# R Coding Demonstration Week 9: Predicting Elections and Sampling from the Voter File

Matthew Blackwell

Gov 51 (Harvard)

# Introduction

- Modern election forecasting usually takes polling averages + Monte Carlo simulation to simulate uncertainty about what exactly the final votes will be in each state
- Allows us to simulate different electoral vote outcomes in presidential races.
- Today, we're going to use the final polling averages from Five Thirty Eight to:
    - (a) get predictions of what will happen and
    - (b) see how much uncertainty there is around that prediction.
- To do this, we'll rely on draws of a random variable (the normal in this case).

# Data

```
polls_2020 <- read.csv("data/polls_2020.csv")
```

| Variable Name | Description |
|---|---|
| state | State (or district) name |
| biden | Polling average for Biden |
| trump | Polling average for Trump |
| biden_lead | Biden's lead in the polling average |
| e_votes | Number of electoral votes for state/district |

```
head(polls_2020, 3)
```

```
##     state biden trump biden_lead e_votes
## 1 Alabama  38.0  57.5     -19.46       9
## 2  Alaska  43.3  50.9      -7.61       3
## 3 Arizona  48.6  45.7       2.93      11
```

# Question 1

Based on the current polling averages, calculate the number of electoral votes that Biden is predicted to win.

# Answer 1

```
biden_winning <- polls_2020$biden_lead > 0
sum(polls_2020$e_votes[biden_winning])
```

```
## [1] 351
```

# Question 2

Suppose now that there are “Shy Trump” voters who refuse to answer the polls or give the wrong answer. Assume these result in a 4 point swing to Biden. Adjusting Biden’s lead for this, how many electoral votes is he predicted to get?

# Answer 2

```r
biden_winning_shy <- (polls_2020$biden_lead - 4) > 0
sum(polls_2020$e_votes[biden_winning_shy])
```

```
## [1] 279
```

# Polling errors

## How accurate have U.S. polls been?

Weighted-average error in polls in final 21 days of the campaign

| | PRESIDENTIAL | | STATE-LEVEL | | | |
|---|---|---|---|---|---|---|
| CYCLE | PRIMARY | GENERAL | GOVERNOR | U.S. SENATE | U.S. HOUSE | COMBINED |
| 2017-18 | – | – | 5.2 | 6.0 | 4.1 | 5.1 |
| 2015-16 | 10.1 | 4.8 | 5.4 | 5.0 | 5.5 | 6.8 |
| 2013-14 | – | – | 4.4 | 5.4 | 6.7 | 5.4 |
| 2011-12 | 8.9 | 3.6 | 4.8 | 4.7 | 4.7 | 5.1 |
| 2009-10 | – | – | 4.9 | 4.8 | 6.9 | 5.7 |
| 2007-08 | 7.4 | 3.6 | 4.1 | 4.7 | 5.7 | 5.4 |
| 2005-06 | – | – | 5.0 | 4.2 | 6.5 | 5.3 |
| 2003-04 | 7.1 | 3.2 | 6.1 | 5.6 | 5.4 | 4.8 |
| 2001-02 | – | – | 5.2 | 4.9 | 5.4 | 5.2 |
| 1999-2000 | 7.6 | 4.4 | 4.9 | 6.1 | 4.4 | 5.5 |
| 1998 | – | – | 8.1 | 7.4 | 6.8 | 7.5 |
| All years | 8.7 | 4.0 | 5.4 | 5.4 | 6.2 | 5.9 |

Pollsters that are banned by FiveThirtyEight because we know or suspect that they faked their data are not included in the averages. Averages are weighted based on the number of polls a particular firm conducted. Specifically, the weights are based on the square root of the number of polls in a particular category that each

# Question 3

Let's use random variables to simulate the electoral college over lots of different possible outcomes. Assume that the polling error for each state is distributed normally with mean 0 and standard deviation 5. Conduct 10,000 simulations of that polling error, use it to recover the "true" Biden lead and electoral votes.

Plot the distribution of the simulated electoral votes for Biden and calculate what proportion of simulations he gets above 270.

# Answer 3

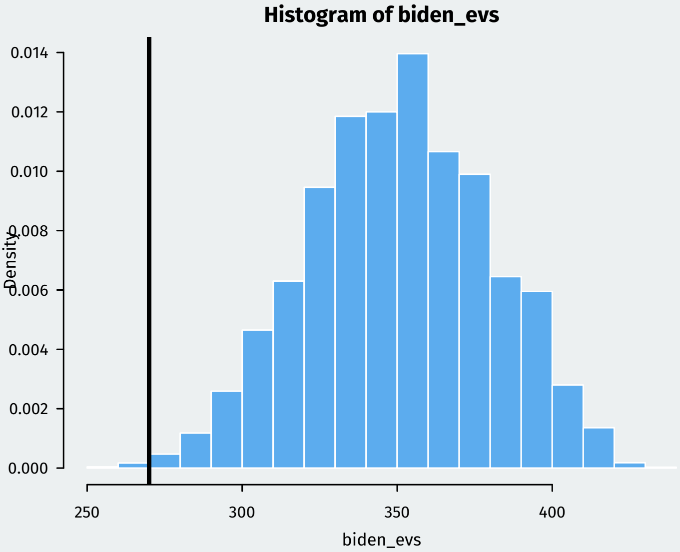
```r
n_sims <- 10000
n_st <- nrow(polls_2020)

biden_evs <- rep(NA, times = n_sims)
for (i in 1:n_sims) {
  polling_noise <- rnorm(n = n_st, mean = 0, sd = 5)
  sim_leads <- polls_2020$biden_lead - polling_noise
  biden_wins <- sim_leads > 0
  biden_evs[i] <- sum(polls_2020$e_votes[biden_wins])
}
mean(biden_evs >= 270)
```

```
## [1] 0.998
```

```r
hist(biden_evs, col = "steelblue2", border = "white",
     freq = FALSE)
abline(v = 270, lwd = 3)
```

# Answer 3 (cont'd)



# Question 4

Now let's add a systematic polling error. In addition to the state-level polling error, assume there is a single error that applies to all states that follows a normal distribution with mean 0 and standard deviation 4. Draw a single systematic error for each iteration of the simulation. Conduct 10,000 simulations of the two polling errors, use them to recover the "true" Biden lead and electoral votes.

Plot the distribution of the simulated electoral votes for Biden and calculate what proportion of simulations he gets above 270.

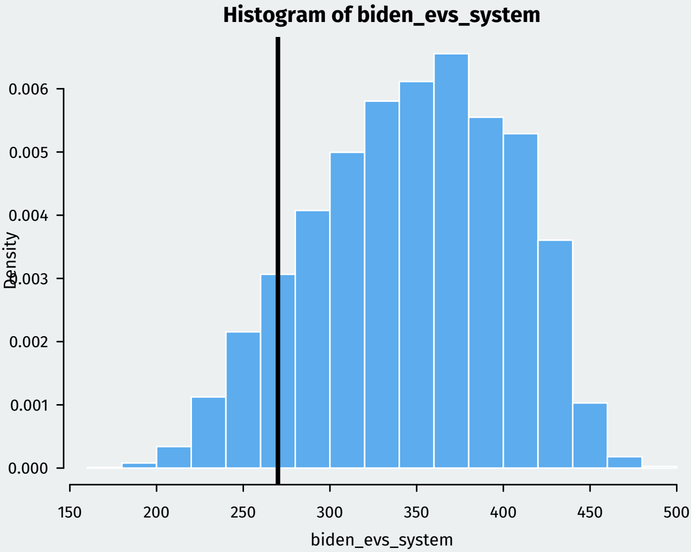
# Answer 4

```
biden_evs_system <- rep(NA, times = n_sims)
for (i in 1:n_sims) {
  system_error <- rnorm(n = 1, mean = 0, sd = 4)
  polling_noise <- rnorm(n = n_st, mean = 0, sd = 5)
  sim_leads <- polls_2020$biden_lead - system_error - polling_noise
  biden_wins <- sim_leads > 0
  biden_evs_system[i] <- sum(polls_2020$e_votes[biden_wins])
}
mean(biden_evs_system >= 270)
```

```
## [1] 0.901
```

```
hist(biden_evs_system, col = "steelblue2", border = "white",
     freq = FALSE)
abline(v = 270, lwd = 3)
```

# Answer 4 (cont'd)



# Sampling from the voter file

- A new way that some pollsters are polling for election is by sampling from the voter file directly.
- Voter files are really big, so we're going to work with one county in FL, Miami-Dade.
- We've stripped identifiable data, but the original had names, addresses, phone numbers, and email addresses.

# Miami-Dade voter file

```
load("data/dade_vf_2020.RData")
```

| Variable | Description |
|---|---|
| voter_id | Voter ID number |
| city | City of residence |
| precinct | Precinct of residence |
| race | Race of registered voter |
| dem | 1=Democrat, 0=otherwise |
| rep | 1=Republican, 0=otherwise |
| female | 1=Female, 0=otherwise (Male/Unknown) |
| age | Registrant age |
| PPP_2016 | 1 = Voted in 2016 presidential primary, 0=didn't vote |
| PRI_2016 | 1 = Voted in 2016 state primary, 0=didn't vote |
| GEN_2016 | 1 = Voted in 2016 general election, 0=didn't vote |

# Question 5

What proportion of Miami-Dade County registered voters are registered as Democrats? Take a sample of size 100 and calculate the sample mean.

# Answer 5

```
mean(dade$dem)
```

```
## [1] 0.407
```

```
dem_sample <- sample(dade$dem, size = 100)
dem_sample
```

```
##   [1] 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 1 1 1 0 0 0 0 1 0 1 1 0 0 0
##  [33] 0 1 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 1 1 0 0 0
##  [65] 1 0 1 0 1 0 0 1 0 0 0 0 0 1 0 0 1 0 1 0 1 1 0 1 0 1 0 1 0 1 1 0
##  [97] 0 0 1 1
```

```
mean(dem_sample)
```

```
## [1] 0.38
```

# Question 6

What is the average age of Miami-Dade County registered voters (that is, what is the population mean)? Take a sample of size 100 ages from the set of registered voters and calculate the sample mean.

# Answer 6

```
mean(dade$age, na.rm = TRUE)
```

```
## [1] 50
```

```
age_sample <- sample(dade$age, size = 100)
age_sample
```

```
##   [1] 27 26 60 63 66 79 64 88 70 26 47 92 26 32 86 53 42 42 31 19 39
##  [22] 20 58 62 68 58 76 78 31 42 31 46 34 62 43 36 36 47 57 70 73 68
##  [43] 56 35 91 70 66 58 46 22 19 53 52 68 85 38 23 46 92 40 44 31 55
##  [64] 25 93 43 34 67 43 20 92 24 41 30 56 56 55 25 64 70 23 41 55 48
##  [85] 31 55 27 40 58 49 25 38 30 27 45 78 69 84 44 94
```

```
mean(age_sample, na.rm = TRUE)
```

```
## [1] 50.6
```

# Question 7

Use a `for` loop to repeat the process of sampling the voter file 10,000 times. In each iteration, take a sample of 100 from `dade$dem` and save the sample mean of that sample.

Compare the mean and standard deviation of the 10,000 sample means to the population mean and population standard deviation of `dem`. Draw a histogram of the means: what distribution do they follow?

# Answer 7

```
n_sims <- 10000

dem_means <- rep(NA, times = n_sims)
for (i in 1:n_sims) {
  dem_means[i] <- mean(sample(dade$dem, size = 100))
}
mean(dem_means)
```

```
## [1] 0.407
```

```
mean(dade$dem)
```

```
## [1] 0.407
```

```
sd(dem_means)
```

```
## [1] 0.0491
```

```
sd(dade$dem) / sqrt(100)
```

```
## [1] 0.0491
```

# Answer 7 (cont'd)

```
hist(dem_means, col = "steelblue2", border = "white", freq = FALSE)
```