Topic 7: (Re)Sampling Redux

Josh Clinton

10/5/2021

Last Time

- Probability can be thought of as the "limit' ' of repeated identical experiments. Instead of calculating expected values of random variables that are independently and identically distributed using mathematics we can often use resampling to calculate the probability an event occurs by computing the proportion of events that occur in the resampling.
- Functions: `sample` and `sample_n`

This Time

- How much data for an "accurate" estimate?
- Given a polling estimate, how quantify the probability that a candidate will win? Or get over a specific threshold? Or win by a given margin?

Applications:

1. Drawing a sample from the PA Voter File
2. Quantifying error in the National Popular Vote

When pollsters do polls they often sample directly from the voter file! If the voter file is "true" (is it?) how accurate are samples from the voter file? [Best case scenario – only source of error is sampling error!]

Load the data file used to interview respondents for the 2020 Pennsylvania Exit Poll and compute some indicator variables .

```
load("data/pa.sample.select.Rdata")
pa.sample <- pa.sample %>%
    mutate(likely.dem = ifelse(likely.party == "D",1,0),
           likely.rep = ifelse(likely.party == "R",1,0),
           likely.ind = ifelse(likely.party == "I",1,0))
glimpse(pa.sample)
```

```
## Rows: 98,548
## Columns: 19
## $ city              <chr> "Aliquippa", "Aliquippa", "Aliquippa", "Aliquippa", ~
## $ likely.party      <fct> D, D, D, R, D, R, R, D, R, D, D, D, D, D, D, R, R, R~
## $ female            <dbl> 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1~
## $ AgeUnder30        <dbl> 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0~
## $ Age3039           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0~
## $ Age4049           <dbl> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0~
## $ Age5059           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0~
## $ Age6074           <dbl> 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1~
## $ Age75p            <dbl> 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

```
## $ imputed.white     <dbl> 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1~
## $ imputed.black     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ imputed.hispanic  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ fips.county       <chr> "42007", "42007", "42007", "42007", "42007", "42007"~
## $ PctDemCtyVote2020 <dbl> 41.04, 41.04, 41.04, 41.04, 41.04, 41.04, 41.04, 41.~
## $ PctDemCtyVote2016 <dbl> 40.31, 40.31, 40.31, 40.31, 40.31, 40.31, 40.31, 40.~
## $ CooperateSurvey   <dbl> NA, 0, NA, NA, NA, NA, NA, NA, NA, 1, NA, NA, NA, NA~
## $ likely.dem        <dbl> 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0~
## $ likely.rep        <dbl> 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1~
## $ likely.ind        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

Create a tibble of the Statewide distribution of partisanship called `PA.pty.breakdown`

We will use this as the "truth" to compare the accuracy and precision/dispersion (uncertainty) of random samples.

```
PA.pty.breakdown <- pa.sample %>%
          summarize(pct.dem = mean(likely.dem),
                    pct.rep = mean(likely.rep),
                    pct.ind = mean(likely.ind))
```

- How large is the tibble `PA.pty.breakdown`?

- What is the meaning of the 2nd value?
- How can you extract just the 2nd value (recall that it is a list!)

Use `sample_n` to sample various sized samples from the tibble

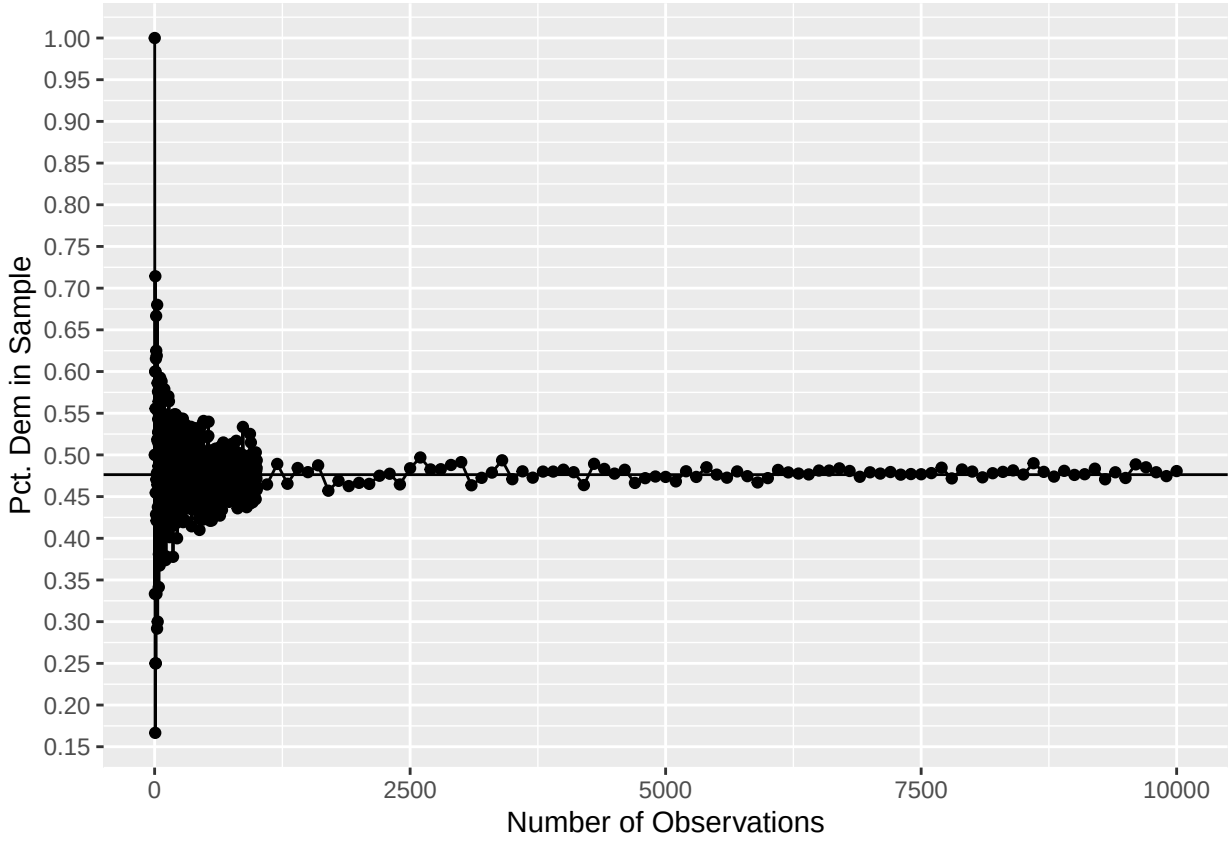
```
samplesize <- c(seq(1,1000),seq(1100,10000,by=100))  # Create a vector of sample sizes (number of respo

pty.est <- NULL # Create a holding vector

for(i in seq_along(samplesize)){  # because our index is not sequential we want to seq_along()

  pty.est <- pa.sample %>%
    sample_n(samplesize[i], replace= TRUE) %>%   # note the indexing of samplesize!  This will change a
    summarize(MeanDem = mean(likely.dem),
              MeanRep = mean(likely.rep),
              MeanInd = mean(likely.ind)) %>%
    mutate(SampleSize = samplesize[i]) %>%
    bind_rows(pty.est)   # add the results to the existing list
}

pty.est %>%
  ggplot() +
  geom_line(aes(x=SampleSize,y=MeanDem)) +       # NOTE: I could have put this in ggplot() but I define
  geom_point(aes(x=SampleSize,y=MeanDem)) +
  labs(x = "Number of Observations") +
  labs(y = "Pct. Dem in Sample") +
  scale_y_continuous(breaks=seq(0,1,by=.05)) +
  geom_hline(yintercept=PA.pty.breakdown[[1]])   # NOTE: because PA.pty.breakdown is a list we need [[]
```



Law of Large Numbers

- As the number of data points being analyzed get larger, the mean of a random sample of that data will get closer and closer to the true mean in the data generating process.
- Importance of random sampling! If every observation has an equal chance of being observed/measured/studied, more data means more accurate results! (But are poll respondents a random sample?)

Aside: accurate relative to what?

How do we define what we mean by "error"?

- Error: difference between the estimate and the truth
- Absolute Error: absolute value between the estimate and the truth
- Squared Error: difference between the estimate and the truth squared

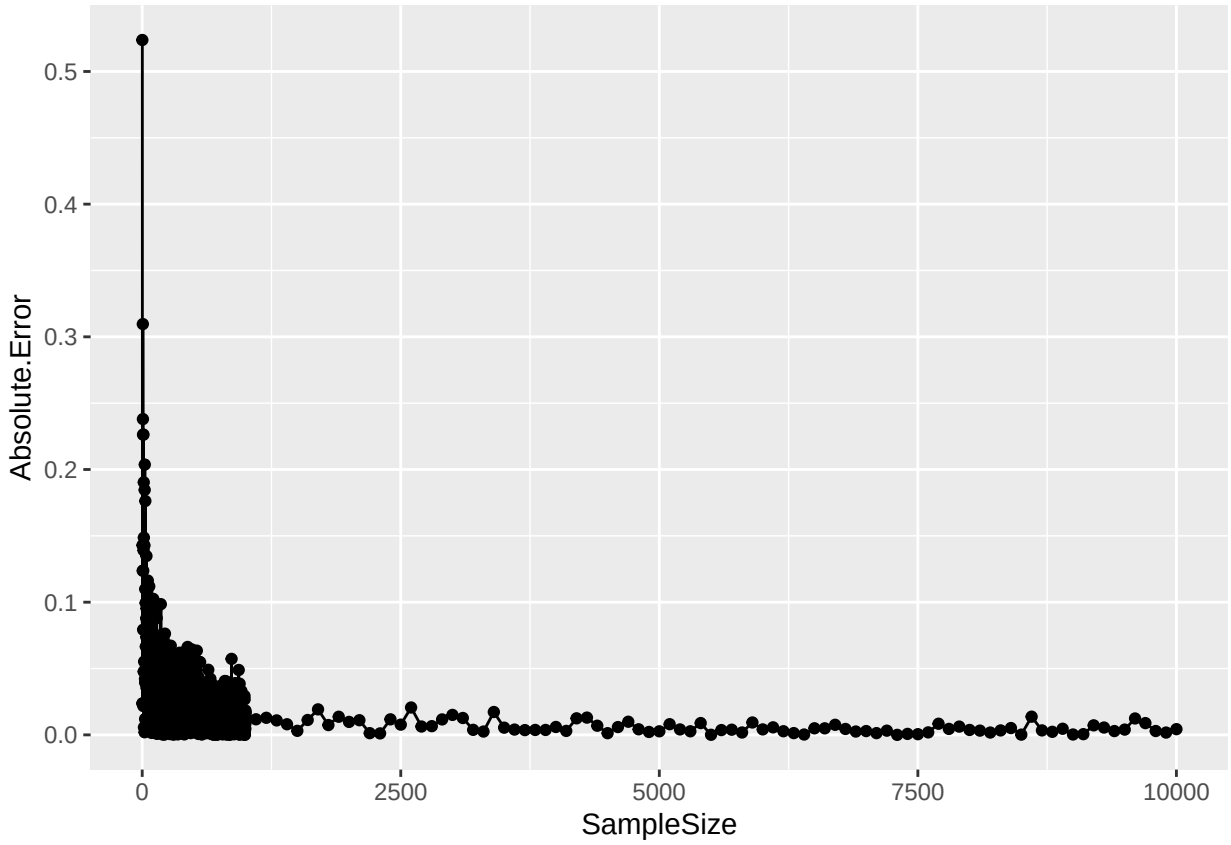
```
pty.est <- pty.est %>%
  mutate(Error = MeanDem - PA.pty.breakdown[[1]],
         Absolute.Error = abs(MeanDem - PA.pty.breakdown[[1]]),
         Squared.Error = (MeanDem - PA.pty.breakdown[[1]])^2)

pty.est %>%
   summarize(MeanError = mean(Error),
     MeanAbsError = mean(Absolute.Error),
     MeanSqdError = mean(Squared.Error))
```

```
##       MeanError MeanAbsError MeanSqdError
## 1 -1.359632e-05   0.02266336  0.001508641
```

- How come `MeanError` is so much smaller than `MeanAbsError`?
- Error *due to random sampling of data* gets smaller as number of data points gets larger!

```
pty.est %>% ggplot(aes(x=SampleSize,y=Absolute.Error)) +  # Here we defined the aesthetic in the ggplot
  geom_point() +
  geom_line()
```



Empirical Bootstrap: General Idea

- General idea is to use our sample as the population – draw repeated samples to learn about the variation in our estimates (if data is iid).
- The basic idea – variation in the samples we draw reflect how any single sample may vary from the true population. This between-sample variation will result in variation in the statistic/function we are interested in that we can then use.
- Proportions = Probability!
- Completely generic – can be applied to *any* function/statistic of the data!
- Completely easy – all we need to do is to be able to sample from our data with replacement!
- Limitation? – computing time/memory.

Implementation:

1. Sample from our data `B` times with replacement.
2. For each sample `b` in the `B` total samples, calculate the statistic of interest.
3. Use the distribution of those `b` statistics to evaluate the estimate (`mean`) and precision (`quantile` or `sd`)!

We get sample of 1000 respondents from the PA voter file.

- Assume voter file `pa.sample` is truth and our sample `my.sample` is an estimate!

```
n.samplesize <- 1000
my.sample <- sample_n(pa.sample,n.samplesize,replace= TRUE)  # draw a sample of `n.samplesize` from `pa

mean(my.sample$likely.dem)*100 # Estimated percentage of Democrats in your 1000 person sample
```

```
## [1] 51
```

```
mean(pa.sample$likely.dem)*100 # Truth
```

```
## [1] 47.62857
```

What is the squared error?

```
(mean(my.sample$likely.dem)*100 - mean(pa.sample$likely.dem)*100)^2  # Error
```

```
## [1] 11.36656
```

Questions you can answer using resampling

- Probability the percentage of likely Democrats is greater than 40%?
- Probability the percentage of likely Democrats in sample is greater than 3.5 times the percentage of Independents?
- Probability that the percentage of likely Democrats in in PA is 10 percentage points larger than the percentage of Republicans?

```
# Fix Sample Size, Vary number of Samples
B <- 1000

bootstrap1000 <- NULL
for(i in 1:B){
  bootstrap1000<- my.sample %>%
    sample_n(nrow(my.sample), replace= TRUE) %>%
    summarize(MeanDem = mean(likely.dem)*100,
              MeanRep = mean(likely.rep)*100,
              MeanInd = mean(likely.ind)*100,
              MeanDemgt40 = ifelse(MeanDem > 40,1,0),
              MeanDemgtRep = ifelse(MeanDem > MeanRep,1,0),
              Dem3Ind = ifelse(MeanDem > 3.5*MeanInd,1,0),
              DR10diff = ifelse(MeanDem-MeanRep > 10,1,0)) %>%
    bind_rows(bootstrap1000)
}
```

Now lets analyze the `bootstrap1000` tibble to get the results! (Note we could look at the quantiles as well to estimate the precision of these point estimates!)

```
bootstrap1000 %>%
  summarize(MeanDemGT40 = mean(MeanDemgt40),
            MeanDemGTRep = mean(MeanDemgtRep),
            Dem3Ind  = mean(Dem3Ind),
            DR10diff  = mean(DR10diff))
```

```
##   MeanDemGT40 MeanDemGTRep Dem3Ind DR10diff
## 1           1            1   0.985    0.911
```

Proportions as a measure of probability

- `rbinom` - how many 1's if we draw `n` samples of `size` observations when the probability of seeing a 1 is `prob` and the probability of seeing a 0 is `1-prob`.
- e.g., number of heads (out of 10) when flipping a fair coin 100 times is:

```
B <- 100
n.flips <- 10
prob.heads <- .5

fair.coin.sample <- rbinom(n=B, size=n.flips, prob=prob.heads)
fair.coin.sample
```

```
##   [1] 5 7 6 5 5 4 4 5 4 8 7 6 6 6 4 3 6 8 6 4 2 8 7 6 6 6 7 6 7 4 5 3 4 6 6 6 5
##  [38] 5 6 6 4 7 5 5 8 8 4 5 4 1 8 5 7 6 5 4 5 3 4 4 4 7 3 5 7 4 5 7 7 2 5 4 8 7
##  [75] 2 3 5 7 3 6 4 4 2 4 7 5 3 6 7 4 4 6 4 4 7 5 6 8 5 6
```

```
fair.coin.sample/n.flips
```

```
##   [1] 0.5 0.7 0.6 0.5 0.5 0.4 0.4 0.5 0.4 0.8 0.7 0.6 0.6 0.6 0.4 0.3 0.6 0.8
##  [19] 0.6 0.4 0.2 0.8 0.7 0.6 0.6 0.6 0.7 0.6 0.7 0.4 0.5 0.3 0.4 0.6 0.6 0.6
##  [37] 0.5 0.5 0.6 0.6 0.4 0.7 0.5 0.5 0.8 0.8 0.4 0.5 0.4 0.1 0.8 0.5 0.7 0.6
##  [55] 0.5 0.4 0.5 0.3 0.4 0.4 0.4 0.7 0.3 0.5 0.7 0.4 0.5 0.7 0.7 0.2 0.5 0.4
##  [73] 0.8 0.7 0.2 0.3 0.5 0.7 0.3 0.6 0.4 0.4 0.2 0.4 0.7 0.5 0.3 0.6 0.7 0.4
##  [91] 0.4 0.6 0.4 0.4 0.7 0.5 0.6 0.8 0.5 0.6
```

```
mean(fair.coin.sample > 7)
```

IN-CLASS: What is the probability that the number of heads is greater than 7?

```
## [1] 0.08
```

Extension to political analysis

- In the 2020 Democratic Primary, a candidate would only receive delegates to the Democratic Convention if they get at least 15%.
- What is the probability that a candidate will get a delegate?
- What if we lack the underlying individual level data?
- 1000 Samples of a poll of 579 respondents where the probability of Sanders Support is .14

```
B <- 1000
n.resp <- 579
prob.Sanders <- .14

SandersSupport <- rbinom(n=B, size=n.resp, prob=prob.Sanders)
summary(SandersSupport)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   55.00   75.00   81.00   81.08   87.00  109.00
```

```
SandersSupport <- SandersSupport/n.resp
summary(SandersSupport)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.09499 0.12953 0.13990 0.14004 0.15026 0.18826
```

***The Economist*/YouGov Poll**
November 3 - 5, 2019 - 1500 US Adult citizens

YouGov

51. Democratic candidate - first choice
If the Democratic presidential primary or caucus in your state were held today, who would you vote for?
Asked of registered voters who say they will vote in the Democratic Presidential primary or caucus in 2020

		Gender		Age (4 category)				Race (4 category)			
	Total	Male	Female	18-29	30-44	45-64	65+	White	Black	Hispanic	Other
Joe Biden	26%	27%	24%	10%	22%	30%	35%	20%	42%	35%	*
Elizabeth Warren	25%	25%	25%	26%	22%	26%	25%	30%	17%	9%	*
Bernie Sanders	14%	14%	15%	27%	24%	9%	3%	13%	14%	21%	*
Pete Buttigieg	8%	7%	8%	4%	3%	6%	16%	10%	0%	6%	*
Kamala Harris	6%	4%	8%	5%	8%	8%	3%	6%	7%	6%	*
Julian Castro	3%	3%	3%	2%	6%	2%	1%	1%	5%	7%	*
Tulsi Gabbard	3%	4%	1%	2%	1%	3%	3%	4%	1%	0%	*
Cory Booker	2%	1%	2%	1%	4%	2%	1%	1%	3%	3%	*
Amy Klobuchar	2%	2%	2%	1%	1%	2%	3%	2%	0%	2%	*
Marianne Williamson	1%	2%	0%	2%	1%	1%	0%	1%	1%	0%	*
Steve Bullock	1%	1%	0%	3%	1%	0%	0%	1%	0%	2%	*
Tom Steyer	1%	1%	0%	2%	0%	1%	1%	1%	0%	0%	*
Andrew Yang	1%	1%	1%	1%	2%	0%	0%	1%	0%	0%	*
John Delaney	1%	1%	1%	1%	1%	1%	0%	0%	0%	6%	*
Michael Bennet	0%	1%	0%	2%	0%	0%	0%	0%	2%	0%	*
Wayne Messam	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	*
Joe Sestak	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	*
Not sure	6%	5%	7%	7%	3%	5%	9%	6%	5%	0%	*
I would not vote	2%	1%	3%	3%	1%	3%	1%	2%	2%	3%	*
Totals	102%	100%	100%	99%	100%	99%	101%	99%	99%	100%	*
Unweighted N	(579)	(251)	(328)	(110)	(102)	(243)	(124)	(371)	(106)	(74)	(28)

Figure 1: YouGov Polling Results

```
sum(as.logical(SandersSupport > .15)/n.resp)
```

```
## [1] 0.4611399
```

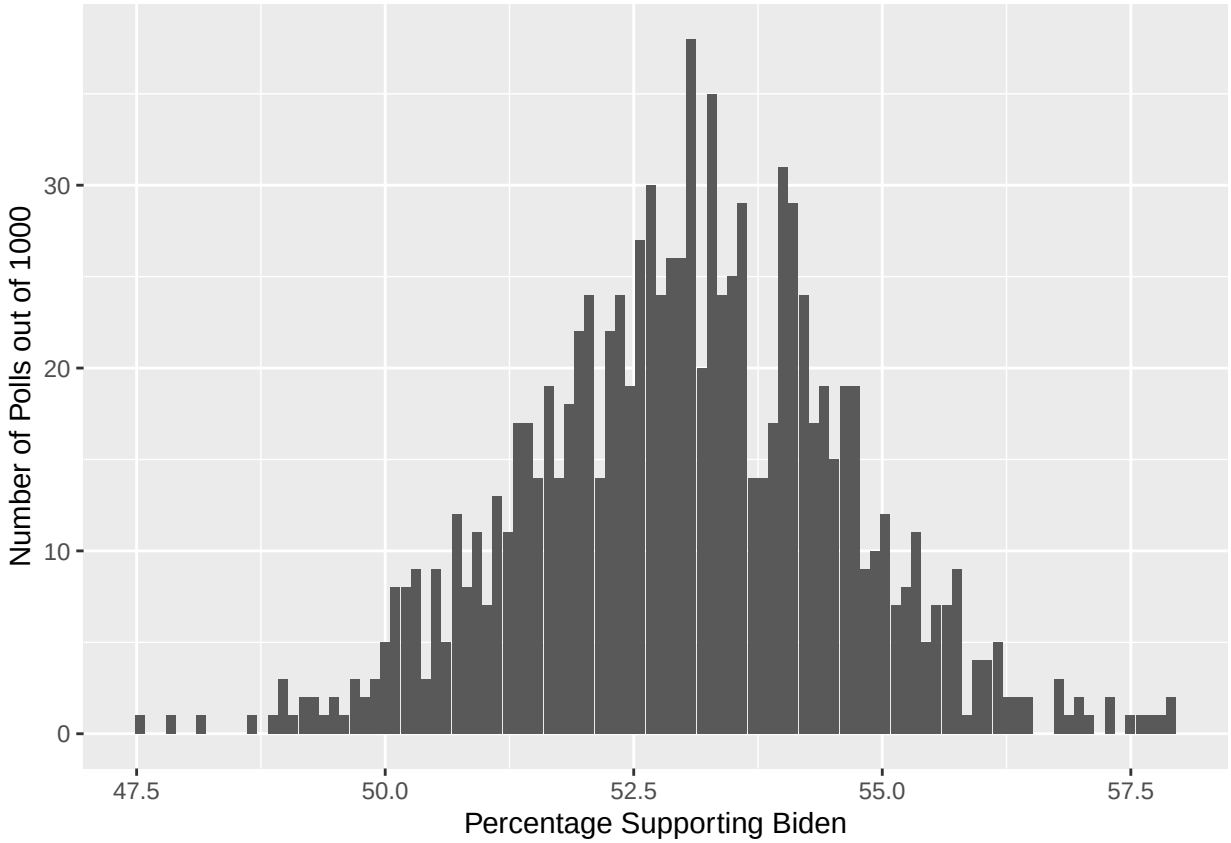
- But we can use a similar process for the presidential polls!
- Resampling to characterize the "margin of error" (due to random sampling):

```
load(file="data/Pres2020.PV.Rdata")
election.day <- as.Date("11/3/2020", "%m/%d/%Y")
Pres2020.PV <- Pres2020.PV %>%
               mutate(EndDate = as.Date(Pres2020.PV$EndDate, "%m/%d/%Y"),
                      StartDate = as.Date(Pres2020.PV$StartDate, "%m/%d/%Y"),
                      DaysToED = as.numeric(election.day - EndDate),
                      margin = Biden - Trump,
                      pollnum = as.numeric(rownames(Pres2020.PV)),
                      BidenError = Biden - DemCertVote,
                      TrumpError = Trump - RepCertVote,
                      SignedError = (Biden-Trump) - (DemCertVote - RepCertVote))
```

- Select 2nd poll (because they actually reported a Margin of Error!)
- Bootstrap the support for Biden 1000 times to see how support for Biden for that poll might vary

```
dat <- Pres2020.PV[2,]
r_sampleBiden <- rbinom(n = 1000, size = dat$SampleSize, prob = dat$Biden/100)/dat$SampleSize
r_sampleBiden <- tibble(r_sampleBiden)  # Make the vector a tibble for ggplot
```

```
r_sampleBiden %>%
  mutate(BidenPct = 100*r_sampleBiden) %>%
    ggplot() +
  geom_bar(aes(x=BidenPct)) +
  xlab("Percentage Supporting Biden") +
  ylab("Number of Polls out of 1000")
```




```
r_sampleBiden %>%
  summarize(pct05 = 100*quantile(r_sampleBiden,.05),
            pct95 = 100*quantile(r_sampleBiden,.95))
```

```
## # A tibble: 1 x 2
##   pct05 pct95
##   <dbl> <dbl>
## 1  50.3  55.6
```

Can you do the same for Trump? How does this compare to the reported "Margin of Error" in the poll?

```
Pres2020.PV$MoE[2]
```

```
## [1] 3.1
```

```
r_sampleBiden %>%
  summarize(MoE = 100*(mean(r_sampleBiden) - quantile(r_sampleBiden,.025)))
```

```
## # A tibble: 1 x 1
##     MoE
```

```
##   <dbl>
## 1  3.02
```

Applying to the universe of polls to characterize racewide uncertainty

Instead of doing it poll-by-poll we can also look at the uncertainty around the distribution of poll results altogether!

- How much does the overall margin vary if we were to redo all of the polls? (What are we assuming here!)
- What is the probability that Biden is leading?
- What is the probability of a tie?
- What is the probability that the poll results exactly predicts the percentage of certified vote a candidate receives?

```
B <- 1000

bootstrap1000 <- NULL
n.polls <- nrow(Pres2020.PV)

for(i in 1:B){

  bootstrap1000 <- Pres2020.PV %>%
    sample_n(n.polls, replace=TRUE) %>%
    summarize(MeanMargin = mean(margin),
            BidenWin = mean(Biden > Trump),
            TrumpWin = mean(Trump > Biden),
            Tie = mean(Biden == Trump),
            CorrectBidenPred = mean(Biden == DemCertVote),
            CorrectTrumpPred = mean(Trump == RepCertVote)) %>%
      bind_rows(bootstrap1000)
}
```

Use this to get an estimate for each quantity.

```
bootstrap1000 %>% summarize(MeanMargin = mean(MeanMargin),
                            ProbBidenWin = mean(BidenWin),
                            ProbTie = mean(Tie),
                            ProbTrumpWin = mean(TrumpWin),
                            ProbBidenCorrect = mean(CorrectBidenPred),
                            ProbTrumpCorrect = mean(CorrectTrumpPred))
```

```
## # A tibble: 1 x 6
##   MeanMargin ProbBidenWin ProbTie ProbTrumpWin ProbBidenCorrect ProbTrumpCorrect
##        <dbl>        <dbl>   <dbl>        <dbl>            <dbl>            <dbl>
## 1       8.02        0.985 0.00571      0.00941            0.128           0.0153
```

But can also get a range of uncertainty for each of these estimates!

```
bootstrap1000 %>%
  summarize(
    Margin = quantile(MeanMargin,seq(0,1,by=.1)),
    PctBidenPollsCorrect = quantile(CorrectBidenPred,seq(0,1,by=.1))) %>%
        mutate(Percentile = seq(0,100,by=10))
```

```
## # A tibble: 11 x 3
##    Margin PctBidenPollsCorrect Percentile
##     <dbl>                <dbl>      <dbl>
##  1   7.57               0.0777          0
##  2   7.83               0.110          10
##  3   7.89               0.117          20
##  4   7.95               0.121          30
##  5   7.98               0.125          40
##  6   8.02               0.129          50
##  7   8.06               0.133          60
##  8   8.10               0.136          70
##  9   8.14               0.140          80
## 10   8.21               0.148          90
## 11   8.48               0.172         100
```

Other questions/quantities:

- What is the probability that the polling average over-predicts candidate support vs. under-predicts candidate support. Symmetric around "Truth"?
- How *much* are the polls off, if at all, for each candidate?
- What does this say about the role of random sampling and polls?

Does the average polling error error:

- Vary by proximity to Election Day?
- Vary by mode of interviewing? (e.g, Online vs. Phone - RDD)
- By pollster?
- Vary by sample size of poll?

ADVANCED: Do you get a different answer using the empirical bootstrap for `margin` vs. the empirical bootstrap for `Biden` and `Trump` separately and differencing the bootstrapped samples for each? Why?

REALLY ADVANCED: We are treating `Biden` and `Trump` as fixed but we know each value is an estimate from a poll! Should we account for the uncertainty in individual polls as well? Can we given available data? How?

REALLY ADVANCED: What if we had 3 polls and they all had Biden up 49-48? Doing what we just did and resampling the three polls and counting the times Biden is leading would give a 100% probability of a Biden win. Does it make sense to go from a 1 point lead to a 100% probability? What is going on?