

Code Apendix

```
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
```

```
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")))

## 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()))

## 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()

# Midterm election results via ABC and 538
# Used in https://538ig.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"))))

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"))

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 incorrectly 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)

# 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 endorsed 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")

# Separate 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)

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

# 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 <- 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)

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

# 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)

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

# 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()

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

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

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

brier_model <- verification::brier(
  obs = hits_model$winner,
  pred = hits_model$prob)

brier_market <- verification::brier(
  obs = hits_market$winner,
  pred = hits_market$prob)

```