Inferences for Categorical Data

Zachary Palmore

2020-10-11

```
library(tidyverse)
library(openintro)
library(infer)
library(DATA606)
```

```
##
## Welcome to CUNY DATA606 Statistics and Probability for Data Analytics
## This package is designed to support this course. The text book used
## is OpenIntro Statistics, 3rd Edition. You can read this by typing
## vignette('os3') or visit www.OpenIntro.org.
##
## The getLabs() function will return a list of the labs available.
##
## The demo(package='DATA606') will list the demos that are available.
```

Pre-exercise

Activating packages and confirming access to the data.

```
glimpse(yrbss)
```

```
## Rows: 13,583
## Columns: 13
## $ age                      <int> 14, 14, 15, 15, 15, 15, 15, 14, 15, 15, 15...
## $ gender                   <chr> "female", "female", "female", "female", "f...
## $ grade                    <chr> "9", "9", "9", "9", "9", "9", "9", "9", "9...
## $ hispanic                 <chr> "not", "not", "hispanic", "not", "not", "n...
## $ race                     <chr> "Black or African American", "Black or Afr...
## $ height                   <dbl> NA, NA, 1.73, 1.60, 1.50, 1.57, 1.65, 1.88...
## $ weight                   <dbl> NA, NA, 84.37, 55.79, 46.72, 67.13, 131.54...
## $ helmet_12m               <chr> "never", "never", "never", "never", "did n...
## $ text_while_driving_30d   <chr> "0", NA, "30", "0", "did not drive", "did ...
## $ physically_active_7d     <int> 4, 2, 7, 0, 2, 1, 4, 4, 5, 0, 0, 0, 4, 7, ...
## $ hours_tv_per_school_day  <chr> "5+", "5+", "5+", "2", "3", "5+", "5+", "5...
## $ strength_training_7d     <int> 0, 0, 0, 0, 1, 0, 2, 0, 3, 0, 3, 0, 0, 7, ...
## $ school_night_hours_sleep <chr> "8", "6", "<5", "6", "9", "8", "9", "6", "...
```

Exercise 1

What are the counts within each category for the amount of days these students have texted while driving within the past 30 days?

```
yrbss %>%
  count(text_while_driving_30d)
```

```
## # A tibble: 9 x 2
##   text_while_driving_30d     n
##   <chr>                  <int>
## 1 0                       4792
## 2 1-2                      925
## 3 10-19                    373
## 4 20-29                    298
## 5 3-5                      493
## 6 30                       827
## 7 6-9                      311
## 8 did not drive           4646
## 9 <NA>                     918
```

Exercise 2

What is the proportion of people who have texted while driving every day in the past 30 days and never wear helmets?

```
yrbss %>%
  filter(helmet_12m == "never") %>%
  count(text_while_driving_30d)
```

```
## # A tibble: 9 x 2
##   text_while_driving_30d     n
##   <chr>                  <int>
## 1 0                       2566
## 2 1-2                      515
## 3 10-19                    207
## 4 20-29                    180
## 5 3-5                      281
## 6 30                       463
## 7 6-9                      175
## 8 did not drive           2116
## 9 <NA>                     474
```

Exercise 3

What is the margin of error for the estimate of the proportion of non-helmet wearers that have texted while driving each day for the past 30 days based on this survey?

The estimate of the proportion of non-helmet wearers that have texted while driving each day for the past 30 days is 0.0664 with margin of error of 0.0227 at a 95% confidence level.

```
no_helmet <- yrbss %>%
  filter(helmet_12m == "never")
no_helmet <- no_helmet %>%
  mutate(text_ind = ifelse(text_while_driving_30d == "30", "yes", "no"))
no_helmet %>%
  specify(response = text_ind, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)
```

```
## Warning: Removed 474 rows containing missing values.
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1   0.0647   0.0778
```

```
no_helmet %>%
  count(text_while_driving_30d) %>%
  mutate(p = n/sum(n)) %>%
  mutate(total = sum(p)) %>%
  # standard error
  mutate(se = sqrt(p * (1 - p)/n)) %>%
  # at a confidence level of 95%
  mutate(me_95 = 1.96 * se)
```

```
## # A tibble: 9 x 6
##   text_while_driving_30d     n      p total      se  me_95
##   <chr>                  <int>  <dbl> <dbl>   <dbl>  <dbl>
## 1 0                       2566 0.368      1 0.00952 0.0187
## 2 1-2                      515 0.0738     1 0.0115  0.0226
## 3 10-19                    207 0.0297     1 0.0118  0.0231
## 4 20-29                    180 0.0258     1 0.0118  0.0232
## 5 3-5                      281 0.0403     1 0.0117  0.0230
## 6 30                       463 0.0664     1 0.0116  0.0227
## 7 6-9                      175 0.0251     1 0.0118  0.0232
## 8 did not drive           2116 0.303      1 0.00999 0.0196
## 9 <NA>                     474 0.0679     1 0.0116  0.0227
```

Exercise 4

Using the infer package, calculate confidence intervals for two other categorical variables (you'll need to decide which level to call "success", and report the associated margins of error. Interpet the interval in context of the data. It may be helpful to create new data sets for each of the two countries first, and then use these data sets to construct the confidence intervals.

One other categorical variable is the proportion of individuals that always wear helmets that have and have not texted while driving each day for the past 30 days. Formally, we can say that, we are 95% confident that the percentage of individuals that always wear helmets and have not texted while driving each day for the past 30 days is between 94.7% and 98.2%.

```
# 95% confidence of not texting while driving for all days
always_helmet_wearers <- yrbss %>%
  filter(helmet_12m == "always")
always_helmet <- always_helmet_wearers %>%
  mutate(text_ind = ifelse(text_while_driving_30d == "30", "yes", "no"))
always_helmet %>%
  specify(response = text_ind, success = "no") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)
```

```
## Warning: Removed 20 rows containing missing values.
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1    0.947    0.984
```

```
# 95% confidence of texting while driving for all days
always_helmet_wearers <- yrbss %>%
  filter(helmet_12m == "always")
always_helmet <- always_helmet_wearers %>%
  mutate(text_ind = ifelse(text_while_driving_30d == "30", "yes", "no"))
always_helmet %>%
  specify(response = text_ind, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)
```

```
## Warning: Removed 20 rows containing missing values.
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1   0.0185   0.0554
```

Alternatively, we could also say with 95% confidence that the percentage of always-helmet wearers that have texted while driving each day for the past 30 days is between 1.85% and 5.54%.

```
# Estimating proportions and margin of error
always_helmet %>%
  count(text_while_driving_30d) %>%
  mutate(p = n/sum(n)) %>%
  mutate(total = sum(p)) %>%
  # standard error
  mutate(se = sqrt(p * (1 - p)/n)) %>%
  # at a confidence level of 95%
  mutate(me_95 = 1.96 * se)
```

```
## # A tibble: 9 x 6
##   text_while_driving_30d     n      p total     se  me_95
##   <chr>                  <int>  <dbl> <dbl>  <dbl>  <dbl>
## 1 0                        172 0.431      1 0.0378 0.0740
## 2 1-2                       18 0.0451     1 0.0489 0.0959
## 3 10-19                      5 0.0125     1 0.0497 0.0975
## 4 20-29                      6 0.0150     1 0.0497 0.0974
## 5 3-5                       10 0.0251     1 0.0494 0.0969
## 6 30                          13 0.0326     1 0.0492 0.0965
## 7 6-9                          5 0.0125     1 0.0497 0.0975
## 8 did not drive              150 0.376      1 0.0395 0.0775
## 9 <NA>                        20 0.0501     1 0.0488 0.0956
```

The estimate of the proportion of always-helmet wearers that have texted while driving each day for the past 30 days is 0.0326 with margin of error of 0.0965 at a 95% confidence level.

Another categorical example could be the proportion of texting in the past 30 days for those who wear their helmet only sometimes. Here, we can formally say that we are 95% confident the percentage of individuals who wear their helmet only sometimes and have texted while driving for the past 30 days is between 1.55% and 5.26%.

```
sometimes_helmet_wearers <- yrbss %>%
  filter(helmet_12m == "sometimes")
sometimes_helmet <- sometimes_helmet_wearers %>%
  mutate(text_ind = ifelse(text_while_driving_30d == "30", "yes", "no"))
sometimes_helmet %>%
  specify(response = text_ind, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)
```

```
## Warning: Removed 18 rows containing missing values.
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1   0.0155   0.0526
```

The estimate of the proportion of sometimes-helmet wearers that have texted while driving each day for the past 30 days is 0.0293 with margin of error of 0.1046 at a 95% confidence level.

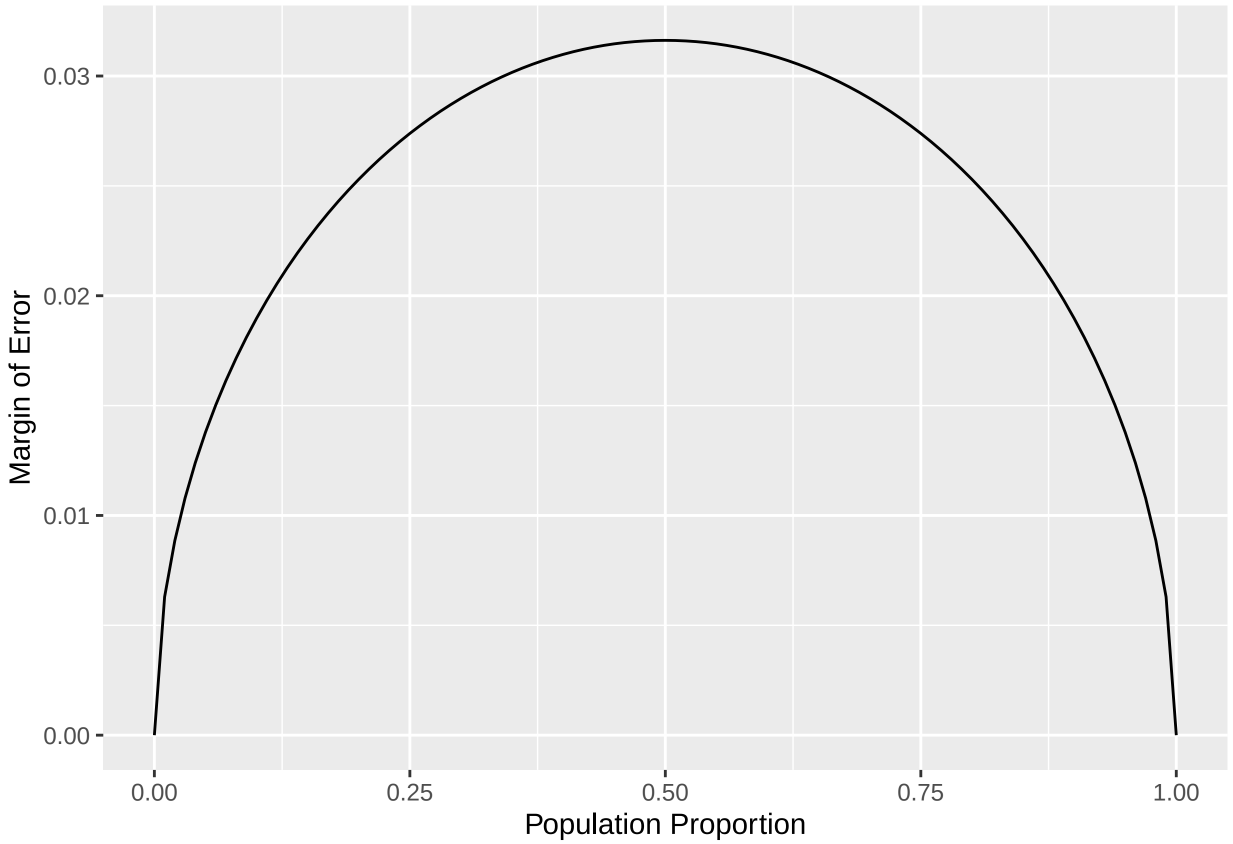
```
sometimes_helmet %>%
  count(text_while_driving_30d) %>%
  mutate(p = n/sum(n)) %>%
  mutate(total = sum(p)) %>%
  # standard error
  mutate(se = sqrt(p * (1 - p)/n)) %>%
  # at a confidence level of 95%
  mutate(me_95 = 1.96 * se)
```

```
## # A tibble: 9 x 6
##   text_while_driving_30d     n      p total     se  me_95
##   <chr>                  <int>  <dbl> <dbl>  <dbl>  <dbl>
## 1 0                        132 0.387      1 0.0424 0.0831
## 2 1-2                       23 0.0674     1 0.0523 0.102
## 3 10-19                      4 0.0117     1 0.0538 0.106
## 4 20-29                      6 0.0176     1 0.0537 0.105
## 5 3-5                       12 0.0352     1 0.0532 0.104
## 6 30                        10 0.0293     1 0.0534 0.105
## 7 6-9                        7 0.0205     1 0.0536 0.105
## 8 did not drive            129 0.378      1 0.0427 0.0837
## 9 <NA>                      18 0.0528     1 0.0527 0.103
```

Exercise 5

Describe the relationship between p and me. Include the margin of error vs. population proportion plot you constructed in your answer. For a given sample size, for which value of p is margin of error maximized?

```r
n <- 1000
p <- seq(from = 0, to = 1, by = 0.01)
me <- 2 * sqrt(p * (1 - p)/n)
dd <- data.frame(p = p, me = me)
ggplot(data = dd, aes(x = p, y = me)) +
  geom_line() +
  labs(x = "Population Proportion", y = "Margin of Error")
```



Margin of error is maximized at a population proportion of about 0.50 assuming the sample size is the same throughout. The relationship is such that as the population proportion increases, so does the margin of error until about 50% of the population proportion. At 50% of the population proportion, the margin of error begins to decrease steadily and symmetrically with the margin of error between 0 and 0.50. when the population proportion is 0 or 1 the margin of error is equal to 0.

Exercise 6

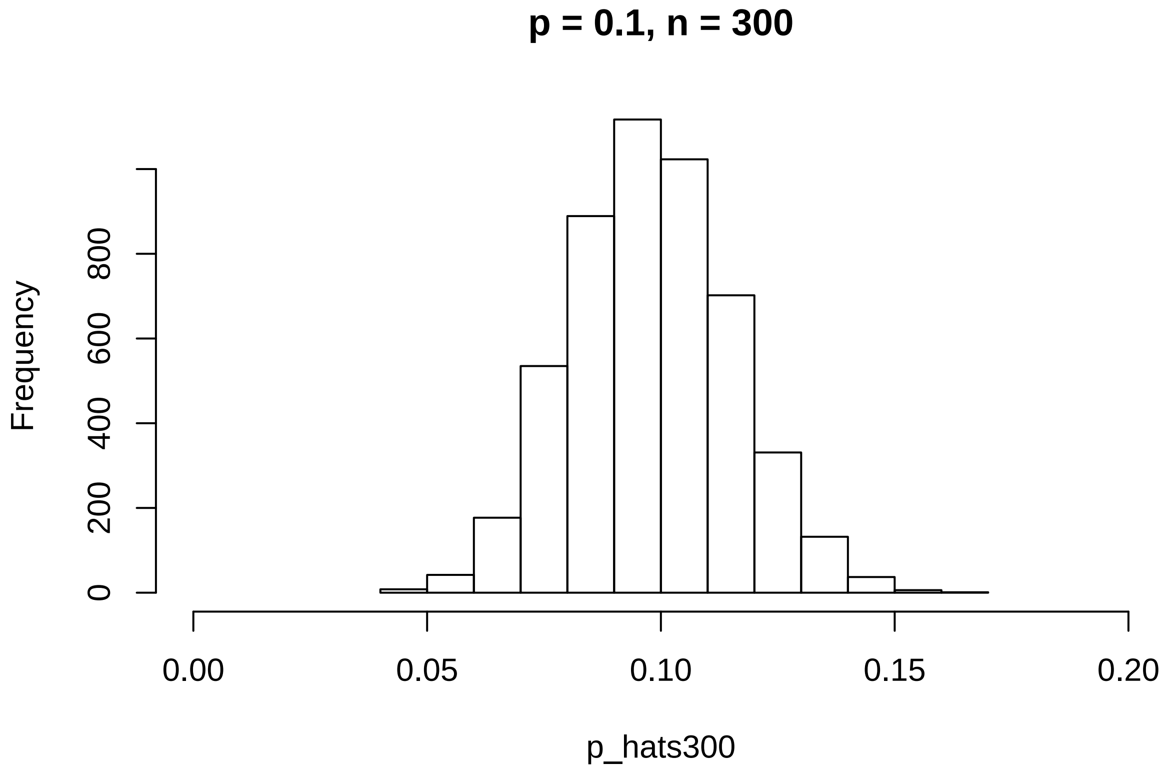
Describe the sampling distribution of sample proportions at n=300 and p=0.1. Be sure to note the center, spread, and shape.

Creating sampling proportions at n = 300 and p = 0.1.

```
# Sampling proportions at n = 300 and p = 0.1
p <- 0.1
n <- 300
p_hats300 <- rep(0, 5000)

set.seed(0010)
for(i in 1:5000){
  samp <- sample(c("atheist", "non_atheist"),
                 n,
                 replace = TRUE,
                 prob = c(p, 1-p))
  p_hats300[i] <- sum(samp == "atheist")/n
}

hist(p_hats300, main = "p = 0.1, n = 300", xlim = c(0,0.20))
```



At sample size 300 where p = 0.1 the sampling distribution centers itself around 0.1 with a symmetric shape. The spread is from 0.047 to 0.163. This makes its range 0.117. It is unimodal and appears normal.

```
# Spread is min - max
min(p_hats300)
```

```
## [1] 0.04666667
```

```
max(p_hats300)
```

```
## [1] 0.1633333
```

```
min_phat300 <- min(p_hats300)
max_phat300 <- max(p_hats300)
rangephat300 <- max_phat300 - min_phat300
rangephat300
```

```
## [1] 0.1166667
```

Exercise 7

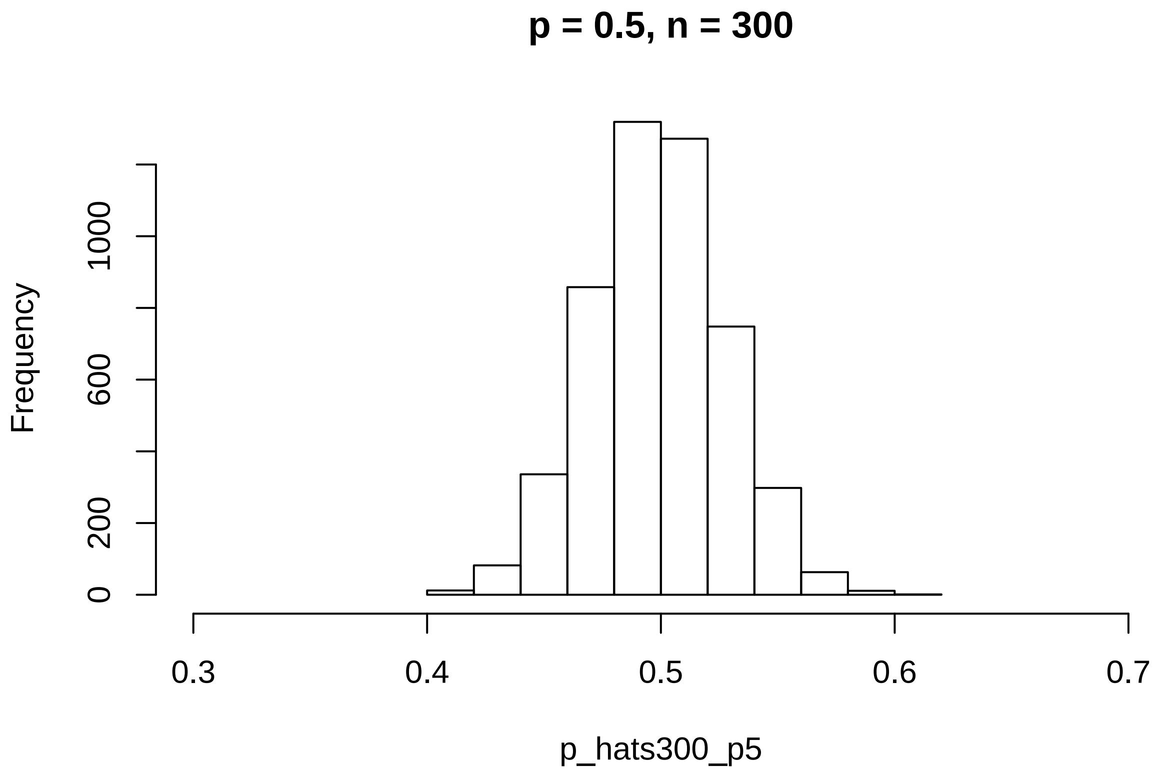
Keep n constant and change p. How does the shape, center, and spread of the sampling distribution vary as p changes. You might want to adjust min and max for the x-axis for a better view of the distribution.

Creating another histogram with p = 0.5.

```
p <- 0.5
n <- 300
p_hats300_p5 <- rep(0, 5000)

set.seed(0011)
for(i in 1:5000){
  samp <- sample(c("atheist", "non_atheist"),
                 n,
                 replace = TRUE,
                 prob = c(p, 1-p))
  p_hats300_p5[i] <- sum(samp == "atheist")/n
}

hist(p_hats300_p5, main = "p = 0.5, n = 300", xlim = c(0.3, .7))
```



At sample size 300 where p = 0.5 the sampling distribution centers itself around 0.5 with a symmetric shape. The spread is from 0.403 to 0.610. Its range is 0.207. It is also unimodal and appears normal.

```r
min_phat300p5 <- min(p_hats300_p5)
max_phat300p5 <- max(p_hats300_p5)
rangephat300p5 <- max_phat300p5 - min_phat300p5
rangephat300p5
```

```
## [1] 0.2066667
```

The distribution changed by centering itself around 0.5 and widening its spread as evident in their range values. The scale of the x-axis also changed to accommodate the wider distribution. The rest of the distribution appears similar in reference to the smaller $p = 0.1$ value.

Exercise 8

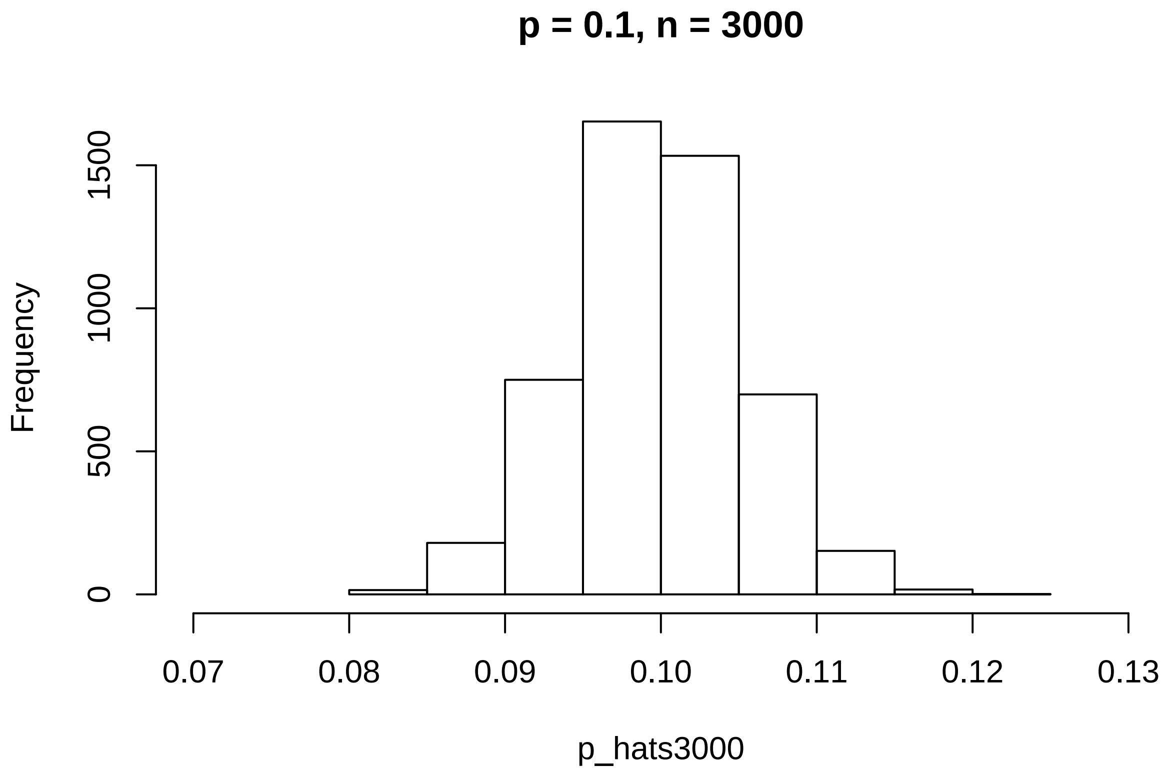
Now also change n. How does n appear to affect the distribution of pˆ?

Creating two new distributions based on the previous proportions 0.1 and 0.5. Changing n to 3000 for both to compare.

```r
# Sampling proportions at n = 3000 and p = 0.1
p <- 0.1
n <- 3000
p_hats3000 <- rep(0, 5000)
```

```
set.seed(0012)
for(i in 1:5000){
  samp <- sample(c("atheist", "non_atheist"),
                 n,
                 replace = TRUE,
                 prob = c(p, 1-p))
  p_hats3000[i] <- sum(samp == "atheist")/n
}

hist(p_hats3000, main = "p = 0.1, n = 3000", xlim = c(0.07,0.13))
```



At sample size 3000 where p = 0.1 the sampling distribution centers itself around 0.1 with a symmetric shape. The spread is from 0.0807 to 0.122. Its range is 0.0413. It is also unimodal and appears normal.

```
min_phat3000 <- min(p_hats3000)
min_phat3000
```

```
## [1] 0.08066667
```

```
max_phat3000 <- max(p_hats3000)
max_phat3000
```

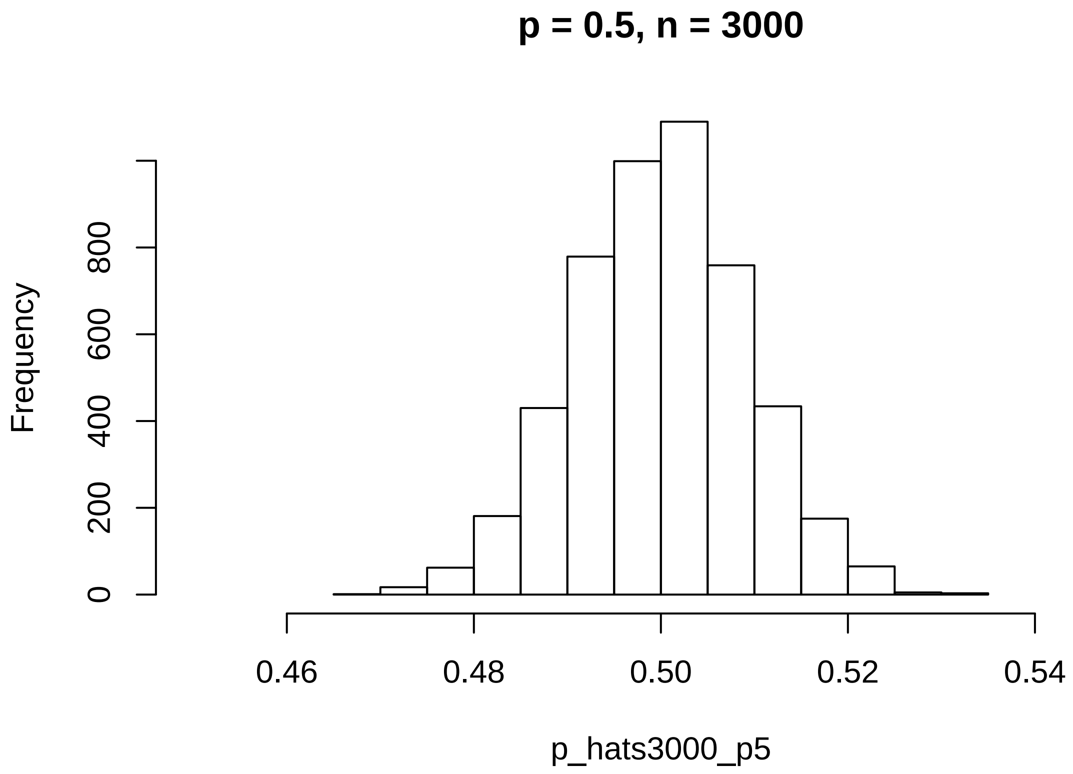
```
## [1] 0.122
```

```
rangephat3000 <- max_phat3000 - min_phat3000
rangephat3000
```

```
## [1] 0.04133333
```

```
p <- 0.5
n <- 3000
p_hats3000_p5 <- rep(0, 5000)
set.seed(0013)
for(i in 1:5000){
  samp <- sample(c("atheist", "non_atheist"),
                 n,
                 replace = TRUE,
                 prob = c(p, 1-p))
  p_hats3000_p5[i] <- sum(samp == "atheist")/n
}

hist(p_hats3000_p5, main = "p = 0.5, n = 3000", xlim = c(0.45, .55))
```



```
min_phat3000_p5 <- min(p_hats3000_p5)
min_phat3000_p5
```

```
## [1] 0.4673333
```

```
max_phat3000_p5 <- max(p_hats3000_p5)
max_phat3000_p5
```

```
## [1] 0.531
```

```
rangephat3000_p5 <- max_phat3000_p5 - min_phat3000_p5
rangephat3000_p5
```

```
## [1] 0.06366667
```

At sample size 3000 where p = 0.5 the sampling distribution centers itself around 0.5 with a symmetric shape. The spread is from 0.467 to 0.531. Its range is 0.0647. It is also unimodal and appears normal.

It appears the larger the sample size, n, the thinner the spread of the distribution of all the data. This is evident from the respective proportions ranges. For example, at p=0.5 and n=3000 the range is 0.0637 and at p=0.5 and n = 300 the range is 0.207. This is a difference of 0.1433 with the wider distribution coming from the smaller sample size.

```
0.207 - 0.0637
```

```
## [1] 0.1433
```

We could also state this in terms of the standard deviation. As the sample size increases, the standard deviation decreases as data concentrates around its center (mean) symmetrically. This assumes the proportion is held constant.

Exercise 9

Is there convincing evidence that those who sleep 10+ hours per day are more likely to strength train every day of the week? As always, write out the hypotheses for any tests you conduct and outline the status of the conditions for inference. If you find a significant difference, also quantify this difference with a confidence interval.

Null hypothesis: Those who sleep 10+ hours per day are no more likely to strength train every day of the week than those who sleep less than 10+ hours per day

Alternative hypothesis: Those who sleep 10+ hours per day are more likely to strength train every day of the week

To determine the significance of this statement we need to find the sample proportions and sample sizes.

```
# Proportion of 10+ hour sleepers that strength train daily
View(yrbss)
yrbss %>%
  filter(school_night_hours_sleep == "10+") %>%
  count(strength_training_7d) %>%
  mutate(p = n / sum(n)) %>%
  mutate(totalP = sum(p)) %>%
  mutate(totalN = sum(n))
```

```
## # A tibble: 9 x 5
##   strength_training_7d     n       p totalP totalN
##                    <int> <int>  <dbl>  <dbl>  <int>
## 1                      0   100 0.316       1    316
## 2                      1    17 0.0538      1    316
## 3                      2    31 0.0981      1    316
## 4                      3    31 0.0981      1    316
## 5                      4    18 0.0570      1    316
## 6                      5    23 0.0728      1    316
## 7                      6     8 0.0253      1    316
## 8                      7    84 0.266       1    316
## 9                     NA     4 0.0127      1    316
```

The proportion of individuals that sleep 10+ hours per day and strength train daily is 0.266 (84 of 316 individuals). Therefore the proportion of individuals that sleep 10+ hours per day and do not strength train daily is 0.734 or 232 of 316 individuals.

```
## [1] 0.734
```

```
# Playing with 95% confidence intervals

# 7 days of strength training wkly and 10+ of sleep
sleepers10 <- yrbss %>%
  filter(school_night_hours_sleep == "10+")
sleepers10 <- sleepers10 %>%
  mutate(text_ind = ifelse(strength_training_7d == "7", "yes", "no"))
sleepers10 %>%
  specify(response = text_ind, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)
```

```
## Warning: Removed 4 rows containing missing values.
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1    0.221    0.317
```

```
# Not 7 days of strength training but still 10+ of sleep
sleepers10 <- yrbss %>%
  filter(school_night_hours_sleep == "10+")
sleepers10 <- sleepers10 %>%
  mutate(text_ind = ifelse(strength_training_7d == "7", "yes", "no"))
sleepers10 %>%
  specify(response = text_ind, success = "no") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)
```

```
## Warning: Removed 4 rows containing missing values.
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1    0.683    0.782
```

The confidence intervals of those who sleep 10+ hours and exercise 7 days a week are 0.221 and 0.317. This is good because our proportion is between these intervals. The same is true of the alternative proportion, those who sleep 10+ hours per day but do not strength train 7 days a week. The confidence intervals of those who sleep 10+ hours but do not exercise 7 days a week are 0.679 and 0.779 and our calculated proportion of 0.734 fits within this interval.

Now, we find the null hypothesis proportion.

```r
sleepersNot10 <- yrbss %>%
  filter(school_night_hours_sleep != "10+")
sleepersNot10 <- sleepersNot10 %>%
  mutate(text_ind = ifelse(strength_training_7d == "7", "yes", "no"))
sleepersNot10 %>%
  specify(response = text_ind, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)
```

```
## Warning: Removed 112 rows containing missing values.
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1    0.158    0.172
```

We can say with 95% confidence that the proportion of those who do not sleep 10+ hours per day but do strength train 7 days a week, is between 15.8% and 17.1%.

```r
# Proportion of population
totalNot10 <- count(sleepersNot10)
sleepersNot10_strength7d <- sleepersNot10 %>%
  filter(strength_training_7d == "7")
total7d_not10 <- count(sleepersNot10_strength7d)
p2 <- (total7d_not10 / totalNot10)
p2
```

```
##           n
## 1 0.1629087
```

We found that the proportion of individuals who do not sleep 10+ hours per day but do strength train 7 days a week is 0.163 or 1958 of 12019 individuals. Based on these proportions of 0.266 and 0.163 we can find p-value using this formula:

```r
p1 <- 0.26582278 # from data table
# p2 remains 0.1629... from last calc of counts
n1 <- 316 # from data table
n2 <- totalNot10
z <- ((p1 - p2) - 0 )/ sqrt(p2*(1-p2)*((1/n1)+(1/n2)))
z
```

```
##          n
## 1 4.890173
```

With this zscore of 4.890 we can look up the p-value in a table of z-scores. The value corresponds to a p-value of 0.00001. At a significance level of 0.05 we reject the null hypothesis in favor of the alternative.

Exercise 10

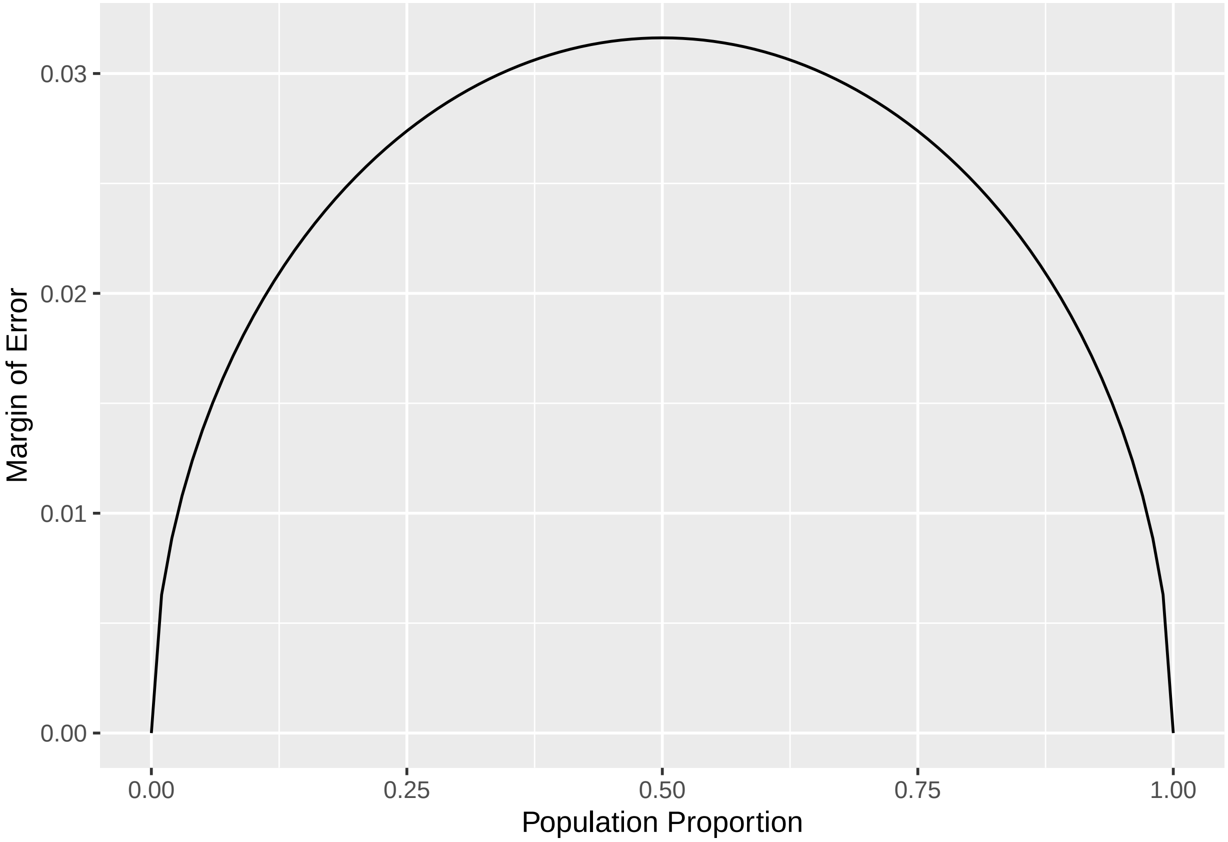
Let's say there has been no difference in likeliness to strength train every day of the week for those who sleep 10+ hours. What is the probablity that you could detect a change (at a significance level of 0.05) simply by chance? Hint: Review the definition of the Type 1 error.

At a significance level of 0.05, the probability of detecting a change simply by chance (or an incorrect rejection of the null hypothesis) is about 5%. The significance level determines the likelihood of committing a type 1 error.

Exercise 11

Suppose you're hired by the local government to estimate the proportion of residents that attend a religious service on a weekly basis. According to the guidelines, the estimate must have a margin of error no greater than 1% with 95% confidence. You have no idea what to expect for p. How many people would you have to sample to ensure that you are within the guidelines? Hint: Refer to your plot of the relationship between p and margin of error. This question does not require using a dataset.

```r
n <- 1000
p <- seq(from = 0, to = 1, by = 0.01)
me <- 2 * sqrt(p * (1 - p)/n)
dd <- data.frame(p = p, me = me)
ggplot(data = dd, aes(x = p, y = me)) +
  geom_line() +
  labs(x = "Population Proportion", y = "Margin of Error")
```



Based on this plot of the relationship between p and me (margin of error), we can visually see that the maximum me is at 0.5 while the lowest is at 0 and 1. By rearranging the me equation we can calculate the estimated population from the given information. As a reminder, the me equation looks like;

$$me = z * SE$$

where SE is the standard error, which can also be represented by the formula;

$$SE = \sqrt(\frac{p(1-p)}{(n)})$$

Bring it together and we have:

$$me = z * \sqrt(\frac{p(1-p)}{(n)})$$

In this case, all but two parameters were given. The z-score that correlates with a 95% confidence interval is 1.96. This is the value for z. The significance of 1% means the margin of error in the population cannot be greater than 0.01. If we consider the worst case scenario, our error would be 0.01 without going over. The standard deviation could be any proportion of the population given in the graph. Thus, if we take the proportion at the maximum margin of error, we can calculate the population we need to have a sample capable of meeting 1% significance at a 95% confidence interval.

We can re-write this as,

$$n = \frac{p(1-p)}{\frac{me}{z}^2}$$

then compute:

```
(.5*(1-.5))/(.01/1.96)^2
```

```
## [1] 9604
```

```
# Or, a little easier
.25*(1.96/0.01)^2
```

```
## [1] 9604
```

We would need 9604 individuals to ensure that we are within the guidelines.

…