Resampling (& Pollling)

Josh Clinton

9/27/2021

Motivating questions:

- What is *random sampling* and why is random sampling so powerful as a method of collecting data?
- Why can the opinions of 1000 people be used to measure the public opinion of 350 million people?
- If my data is a random sample, how many observations do I need to calculate an accurate average?
- How much better is a random sample of size 1000 than a sample of size 100? In what ways?

Also, quantifying uncertainty is critical for science

- Without knowing how much your results may change it is hard to describe, predict, or understand relationships.
- Lots of ways that uncertainty/error can arise! Data, Modelling, User Error, ...
- Focus on on "best case" - what if our data is a random sample from the population of interest, how much would the results change if we did everything we just did again? Variation due to *random sampling*!
- NOTE: Very rarely are we in this condition (hence the need for statistical modelling).

Some Simple Sampling

- The function `sample(X, Y, replace = TRUE, prob = P)` will sample `Y` units from a vector `X` with or without replacement (`replace = TRUE` or `replace = FALSE`) using a vector of probability `P` (the default is equal probability).

```
Z <- seq(1,8)

set.seed(42)
## Randomly draw 8 samples from Z, with replacement
sample(Z, 8, replace = TRUE)
## Randomly draw 8 samples from Z, without replacement
sample(Z, 8, replace = FALSE)
## Randomly draw 8 samples from Z, with replacement
sample(Z, 8, replace = TRUE, prob = Z/sum(Z))
```

Some Simple Sampling

```
Z <- seq(1,8)
set.seed(42)
## Randomly draw 8 samples from Z, with replacement
sample(Z, 8, replace = TRUE)
## [1] 1 5 1 1 2 4 2 2
## Randomly draw 8 samples from Z, without replacement
sample(Z, 8, replace = FALSE)
## [1] 1 7 4 8 5 2 3 6
## Randomly draw 8 samples from Z, with replacement
sample(Z, 8, replace = TRUE, prob = Z/sum(Z))
## [1] 6 3 8 1 2 8 6 7
```

Calculating Probability through Simulation

- Probability can be thought of as the "limit" of repeated identical experiments.
- Use loops to repeat an experiment and calculate the probability of certain events.

Birthday Problem

- How many people do you need for the probability that at least two people have the same birthday exceeds 0.5?

Birthday Problem

- How many people do you need for the probability that at least two people have the same birthday exceeds 0.5?

  How would you study this using a simulation?

Two Functions for vectors

- `unique`: the number of unique values in a vector

```
values <- c(1,2,3,3)
length(values)
```

```
## [1] 4
```

Two Functions for vectors

- `unique`: the number of unique values in a vector

```
values <- c(1,2,3,3)
length(values)
```

```
## [1] 4
```

- `length`: the length of a vector

```
unique(values)
```

```
## [1] 1 2 3
```

Birthday Problem: Solving via Simulation

```
sims <- 10000 ## number of sims
bday <- 1:365 ## possible bdays
answer <- NULL ## placeholder

for (k in 1:25) {
  count <- 0  ## counter
  for (i in 1:sims) {
    class <- sample(bday, k, replace = TRUE)

    if (length(unique(class)) < length(class)) {
      count <- count + 1
    }}
  ## printing the estimate
  cat("The estimated probability for", k,"people is:",
      count/sims, "\n")
  answer[k] <- count/sims # store the answers
}
```

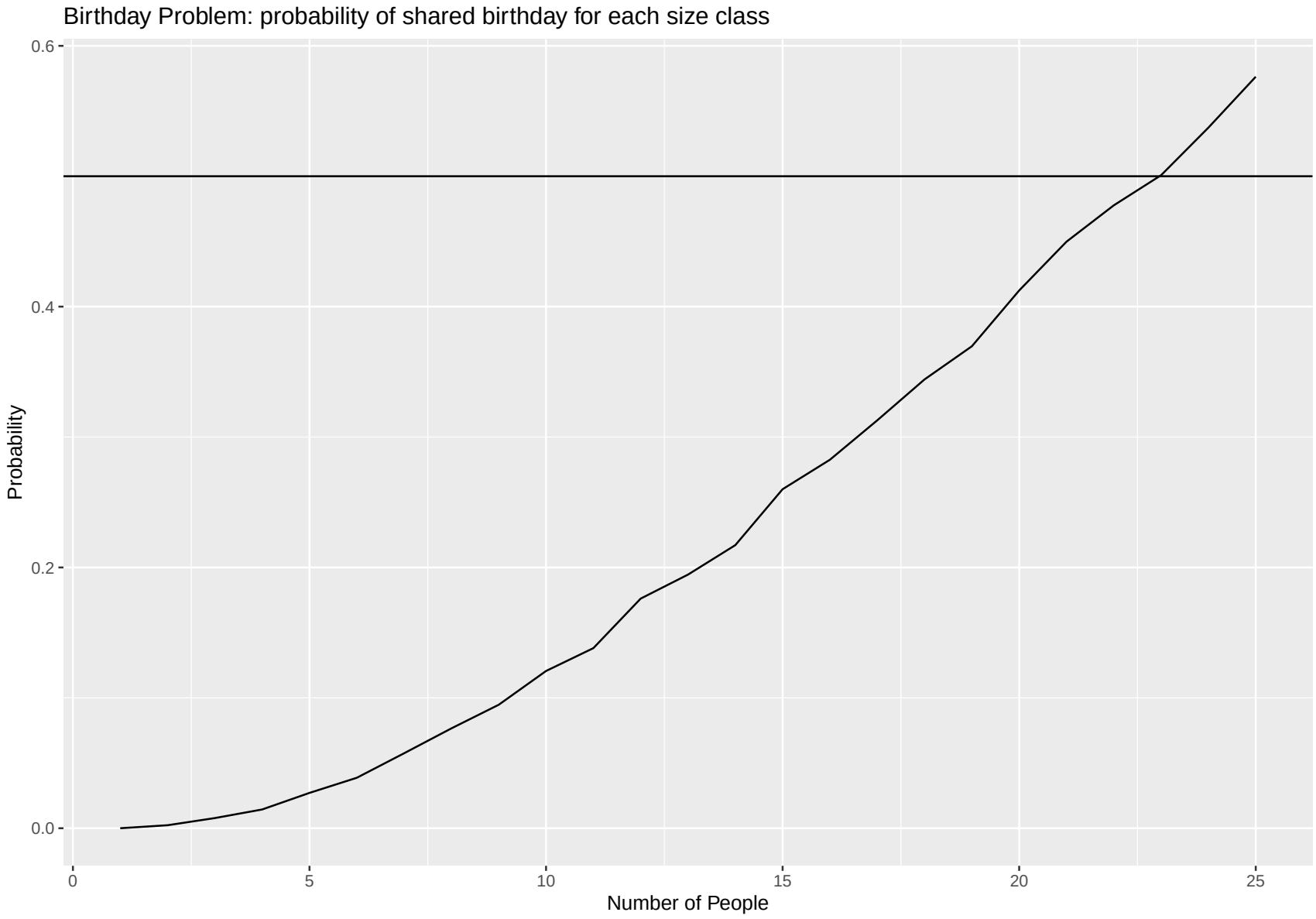
Birthday Problem

```
## The estimated probability for 1 people is: 0
## The estimated probability for 2 people is: 0.0023
## The estimated probability for 3 people is: 0.0078
## The estimated probability for 4 people is: 0.0144
## The estimated probability for 5 people is: 0.0271
## The estimated probability for 6 people is: 0.0387
## The estimated probability for 7 people is: 0.0575
## The estimated probability for 8 people is: 0.0766
## The estimated probability for 9 people is: 0.0947
## The estimated probability for 10 people is: 0.1206
## The estimated probability for 11 people is: 0.1381
## The estimated probability for 12 people is: 0.1761
## The estimated probability for 13 people is: 0.1945
## The estimated probability for 14 people is: 0.2171
## The estimated probability for 15 people is: 0.26
## The estimated probability for 16 people is: 0.2826
## The estimated probability for 17 people is: 0.3128
## The estimated probability for 18 people is: 0.3442
```

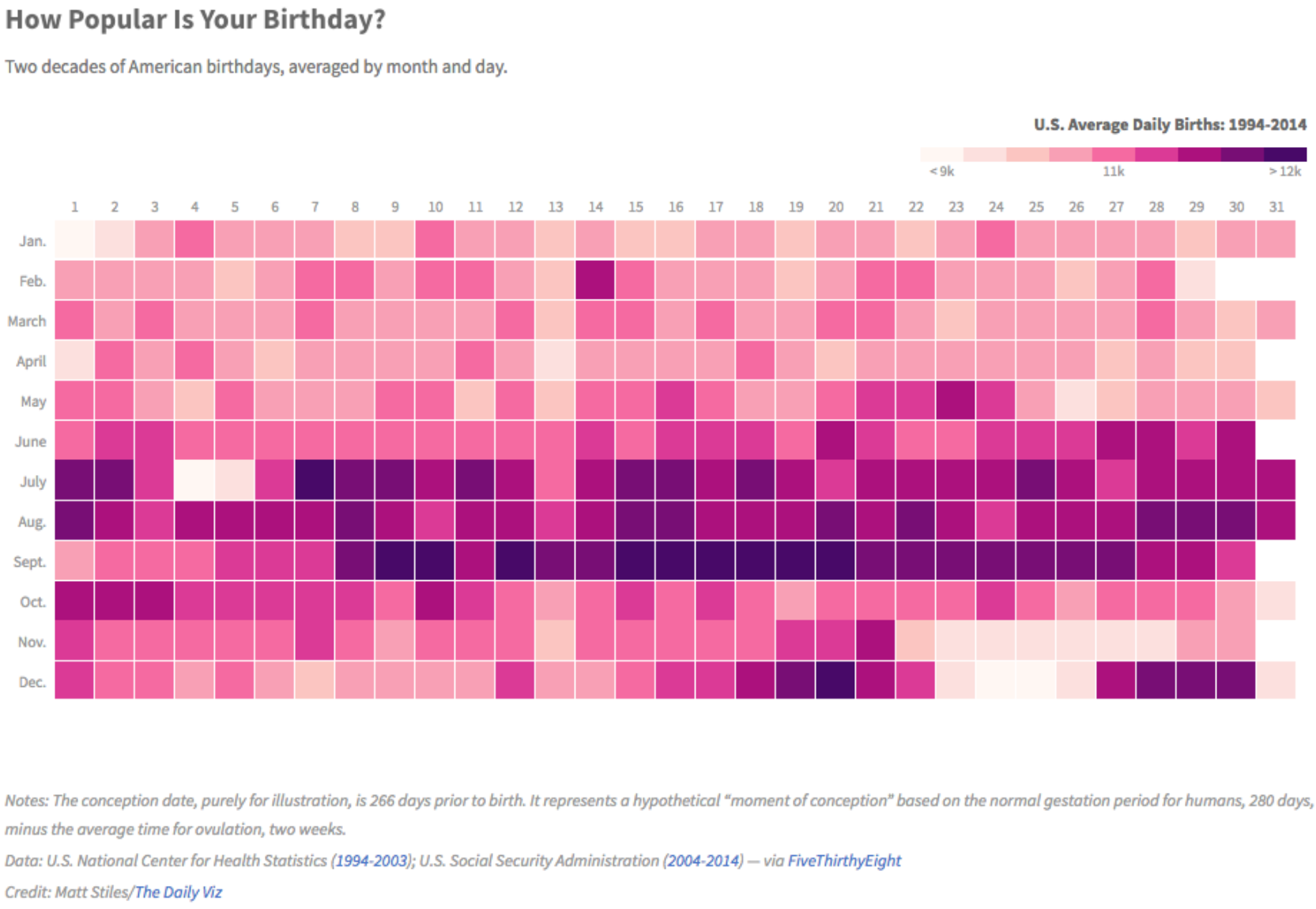
Birthday Problem: Visualize

```
dat <- bind_cols(npeople=seq(1,25),answer)

ggplot(dat,aes(x=npeople,y=answer)) +
  geom_line() +
  labs(x = "Number of People") +
  labs(y="Probability") +
  labs(title = "Prob. of shared bday for each size") +
  geom_hline(yintercept=.5)
```

Birthday Problem: Visualize



But Uniform (Equal) Probability of Birth?



from https://github.com/fivethirtyeight/data/tree/master/births

Voter Files

Many political parties, interest groups, and media organizations rely on voter files to predict elections.

Let's go!

```
load("data/pa.sample.select.Rdata")
glimpse(pa.sample)
```

```
## Rows: 98,548
## Columns: 16
## $ city              <chr> "Aliquippa", "Aliquippa", "Al
## $ likely.party      <fct> D, D, D, R, D, R, R, D, R, D,
## $ female            <dbl> 1, 0, 1, 1, 0, 1, 0, 0, 0, 0,
## $ AgeUnder30        <dbl> 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
## $ Age3039           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
## $ Age4049           <dbl> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
## $ Age5059           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
## $ Age6074           <dbl> 0, 1, 1, 0, 1, 0, 0, 0, 1, 0,
## $ Age75p            <dbl> 1, 0, 0, 0, 0, 1, 0, 1, 0, 0,
## $ imputed.white     <dbl> 1, 1, 1, 1, 0, 0, 0, 1, 1, 0,
## $ imputed.black     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
## $ imputed.hispanic  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
## $ fips.county       <chr> "42007", "42007", "42007", "42
## $ RatDemQtyVote2020 <dbl> 41.04, 41.04, 41.04, 41.04, 41
```

Partisanship

```
pa.sample %>%
  count(likely.party)
```

```
##   likely.party     n
## 1            D 46937
## 2            I 12830
## 3            R 38781
```

Mutate!

```
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))
```

Mutate!

```
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))
```

```
pa.sample %>%
  count(likely.dem)
```

```
##   likely.dem     n
## 1          0 51611
## 2          1 46937
```

Data Wrangling: Transmute?

```
pa.sample.new <- pa.sample %>%
    transmute(likely.dem = ifelse(likely.party == "D",1,0),
              likely.rep = ifelse(likely.party == "R",1,0),
              likely.ind = ifelse(likely.party == "I",1,0))
```

Data Wrangling: Transmute?

```
pa.sample.new <- pa.sample %>%
    transmute(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.new)
```

```
## Rows: 98,548
## Columns: 3
## $ likely.dem <dbl> 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1
## $ likely.rep <dbl> 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0
## $ likely.ind <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
```

Summarize

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

PA.pty.breakdown
```

```
##      pct.dem  pct.rep   pct.ind
## 1 0.4762857 0.393524 0.1301904
```

sample_n

- `sample` is a "base R" function that samples from a vector
- `sample_n` is a `tidyverse` version that samples rows from a tibble.

NOTE: need `dplyr` for `sample_n`

Using sample_n

```
#library(dplyr)
sample_n(pa.sample,5,replace= TRUE)
```

```
##             city likely.party female AgeUnder30 Age3039 A
## 1          Erie             D      1          0       1
## 2    Pittsburgh             D      0          0       1
## 3 Connellsville             D      0          0       0
## 4 Elizabethtown             D      0          0       0
## 5  Philadelphia             D      1          0       0
##   Age75p imputed.white imputed.black imputed.hispanic fi
## 1      0             1             0                 0
## 2      0             0             1                 0
## 3      0             0             0                 0
## 4      0             1             0                 0
## 5      0             1             0                 0
##   PctDemCtyVote2020 PctDemCtyVote2016 CooperateSurvey li
## 1             50.52             49.17               NA
## 2             60.36             58.62               NA
```

Looping

```
samplesize <- seq(1,10)

for(i in seq_along(samplesize)){
  rand.dat <- sample_n(pa.sample, i , replace= TRUE)
}

dim(rand.dat)
```

```
## [1] 10 19
```

- What is `rand.dat`?
- How big is `rand.dat` at each `i`?

Using Samples in a loop

```
dem.pty.est <- NULL
samplesize <- c(3,10,34,567,4762)

for(i in seq_along(samplesize)){
  rand.dat <- sample_n(pa.sample,
                       samplesize[i],replace= TRUE)

  dem.pty.est[i] <- mean(rand.dat$likely.dem)
}
```

Using Samples in a loop

```
dem.pty.est <- NULL
samplesize <- c(3,10,34,567,4762)

for(i in seq_along(samplesize)){
  rand.dat <- sample_n(pa.sample,
                       samplesize[i],replace= TRUE)

  dem.pty.est[i] <- mean(rand.dat$likely.dem)
}
```

```
dem.pty.est
```

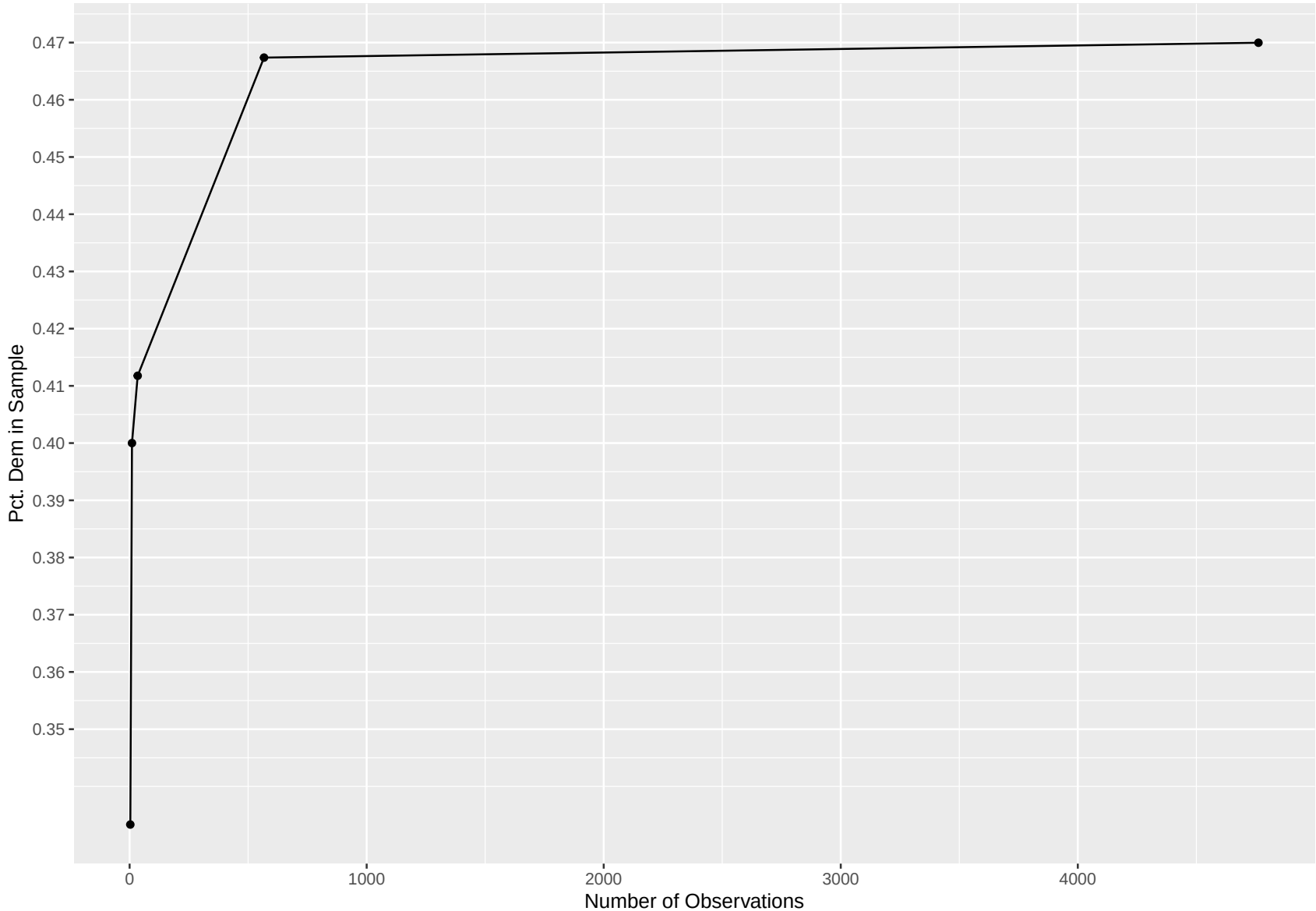
```
## [1] 0.3333333 0.4000000 0.4117647 0.4673721 0.4699706
```

Sample Size and percentage?

```
dat <- bind_cols(samplesize,dem.pty.est)

pasample.plot <- dat %>%
  ggplot(aes(x=samplesize,y=dem.pty.est)) +
  geom_line() +
  geom_point() +
  scale_y_continuous(breaks=seq(.35,.48,by=.01)) +
  labs(x = "Number of Observations") +
  labs(y = "Pct. Dem in Sample")
```

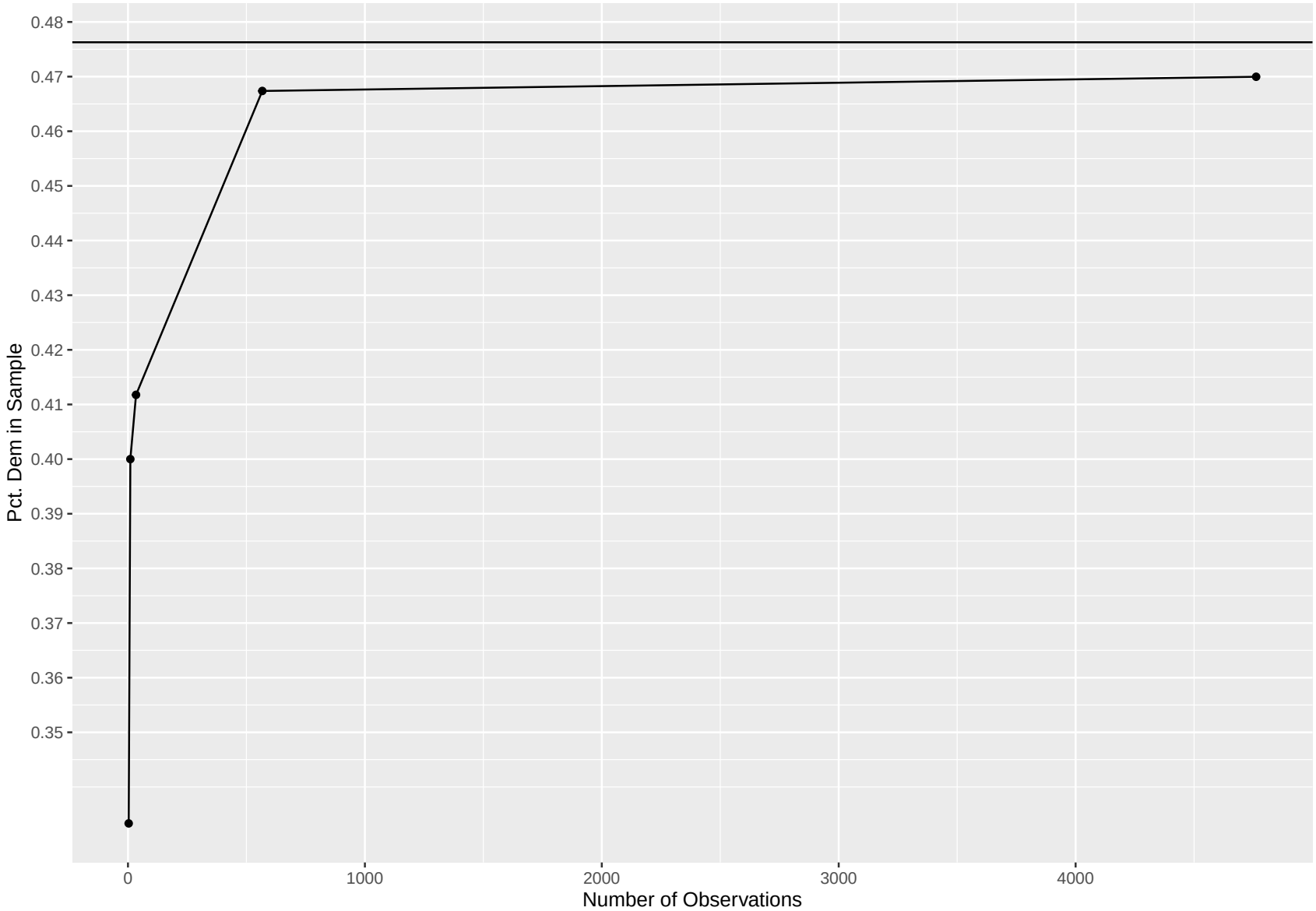
Sample Size and percentage?

```
pasample.plot
```



Context?

```
pasample.plot+geom_hline(yintercept=PA.pty.breakdown[[1]])
```

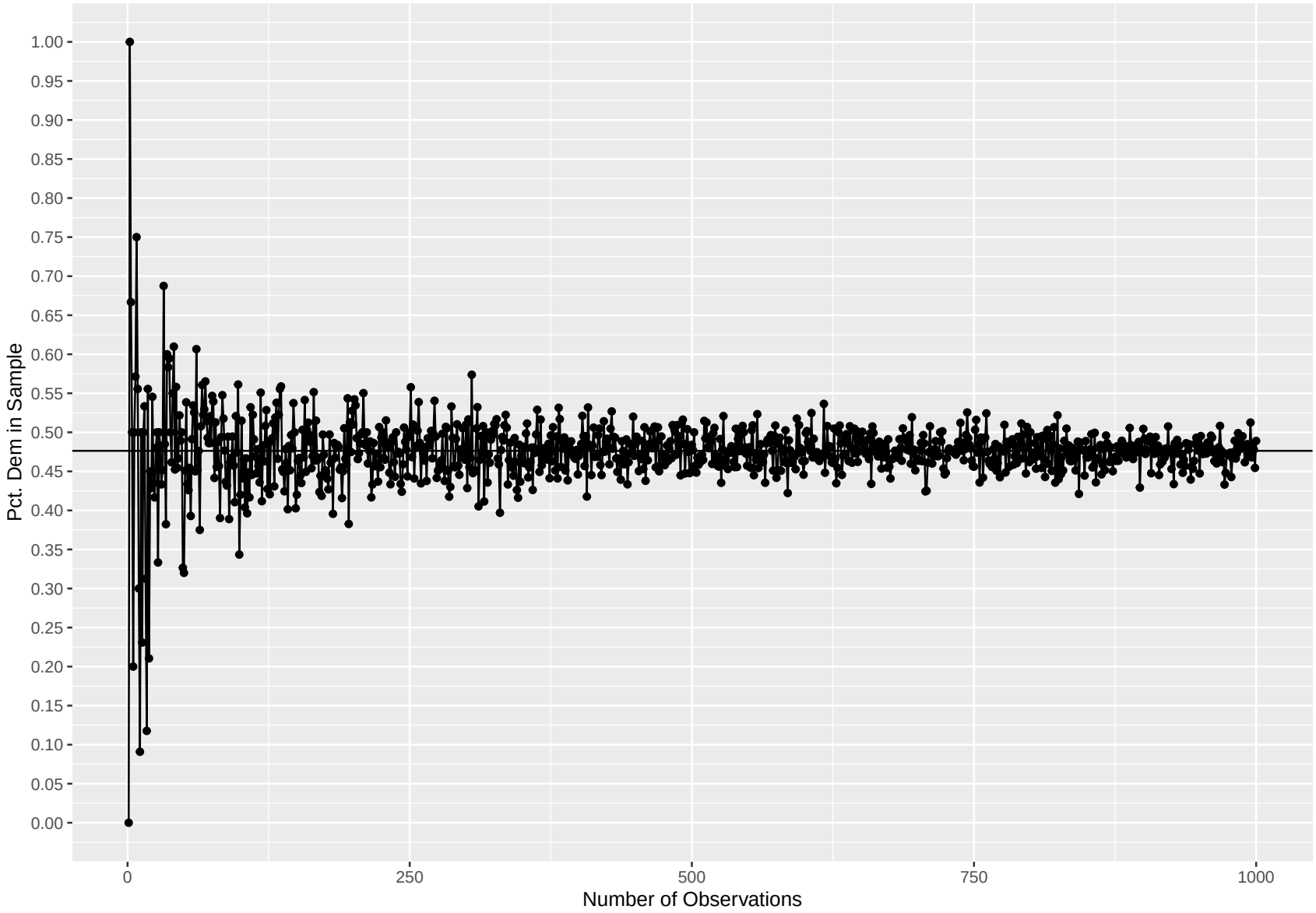


More N!

```
samplesize <- seq(1,1000)
for(i in seq_along(samplesize)){
  rand.dat <- sample_n(pa.sample,i,replace= TRUE)
  dem.pty.est[i] <- mean(rand.dat$likely.dem)
}

dat <- bind_cols(samplesize,dem.pty.est)

pasample.plot <- dat %>% ggplot(aes(x=samplesize,y=dem.pty
  geom_line() +
  geom_point() +
  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]])
```

More N!

```
pasample.plot
```

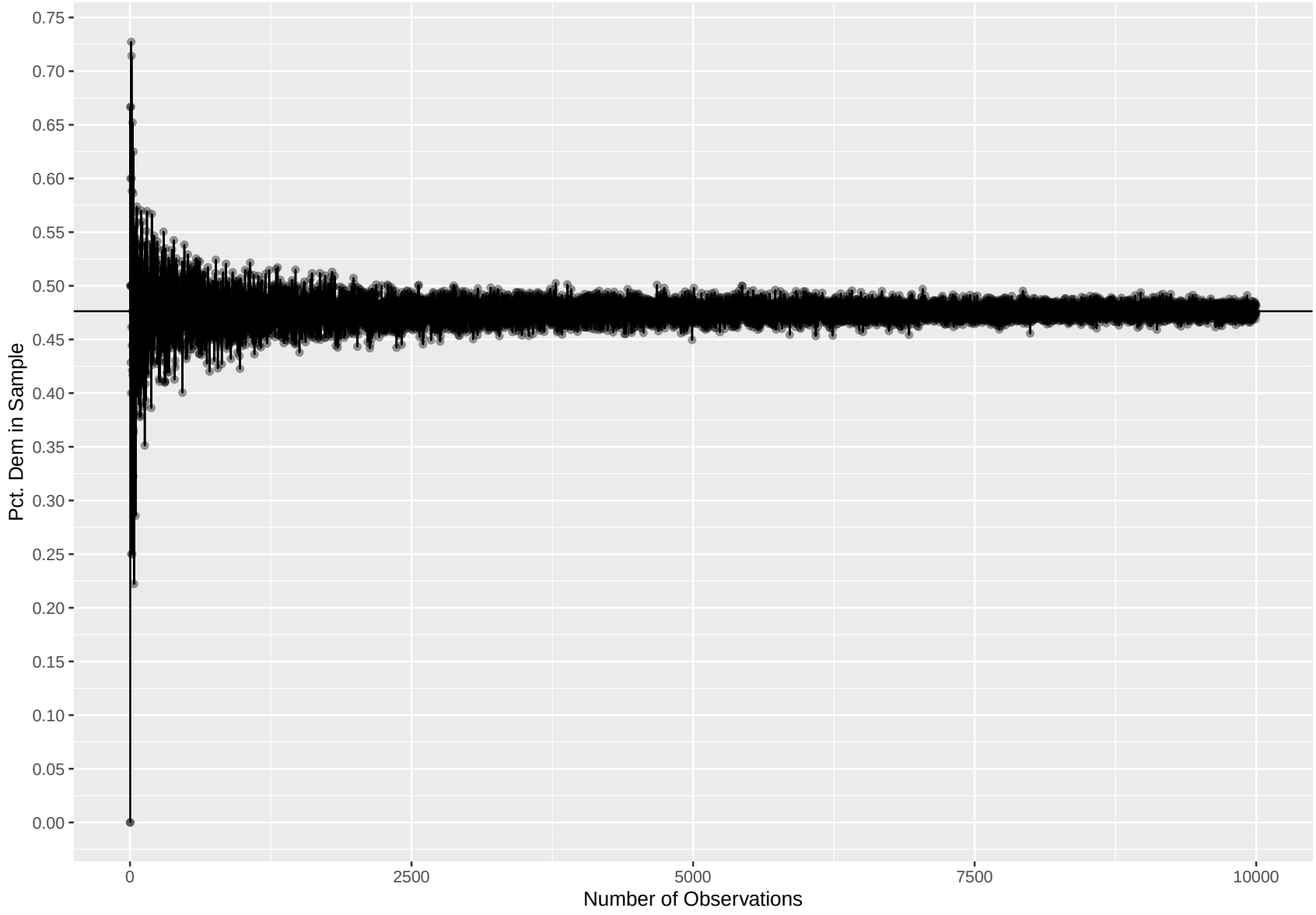


And More!

```
samplesize2 <- seq(1,10000)
dem.pty.est2 <- NULL

for(i in seq_along(samplesize2)){
  rand.dat <- sample_n(pa.sample,i,replace= TRUE)
  dem.pty.est2[i] <- mean(rand.dat$likely.dem)
}

dat <- bind_cols(samplesize2,dem.pty.est2)

pasample.plot <- dat %>% ggplot(aes(x=samplesize2,y=dem.pty
  geom_line() +
  geom_point(alpha=.4) +
  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]])
```

And More!

```
pasample.plot
```



So how big of a sample do we need?

- How does the precision of our estimate change as the sample size increases?
- How many data points do we need to give "accurate" estimates? (Assuming everything else is OK!)

Helpful function

- `cut`: takes a vector and creates a factor that labels groups based on dividing vector into equal sizes based on `breaks`

Helpful function

- `cut`: takes a vector and creates a factor that labels groups based on dividing vector into equal sizes based on `breaks`

```
x <- c(1,2,3,4,5,6)
cut(x,breaks=2)
```

```
## [1] (0.995,3.5] (0.995,3.5] (0.995,3.5] (3.5,6]     (3.5
## Levels: (0.995,3.5] (3.5,6]
```

Thinking about error

```
dat <- dat %>%
  mutate(Absolute.Error =
           abs(dem.pty.est2 - PA.pty.breakdown[[1]]),
         Squared.Error =
           (dem.pty.est2 - PA.pty.breakdown[[1]])^2,
         cut = cut(samplesize2,breaks = 10))

table(dat$cut)
```

```
##
##    (-9,1e+03] (1e+03,2e+03] (2e+03,3e+03] (3e+03,4e+03]
##          1000          1000          1000          1000
## (5e+03,6e+03] (6e+03,7e+03] (7e+03,8e+03] (8e+03,9e+03]
##          1000          1000          1000          1000
```

Absolute Error

```
dat %>%
  group_by(cut) %>%
  summarize(avgae = mean(Absolute.Error),
            sdae = sd(Absolute.Error))
```

```
## # A tibble: 10 x 3
##    cut               avgae    sdae
##    <fct>             <dbl>   <dbl>
##  1 (-9,1e+03]     0.0252  0.0344
##  2 (1e+03,2e+03] 0.0106  0.00809
##  3 (2e+03,3e+03] 0.00802 0.00610
##  4 (3e+03,4e+03] 0.00663 0.00495
##  5 (4e+03,5e+03] 0.00609 0.00462
##  6 (5e+03,6e+03] 0.00540 0.00429
##  7 (6e+03,7e+03] 0.00511 0.00392
##  8 (7e+03,8e+03] 0.00445 0.00349
##  9 (8e+03,9e+03] 0.00417 0.00314
## 10 (9e+03,1e+04] 0.00396 0.00310
```

Signed Error

```
dat %>%
  group_by(cut) %>%
  summarize(avgae = mean(Squared.Error),
            sdae = sd(Squared.Error))
```

```
## # A tibble: 10 x 3
##    cut                avgae       sdae
##    <fct>              <dbl>      <dbl>
##  1 (-9,1e+03]     0.00182   0.0112
##  2 (1e+03,2e+03] 0.000178  0.000258
##  3 (2e+03,3e+03] 0.000102  0.000145
##  4 (3e+03,4e+03] 0.0000685 0.0000943
##  5 (4e+03,5e+03] 0.0000584 0.0000820
##  6 (5e+03,6e+03] 0.0000475 0.0000717
##  7 (6e+03,7e+03] 0.0000415 0.0000610
##  8 (7e+03,8e+03] 0.0000320 0.0000472
##  9 (8e+03,9e+03] 0.0000272 0.0000377
## 10 (9e+03,1e+04] 0.0000253 0.0000370
```

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, is the data really a random sample? Is random sampling the largest source of error?

Amazing Result 2: Central Limit Theorem

So we know that a large random sample of data is expected to give a sample average close to the true average (Law of Large Numbers).

Amazing Result 2: Central Limit Theorem

So we know that a large random sample of data is expected to give a sample average close to the true average (Law of Large Numbers).

- But can we say anything about the distribution of the sample mean?
- i.e., What if we take repeated observations from the same data generating process, calculate the mean, and then look at the shape of the result histogram? What will the histogram of means look like?
- How does the answer depend on if our data is binary (0,1), categorical (e.g., 1,2,3,4), or continuous?

Amazing Result 2: Central Limit Theorem

So we know that a large random sample of data is expected to give a sample average close to the true average (Law of Large Numbers).

- But can we say anything about the distribution of the sample mean?
- i.e., What if we take repeated observations from the same data generating process, calculate the mean, and then look at the shape of the result histogram? What will the histogram of means look like?
- How does the answer depend on if our data is binary (0,1), categorical (e.g., 1,2,3,4), or continuous?
- Hard case? Distribution of the mean of a binary variaable: "likely Dem" (1), or not (0)

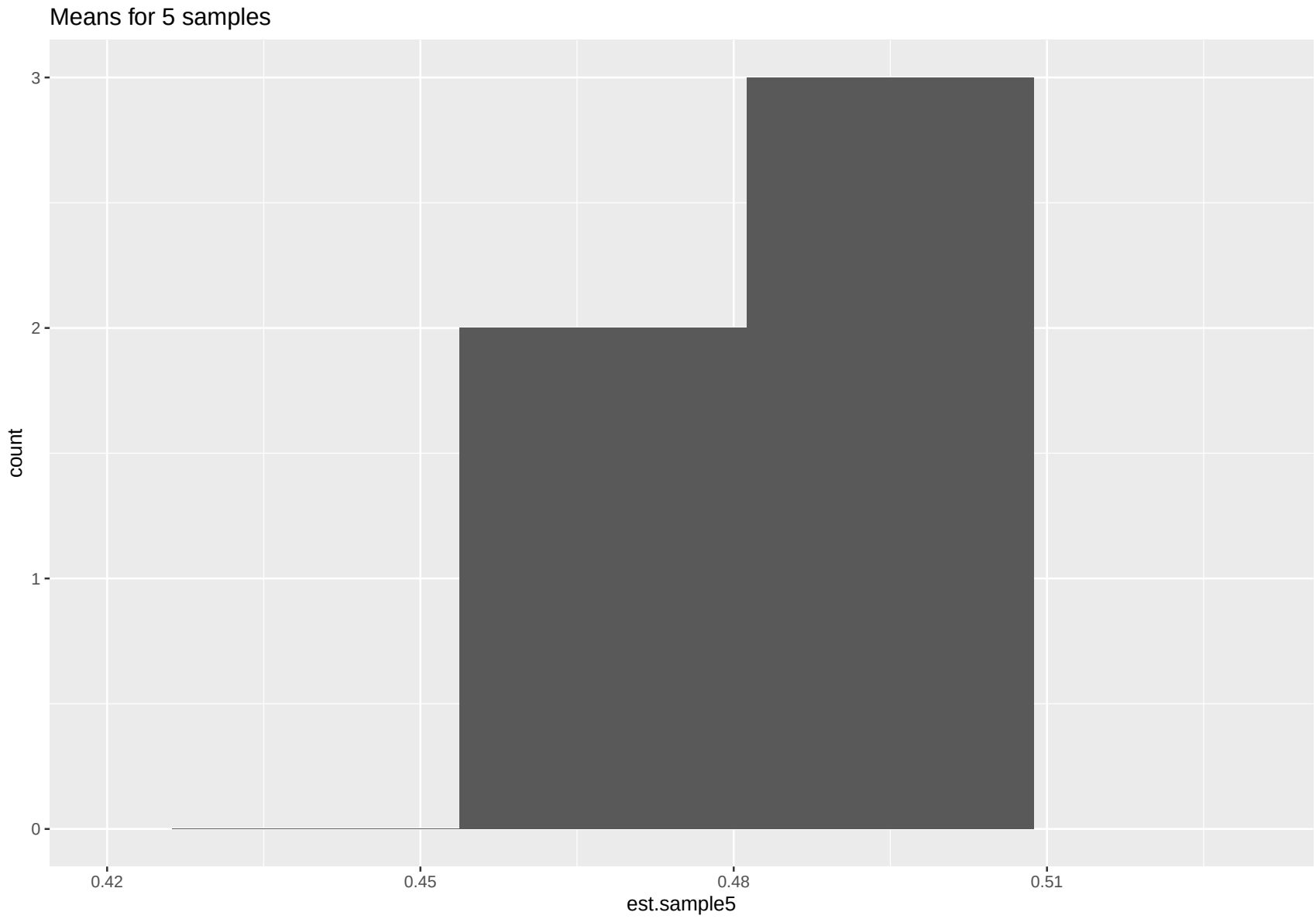
Resampling Means

```
# Fix Sample Size, Vary number of Samples
n.samplesize <- 1000
n.samples <- 5

est.sample5 <- NULL
for(i in 1:5){
    rand.dat <- sample_n(pa.sample,n.samplesize,replace= TR
    est.sample5[i] <- mean(rand.dat$likely.dem)
}

est.sample5 <- tibble(est.sample5)
```

Plot

```
est.sample5 %>% ggplot(aes(x=est.sample5)) +
  geom_histogram(bins=5) +
  labs(title="Means for 5 samples") +
  xlim(.42,.53)
```

Plot



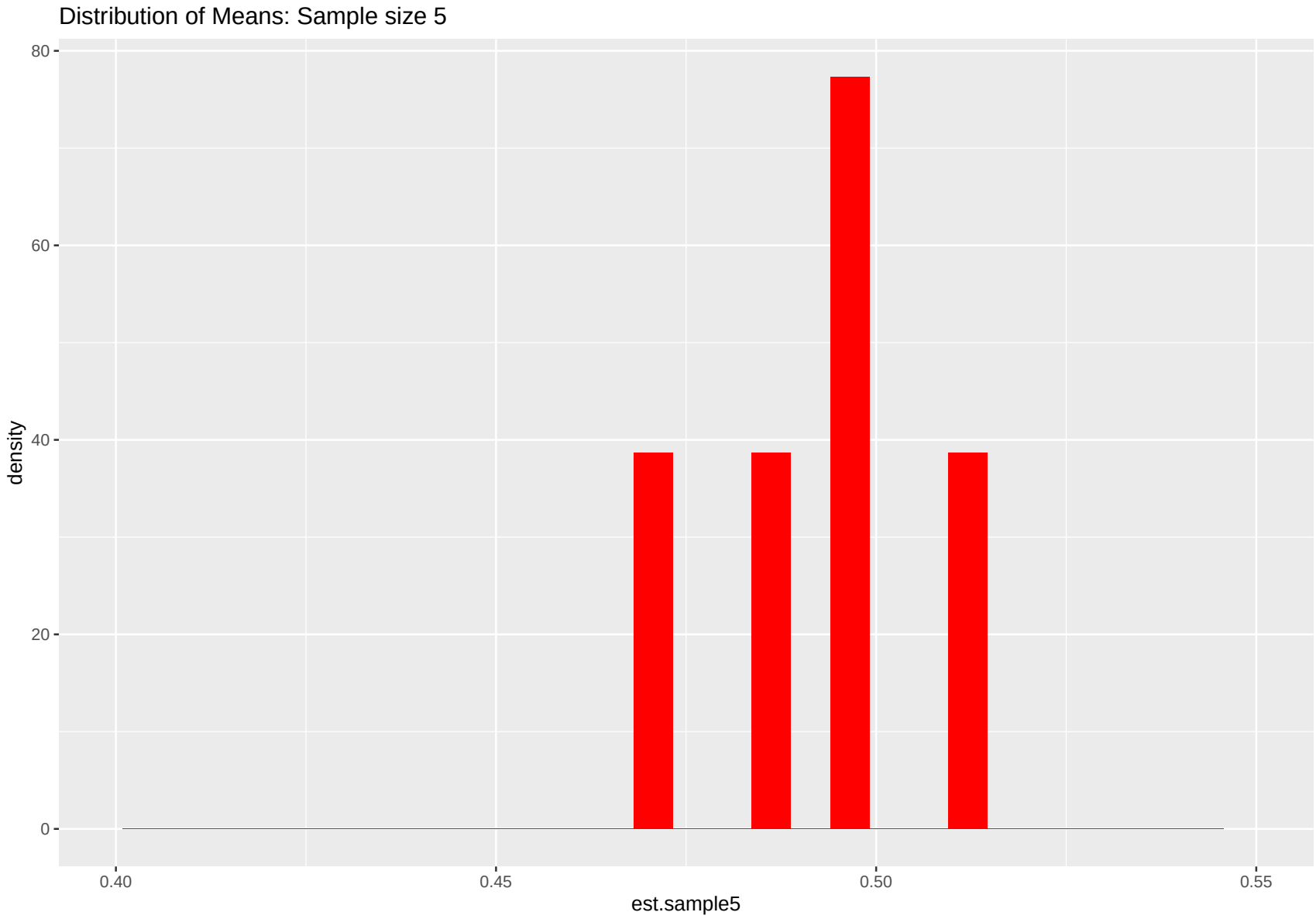
More Resampling

- Now lets consider the shape of the histogram of means of size 1000 samples that are done 5 times, 10, times, 20 times, 40 times, 80 times, and 160 times.
- Think of this as the number of polls being done each week.
- Or number of quality assurance tests being done on a product to test for manufacturing defects.

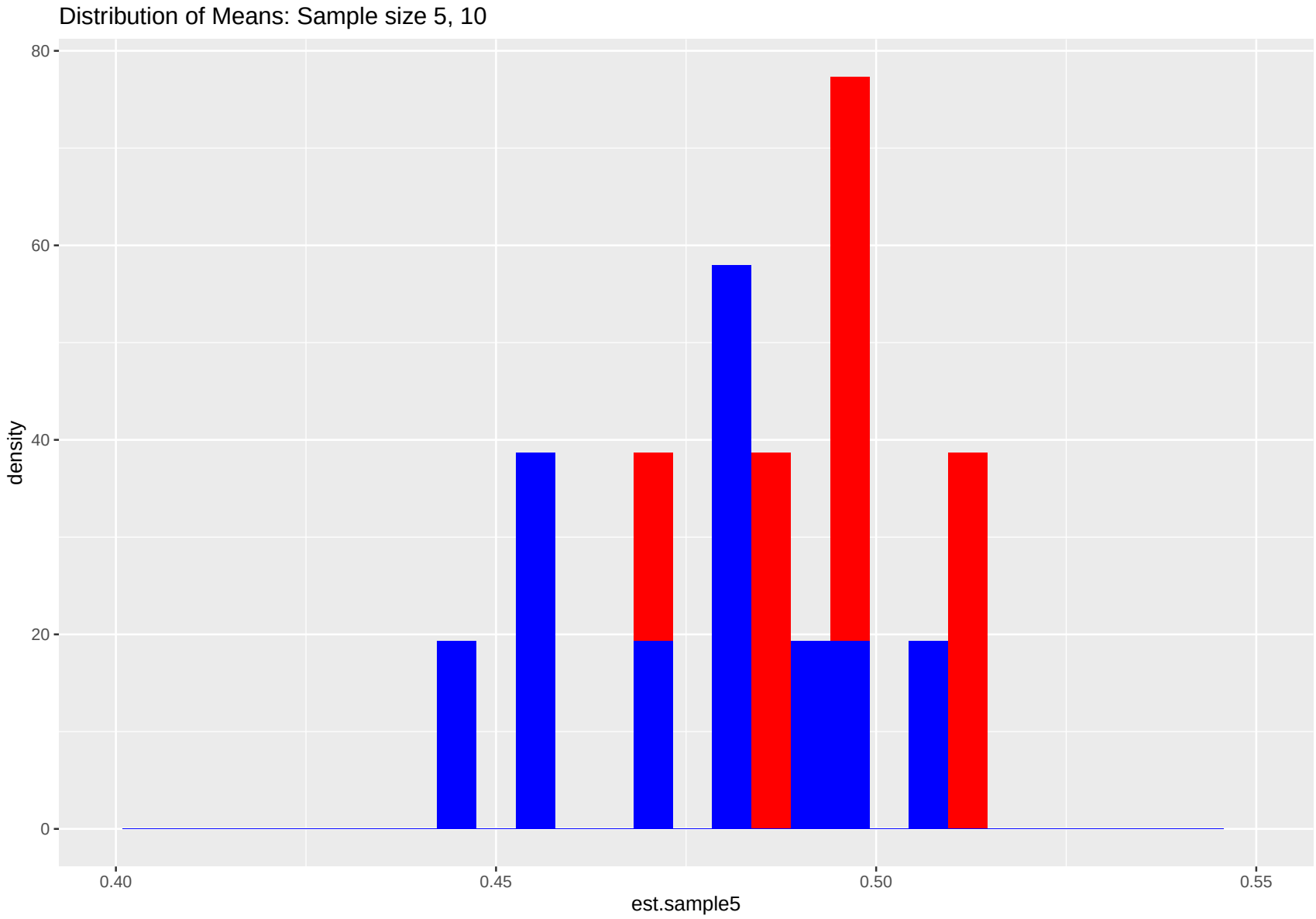
Clunky Sampling Code

```
est.sample5 <- NULL
for(i in 1:5){
    rand.dat <- sample_n(pa.sample,n.samplesize,replace= TR
    est.sample5[i] <- mean(rand.dat$likely.dem)
}

est.sample10 <- NULL
for(i in 1:10){
  rand.dat <- sample_n(pa.sample,n.samplesize,replace= TRU
  est.sample10[i] <- mean(rand.dat$likely.dem)
}

est.sample20 <- NULL
for(i in 1:20){
  rand.dat <- sample_n(pa.sample,n.samplesize,replace= TRU
  est.sample20[i] <- mean(rand.dat$likely.dem)
}
```

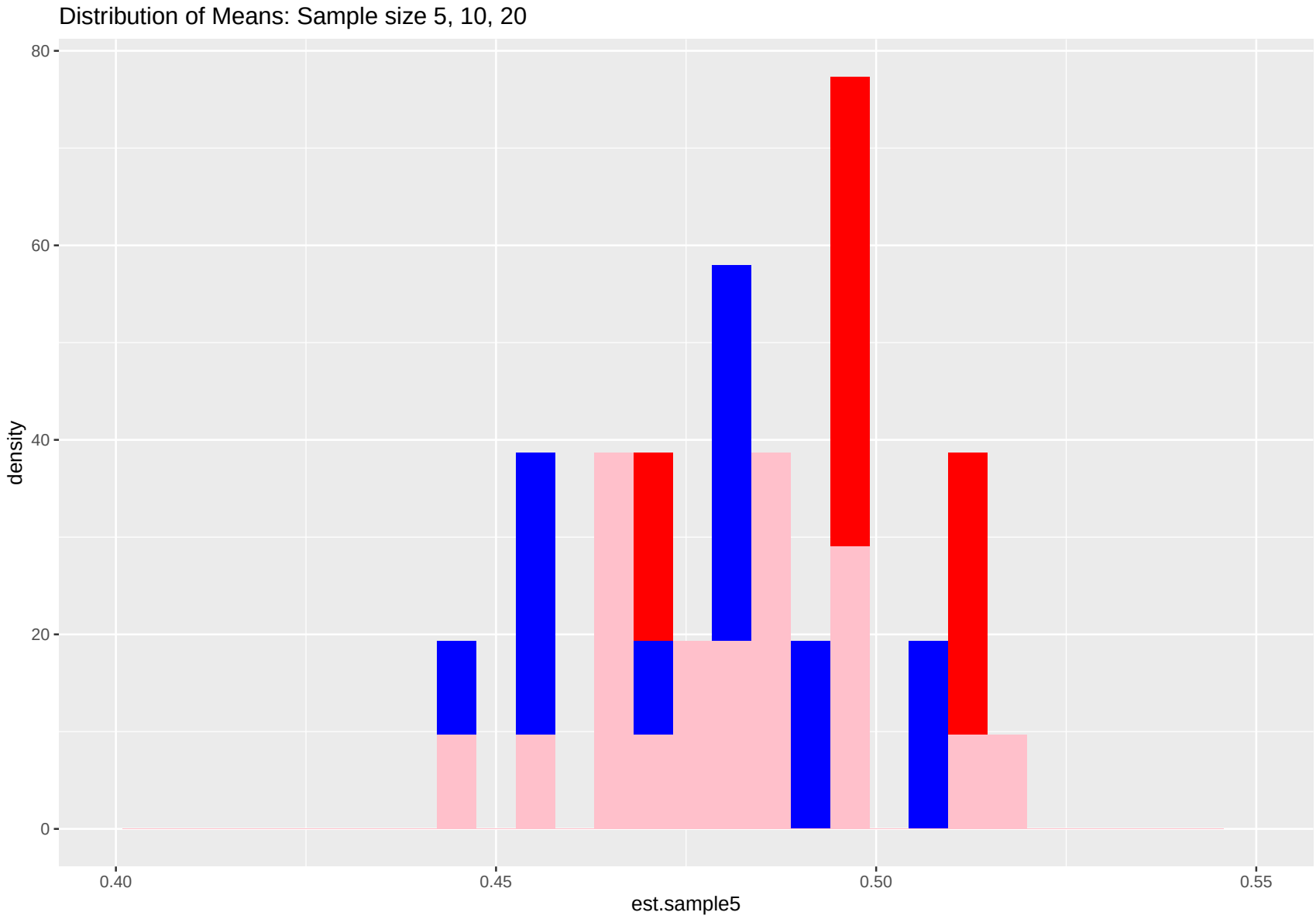
More Data, More Normal?



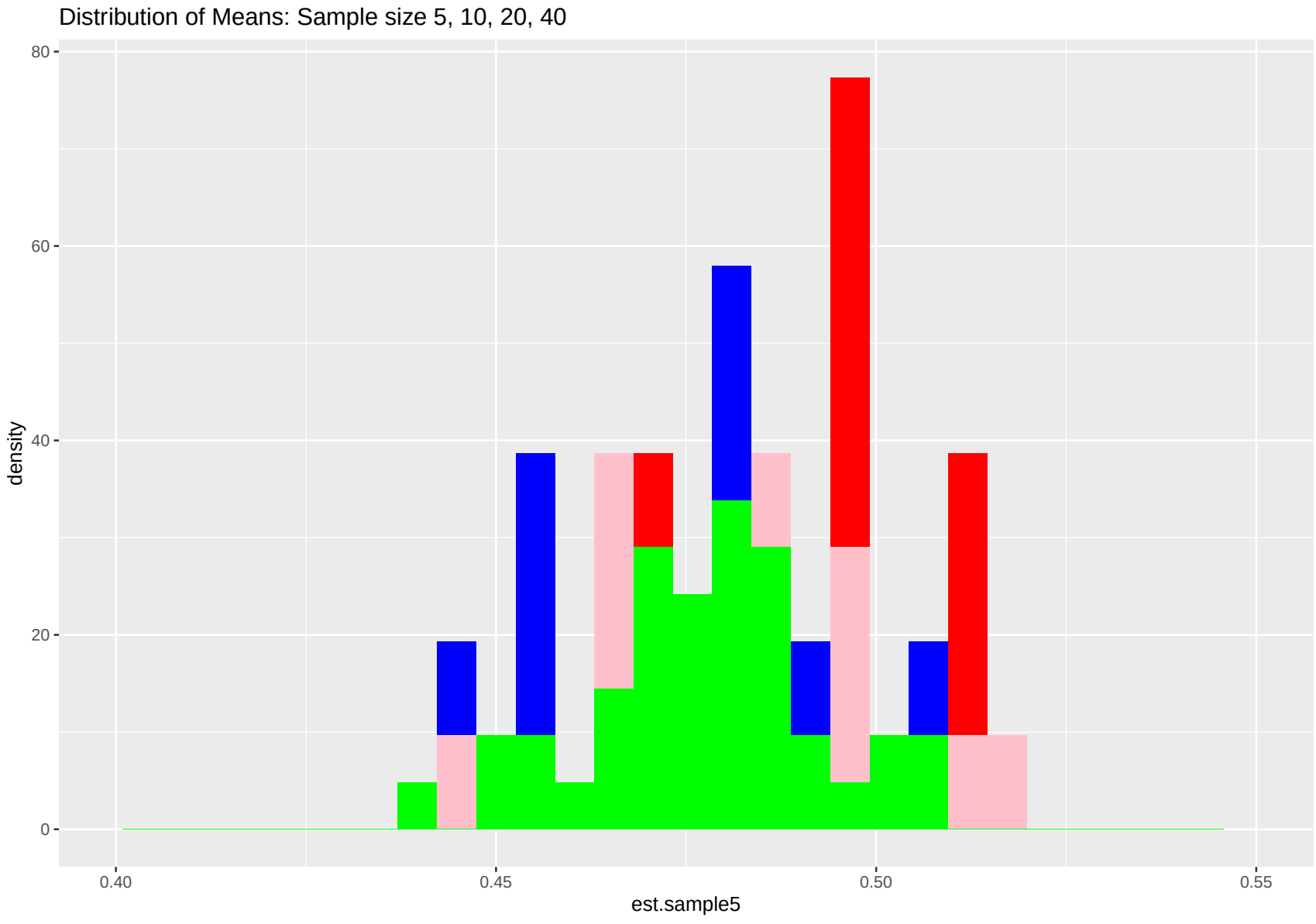
More Data, More Normal?



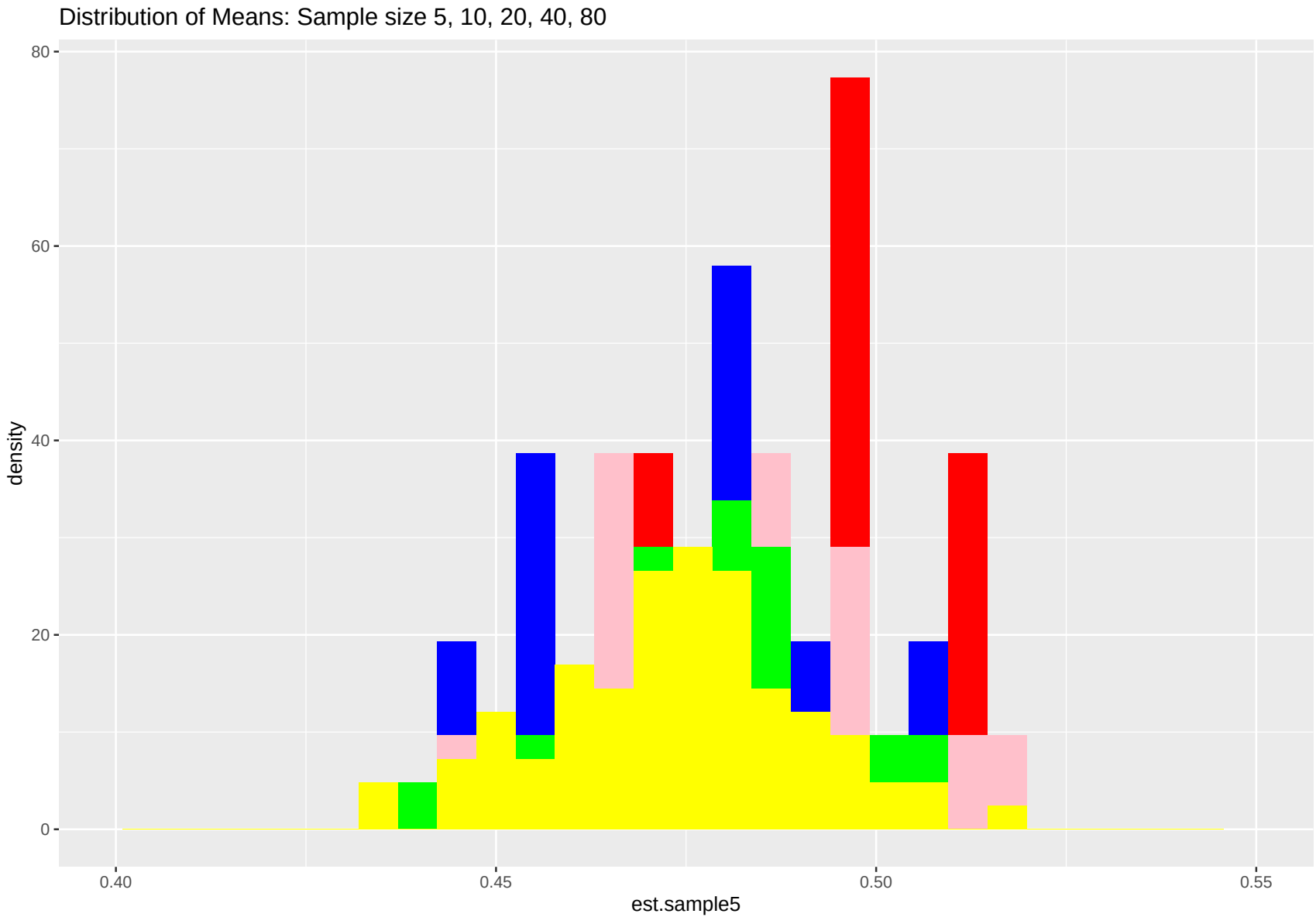
More Data, More Normal?



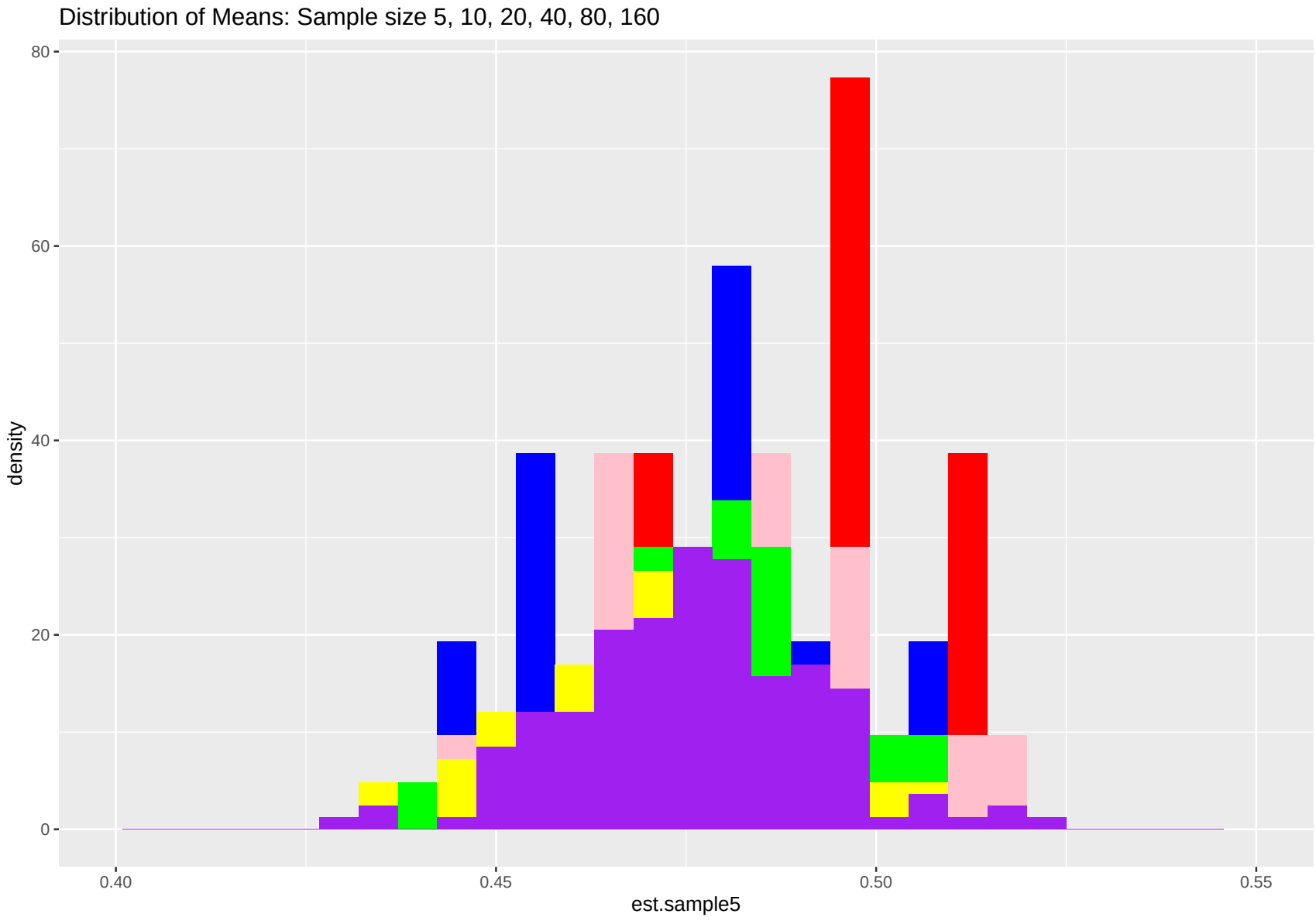
More Data, More Normal?



More Data, More Normal?



More Data, More Normal?

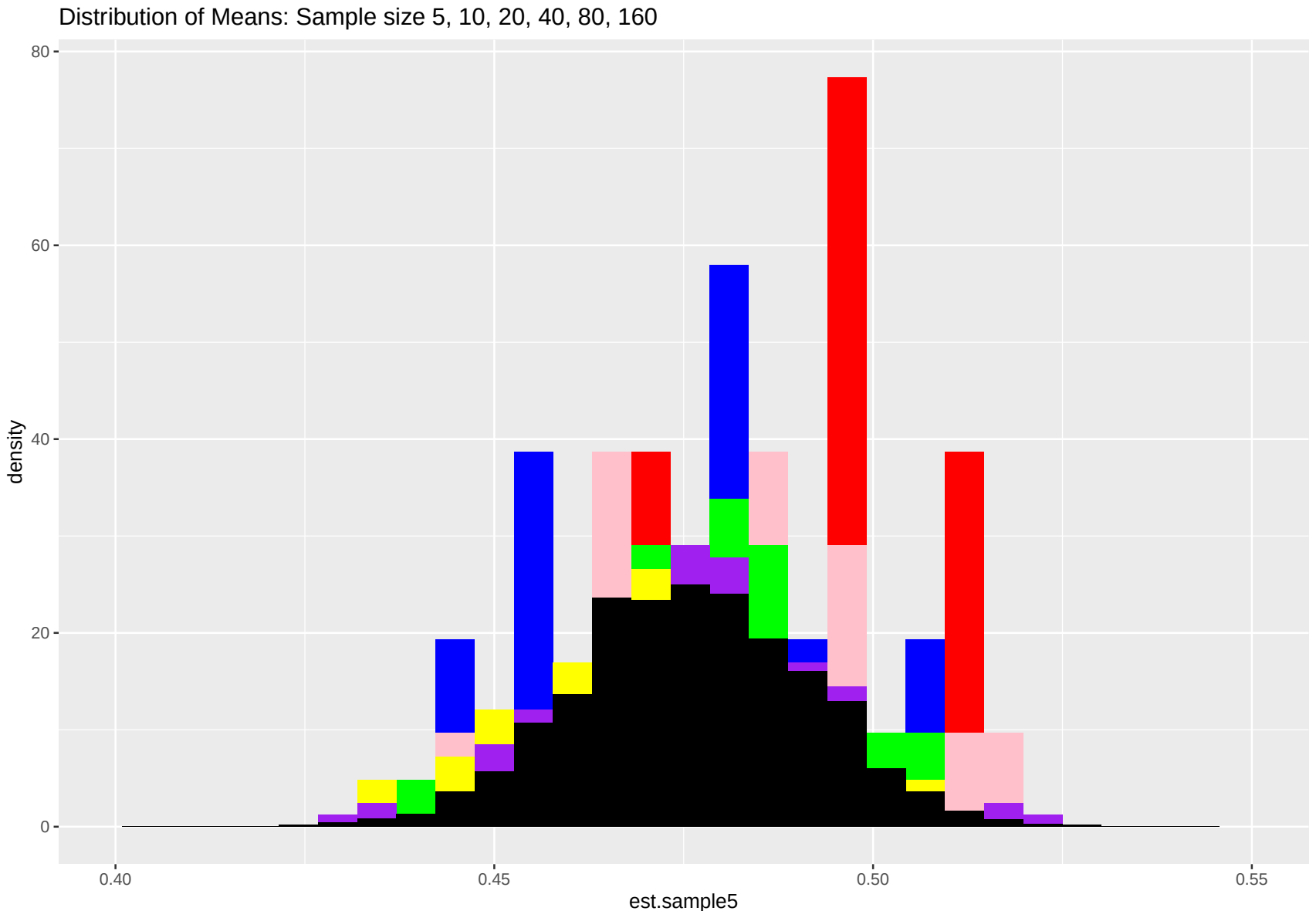


Central Limit Theorem

```
est.sample5000 <- NULL
for(i in 1:5000){
  rand.dat <- sample_n(pa.sample,n.samplesize,replace= TRUE
  est.sample5000[i] <- mean(rand.dat$likely.dem)
}
```

Central Limit Theorem

```
clt1 + geom_histogram(aes(x=est.sample5000,y = ..density..)
                      fill="black")
```



Central Limit Theorem

- The distribution of means from random samples from a population will approach a normal distribution as the number of random samples being drawn (and analyzed) increases.
- This is why the normal distribution is so special!
- Basis of a lot of statistics: e.g., difference of means tests (Z-test, T-test)

Application: Margin of Error in Polls

```
# Do the study
n.samplesize <- 1000
my.sample <- sample_n(pa.sample,n.samplesize,
                      replace= TRUE)

mean(my.sample$likely.dem)
```

```
## [1] 0.491
```

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

```
## [1] 0.4762857
```

```
mean(my.sample$likely.dem) - mean(pa.sample$likely.dem) #
```

```
## [1] 0.01471433
```

Resampling

```
B <- 5000
resample5000 <- NULL
for(i in 1:B){
  rand.dat <- sample_n(my.sample,n.samplesize,
                       replace= TRUE)
  resample5000[i] <- mean(rand.dat$likely.dem)
}
```

Resampling

```
quantile(resample5000,c(.25,.75)) # Contains 50% of the me
```

```
##   25%   75%
## 0.481 0.501
```

```
quantile(resample5000,c(.05,.95)) # Contains 90% of the me
```

```
##    5%   95%
## 0.465 0.517
```

```
quantile(resample5000,c(.025,.975)) # Contains 95% of the
```

```
##  2.5% 97.5%
## 0.459 0.523
```

```
# Notice True value is solidly within!
```

Margin of Error +/- What?

```
quantile(resample5000,c(.025,.975)) - mean(my.sample$likely
```

```
##    2.5%   97.5%
## -0.032   0.032
```

Error for 5000

```
my.sample <- sample_n(pa.sample,100,replace= TRUE)
resample5000 <- NULL
for(i in 1:5000){
  rand.dat <- sample_n(my.sample,100,replace= TRUE)
  resample5000[i] <- mean(rand.dat$likely.dem)
}

# "Margin of Error?"
quantile(resample5000,c(.025,.975)) - mean(my.sample$likely
```

```
##  2.5% 97.5%
##  -0.1   0.1
```

Error for 10000

```
my.sample <- sample_n(pa.sample,10000,replace= TRUE)
resample1000 <- rep(NA,times=1000)
for(j in 1:1000){
  rand.dat <- sample_n(my.sample,10000,replace= TRUE)
  resample1000[j] <- mean(rand.dat$likely.dem, na.rm=TRUE)
}
quantile(resample1000,c(.975)) - mean(my.sample$likely.dem)
```

```
##      97.5%
## 0.0102025
```

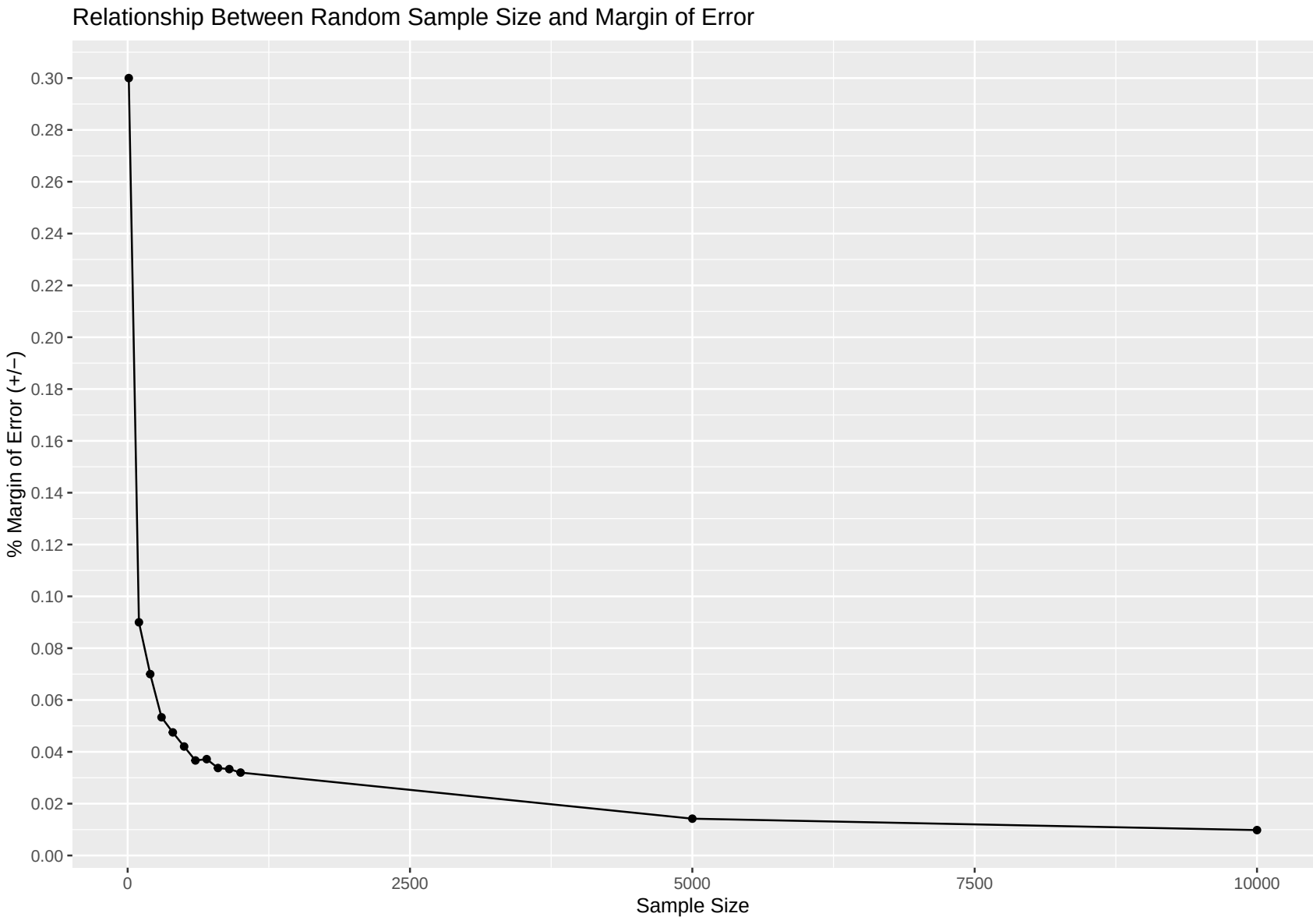
Effect of Sample Size

```
samplesizes <- c(10,100,200,300,400,500,600,700,800,900,100
moe <- rep(NA,length(samplesizes))

for(i in seq_along(samplesizes)){
  my.sample <- sample_n(pa.sample,samplesizes[i],
                        replace= TRUE)

  resample1000 <- rep(NA,times=1000)
  for(j in 1:1000){
    rand.dat <- sample_n(my.sample,samplesizes[i],
                         replace= TRUE)
    resample1000[j] <- mean(rand.dat$likely.dem)
  }
  moe[i] <- quantile(resample1000,c(.975)) - mean(my.sample
}
```

Plot!

```
dat <- bind_cols(samplesizes,moe)

ggplot(dat) +
  geom_point(aes(x=samplesizes,y=moe)) +
  labs(x="Sample Size") +
  labs(title = "Relationship Between Random Sample Size and
  scale_y_continuous(breaks=seq(0,.3,by=.02)) +
  labs(y="% Margin of Error (+/-)")
```

Plot!


Relationship Between Random Sample Size and Margin of Error
% Margin of Error (+/–)
0.30
0.28
0.26
0.24
0.22
0.20
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0
2500
5000
7500
10000
Sample Size

Bootstrap: General Idea

- General idea is to use our sample as the population – draw repeated samples to learn about the variation in our estimates.
- 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!
- 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.
- Limitation? – computing time/memory.

Bootstrap: General Idea

- General idea is to use our sample as the population – draw repeated samples to learn about the variation in our estimates.
- 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!
- 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.
- Limitation? – computing time/memory.

Technical Note: Assumes that our samples are identically and independently distributed.

Bootstrap: Conceptual Map

1. Sample from our data B times with replacement.
2. For each sample b in B, calculate the statistic of interest.
3. Use the distribution of those b statistics to evaluate precision!

Bootstrapping the Sample Mean

We get an estimate for some survey responses.

- Is it different from .5?
- Probability it is greater than .4?

```
SurveyResponses <- sample(0:1, 1000, replace=TRUE,
                          prob=c(.6,.4))
mean(SurveyResponses)
## [1] 0.409
```

Create Bootstrap Samples

```
B <- 1000
b.mean <- NULL

for(b in 1:B){
    dat<-sample(SurveyResponses,replace=TRUE)

    b.mean[b] <- mean(dat)
  }
```

Evaluate Bootstrap Samples

```
mean(b.mean)
## [1] 0.409298
quantile(b.mean,.025)
##  2.5%
## 0.379
quantile(b.mean,.975)
## 97.5%
## 0.439
```

Evaluate Bootstrap Samples

```
mean(b.mean)
## [1] 0.409298
quantile(b.mean,.025)
##  2.5%
## 0.379
quantile(b.mean,.975)
## 97.5%
## 0.439
```

Because 0.50 $>$ the 97.5% quantile, we can conclude it is very unlikely that .5 is contained in the estimate!

Probability Greater than .4?

```
mean(ifelse(b.mean > .4,1,0))
## [1] 0.725
```

Doing even more!

In the 2020 Democratic Primary, a candidate will only receive delegates to the Democratic Convention if they get at least 15%.

What is the probability that a candidate will get a delegate?

Political Polling

***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)

Simulating Sanders

- `rbinom` - how many 1's if we draw `n` samples of `size` observations with the probability of seeing a 1 is `prob` and the probability of seeing a 0 is `1-prob`.

```
# 1000 Samples of a poll of 579 with Sanders Support of .1
SandersSupport <- rbinom(n=1000, size=579, prob=.14)
summary(SandersSupport)
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   51.00   75.00   81.00   80.79   86.00  108.00

SandersSupport <- SandersSupport/579
summary(SandersSupport)
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.08808 0.12953 0.13990 0.13953 0.14853 0.18653
```

What is the probability that Sanders gets some delegates?

How can we calculate the probability that Sanders is $> .15$?

What is the probability that Sanders gets some delegates?

How can we calculate the probability that Sanders is $>$ .15?

```
sum(as.logical(SandersSupport > .15)/length(SandersSupport)
## [1] 0.249
```

What is the probability that Sanders gets some delegates?

How can we calculate the probability that Sanders is $>$ .15?

```
sum(as.logical(SandersSupport > .15)/length(SandersSupport)
## [1] 0.249
```

Now you can compute the probability that a candidate is above any threshold!

Questions for Review:

1. Why is random sampling important?
2. What does the Law of Large Numbers say?
3. Why is the Central Limit Theorem important?
4. The interquartile range is the interval containing the 25th and 75th percentiles of the data. Can you adapt the code to show how width of the interquartile range varies as the sample size increases?
5. Can you replicate the margin of error calculations for the percentage of independents? Compared to the size of the margin of error for Likely Democrats, how does the size of the margin of error for the percentage of independents compare?