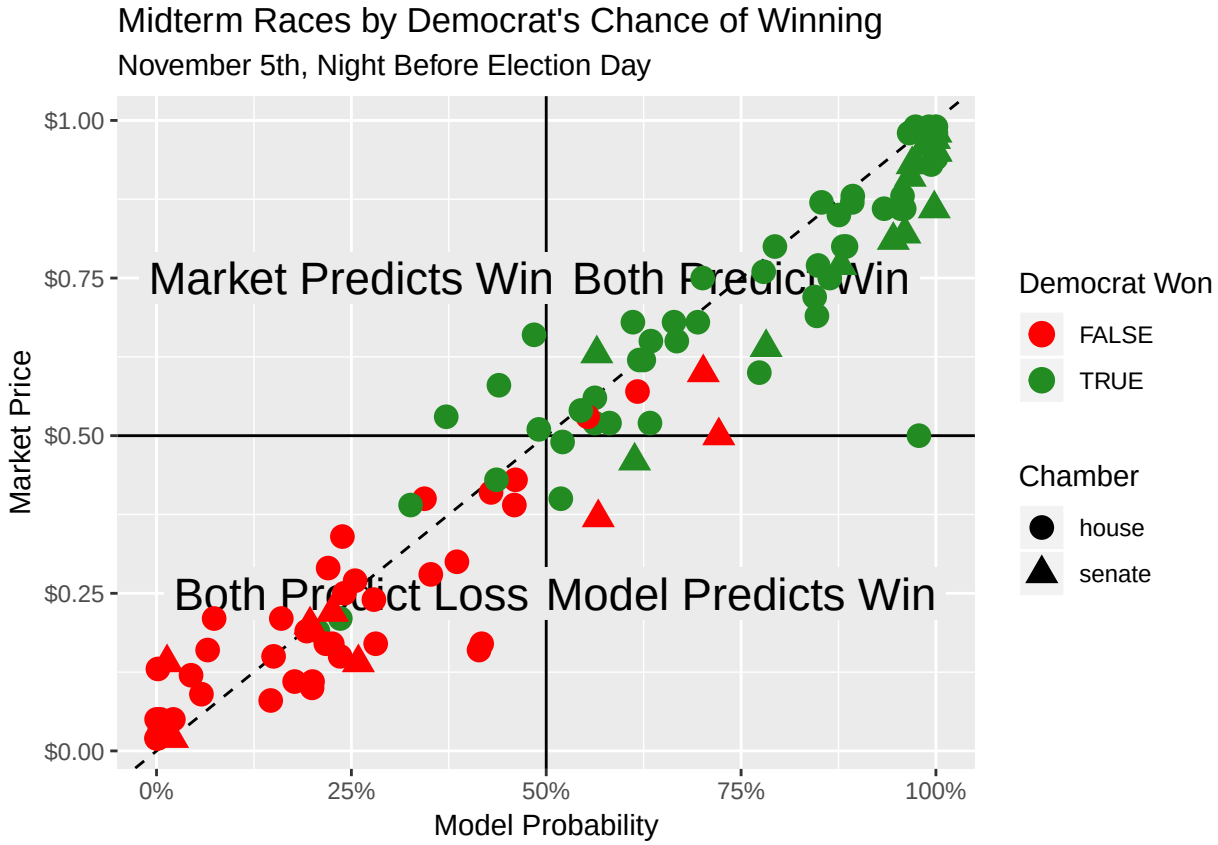
Explore Data Visually

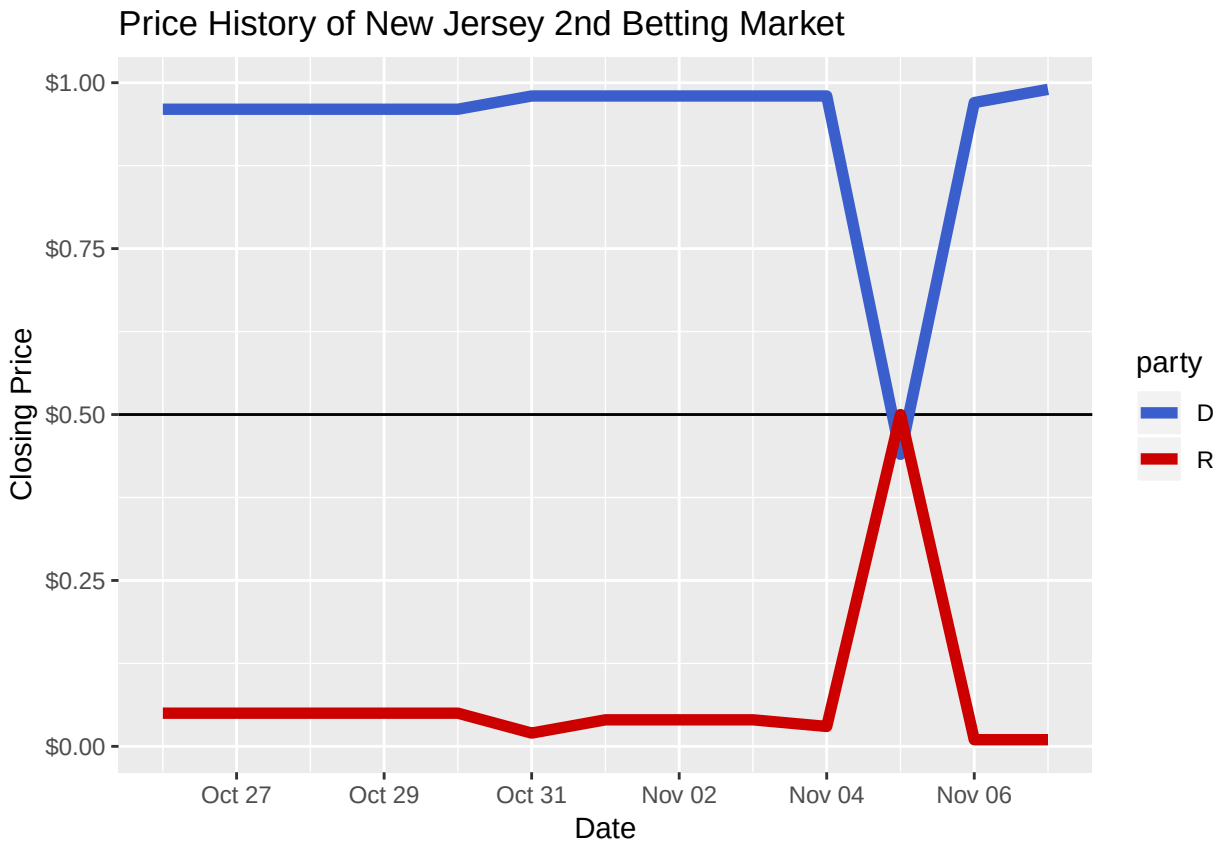


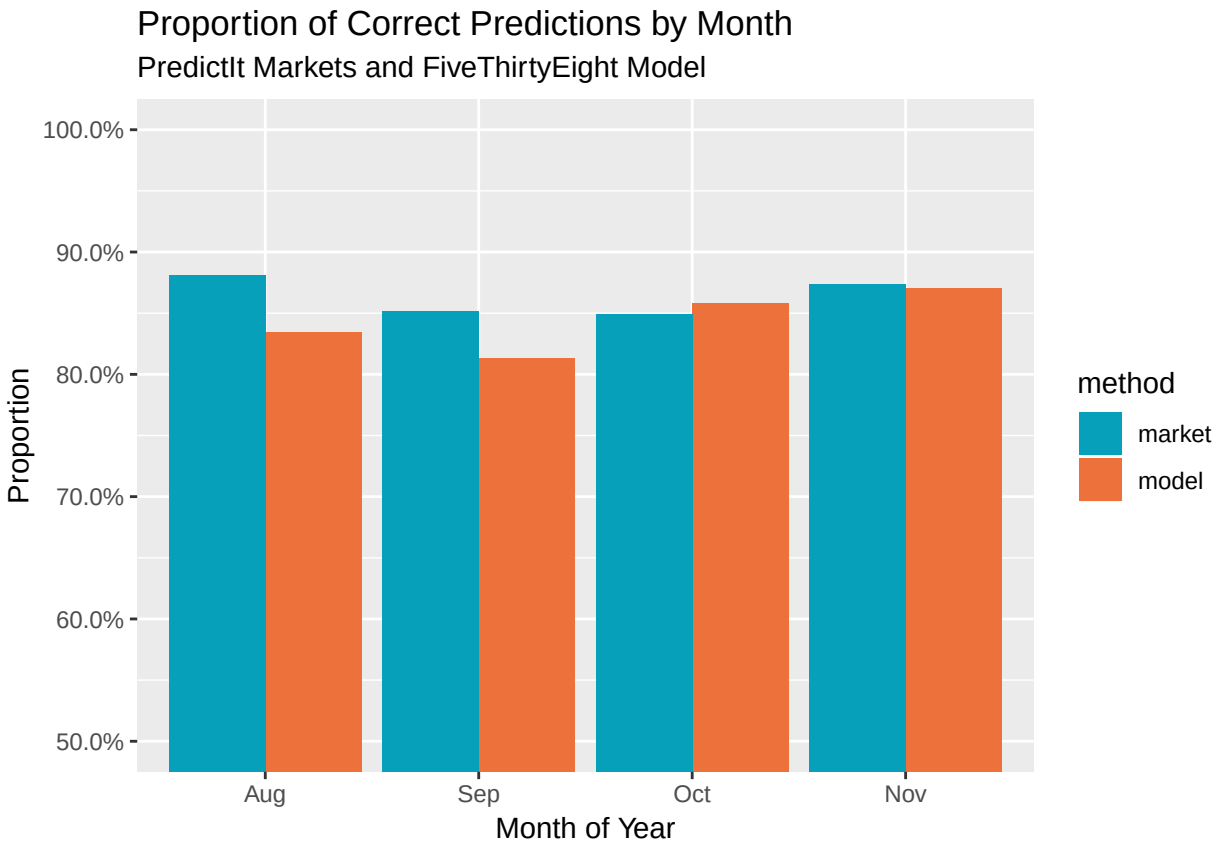

Cumulative Dollars Traded on Election Markets
$4,000,000
$3,000,000
$2,000,000
$1,000,000
$0
Dollars Traded to Date
Jan
Apr
Jul
Oct
Date

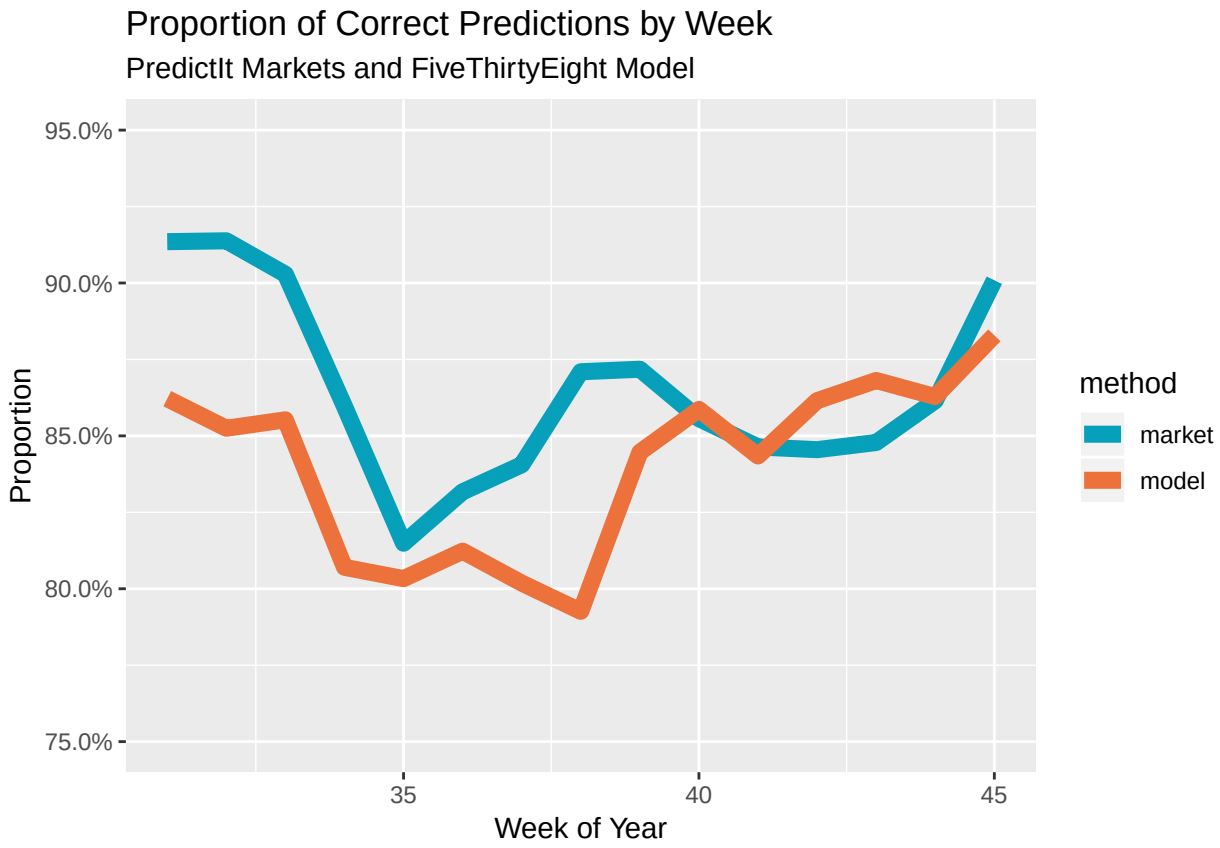

Cumulative Number of Election Markets
150
100
50
Markets to Date
Jan
Apr
Jul
Oct
Date

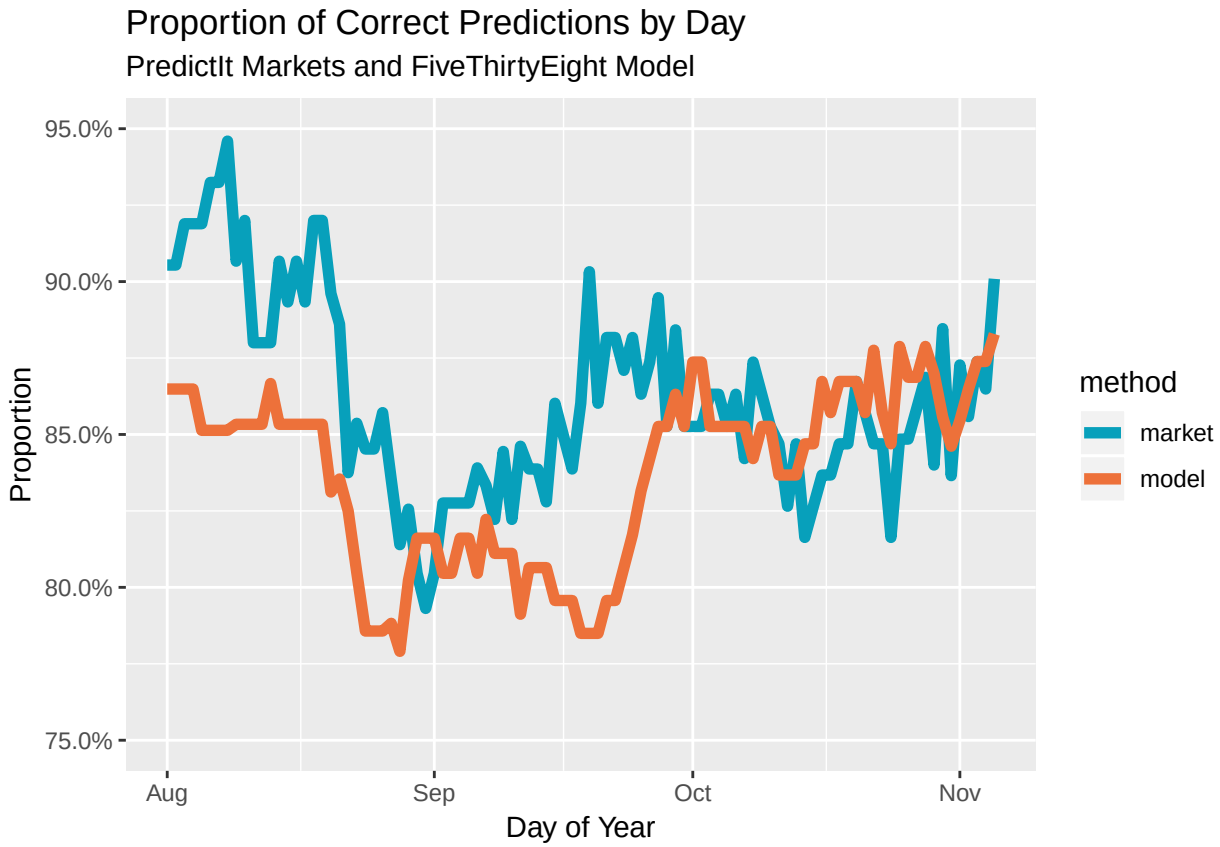

Midterm Races by Democrat's Chance of Winning
November 5th, Night Before Election Day
Market Price
$1.00
$0.75
$0.50
$0.25
$0.00
0%
25%
50%
75%
100%
Model Probability
VT−S1
WI−S1
VA−10
WV−S1
MN−03
WA−08
MT−S1
NM−02
TX−32
NJ−03
NY−22
IN−S1
NJ−02
VA−07
VA−02
VA−05
MT−01
NC−13
WA−03
WV−03
TX−S1
WI−01
NE−02
TX−21
VA−06
winner
a
FALSE
a
TRUE

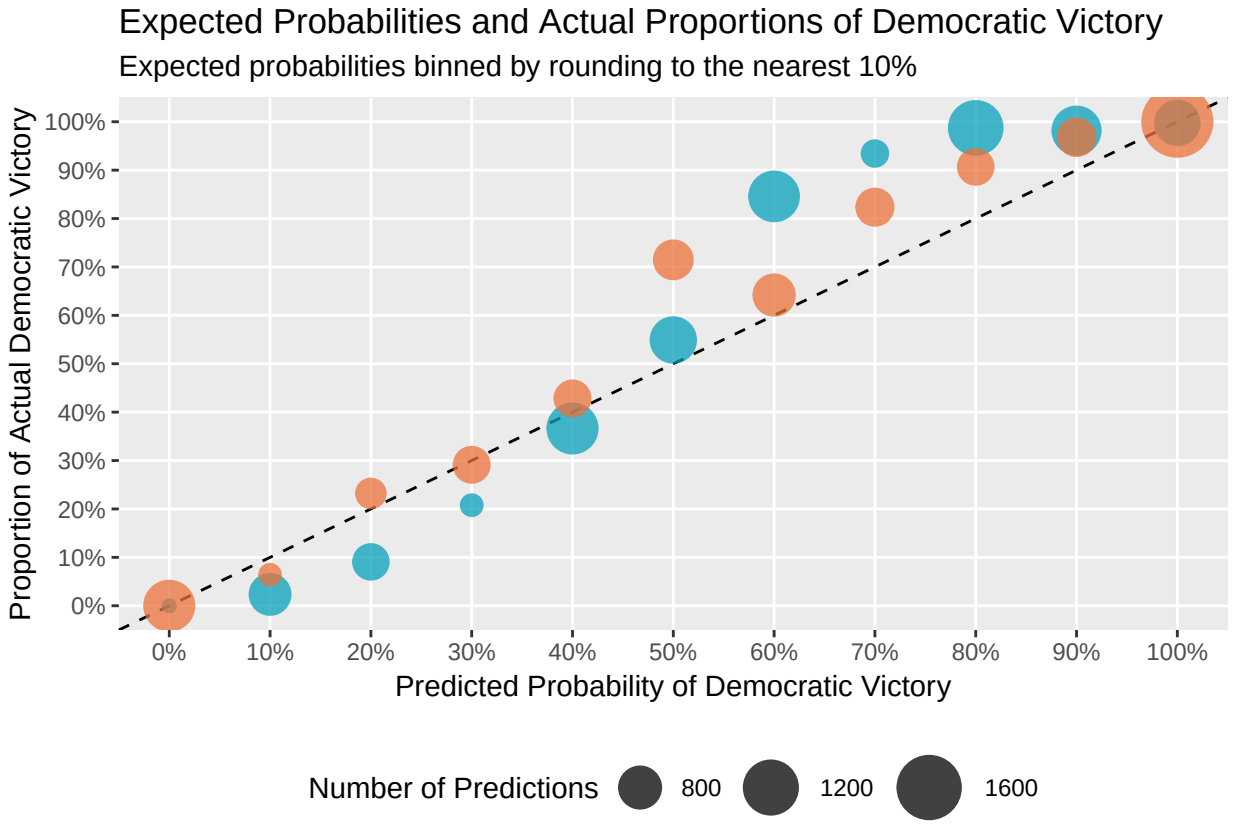

Midterm Races by Democrat's Chance of Winning
November 5th, Night Before Election Day
Market Price
$1.00
$0.75
$0.50
$0.25
$0.00
0%
25%
50%
75%
100%
Model Probability
Market Predicts Win
Both Predict Win
Both Predict Loss
Model Predicts Win
Democrat Won
FALSE
TRUE
Chamber
house
senate


Price History of New Jersey 2nd Betting Market
$1.00
$0.75
$0.50
$0.25
$0.00
Closing Price
Oct 27
Oct 29
Oct 31
Nov 02
Nov 04
Nov 06
Date
party
D
R


Proportion of Correct Predictions by Month
PredictIt Markets and FiveThirtyEight Model
100.0%
90.0%
80.0%
70.0%
60.0%
50.0%
Proportion
Aug
Sep
Oct
Nov
Month of Year
method
market
model


Proportion of Correct Predictions by Week
PredictIt Markets and FiveThirtyEight Model
95.0%
90.0%
85.0%
80.0%
75.0%
Proportion
35
40
45
Week of Year
method
market
model


Proportion of Correct Predictions by Day
PredictIt Markets and FiveThirtyEight Model
95.0%
90.0%
85.0%
80.0%
75.0%
Proportion
Aug
Sep
Oct
Nov
Day of Year
method
market
model


Expected Probabilities and Actual Proportions of Democratic Victory
Expected probabilities binned by rounding to the nearest 10%
Proportion of Actual Democratic Victory
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Predicted Probability of Democratic Victory
Number of Predictions
800
1200
1600