Markets and Models

Comparing 2018 Midterm Predictions

Kiernan Nicholls

May 07, 2019

Contents

Overview

The forecast model has become a staple of political punditry in recent years. Popularized by the data journalism site FiveThirtyEight, the forecasting model is a statistical tool used to incorporate a number of quantitative inputs to generate a *probabilistic* view of all possible outcomes.

Prediction markets can be used as alternative method of generating similarly probabilistic views of election outcomes. Markets utilize the economic forces of price discovery and risk aversion to overcome the ideological bias of self-interested traders on a binary options exchange.

Can markets help us predict elections better than the models? If so, under what conditions?

I propose a null hypothesis of no difference between the proportion of accurate predictions made by forecasting models and prediction markets in the 2018 congressional midterm elections.

Reproduce

All data used in this project is freely available for academic research. An archived version of all public information has been created on the free Internet Archive.

Additionally, the source code for this summary, a more detailed academic manuscript, and all analysis is hosted on a public GitHub repository, which can be cloned to reproduce findings exactly. All software needed to produce the same results is free and open source.

Packages

All data sets is collected, formatted, combined, and analyzed using the statistical computing language R (R Core Team 2018a) and a handful of specialized packages, mostly from the tidyverse ecosystem.

```r
library(devtools) # for package managment
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
```

```r
library(readr)     # reading data
library(dplyr)     # wrangling data
library(tidyr)     # tidying data
library(knitr)     # pritning tables
library(tibble)    # rectangle data
```

```r
library(pander)    # printing tests
library(stringr)   # character strings
library(wayback)   # reading archives
library(ggplot2)   # plotting data
library(magrittr)  # piping data
library(lubridate) # dates strings
```

Read Data

Reading data is handles by the `readr` (Wickham, Hester, and Francois 2018) and `wayback` (Rudis 2017) packages. Data public on the Internet has been archived for posterity and stored as "mementos" accessible by the Wayback Machine. Data is read as a raw file and formatted by `read_delim()`.

Read Markets

Market data is courtesy of PredictIt.org, an exchange owned and operated by the Victoria University of Wellington, New Zealand. The past 90 days of all market data can be scraped, but more in depth data is provided to partnered academic researchers. The data for this project can be found in the `/data` directory of the GitHub repository.

```r
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")))
```

Table 1: Input Market Data (Sample)

MarketSymbol	ContractSymbol	Date	OpenPrice	ClosePrice	Volume
SANDERS.VTSENATE.2018	NA	2017-09-10	0.90	0.90	0
WA08.2018	DEM.WA08.2018	2018-01-28	0.79	0.79	0
MS03.2018	DEM.MS03.2018	2018-02-07	0.15	0.15	0
DENH.CA10.2018	NA	2018-02-22	0.27	0.30	5
ISSA.CA49.2018	NA	2018-03-14	0.01	0.01	0
TX29.2018	GOP.TX29.2018	2018-03-21	0.05	0.05	0
PA09.2018	GOP.PA09.2018	2018-06-15	0.86	0.86	0
TEST.MTSENATE.2018	NA	2018-06-26	0.62	0.65	705
KNIG.CA25.2018	NA	2018-08-13	0.35	0.35	0

MarketSymbol	ContractSymbol	Date	OpenPrice	ClosePrice	Volume
SHEA.NH01.2018	NA	2018-09-02	0.01	0.01	0

Read Members

Market data lacks all the data needed to join it with model data, namely party association for members of the Senate. This data can be found in the (the \ **???** project's)05 legislators data set (Project 2018).

```
## 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()))
```

Table 2: Input Member Data (Sample)

last_name	birthday	gender	type	state	district	party
Amata	1947-12-29	F	rep	AS	0	Republican
Crapo	1951-05-20	M	sen	ID	NA	Republican
Donnelly	1955-09-28	M	sen	IN	NA	Democrat
Gallego	1979-11-20	M	rep	AZ	7	Democrat
Lee	1971-06-04	M	sen	UT	NA	Republican
Massie	1971-01-13	M	rep	KY	4	Republican
Royce	1951-10-12	M	rep	CA	39	Republican
Stivers	1965-03-24	M	rep	OH	15	Republican
Thompson	1959-07-27	M	rep	PA	5	Republican
Wasserman Schultz	1966-09-27	F	rep	FL	23	Democrat

Read Models

While the FiveThirtyEight model proprietary, they release top level output data free to the public as two separate files, one for the House (FiveThirtyEight 2018a) and one for the Senate (FiveThirtyEight 2018b).

```
## District level 538 House model history
## Updated:  2018-11-06 at 01:56
## Archived: 2018-11-06 at 12:06
house_district_forecast <-
```

```r
  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()
```

Table 3: Input House Model Data (Sample)

forecastdate	state	district	party	incumbent	win_probability	voteshare
2018-08-14	NY	6	D	TRUE	1.000	84.02
2018-08-17	NY	14	WOF	TRUE	0.000	3.72
2018-08-18	ME	1	IND	FALSE	0.000	3.44
2018-09-19	TX	5	R	FALSE	0.981	61.92
2018-09-25	MD	5	D	TRUE	1.000	73.96
2018-10-04	KS	4	D	FALSE	0.007	40.96
2018-10-09	NM	1	D	FALSE	0.896	52.73
2018-10-13	OH	16	D	FALSE	0.041	43.01
2018-10-29	CA	15	R	FALSE	0.000	22.68
2018-11-05	NC	12	D	TRUE	1.000	72.74

Read Results

Results come courtesy of FiveThirtyEight and the Decision Desk at their parent company, ABC News. Used in the article *How FiveThirtyEight's 2018 Midterm Forecasts Did* (FiveThirtyEight 2018c)

```r
# 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"))))
```

Table 4: Input Results Data (Sample)

branch	race	version	Democrat_WinProbability	category	Democrat_Won
House	CA-29	classic	1.000	Solid D	TRUE
Senate	CA-S1	classic	1.000	Solid D	TRUE
House	LA-4	lite	0.019	Safe R	FALSE
House	MA-6	classic	1.000	Solid D	TRUE
House	ME-1	deluxe	1.000	Solid D	TRUE
House	NJ-4	deluxe	0.025	Safe R	FALSE
House	NY-14	lite	1.000	Solid D	TRUE
House	OR-1	lite	1.000	Solid D	TRUE
House	PA-13	classic	0.000	Safe R	FALSE
House	TX-15	classic	1.000	Solid D	TRUE

Format Data

The objective of formatting is to create the neccesary variables needed to perform the relational join for method comparison.

Formatting is done using `dplyr` [dplyr] and `tidyr` (Wickham and Henry 2019). Character values are formatted with `stringr` (Wickham 2019) and `lubridate` (Grolemund and Wickham 2011) for dates.

The `race` variable comes from the state abbreviation and race number (e.g., VT-01, AZ-S1, MO-S2).

Together with `date`, these two variables can be used to match daily predictions together for comparison.

We will also need a `party` variable to filter out redundant observations.

Format Members

```
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")
```

```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"))
```

Table 5: Formatted Member Data (Sample)

name	gid	birth	race	party	gender	chamber	ideology	leadership
Bill Johnson	412460	1954-11-10	OH-06	R	M	house	0.885	0.489
Cheri Bustos	412537	1961-10-17	IL-17	D	F	house	0.419	0.503
David Schweikert	412399	1962-03-03	AZ-06	R	M	house	0.856	0.471
Joseph Kennedy	412543	1980-10-04	MA-04	D	M	house	0.326	0.656
Juan Vargas	412522	1961-03-07	CA-51	D	M	house	0.293	0.357
Kevin Yoder	412430	1976-01-08	KS-03	R	M	house	0.783	0.707
Mitch McConnell	300072	1942-02-20	KY-S2	R	M	senate	0.795	0.926
Scott Taylor	412727	1979-06-27	VA-02	R	M	house	0.561	0.380
Susan Collins	300025	1952-12-07	ME-S2	R	F	senate	0.445	0.753
Tim Scott	412471	1965-09-19	SC-S3	R	M	senate	0.869	0.500

Format Markets

For market data, `race` comes from the the MarketID, which other contains the candidate name of code itself.

```r
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
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
```

```r
# 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)
```

Table 6: Formatted Market Data (Sample)

date	mid	race	party	open	low	high	close	volume
2018-01-30	2998	IN-S1	D	0.60	0.60	0.60	0.60	0
2018-02-17	3485	WI-S1	D	0.74	0.74	0.74	0.74	4
2018-02-25	4015	MD-06	R	0.09	0.09	0.09	0.09	0
2018-05-26	4156	FL-17	D	0.18	0.18	0.18	0.18	0
2018-08-23	4255	MN-03	D	0.63	0.63	0.63	0.63	0
2018-09-06	3886	VA-02	R	0.60	0.60	0.60	0.60	100
2018-09-08	3886	VA-02	R	0.66	0.58	0.66	0.58	25
2018-09-23	3738	FL-27	D	0.78	0.67	0.78	0.67	64

date	mid	race	party	open	low	high	close	volume
2018-10-02	4886	KY-06	D	0.46	0.30	0.48	0.48	71
2018-10-09	4843	AZ-02	R	0.12	0.10	0.17	0.10	216

Format Models

```
# 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")
```

```r
# 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)
```

Table 7: Formatted Model Data (Sample)

date	race	party	chamber	special	incumbent	prob	voteshare
2018-08-08	AL-05	R	house	FALSE	TRUE	0.998	0.637
2018-08-20	AZ-04	R	house	FALSE	TRUE	0.999	0.647
2018-09-01	TN-S1	D	senate	FALSE	FALSE	0.298	0.460
2018-09-18	SC-04	R	house	FALSE	FALSE	1.000	0.651
2018-09-28	NY-05	D	house	FALSE	TRUE	1.000	1.000
2018-09-29	WV-02	R	house	FALSE	TRUE	0.946	0.541
2018-10-20	MO-02	D	house	FALSE	FALSE	0.143	0.449
2018-10-26	TN-02	R	house	FALSE	FALSE	1.000	0.641
2018-11-01	GA-09	D	house	FALSE	FALSE	0.000	0.204
2018-11-03	MS-04	REF	house	FALSE	FALSE	0.000	0.025

Format Results

```r
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
```

Combine Data

Not all races contain predictions for each party's probability. I will be using only Democratic (or Independent) data. For the races with only Republican predictions, the probability can simply be inverted.

```r
# Take the complimentary probability if only GOP data
# Find race codes for markets with data on only one candidate
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_gop <- markets %>%
```

```
  filter(race %in% single_party_markets,
         party == "R") %>%
  mutate(close = 1 - 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)
```

Table 8: Messy Joined Data (Head)

date	race	market	model
2018-08-01	AZ-S1	0.66	0.738
2018-08-01	CA-12	0.91	1.000
2018-08-01	CA-22	0.30	0.049
2018-08-01	CA-25	0.61	0.745
2018-08-01	CA-39	0.61	0.377
2018-08-01	CA-48	0.72	0.666
2018-08-01	CA-49	0.74	0.795
2018-08-01	CA-S1	0.94	1.000
2018-08-01	CO-05	0.06	0.027
2018-08-01	CO-06	0.58	0.648

```
# Make tidy, with a row for each prediction
# Add in results to determine binary hits/misses
hits <- messy %>%
  gather(model, market,
         key = method,
```

```
        value = prob) %>%
  mutate(pred = prob > 0.5) %>%
  inner_join(results, by = "race") %>%
  mutate(hit = pred == winner) %>%
  select(date, race, method, prob, pred, winner, hit)
```

Table 9: Tidy Joined Data (Head)

date	race	method	prob	pred	winner	hit
2018-08-01	AZ-S1	model	0.738	TRUE	TRUE	TRUE
2018-08-01	CA-12	model	1.000	TRUE	TRUE	TRUE
2018-08-01	CA-22	model	0.049	FALSE	FALSE	TRUE
2018-08-01	CA-25	model	0.745	TRUE	TRUE	TRUE
2018-08-01	CA-39	model	0.377	FALSE	TRUE	FALSE
2018-08-01	CA-48	model	0.666	TRUE	TRUE	TRUE
2018-08-01	CA-49	model	0.795	TRUE	TRUE	TRUE
2018-08-01	CA-S1	model	1.000	TRUE	TRUE	TRUE
2018-08-01	CO-05	model	0.027	FALSE	FALSE	TRUE
2018-08-01	CO-06	model	0.648	TRUE	TRUE	TRUE

Results

Once predictions are combined, tidy-ed, and compared with election results, the predictive ability of each method can be assessed.

A test for equal proportion using the `stats` package shows a statistically significant difference (Table 10) (R Core Team 2018b).

A test of forecast skill using the `verification` package shows no statistical difference in Brier Scores (Table 12) (Research Applications Laboratory 2015).

```
# 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)) %>%
  pander()
```

Table 10: 2-sample test for equality of proportions with continuity correction: `. out of nrow(hits)/2 %>% rep(2)`

Test statistic	df	P value	Alternative hypothesis	prop 1	prop 2
16.79	1	4.166e-05 * * *	two.sided	0.8603	0.8381

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

```
  kable(digits = 3,
        caption = "Mean Probabilities by Prediction Accuracy")
```

Table 11: Mean Probabilities by Prediction Accuracy

pred	winner	market	model
FALSE	FALSE	0.230	0.168
FALSE	TRUE	0.406	0.365
TRUE	FALSE	0.593	0.637
TRUE	TRUE	0.795	0.845

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

Table 12: Welch Two Sample t-test: `brier_score` by `method` (continued below)

Test statistic	df	P value	Alternative hypothesis
-0.339	16943	0.7346	two.sided

mean in group market	mean in group model
0.1084	0.1091

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

brier_model <- verification::brier(
  obs = hits_model$winner,
  pred = hits_model$prob) %>%
  unlist() %>%
  enframe() %>%
  slice(2:7)

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

brier_market <- verification::brier(
  obs = hits_market$winner,
  pred = hits_market$prob) %>%
  unlist() %>%
  enframe() %>%
  slice(2:7)
```

```
left_join(brier_market, brier_model, by = "name") %>%
  rename(market = value.x,
         model = value.y) %>%
  kable(digits = 3,
        caption = "Brier Test Comparison")
```

Table 14: Brier Test Comparison

name	market	model
bs	0.109	0.109
bs.baseline	0.237	0.237
ss	0.541	0.538
bs.reliability	0.019	0.006
bs.resol	0.147	0.134
bs.uncert	0.237	0.237

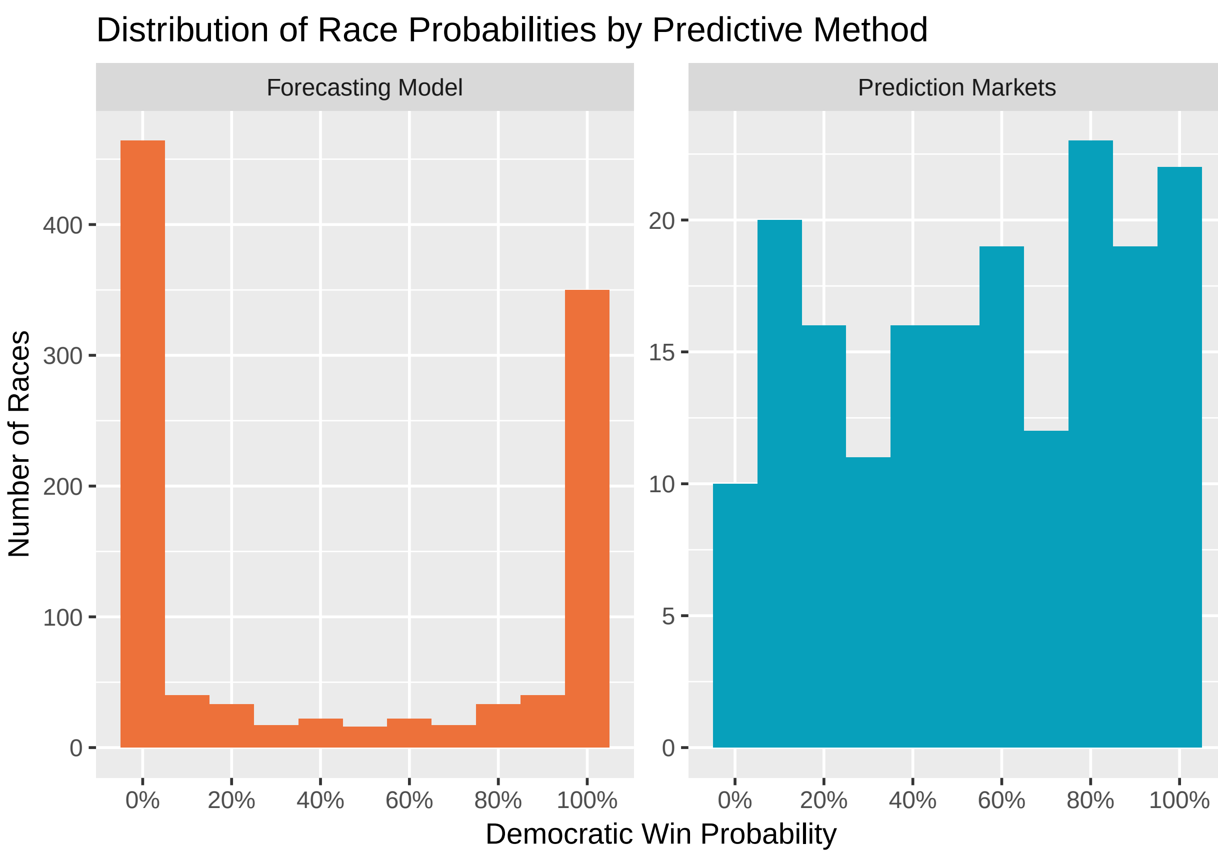
Application

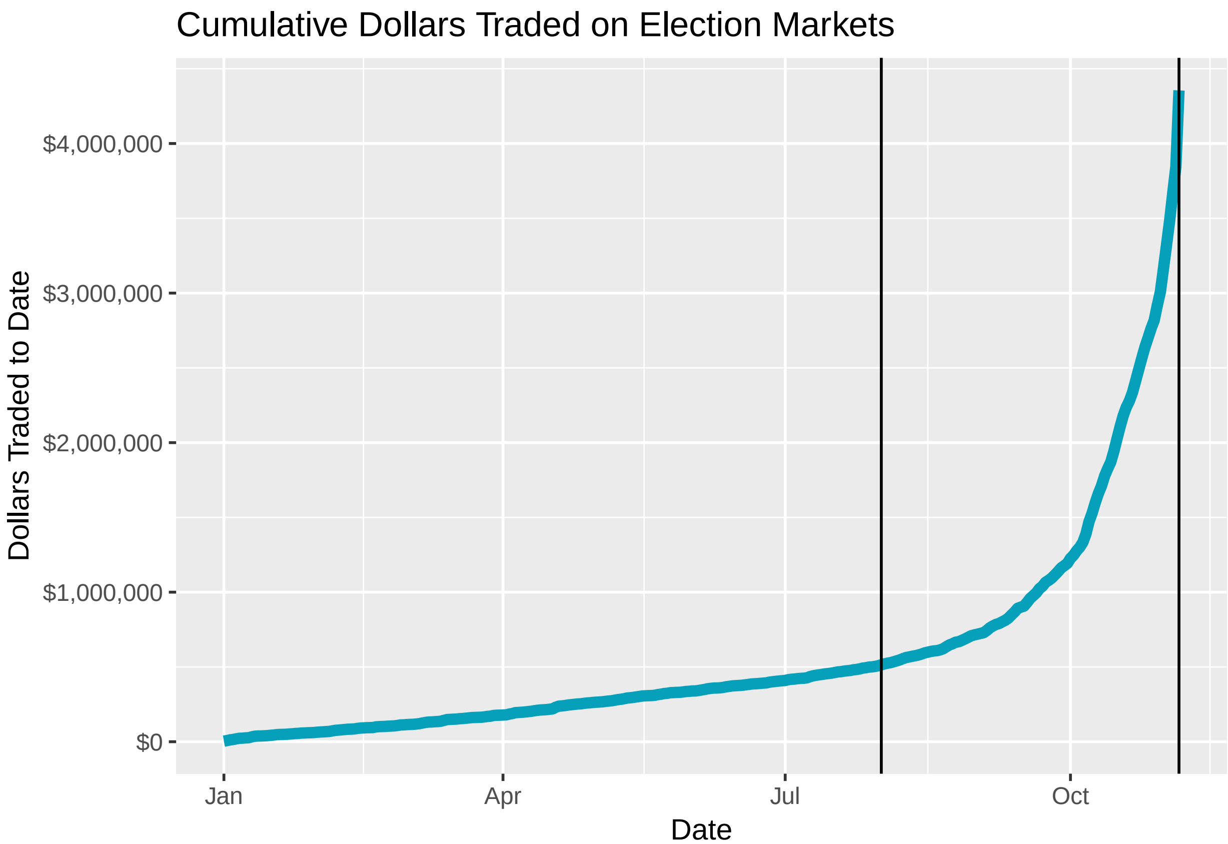
The probabilities for the races can be visualized using the `shiny` package (Chang et al. 2019).

The application is hosted online at https://kiernan.shinyapps.io/predictr/

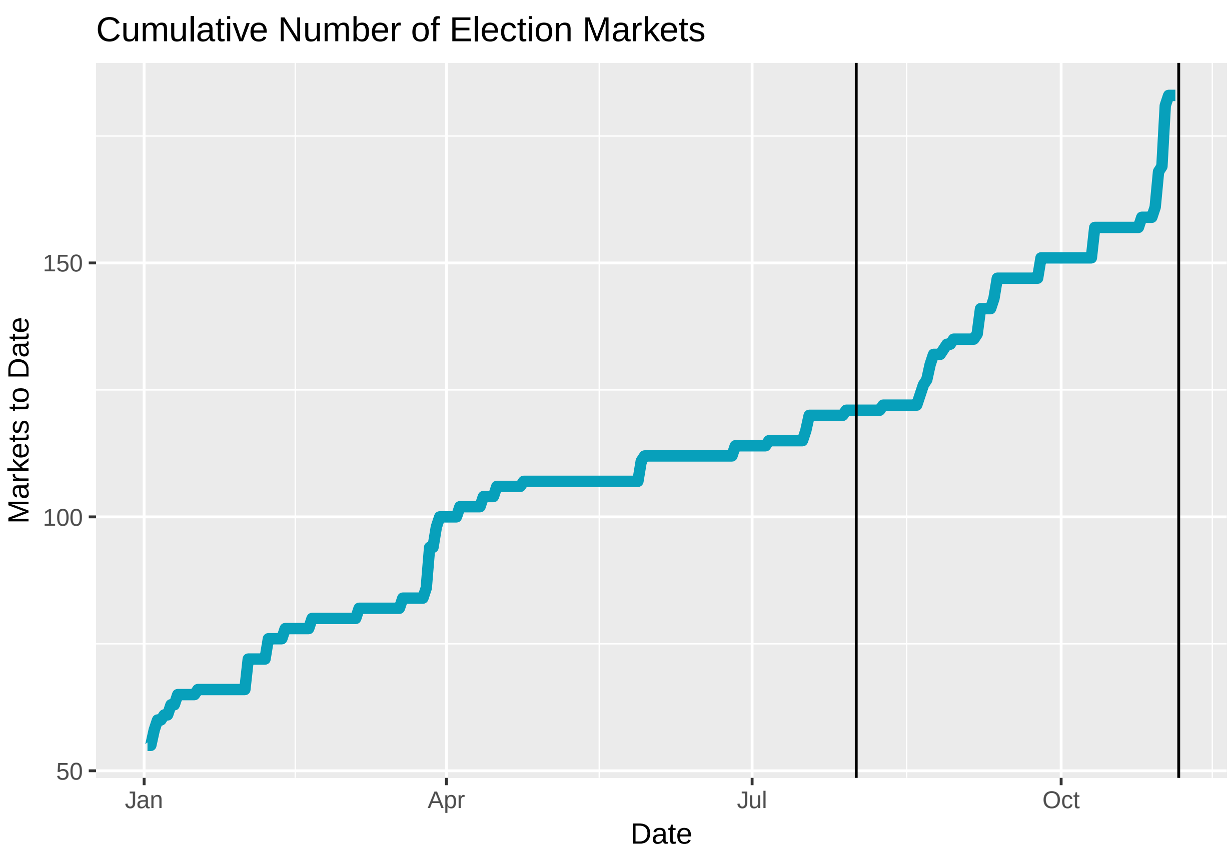
Visualize

Exploratory visualizations are made using the `ggplot2` package (Wickham 2016).

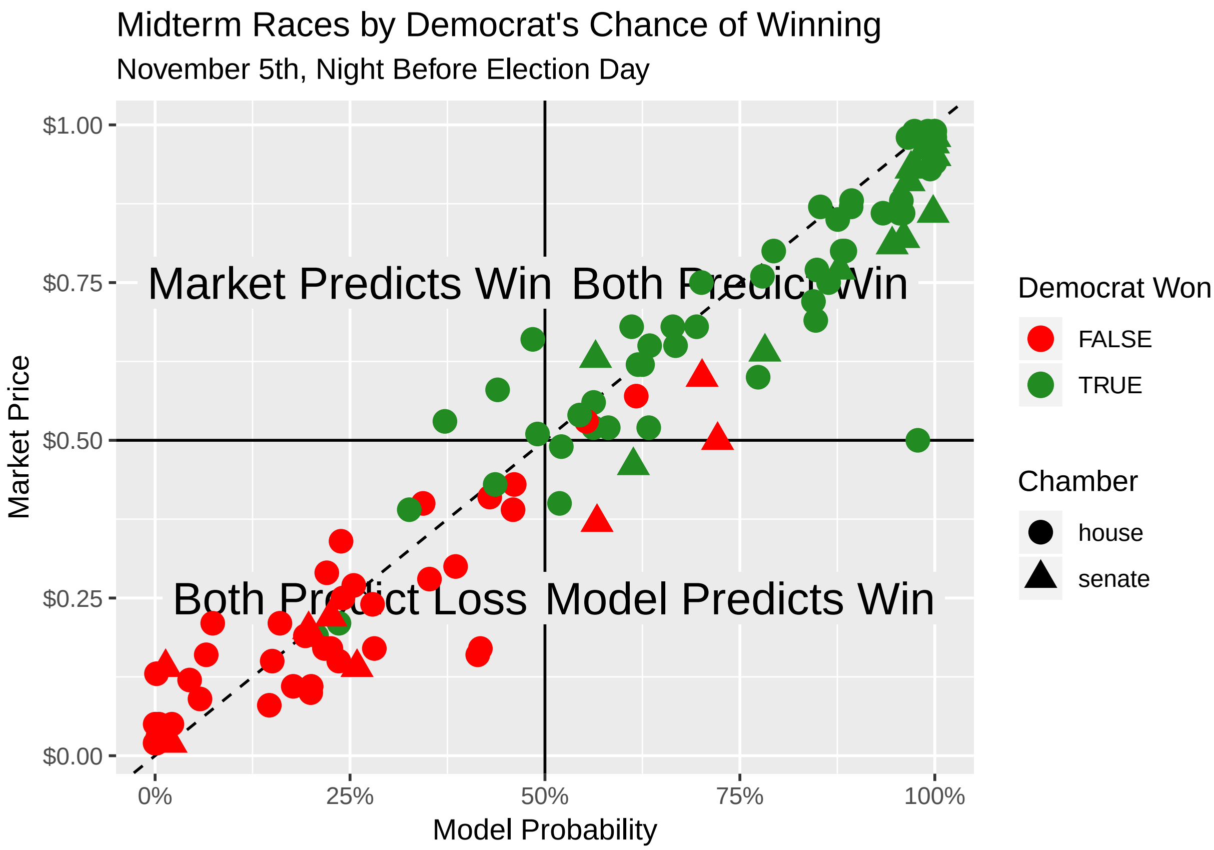


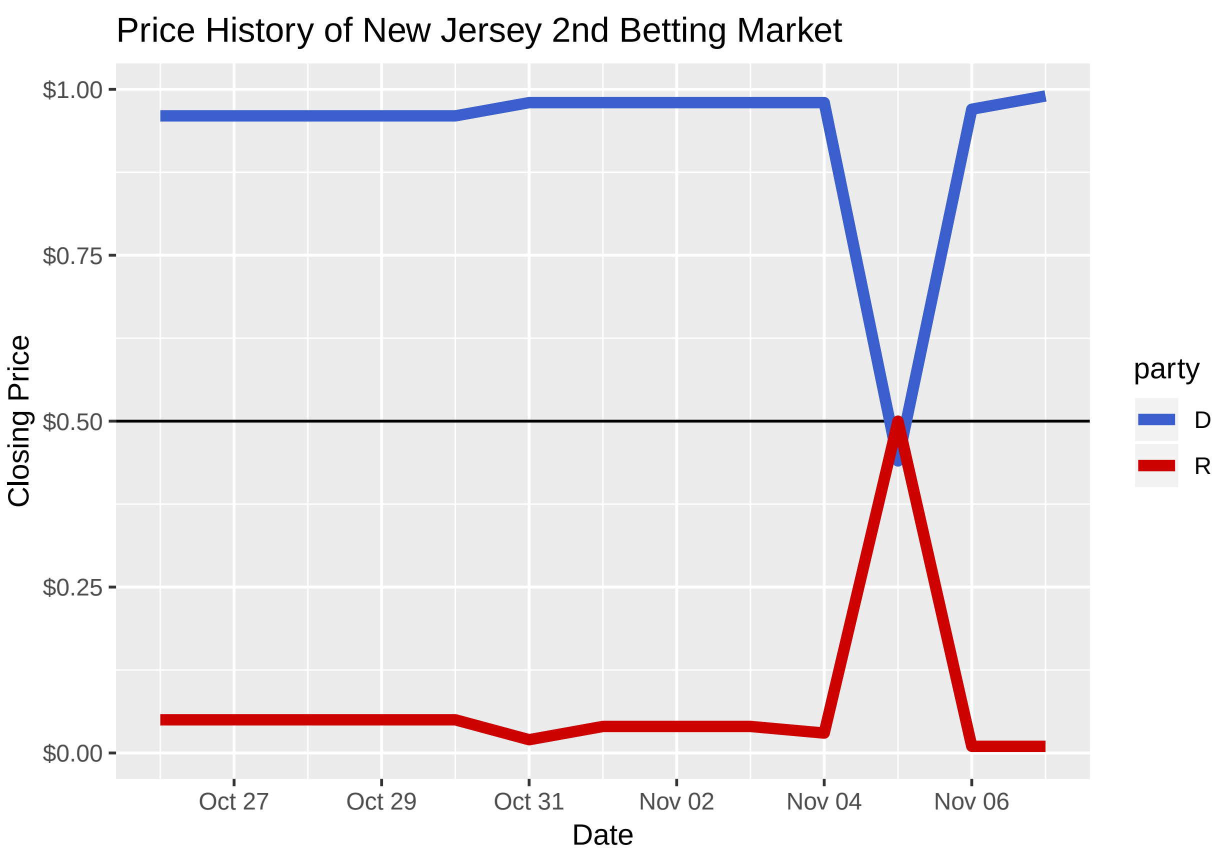


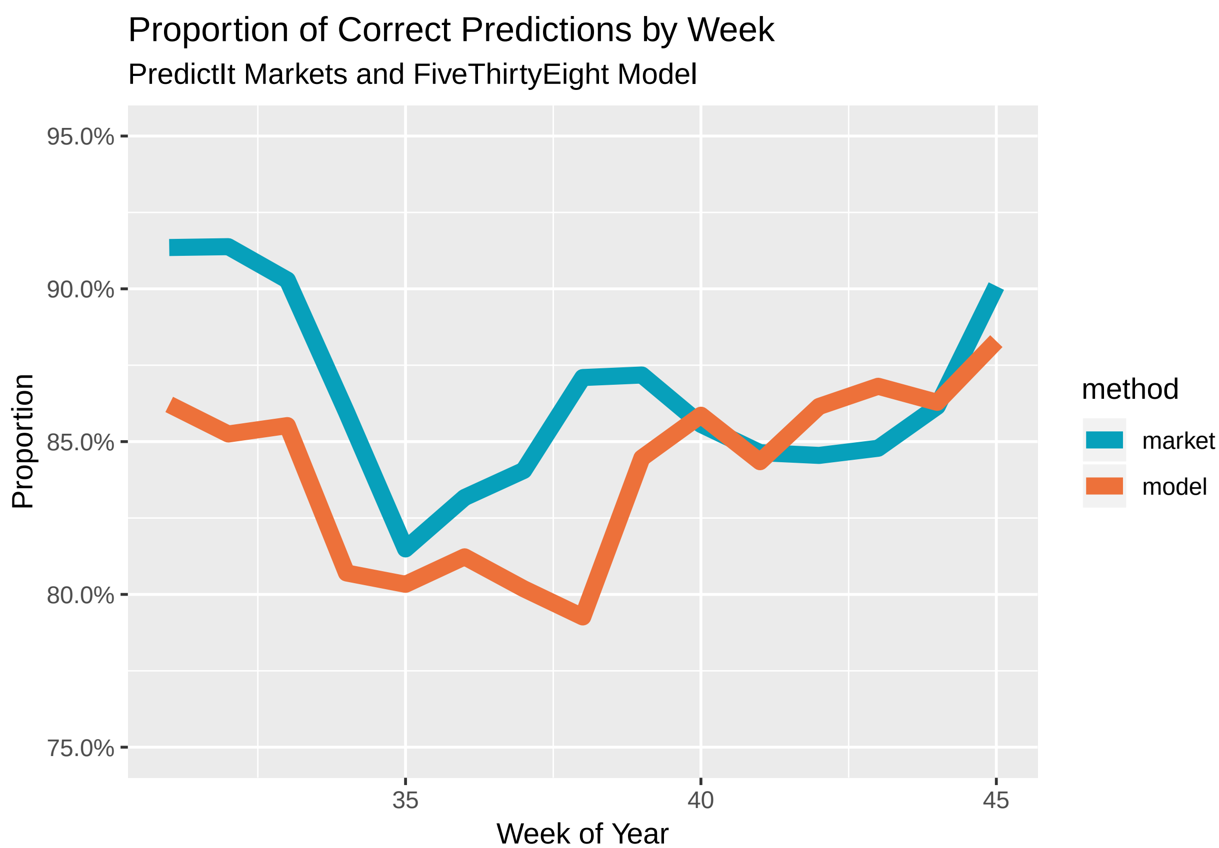
Cumulative Dollars Traded on Election Markets
Dollars Traded to Date
$4,000,000
$3,000,000
$2,000,000
$1,000,000
$0
Jan
Apr
Jul
Oct
Date



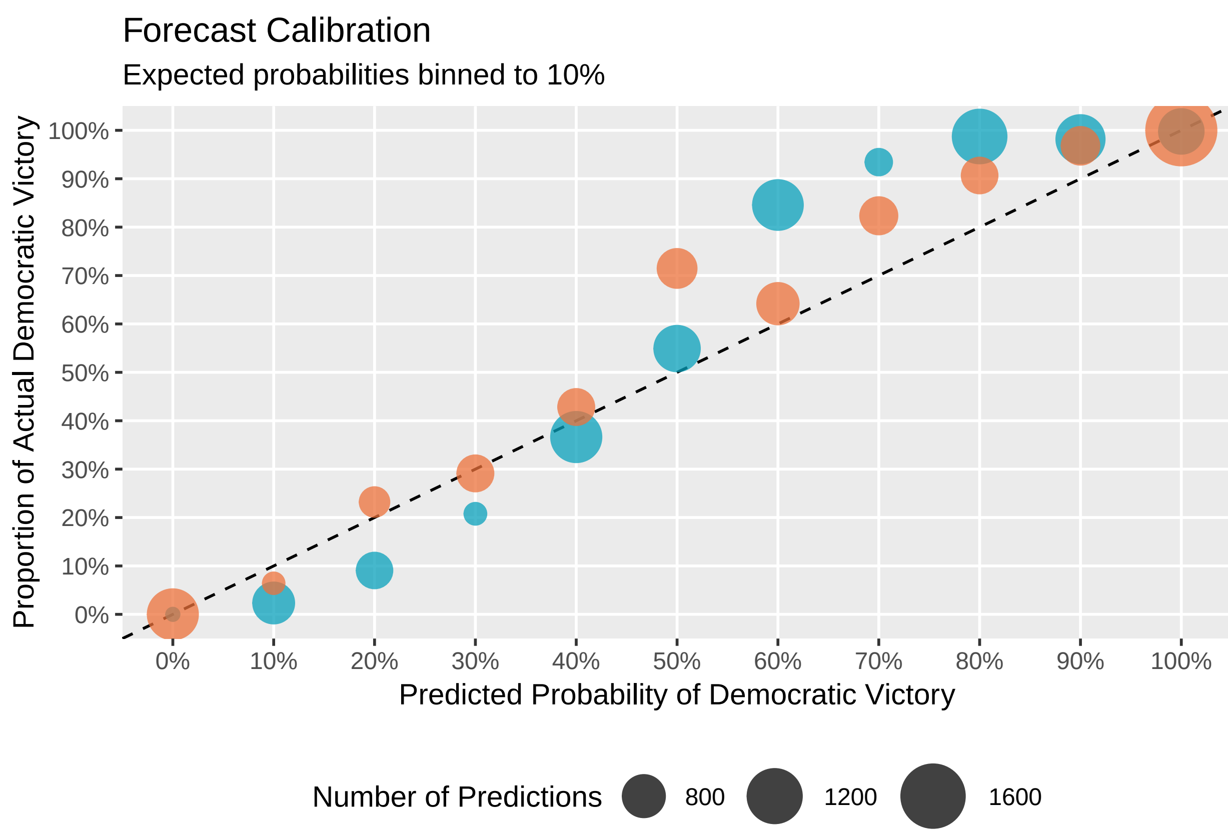
Cumulative Number of Election Markets
Markets to Date
150
100
50
Jan
Apr
Jul
Oct
Date


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

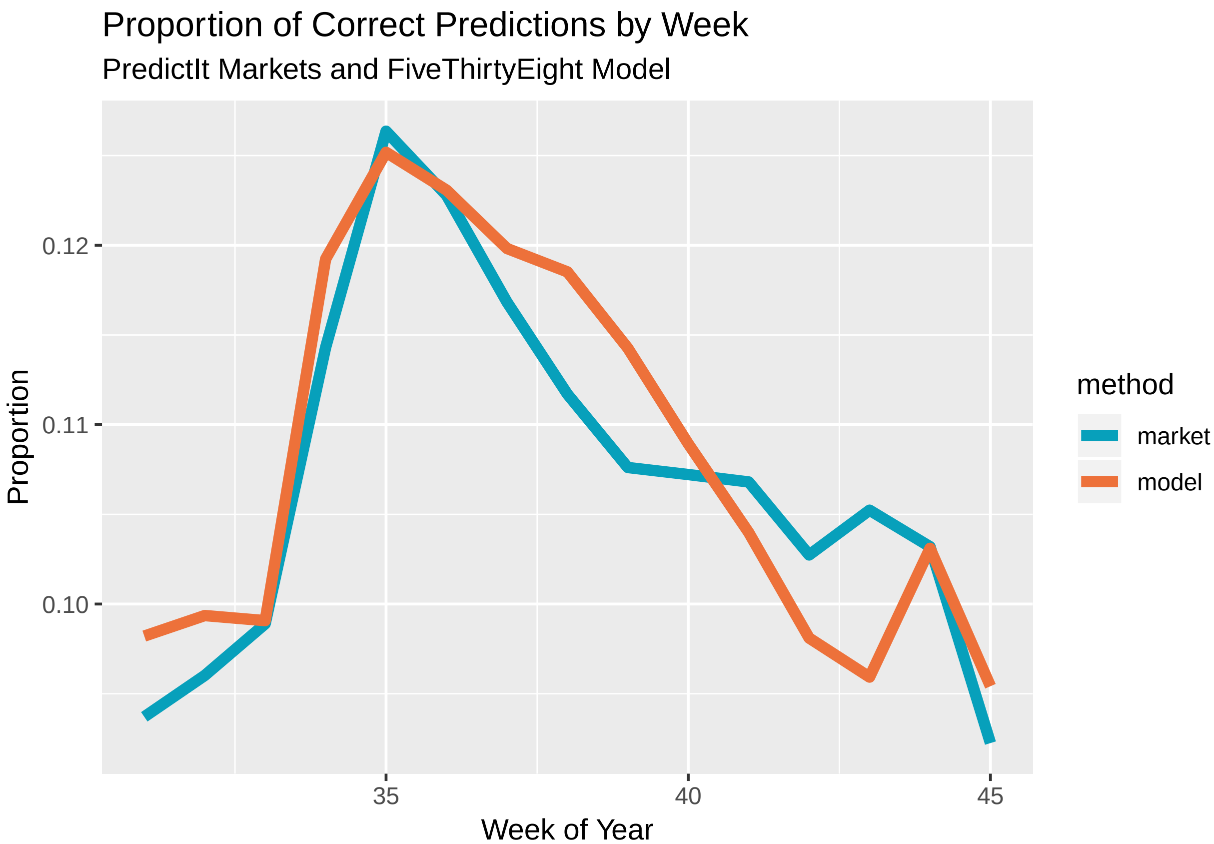

Price History of New Jersey 2nd Betting Market
Closing Price
$1.00
$0.75
$0.50
$0.25
$0.00
Oct 27
Oct 29
Oct 31
Nov 02
Nov 04
Nov 06
Date
party
D
R



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



Forecast Calibration
Expected probabilities binned to 10%
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Proportion of Actual Democratic Victory
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Predicted Probability of Democratic Victory
Number of Predictions
800
1200
1600



Bibliography

Chang, Winston, Joe Cheng, JJ Allaire, Yihui Xie, and Jonathan McPherson. 2019. *Shiny: Web Application Framework for R*. https://CRAN.R-project.org/package=shiny.

FiveThirtyEight. 2018a. "House District Forecast." https://projects.fivethirtyeight.com/2018-midterm-election-forecast/house/.

———. 2018b. "Senate Seat Forecast." https://projects.fivethirtyeight.com/2018-midterm-election-forecast/senate/.

———. 2018c. "2018 Midterm Election Results." https://53eig.ht/2PiFb0f.

Grolemund, Garrett, and Hadley Wickham. 2011. "Dates and Times Made Easy with lubridate." *Journal of Statistical Software* 40 (3): 1–25. http://www.jstatsoft.org/v40/i03/.

Project, United States. 2018. "Current Legislators." https://theunitedstates.io/congress-legislators/.

R Core Team. 2018a. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

———. 2018b. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Research Applications Laboratory, NCAR -. 2015. *Verification: Weather Forecast Verification Utilities*. https://CRAN.R-project.org/package=verification.

Rudis, Bob. 2017. *Wayback: Tools to Work with Internet Archive Wayback Machine Apis*. https://github.com/hrbrmstr/wayback.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org.

———. 2019. *Stringr: Simple, Consistent Wrappers for Common String Operations*. https://CRAN.R-project.org/package=stringr.

Wickham, Hadley, and Lionel Henry. 2019. *Tidyr: Easily Tidy Data with 'Spread()' and 'Gather()' Functions*. https://CRAN.R-project.org/package=tidyr.

Wickham, Hadley, Jim Hester, and Romain Francois. 2018. *Readr: Read Rectangular Text Data*. https://CRAN.R-project.org/package=readr.