R Coding Demonstration Week 9: Polling Errors and Sampling from the Voter File (Tidy)

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.
- But polling averages can be wrong! Let's use the 2020 results (so far!) to investigate how off the polling errors were.

Data

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

Variable Name	Description
`state`	State (or district) name
`biden_polls`	Polling average for Biden
`trump_polls`	Polling average for Trump
`biden_poll_lead`	Biden's lead in the polling average
`e_votes`	Number of electoral votes for state/district
`biden_pct`	Preliminary percent of votes for Biden
`trump_pct`	Preliminary percent of votes for Trump
`biden_lead`	Preliminary lead for Biden

Question 1

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

Answer 1

```
polls_2020 %>%
  mutate(biden_winning_shy = biden_poll_lead > 0) %>%
  summarize(total_ev = sum(e_votes[biden_winning_shy]))
```

```
##   total_ev
## 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

```
polls_2020 %>%
  mutate(biden_winning_shy = biden_poll_lead - 4 > 0) %>%
  summarize(total_ev = sum(e_votes[biden_winning_shy]))
```

```
##   total_ev
## 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 polling errors and compare them to the actual polling error. 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 and add it to the Biden lead in Florida to get simulations of different possible polling averages.

Plot the distribution of the simulated polling average leads for Biden and calculate what proportion of these leads are bigger in magnitude than the true polling error.

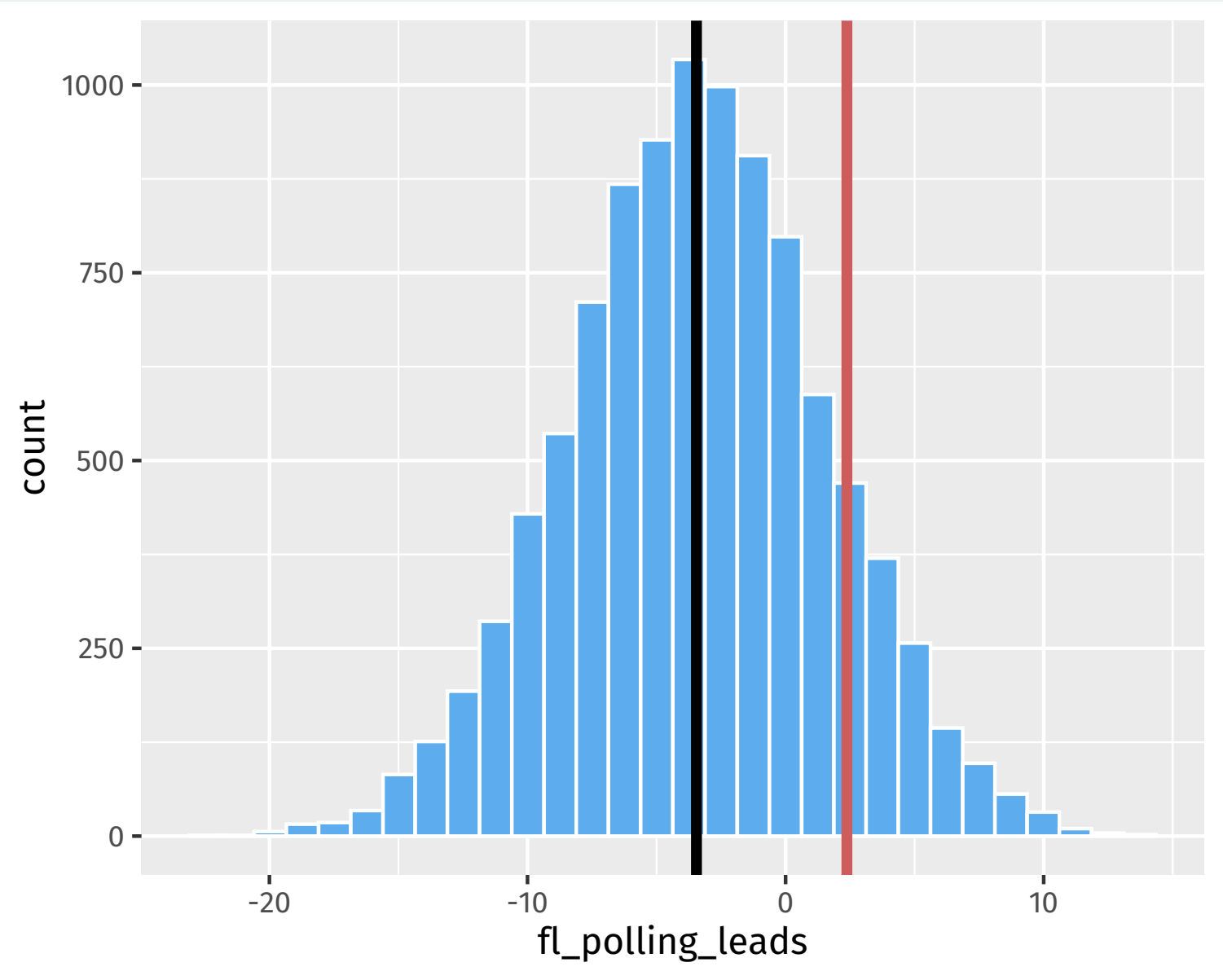
Answer 3

```
n_sims <- 10000
polling_noise <- rnorm(n = n_sims, mean = 0, sd = 5)

fl_2020 <- polls_2020 %>%
  filter(state == "Florida") %>%
  pull(biden_lead)

fl_2020_polls <- polls_2020 %>%
  filter(state == "Florida") %>%
  pull(biden_poll_lead)

fl_polling_leads <- fl_2020 + polling_noise
mean(abs(polling_noise) >= abs(fl_2020 - fl_2020_polls))
```

```
## [1] 0.246
```

```
ggplot(mapping = aes(x = fl_polling_leads)) +
  geom_histogram(fill = "steelblue2", col = "white") +
  geom_vline(xintercept = fl_2020_polls, size = 1.5,
             color = "indianred") +
  geom_vline(xintercept = fl_2020, size = 1.5)
```

Answer 3 (cont'd)



Question 4

Let's use random variables to simulate polling errors and compare them to the actual polling error. 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 and add it to the Biden lead in Wisconsin to get simulations of different possible polling averages.

Plot the distribution of the simulated polling average leads for Biden and calculate what proportion of these leads are bigger in magnitude than the true polling error.

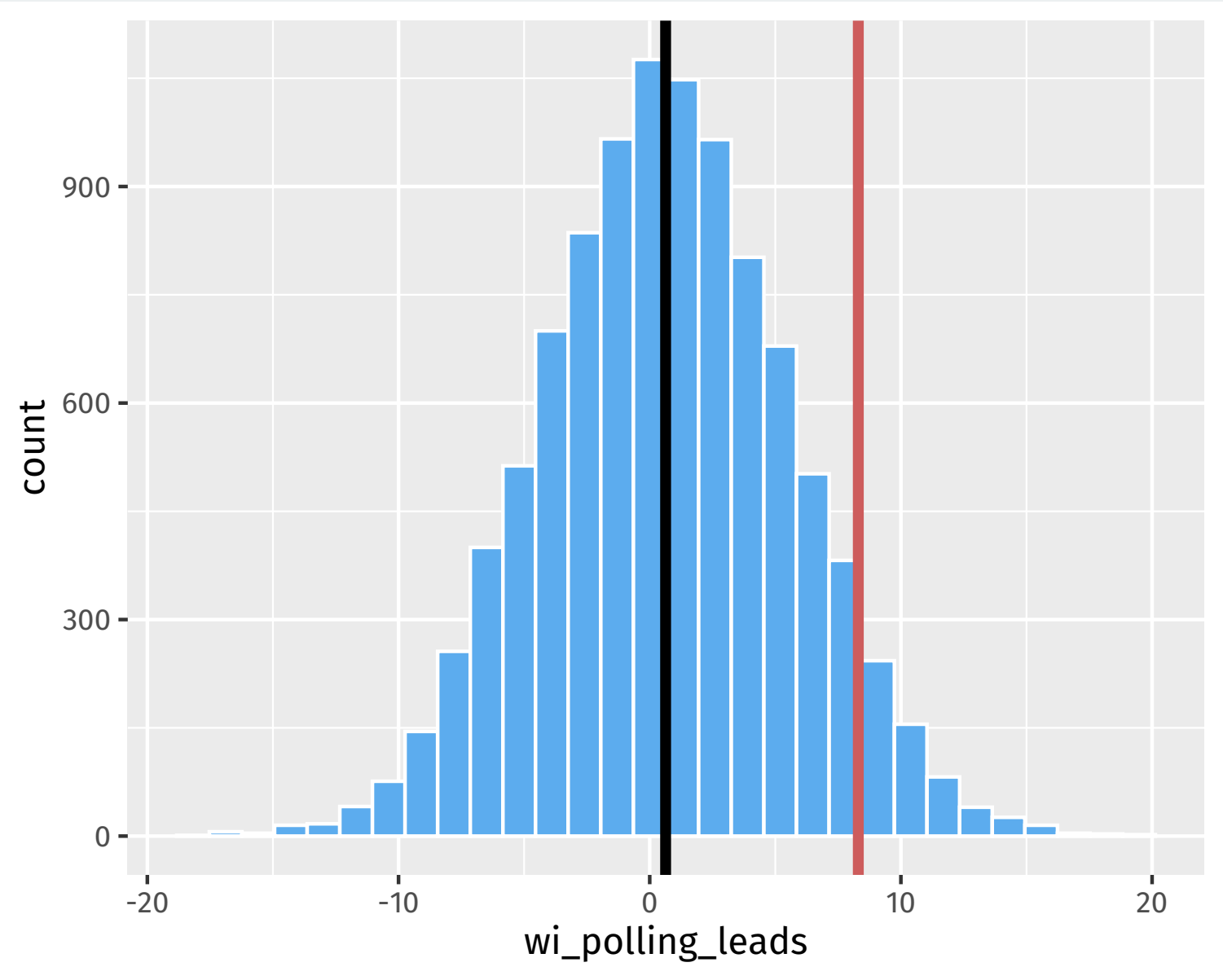
Answer 4

```
polling_noise <- rnorm(n = n_sims, mean = 0, sd = 5)

wi_2020 <- polls_2020 %>%
  filter(state == "Wisconsin") %>%
  pull(biden_lead)

wi_2020_polls <- polls_2020 %>%
  filter(state == "Wisconsin") %>%
  pull(biden_poll_lead)

wi_polling_leads <- wi_2020 + polling_noise
mean(abs(polling_noise) >= abs(wi_2020 - wi_2020_polls))
```

```
## [1] 0.12
```

```
ggplot(mapping = aes(x = wi_polling_leads)) +
  geom_histogram(fill = "steelblue2", col = "white") +
  geom_vline(xintercept = wi_2020_polls, size = 1.5,
             color = "indianred") +
  geom_vline(xintercept = wi_2020, size = 1.5)
```

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

```
mean(dem_sample)
```

```
## [1] 0.35
```

Answer 5 (alt tidy version)

```
dade %>%
  sample_n(size = 100) %>%
  pull(dem) %>%
  mean()
```

```
## [1] 0.3
```

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] 26 23 58 35 66 35 27 23 81 72 25 47 26 20 48 47 22 75 73 19 53
##  [22] 32 58 49 30 54 24 20 36 38 52 68 60 50 21 50 85 48 52 31 56 53
##  [43] 19 28 50 84 34 63 51 68 84 87 19 72 48 65 51 54 26 66 29 20 59
##  [64] 22 33 19 28 24 28 79 20 70 49 39 25 63 36 45 26 97 33 53 43 67
##  [85] 43 82 58 72 28 67 57 29 19 69 40 23 63 79 21 59
```

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

```
## [1] 46.5
```

Answer 5 (alt tidy version)

```
dade %>%
  sample_n(size = 100) %>%
  pull(age) %>%
  mean()
```

```
## [1] 49.1
```

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

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

```
## [1] 0.406
```

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

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

```
sd(samples$sample_means)
```

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

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

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

Answer 7 (cont'd)

```
ggplot(samples, aes(x = sample_means)) +
  geom_histogram(fill = "steelblue2", color = "white", bins = 15)
```