Data Science Inference and Modeling

The textbook for the Data Science course series is freely available online.

This course corresponds to the textbook chapters Statistical Inference and Statistical Models.

Learning Objectives

- The concepts necessary to define estimates and margins of errors of populations, parameters, estimates, and standard errors in order to make predictions about data
- How to use models to aggregate data from different sources
- The very basics of Bayesian statistics and predictive modeling

Course Overview

Section 1: Parameters and Estimates

You will learn how to estimate population parameters.

Section 2: The Central Limit Theorem in Practice

You will apply the central limit theorem to assess how close a sample estimate is to the population parameter of interest.

Section 3: Confidence Intervals and p-Values

You will learn how to calculate confidence intervals and learn about the relationship between confidence intervals and p-values.

Section 4: Statistical Models

You will learn about statistical models in the context of election forecasting.

Section 5: Bayesian Statistics

You will learn about Bayesian statistics through looking at examples from rare disease diagnosis and baseball.

Section 6: Election Forecasting

You will learn about election forecasting, building on what you've learned in the previous sections about statistical modeling and Bayesian statistics.

Section 7: Association Tests

You will learn how to use association and chi-squared tests to perform inference for binary, categorical, and ordinal data through an example looking at research funding rates.

Introduction to Inference

The textbook for this section is available here

In this course, we will learn:

- *statistical inference*, the process of deducing characteristics of a population using data from a random sample
- the statistical concepts necessary to define *estimates* and *margins of errors*
- how to *forecast future results* and estimate the precision of our forecast
- how to calculate and interpret *confidence intervals and p-values*

Key points

- Information gathered from a small random sample can be used to infer characteristics of the entire population.
- Opinion polls are useful when asking everyone in the population is impossible.
- A common use for opinion polls is determining voter preferences in political elections for the purposes of forecasting election results.
- The *spread* of a poll is the estimated difference between support two candidates or options.

Section 1 Overview

Section 1 introduces you to parameters and estimates.

After completing Section 1, you will be able to:

- Understand how to use a sampling model to perform a poll.
- Explain the terms **population**, **parameter**, and **sample** as they relate to statistical inference.
- Use a sample to estimate the population proportion from the sample average.
- Calculate the expected value and standard error of the sample average.

Sampling Model Parameters and Estimates

The textbook for this section is available here and here; first part

Key points

- The task of statistical inference is to estimate an unknown population parameter using observed data from a sample.
- In a sampling model, the collection of elements in the urn is called the *population*.
- A *parameter* is a number that summarizes data for an entire population.
- A *sample* is observed data from a subset of the population.
- An *estimate* is a summary of the observed data about a parameter that we believe is informative. It is a data-driven guess of the population parameter.
- We want to predict the proportion of the blue beads in the urn, the parameter p . The proportion of red beads in the urn is $1 - p$ and the *spread* is $2p - 1$.

- The sample proportion is a random variable. Sampling gives random results drawn from the population distribution.

Code: Function for taking a random draw from a specific urn

The **dslabs** package includes a function for taking a random draw of size n from the urn:

```
if(!require(tidyverse)) install.packages("tidyverse")
```

```
## Loading required package: tidyverse
```

```
## -- Attaching packages --------------------------------------------------------------------------------
```

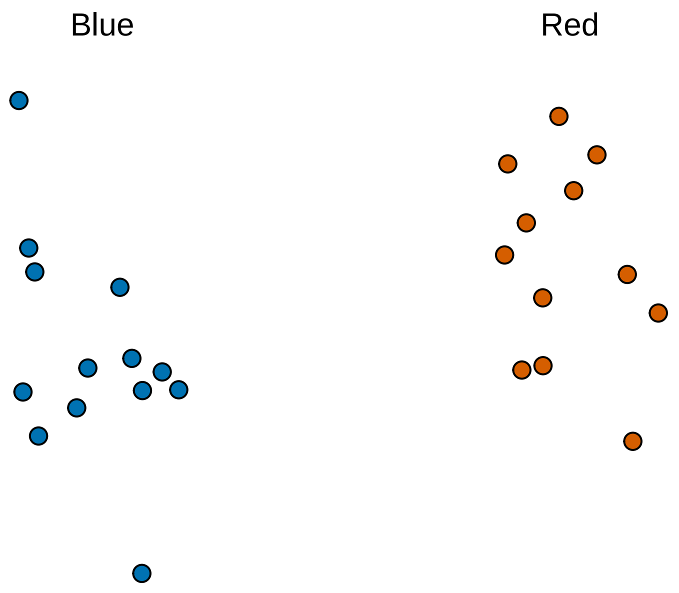
```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.1
## v tidyr   1.1.1     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts -----------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
if(!require(dslabs)) install.packages("dslabs")
```

```
## Loading required package: dslabs
```

```
library(tidyverse)
library(dslabs)
take_poll(25)    # draw 25 beads
```



The Sample Average

The textbook for this section is available here and here

Key points

- Many common data science tasks can be framed as estimating a parameter from a sample.
- We illustrate statistical inference by walking through the process to estimate p. From the estimate of p, we can easily calculate an estimate of the spread, $2p - 1$.
- Consider the random variable X that is 1 if a blue bead is chosen and 0 if a red bead is chosen. The proportion of blue beads in N draws is the average of the draws $X_1, ..., X_N$.
- $\overline{X}$ is the *sample average*. In statistics, a bar on top of a symbol denotes the average. $\overline{X}$ is a random variable because it is the average of random draws - each time we take a sample, $\overline{X}$ is different.

$\overline{X} = \frac{X_1 + X_2 + ... + X_N}{N}$

- The number of blue beads drawn in N draws, $N\overline{X}$, is N times the proportion of values in the urn. However, we do not know the true proportion: we are trying to estimate this parameter p.

Polling versus Forecasting

The textbook for this section is available here

Key points

- A poll taken in advance of an election estimates p for that moment, not for election day.
- In order to predict election results, forecasters try to use early estimates of p to predict p on election day. We discuss some approaches in later sections.

Properties of Our Estimate

The textbook for this section is available here

Key points

- When interpreting values of $\overline{X}$, it is important to remember that $\overline{X}$ is a random variable with an expected value and standard error that represents the sample proportion of positive events.
- The expected value of $\overline{X}$ is the parameter of interest p. This follows from the fact that $\overline{X}$ is the sum of independent draws of a random variable times a constant $1/N$.

$E(\overline{X}) = p$

- As the number of draws N increases, the standard error of our estimate $\overline{X}$ decreases. The standard error of the average of $\overline{X}$ over N draws is:

$SE(\overline{X}) = \sqrt{p(1-p)/N}$

- In theory, we can get more accurate estimates of p by increasing N. In practice, there are limits on the size of N due to costs, as well as other factors we discuss later.
- We can also use other random variable equations to determine the expected value of the sum of draws $E(S)$ and standard error of the sum of draws $SE(S)$.

$E(S) = Np$

$SE(S) = \sqrt{Np(1-p)}$

Assessment - Parameters and Estimates

1. Suppose you poll a population in which a proportion p of voters are Democrats and $1-p$ are Republicans.

Your sample size is $N = 25$. Consider the random variable S, which is the **total** number of Democrats in your sample.

What is the expected value of this random variable S?

- ☐ A. $E(S) = 25(1-p)$
- ☒ B. $E(S) = 25p$
- ☐ C. $E(S) = \sqrt{25p(1-p)}$
- ☐ D. $E(S) = p$

2. Again, consider the random variable S, which is the **total** number of Democrats in your sample of 25 voters.

The variable p describes the proportion of Democrats in the sample, whereas $1-p$ describes the proportion of Republicans.

What is the standard error of S?

- ☐ A. $SE(S) = 25p(1-p)$
- ☐ B. $SE(S) = \sqrt{25p}$
- ☐ C. $SE(S) = 25(1-p)$
- ☒ D. $SE(S) = \sqrt{25p(1-p)}$

3. Consider the random variable S/N, which is equivalent to the sample average that we have been denoting as $\overline{X}$.

The variable N represents the sample size and p is the proportion of Democrats in the population.

What is the expected value of $\overline{X}$?

- ☒ A. $E(\overline{X}) = p$
- ☐ B. $E(\overline{X}) = Np$
- ☐ C. $E(\overline{X}) = N(1-p)$
- ☐ D. $E(\overline{X}) = 1-p$

4. What is the standard error of the sample average, $\overline{X}$?

The variable N represents the sample size and p is the proportion of Democrats in the population.

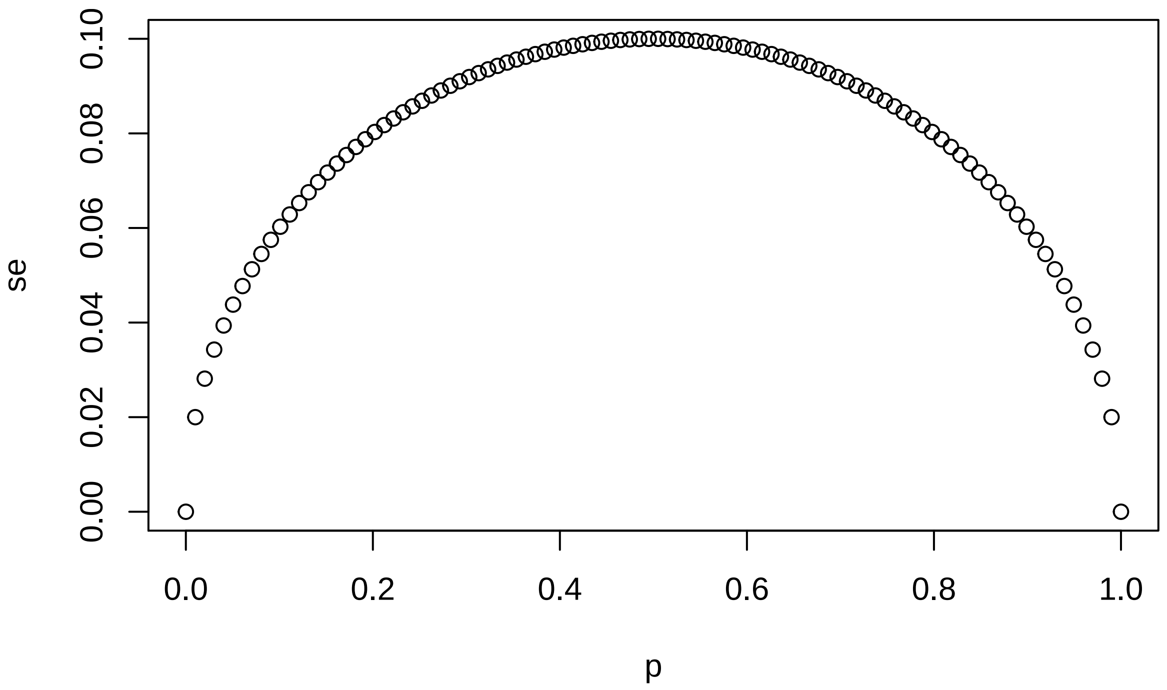
- ☐ A. $SE(\overline{X}) = \sqrt{Np(1-p)}$
- ☒ B. $SE(\overline{X}) = \sqrt{p(1-p)/N}$
- ☐ C. $SE(\overline{X}) = \sqrt{p(1-p)}$
- ☐ D. $SE(\overline{X}) = \sqrt{N}$

5. Write a line of code that calculates the standard error `se` of a sample average when you poll 25 people in the population.

Generate a sequence of 100 proportions of Democrats `p` that vary from 0 (no Democrats) to 1 (all Democrats).

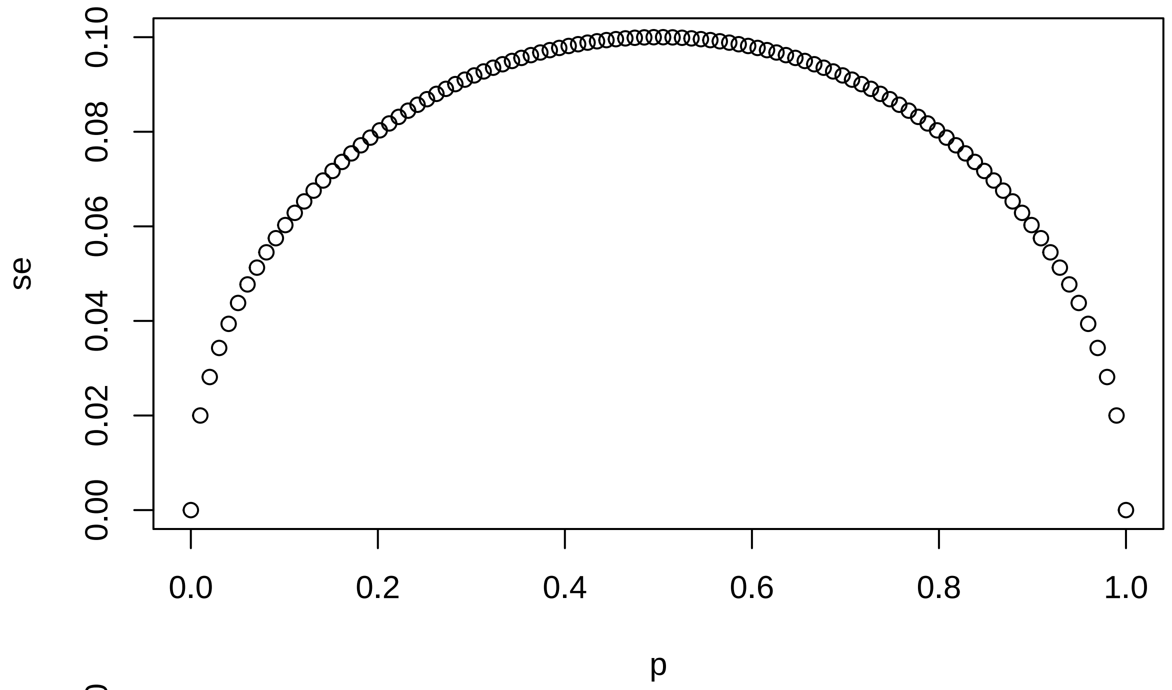
Plot `se` versus `p` for the 100 different proportions.

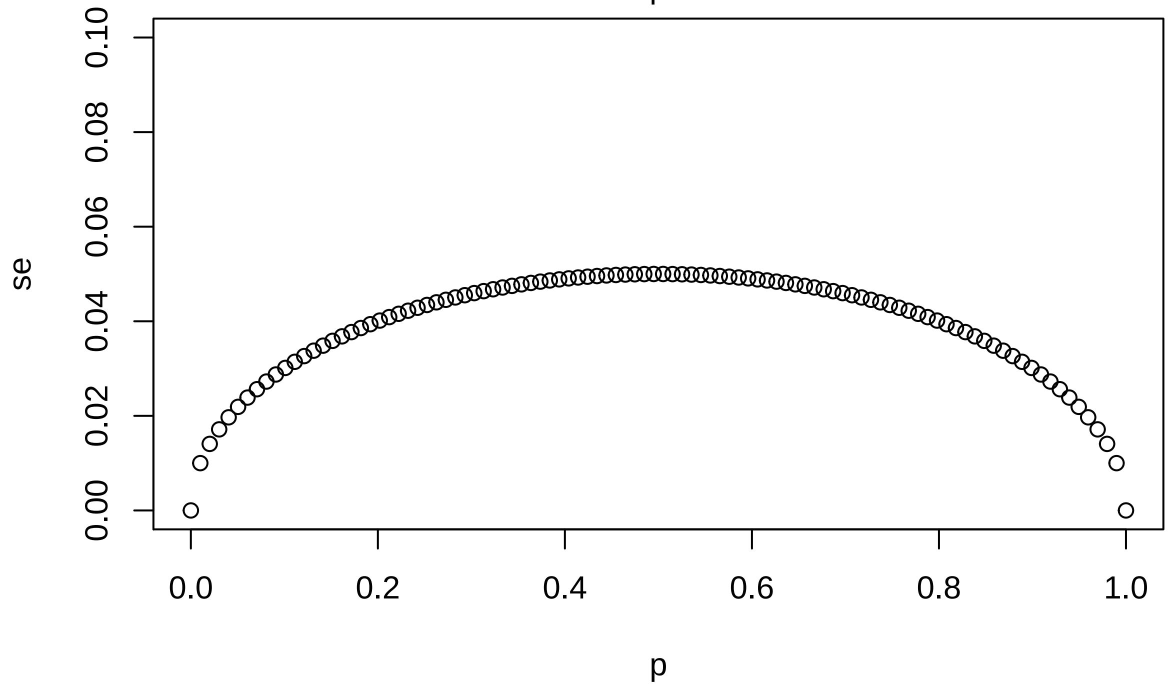
```r
# `N` represents the number of people polled
N <- 25

# Create a variable `p` that contains 100 proportions ranging from 0 to 1 using the `seq` function
p <- seq(0, 1, length.out = 100)

# Create a variable `se` that contains the standard error of each sample average
se <- sqrt(p * (1 - p)/N)

# Plot `p` on the x-axis and `se` on the y-axis
plot(p,se)
```

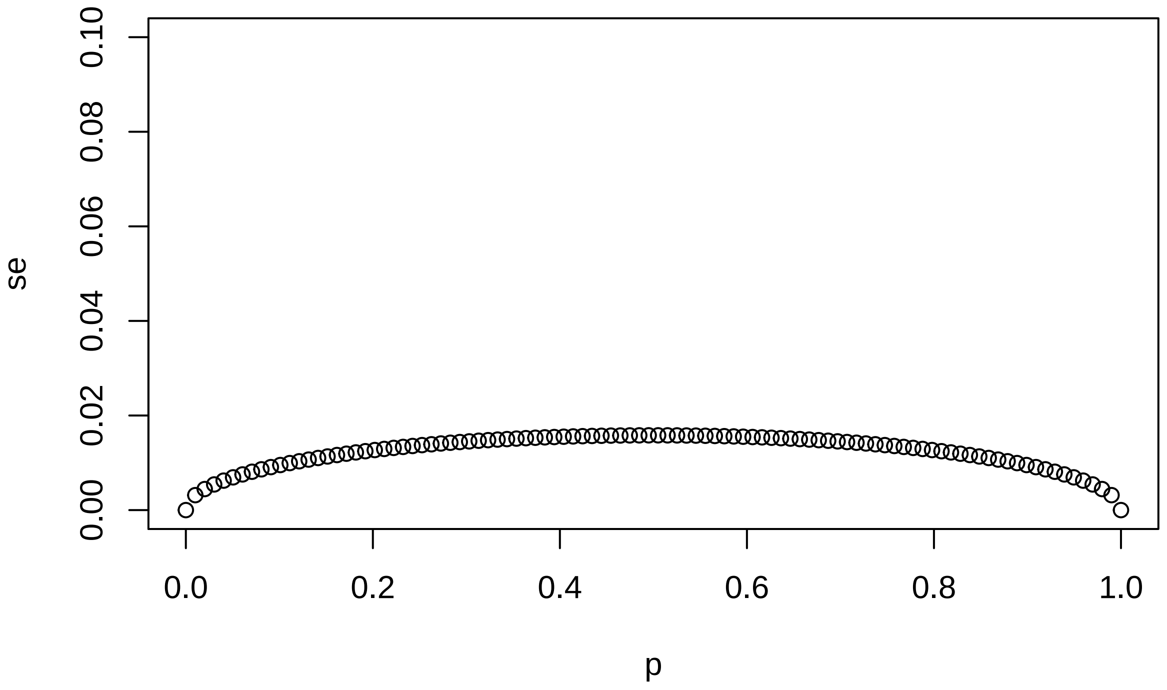


6. Using the same code as in the previous exercise, create a for-loop that generates three plots of `p` versus `se` when the sample sizes equal $N = 25$, $N = 100$, and $N = 1000$.

```r
# The vector `p` contains 100 proportions of Democrats ranging from 0 to 1 using the `seq` function
p <- seq(0, 1, length = 100)

# The vector `sample_sizes` contains the three sample sizes
sample_sizes <- c(25, 100, 1000)

# Write a for-loop that calculates the standard error `se` for every value of `p` for each of the three
for (N in sample_sizes)
{
se <- sqrt(p * (1 - p)/N)
plot(p,se,ylim = c(0,0.1))
}
```


se
0.00
0.02
0.04
0.06
0.08
0.10
0.0
0.2
0.4
0.6
0.8
1.0
p


se
0.00
0.02
0.04
0.06
0.08
0.10
0.0
0.2
0.4
0.6
0.8
1.0
p



7. Our estimate for the difference in proportions of Democrats and Republicans is $d = \overline{X} - (1 - \overline{X})$.

Which derivation correctly uses the rules we learned about sums of random variables and scaled random variables to derive the expected value of d

- ☐ A. $E\left[\overline{X} - (1 - \overline{X})\right] = E\left[2\overline{X} - 1\right] = 2E\left[\overline{X}\right] - 1 = N(2p - 1) = Np - N(1 - p)$
- ☐ B. $E\left[\overline{X} - (1 - \overline{X})\right] = E\left[\overline{X} - 1\right] = E\left[\overline{X}\right] - 1 = p - 1$
- ☐ C. $E\left[\overline{X} - (1 - \overline{X})\right] = E\left[2\overline{X} - 1\right] = 2E\left[\overline{X}\right] - 1 = 2\sqrt{p(1 - p)} - 1 = p - (1 - p)$
- ☒ D. $E\left[\overline{X} - (1 - \overline{X})\right] = E\left[2\overline{X} - 1\right] = 2E\left[\overline{X}\right] - 1 = 2p - 1 = p - (1 - p)$

8. Our estimate for the difference in proportions of Democrats and Republicans is $d = \overline{X} - (1 - \overline{X})$.

Which derivation correctly uses the rules we learned about sums of random variables and scaled random variables to derive the standard error of d?

- ☐ A. $SE\left[\overline{X} - (1 - \overline{X})\right] = SE\left[2\overline{X} - 1\right] = 2SE\left[\overline{X}\right] = 2\sqrt{p/N}$
- ☐ B. $SE\left[\overline{X} - (1 - \overline{X})\right] = SE\left[2\overline{X} - 1\right] = 2SE\left[\overline{X} - 1\right] = 2\sqrt{p(1 - p)/N} - 1$
- ☒ C. $SE\left[\overline{X} - (1 - \overline{X})\right] = SE\left[2\overline{X} - 1\right] = 2SE\left[\overline{X}\right] = 2\sqrt{p(1 - p)/N}$
- ☐ D. $SE\left[\overline{X} - (1 - \overline{X})\right] = SE\left[\overline{X} - 1\right] = SE\left[\overline{X}\right] = \sqrt{p(1 - p)/N}$

9. Say the actual proportion of Democratic voters is $p = 0.45$.

In this case, the Republican party is winning by a relatively large margin of $d = -0.1$, or a 10% margin of victory. What is the standard error of the spread $2\overline{X} - 1$ in this case?

```r
# `N` represents the number of people polled
N <- 25
```

```r
# `p` represents the proportion of Democratic voters
p <- 0.45

# Calculate the standard error of the spread. Print this value to the console.
2*sqrt((p*(1-p)/N))
```

```
## [1] 0.1989975
```

10. So far we have said that the difference between the proportion of Democratic voters and Republican voters is about 10% and that the standard error of this spread is about 0.2 when $N = 25$.

Select the statement that explains why this sample size is sufficient or not.

- ☐ A. This sample size is sufficient because the expected value of our estimate $2\overline{X} - 1$ is d so our prediction will be right on.
- ☒ B. This sample size is too small because the standard error is larger than the spread.
- ☐ C. This sample size is sufficient because the standard error of about 0.2 is much smaller than the spread of 10%.
- ☐ D. Without knowing `p`, we have no way of knowing that increasing our sample size would actually improve our standard error.

Section 2 Overview

In Section 2, you will look at the Central Limit Theorem in practice.

After completing Section 2, you will be able to:

- Use the Central Limit Theorem to calculate the probability that a sample estimate $\overline{X}$ is close to the population proportion p.
- Run a Monte Carlo simulation to corroborate theoretical results built using probability theory.
- Estimate the spread based on estimates of $\overline{X}$ and $\hat{SE}(\overline{X})$.
- Understand why bias can mean that larger sample sizes aren't necessarily better.

The Central Limit Theorem in Practice

The textbook for this section is available here

Key points

- Because $\overline{X}$ is the sum of random draws divided by a constant, the distribution of $\overline{X}$ is approximately normal.
- We can convert $\overline{X}$ to a standard normal random variable Z:

$Z = \frac{\overline{X} - E(\overline{X})}{SE(\overline{X})}$

- The probability that $\overline{X}$ is within .01 of the actual value of p is:

$Pr(Z \leq .01/\sqrt{p(1-p)/N}) - Pr(Z \leq -.01/\sqrt{p(1-p)/N})$

- The Central Limit Theorem (CLT) still works if $\overline{X}$ is used in place of p. This is called a *plug-in estimate*. Hats over values denote estimates. Therefore:

$\hat{SE}(\overline{X}) = \sqrt{\overline{X}(1-\overline{X})/N}$

Using the CLT, the probability that $\overline{X}$ is within .01 of the actual value of p is:

$Pr(Z \leq .01/\sqrt{\overline{X}(1-\overline{X})/N}) - Pr(Z \leq -.01/\sqrt{\overline{X}(1-\overline{X})/N})$

Code: Computing the probability of $\overline{X}$ being within .01 of p

```
X_hat <- 0.48
se <- sqrt(X_hat*(1-X_hat)/25)
pnorm(0.01/se) - pnorm(-0.01/se)
```

```
## [1] 0.07971926
```

Margin of Error

The textbook for this section is available here

Key points

- The *margin of error* is defined as 2 times the standard error of the estimate $\overline{X}$.
- There is about a 95% chance that $\overline{X}$ will be within two standard errors of the actual parameter p.

A Monte Carlo Simulation for the CLT

The textbook for this section is available here

Key points

- We can run Monte Carlo simulations to compare with theoretical results assuming a value of p.
- In practice, p is unknown. We can corroborate theoretical results by running Monte Carlo simulations with one or several values of p.
- One practical choice for p when modeling is $\overline{X}$, the observed value of $\hat{X}$ in a sample.

Code: Monte Carlo simulation using a set value of p

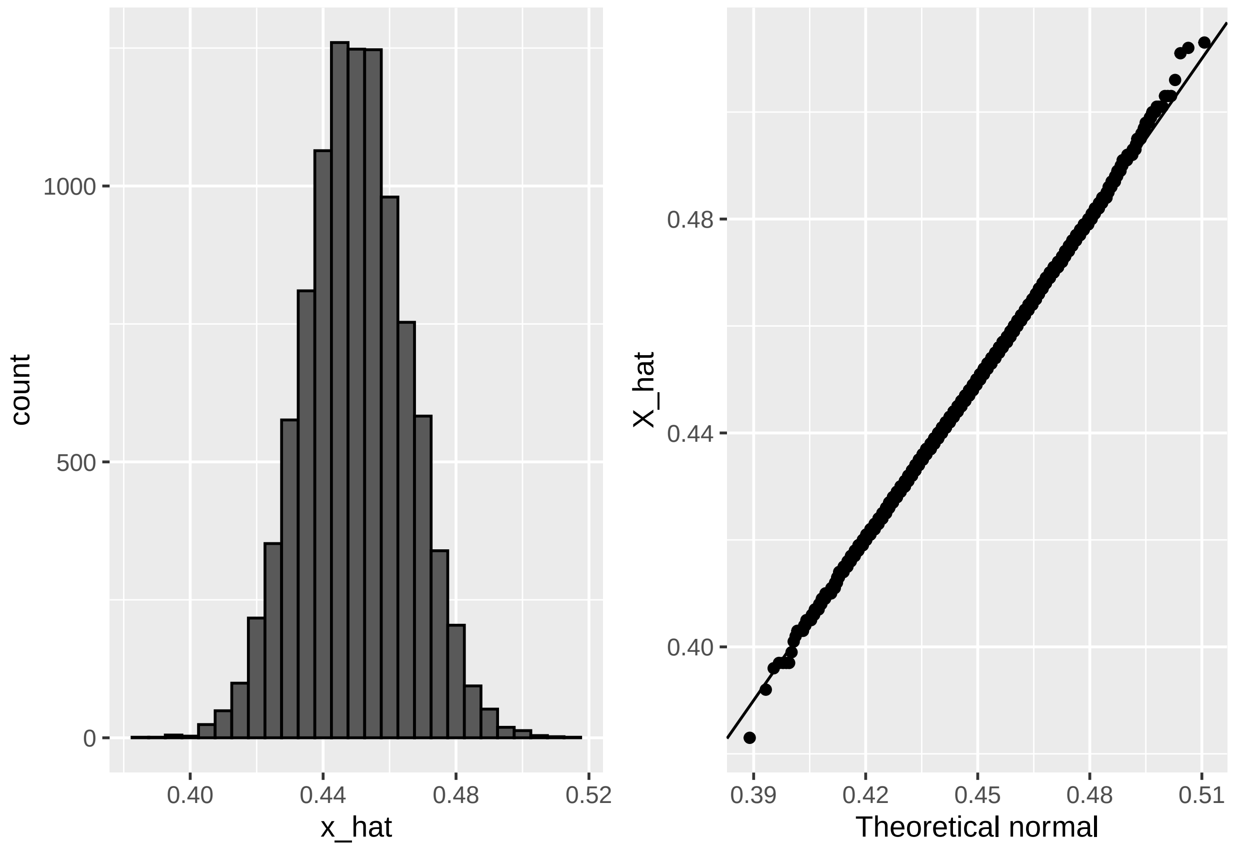
```
p <- 0.45    # unknown p to estimate
N <- 1000

# simulate one poll of size N and determine x_hat
x <- sample(c(0,1), size = N, replace = TRUE, prob = c(1-p, p))
x_hat <- mean(x)

# simulate B polls of size N and determine average x_hat
B <- 10000    # number of replicates
N <- 1000    # sample size per replicate
x_hat <- replicate(B, {
    x <- sample(c(0,1), size = N, replace = TRUE, prob = c(1-p, p))
    mean(x)
})
```

Code: Histogram and QQ-plot of Monte Carlo results

```r
if(!require(gridExtra)) install.packages("gridExtra")
```

```
## Loading required package: gridExtra

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
library(gridExtra)
p1 <- data.frame(x_hat = x_hat) %>%
    ggplot(aes(x_hat)) +
    geom_histogram(binwidth = 0.005, color = "black")
p2 <- data.frame(x_hat = x_hat) %>%
    ggplot(aes(sample = x_hat)) +
    stat_qq(dparams = list(mean = mean(x_hat), sd = sd(x_hat))) +
    geom_abline() +
    ylab("X_hat") +
    xlab("Theoretical normal")
grid.arrange(p1, p2, nrow=1)
```



The Spread

The textbook for this section is available here

Key points

- The spread between two outcomes with probabilities p and $1-p$ is $2p-1$.
- The expected value of the spread is $2\overline{X}-1$.
- The standard error of the spread is $2\hat{SE}(\overline{X})$.
- The margin of error of the spread is 2 times the margin of error of $\overline{X}$.

Bias: Why Not Run a Very Large Poll?

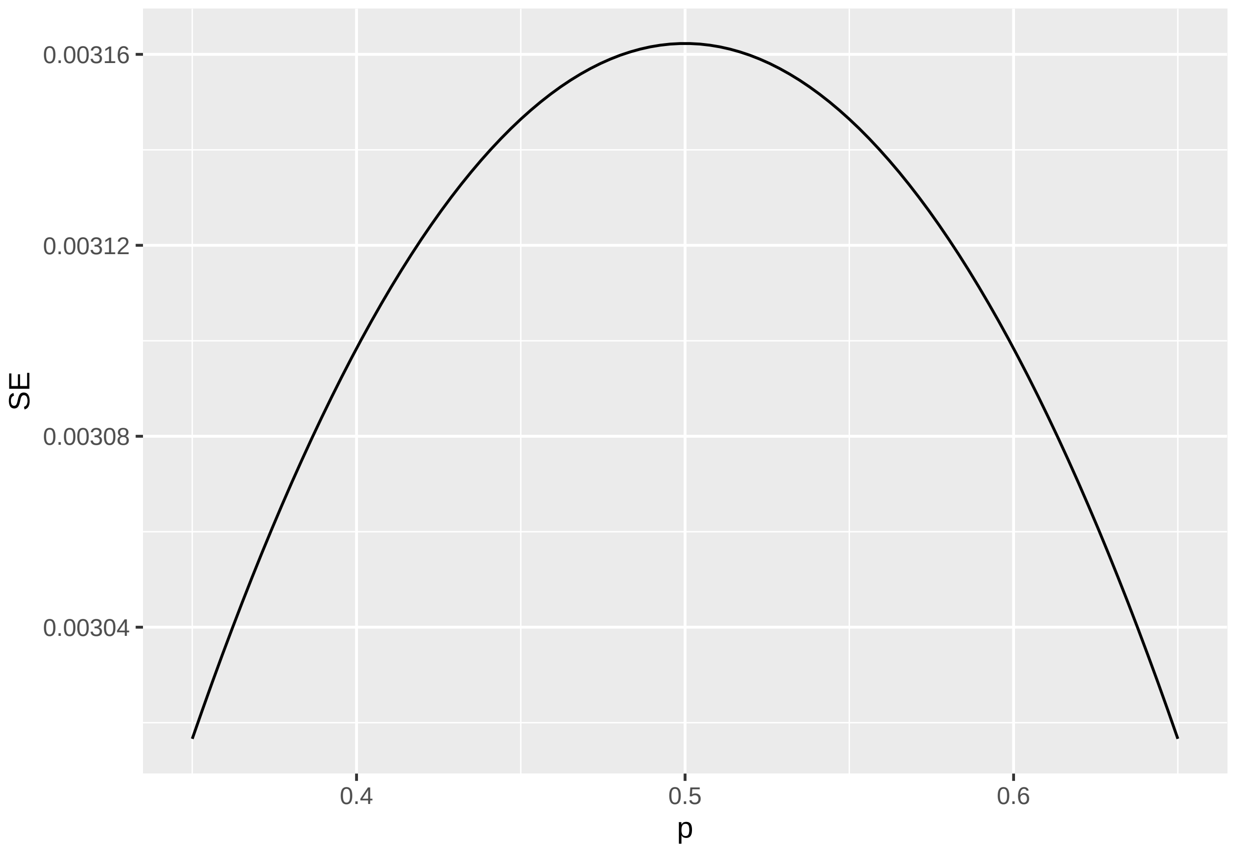
The textbook for this section is available here

Key points

- An extremely large poll would theoretically be able to predict election results almost perfectly.
- These sample sizes are not practical. In addition to cost concerns, polling doesn't reach everyone in the population (eventual voters) with equal probability, and it also may include data from outside our population (people who will not end up voting).
- These systematic errors in polling are called *bias*. We will learn more about bias in the future.

Code: Plotting margin of error in an extremely large poll over a range of values of p

```
N <- 100000
p <- seq(0.35, 0.65, length = 100)
SE <- sapply(p, function(x) 2*sqrt(x*(1-x)/N))
data.frame(p = p, SE = SE) %>%
    ggplot(aes(p, SE)) +
    geom_line()
```



Assessment - Introduction to Inference

1. Write function called `take_sample` that takes the proportion of Democrats p and the sample size N as arguments and returns the sample average of Democrats (1) and Republicans (0).

Calculate the sample average if the proportion of Democrats equals 0.45 and the sample size is 100.

```
# Write a function called `take_sample` that takes `p` and `N` as arguements and returns the average va
take_sample <- function(p, N) {
  x <- sample(c(0,1), size = N, replace = TRUE, prob = c(1-p, p))
  return(mean(x))
}

# Use the `set.seed` function to make sure your answer matches the expected result after random samplin
set.seed(1)

# Define `p` as the proportion of Democrats in the population being polled
p <- 0.45

# Define `N` as the number of people polled
N <- 100

# Call the `take_sample` function to determine the sample average of `N` randomly selected people from
take_sample(p, N)
```

```
## [1] 0.46
```

2. Assume the proportion of Democrats in the population p equals 0.45 and that your sample size N is 100 polled voters.

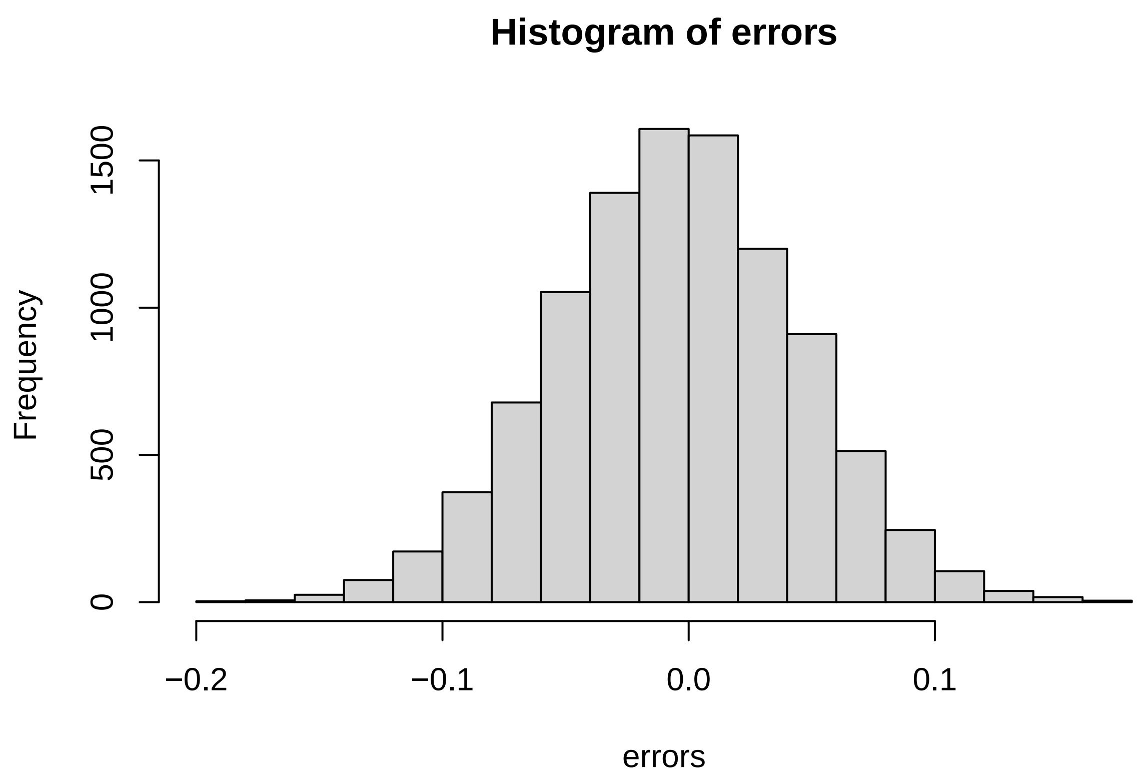
The `take_sample` function you defined previously generates our estimate, $\overline{X}$.

Replicate the random sampling 10,000 times and calculate $p - \overline{X}$ for each random sample. Save these differences as a vector called `errors`. Find the average of `errors` and plot a histogram of the distribution.

```r
# Define `p` as the proportion of Democrats in the population being polled
p <- 0.45

# Define `N` as the number of people polled
N <- 100

# The variable `B` specifies the number of times we want the sample to be replicated
B <- 10000

# Use the `set.seed` function to make sure your answer matches the expected result after random samplin
set.seed(1)

# Create an objected called `errors` that replicates subtracting the result of the `take_sample` functi
errors <- replicate(B, p - take_sample(p, N))

# Calculate the mean of the errors. Print this value to the console.
mean(errors)
```

```
## [1] -4.9e-05
```

```r
hist(errors)
```



3. In the last exercise, you made a vector of differences between the actual value for p and an estimate, $\overline{X}$.

We called these differences between the actual and estimated values `errors`.

The `errors` object has already been loaded for you. Use the `hist` function to plot a histogram of the values contained in the vector `errors`. Which statement best describes the distribution of the errors?

- ☐ A. The errors are all about 0.05.
- ☐ B. The errors are all about -0.05.
- ☒ C. The errors are symmetrically distributed around 0.
- ☐ D. The errors range from -1 to 1.

4. The error $p - \overline{X}$ is a random variable.

In practice, the error is not observed because we do not know the actual proportion of Democratic voters, p. However, we can describe the size of the error by constructing a simulation.

What is the average size of the error if we define the size by taking the absolute value $|p - \overline{X}|$?

```r
# Define `p` as the proportion of Democrats in the population being polled
p <- 0.45

# Define `N` as the number of people polled
N <- 100

# The variable `B` specifies the number of times we want the sample to be replicated
B <- 10000

# Use the `set.seed` function to make sure your answer matches the expected result after random samplin
set.seed(1)

# We generated `errors` by subtracting the estimate from the actual proportion of Democratic voters
errors <- replicate(B, p - take_sample(p, N))

# Calculate the mean of the absolute value of each simulated error. Print this value to the console.
mean(abs(errors))
```

```
## [1] 0.039267
```

5. The standard error is related to the typical **size** of the error we make when predicting.

We say **size** because, as we just saw, the errors are centered around 0. In that sense, the typical error is 0. For mathematical reasons related to the central limit theorem, we actually use the standard deviation of `errors` rather than the average of the absolute values.

As we have discussed, the standard error is the square root of the average squared distance $(\overline{X} - p)^2$. The standard deviation is defined as the square root of the distance squared.

Calculate the standard deviation of the spread.

```r
# Define `p` as the proportion of Democrats in the population being polled
p <- 0.45

# Define `N` as the number of people polled
N <- 100

# The variable `B` specifies the number of times we want the sample to be replicated
B <- 10000

# Use the `set.seed` function to make sure your answer matches the expected result after random samplin
set.seed(1)

# We generated `errors` by subtracting the estimate from the actual proportion of Democratic voters
errors <- replicate(B, p - take_sample(p, N))

# Calculate the standard deviation of `errors`
sqrt(mean(errors^2))
```

```
## [1] 0.04949939
```

6. The theory we just learned tells us what this standard deviation is going to be because it is the standard error of $\overline{X}$.

Estimate the standard error given an expected value of 0.45 and a sample size of 100.

```r
# Define `p` as the expected value equal to 0.45
p <- 0.45

# Define `N` as the sample size
N <- 100

# Calculate the standard error
sqrt(p*(1-p)/N)
```

```
## [1] 0.04974937
```

7. In practice, we don't know p, so we construct an estimate of the theoretical prediction based by plugging in $\overline{X}$ for p. Calculate the standard error of the estimate: $\hat{SE}(\overline{X})$

```r
# Define `p` as a proportion of Democratic voters to simulate
p <- 0.45

# Define `N` as the sample size
N <- 100

# Use the `set.seed` function to make sure your answer matches the expected result after random samplin
set.seed(1)

# Define `X` as a random sample of `N` voters with a probability of picking a Democrat ('1') equal to `
X <- sample(c(0,1), size = N, replace = TRUE, prob = c(1-p, p))
```

```r
# Define `X_bar` as the average sampled proportion
X_bar <- mean(X)

# Calculate the standard error of the estimate. Print the result to the console.
se <- sqrt((X_bar*(1-X_bar)/N))
se
```

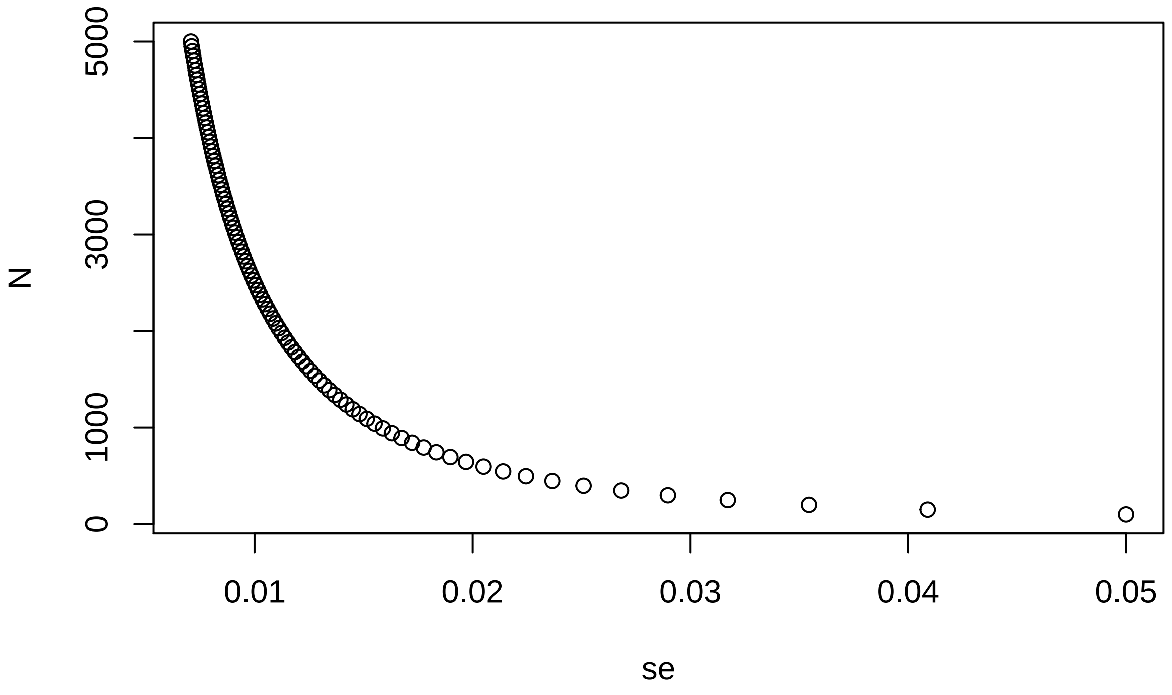
```
## [1] 0.04983974
```

8. The standard error estimates obtained from the Monte Carlo simulation, the theoretical prediction, and the estimate of the theoretical prediction are all very close, which tells us that the theory is working.

This gives us a practical approach to knowing the typical error we will make if we predict p with $\hat{X}$. The theoretical result gives us an idea of how large a sample size is required to obtain the precision we need. Earlier we learned that the largest standard errors occur for $p = 0.5$.

Create a plot of the largest standard error for N ranging from 100 to 5,000.

```r
N <- seq(100, 5000, len = 100)
p <- 0.5
se <- sqrt(p*(1-p)/N)
plot(se, N)
```



Based on this plot, how large does the sample size have to be to have a standard error of about 1%?

- ☐ A. 100
- ☐ B. 500
- ☒ C. 2,500
- ☐ D. 4,000

9. For $N = 100$, the central limit theorem tells us that the distribution of $\hat{X}$ is…

☐ A. practically equal to p.
☒ B. approximately normal with expected value p and standard error $\sqrt{p(1-p)/N}$.
☐ C. approximately normal with expected value $\overline{X}$ and standard error $\sqrt{\overline{X}(1-\overline{X}/N}$.
☐ D. not a random variable.

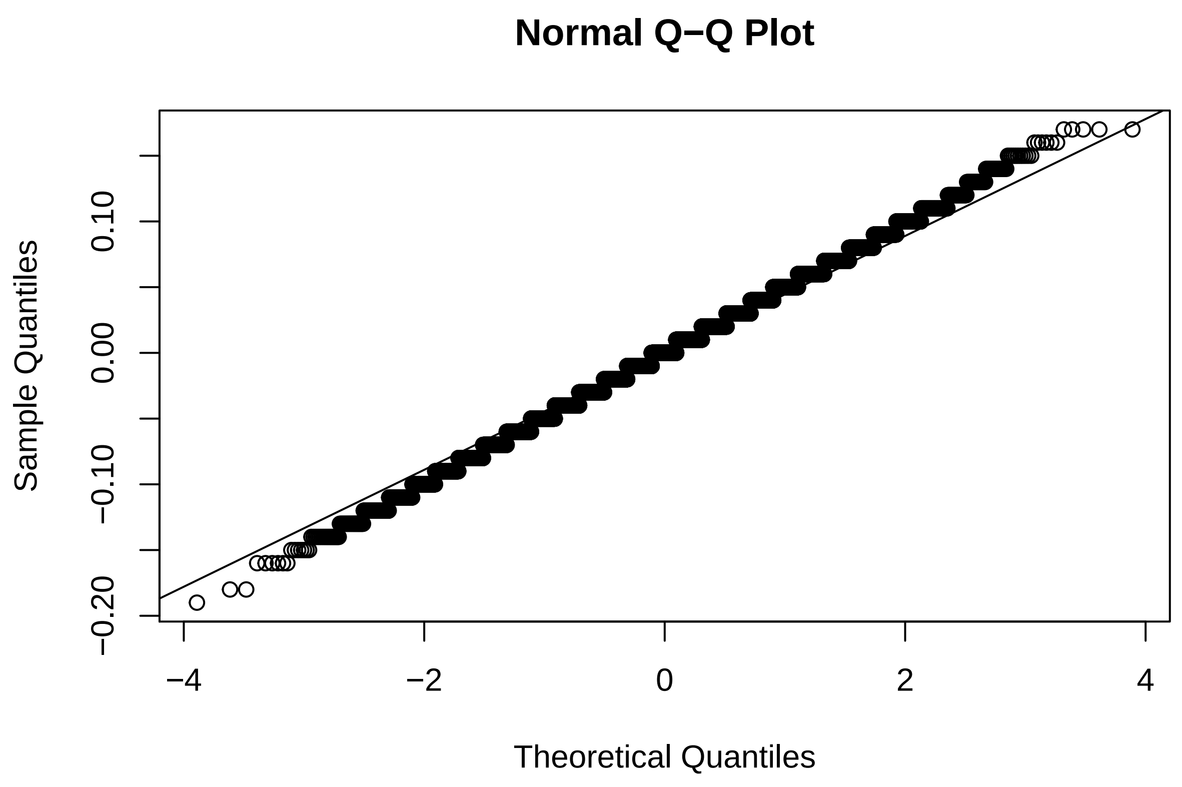
10. We calculated a vector `errors` that contained, for each simulated sample, the difference between the actual value `p` and our estimate $\hat{X}$.

The errors $\overline{X} - p$ are:

☐ A. practically equal to 0.
☒ B. approximately normal with expected value 0 and standard error $\sqrt{p(1-p)/N}$.
☐ C. approximately normal with expected value p and standard error $\sqrt{p(1-p)/N}$.
☐ D. not a random variable.

11. Make a qq-plot of the `errors` you generated previously to see if they follow a normal distribution.

```
# Define `p` as the proportion of Democrats in the population being polled
p <- 0.45

# Define `N` as the number of people polled
N <- 100

# The variable `B` specifies the number of times we want the sample to be replicated
B <- 10000

# Use the `set.seed` function to make sure your answer matches the expected result after random samplin
set.seed(1)

# Generate `errors` by subtracting the estimate from the actual proportion of Democratic voters
errors <- replicate(B, p - take_sample(p, N))

# Generate a qq-plot of `errors` with a qq-line showing a normal distribution
qqnorm(errors)
qqline(errors)
```



12. If $p = 0.45$ and $N = 100$, use the central limit theorem to estimate the probability that $\overline{X} > 0.5$.

```
# Define `p` as the proportion of Democrats in the population being polled
p <- 0.45

# Define `N` as the number of people polled
N <- 100

# Calculate the probability that the estimated proportion of Democrats in the population is greater tha

1-pnorm(0.5, p, sqrt(p*(1-p)/N))
```

```
## [1] 0.1574393
```

13. Assume you are in a practical situation and you don't know p.

Take a sample of size $N = 100$ and obtain a sample average of $\overline{X} = 0.51$.

What is the CLT approximation for the probability that your error size is equal or larger than 0.01?

```
# Define `N` as the number of people polled
N <-100

# Define `X_hat` as the sample average
X_hat <- 0.51

# Define `se_hat` as the standard error of the sample average
se_hat <- sqrt(X_hat*(1-X_hat)/N)
```

```r
# Calculate the probability that the error is 0.01 or larger
1-pnorm(0.01,0,se_hat) + pnorm(-0.01,0,se_hat)
```

```
## [1] 0.8414493
```

Section 3 Overview

In Section 3, you will look at confidence intervals and p-values.

After completing Section 3, you will be able to:

- Calculate confidence intervals of difference sizes around an estimate.
- Understand that a confidence interval is a random interval with the given probability of falling on top of the parameter.
- Explain the concept of "power" as it relates to inference.
- Understand the relationship between p-values and confidence intervals and explain why reporting confidence intervals is often preferable.

Confidence Intervals

The textbook for this section is available here

Key points

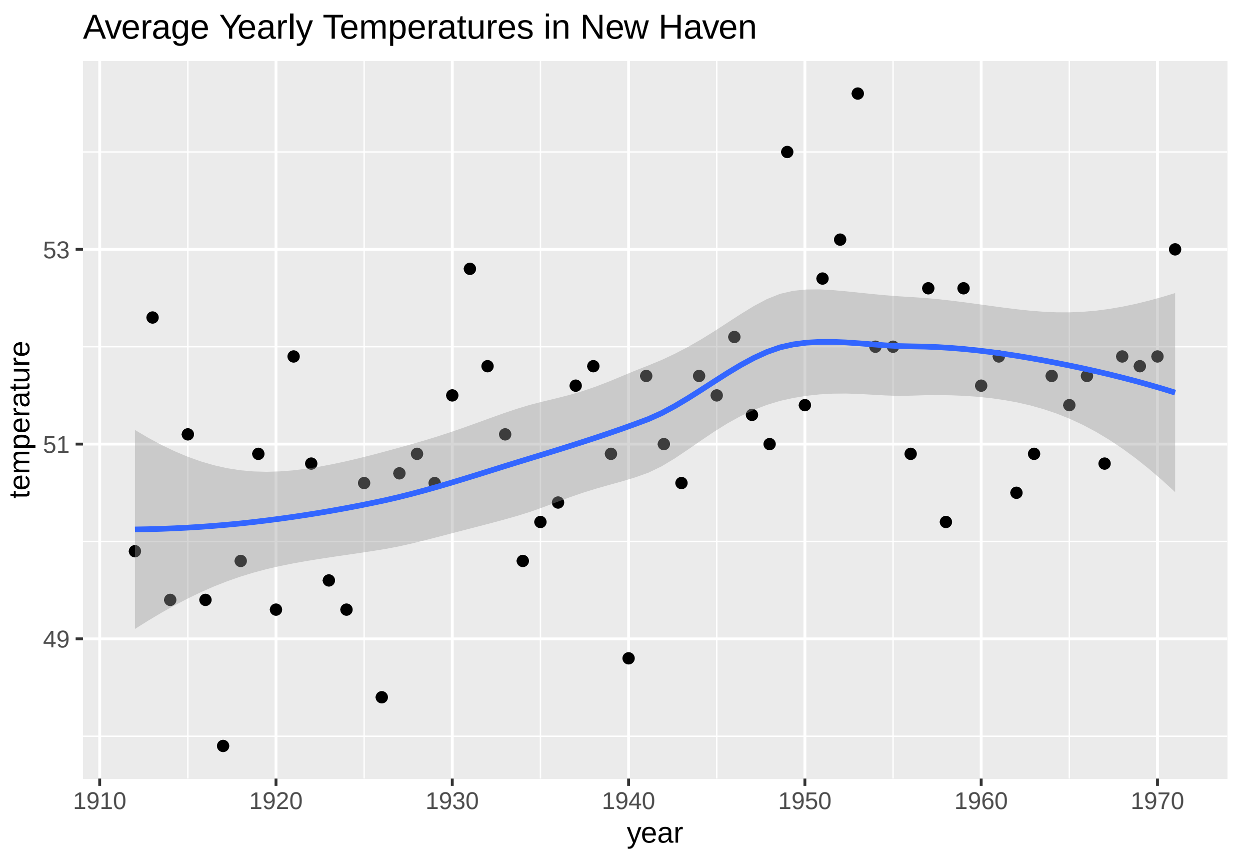
- We can use statistical theory to compute the probability that a given interval contains the true parameter p.
- 95% confidence intervals are intervals constructed to have a 95% chance of including p. The margin of error is approximately a 95% confidence interval.
- The start and end of these confidence intervals are random variables.
- To calculate any size confidence interval, we need to calculate the value z for which $Pr(-z \leq Z \leq z)$ equals the desired confidence. For example, a 99% confidence interval requires calculating z for $Pr(-z \leq Z \leq z) = 0.99$.
- For a confidence interval of size q, we solve for $z = 1 - \frac{1-q}{2}$.
- To determine a 95% confidence interval, use `z <- qnorm(0.975)`. This value is slightly smaller than 2 times the standard error.

Code: geom_smooth confidence interval example

The shaded area around the curve is related to the concept of confidence intervals.

```r
data("nhtemp")
data.frame(year = as.numeric(time(nhtemp)), temperature = as.numeric(nhtemp)) %>%
    ggplot(aes(year, temperature)) +
    geom_point() +
    geom_smooth() +
    ggtitle("Average Yearly Temperatures in New Haven")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Code: Monte Carlo simulation of confidence intervals

Note that to compute the exact 95% confidence interval, we would use `qnorm(.975)*SE_hat` instead of `2*SE_hat`.

```
p <- 0.45
N <- 1000
X <- sample(c(0,1), size = N, replace = TRUE, prob = c(1-p, p))    # generate N observations
X_hat <- mean(X)    # calculate X_hat
SE_hat <- sqrt(X_hat*(1-X_hat)/N)    # calculate SE_hat, SE of the mean of N observations
c(X_hat - 2*SE_hat, X_hat + 2*SE_hat)    # build interval of 2*SE above and below mean
```

```
## [1] 0.4135691 0.4764309
```

Code: Solving for z with `qnorm`

```
z <- qnorm(0.995)    # calculate z to solve for 99% confidence interval
pnorm(qnorm(0.995))    # demonstrating that qnorm gives the z value for a given probability
```

```
## [1] 0.995
```

```
pnorm(qnorm(1-0.995))    # demonstrating symmetry of 1-qnorm
```

```
## [1] 0.005
```

```
pnorm(z) - pnorm(-z)    # demonstrating that this z value gives correct probability for interval
```

```
## [1] 0.99
```

A Monte Carlo Simulation for Confidence Intervals

The textbook for this section is available here

Key points

- We can run a Monte Carlo simulation to confirm that a 95% confidence interval contains the true value of p 95% of the time.
- A plot of confidence intervals from this simulation demonstrates that most intervals include p, but roughly 5% of intervals miss the true value of p.

Code: Monte Carlo simulation

Note that to compute the exact 95% confidence interval, we would use `qnorm(.975)*SE_hat` instead of `2*SE_hat`.

```
B <- 10000
inside <- replicate(B, {
    X <- sample(c(0,1), size = N, replace = TRUE, prob = c(1-p, p))
    X_hat <- mean(X)
    SE_hat <- sqrt(X_hat*(1-X_hat)/N)
    between(p, X_hat - 2*SE_hat, X_hat + 2*SE_hat)    # TRUE if p in confidence interval
})
mean(inside)
```

```
## [1] 0.9566
```

The Correct Language

The textbook for this section is available here

Key points

- The 95% confidence intervals are random, but p is not random.
- 95% refers to the probability that the random interval falls on top of p.
- It is technically incorrect to state that p has a 95% chance of being in between two values because that implies p is random.

Power

The textbook for this section is available here

Key points

- If we are trying to predict the result of an election, then a confidence interval that includes a spread of 0 (a tie) is not helpful.

- A confidence interval that includes a spread of 0 does not imply a close election, it means the sample size is too small.
- Power is the probability of detecting an effect when there is a true effect to find. Power increases as sample size increases, because larger sample size means smaller standard error.

Code: Confidence interval for the spread with sample size of 25

Note that to compute the exact 95% confidence interval, we would use `c(-qnorm(.975), qnorm(.975))` instead of 1.96.

```
N <- 25
X_hat <- 0.48
(2*X_hat - 1) + c(-2, 2)*2*sqrt(X_hat*(1-X_hat)/N)
```

```
## [1] -0.4396799  0.3596799
```

p-Values

The textbook for this section is available here

Key points

- The null hypothesis is the hypothesis that there is no effect. In this case, the null hypothesis is that the spread is 0, or $p = 0.5$.
- The p-value is the probability of detecting an effect of a certain size or larger when the null hypothesis is true.
- We can convert the probability of seeing an observed value under the null hypothesis into a standard normal random variable. We compute the value of z that corresponds to the observed result, and then use that z to compute the p-value.
- If a 95% confidence interval does not include our observed value, then the p-value must be smaller than 0.05.
- It is preferable to report confidence intervals instead of p-values, as confidence intervals give information about the size of the estimate and p-values do not.

Code: Computing a p-value for observed spread of 0.02

```
N <- 100    # sample size
z <- sqrt(N) * 0.02/0.5    # spread of 0.02
1 - (pnorm(z) - pnorm(-z))
```

```
## [1] 0.6891565
```

Another Explanation of p-Values

The p-value is the probability of observing a value as extreme or more extreme than the result given that the null hypothesis is true.

In the context of the normal distribution, this refers to the probability of observing a Z-score whose absolute value is as high or higher than the Z-score of interest.

Suppose we want to find the p-value of an observation 2 standard deviations larger than the mean. This means we are looking for anything with $|z| \geq 2$.

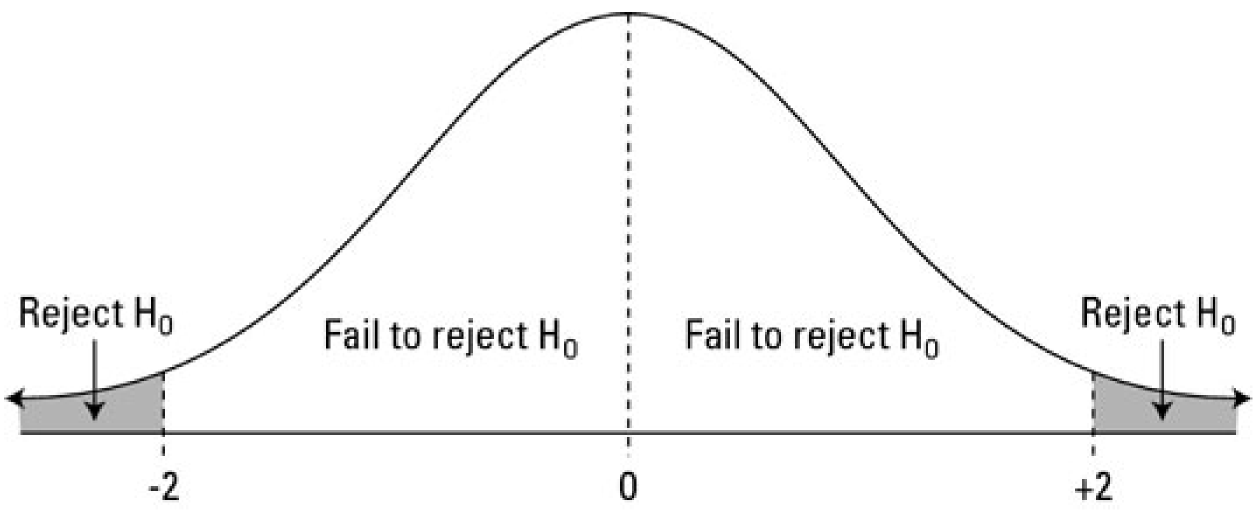


Figure 1: Standard normal distribution (centered at z=0 with a standard deviation of 1

Graphically, the p-value gives the probability of an observation that's at least as far away from the mean or further. This plot shows a standard normal distribution (centered at $z = 0$) with a standard deviation of 1). The shaded tails are the region of the graph that are 2 standard deviations or more away from the mean.

The p-value is the proportion of area under a normal curve that has z-scores as extreme or more extreme than the given value - the tails on this plot of a normal distribution are shaded to show the region corresponding to the p-value.

The right tail can be found with `1-pnorm(2)`. We want to have both tails, though, because we want to find the probability of any observation as far away from the mean or farther, in either direction. (This is what's meant by a two-tailed p-value.) Because the distribution is symmetrical, the right and left tails are the same size and we know that our desired value is just `2*(1-pnorm(2))`.

Recall that, by default, `pnorm()` gives the CDF for a normal distribution with a mean of $\mu = 0$ and standard deviation of $\sigma = 1$. To find p-values for a given z-score z in a normal distribution with mean `mu` and standard deviation `sigma`, use `2*(1-pnorm(z, mu, sigma))` instead.

Assessment - Confidence Intervals and p-Values

1. For the following exercises, we will use actual poll data from the 2016 election.

The exercises will contain pre-loaded data from the `dslabs` package.

```r
library(dslabs)
data("polls_us_election_2016")
```

We will use all the national polls that ended within a few weeks before the election.

Assume there are only two candidates and construct a 95% confidence interval for the election night proportion p.

```r
# Load the data
data(polls_us_election_2016)

# Generate an object `polls` that contains data filtered for polls that ended on or after October 31, 2
```

```r
polls <- filter(polls_us_election_2016, enddate >= "2016-10-31" & state == "U.S.")

# How many rows does `polls` contain? Print this value to the console.
nrow(polls)
```

```
## [1] 70
```

```r
# Assign the sample size of the first poll in `polls` to a variable called `N`. Print this value to the
N <- head(polls$samplesize,1)
N
```

```
## [1] 2220
```

```r
# For the first poll in `polls`, assign the estimated percentage of Clinton voters to a variable called
X_hat <- (head(polls$rawpoll_clinton,1)/100)
X_hat
```

```
## [1] 0.47
```

```r
# Calculate the standard error of `X_hat` and save it to a variable called `se_hat`. Print this value t
se_hat <- sqrt(X_hat*(1-X_hat)/N)
se_hat
```

```
## [1] 0.01059279
```

```r
# Use `qnorm` to calculate the 95% confidence interval for the proportion of Clinton voters. Save the l
ci <- c(X_hat - qnorm(0.975)*se_hat, X_hat + qnorm(0.975)*se_hat)
ci
```

```
## [1] 0.4492385 0.4907615
```

2. Create a new object called `pollster_results` that contains the pollster's name, the end date of the poll, the proportion of voters who declared a vote for Clinton, the standard error of this estimate, and the lower and upper bounds of the confidence interval for the estimate.

```r
# The `polls` object that filtered all the data by date and nation has already been loaded. Examine it
head(polls)
```

```
##   state  startdate    enddate
## 1  U.S. 2016-11-03 2016-11-06
## 2  U.S. 2016-11-01 2016-11-07
## 3  U.S. 2016-11-02 2016-11-06
## 4  U.S. 2016-11-04 2016-11-07
## 5  U.S. 2016-11-03 2016-11-06
## 6  U.S. 2016-11-03 2016-11-06
##                                                    pollster grade samplesize
## 1                                  ABC News/Washington Post    A+       2220
## 2                                   Google Consumer Surveys     B      26574
## 3                                                     Ipsos    A-       2195
## 4                                                     YouGov     B       3677
## 5                                           Gravis Marketing    B-      16639
## 6 Fox News/Anderson Robbins Research/Shaw & Company Research     A       1295
##   population rawpoll_clinton rawpoll_trump rawpoll_johnson rawpoll_mcmullin
## 1         lv           47.00         43.00            4.00               NA
## 2         lv           38.03         35.69            5.46               NA
## 3         lv           42.00         39.00            6.00               NA
## 4         lv           45.00         41.00            5.00               NA
## 5         rv           47.00         43.00            3.00               NA
## 6         lv           48.00         44.00            3.00               NA
##   adjpoll_clinton adjpoll_trump adjpoll_johnson adjpoll_mcmullin
## 1        45.20163      41.72430        4.626221               NA
## 2        43.34557      41.21439        5.175792               NA
## 3        42.02638      38.81620        6.844734               NA
## 4        45.65676      40.92004        6.069454               NA
## 5        46.84089      42.33184        3.726098               NA
## 6        49.02208      43.95631        3.057876               NA
```

```r
# Create a new object called `pollster_results` that contains columns for pollster name, end date, X_ha
polls <- mutate(polls, X_hat = polls$rawpoll_clinton/100, se_hat = sqrt(X_hat*(1-X_hat)/polls$samplesiz
pollster_results <- select(polls, pollster, enddate, X_hat, se_hat, lower, upper)
pollster_results
```

```
##                                                      pollster    enddate  X_hat
## 1                                    ABC News/Washington Post 2016-11-06 0.4700
## 2                                     Google Consumer Surveys 2016-11-07 0.3803
## 3                                                       Ipsos 2016-11-06 0.4200
## 4                                                      YouGov 2016-11-07 0.4500
## 5                                            Gravis Marketing 2016-11-06 0.4700
## 6  Fox News/Anderson Robbins Research/Shaw & Company Research 2016-11-06 0.4800
## 7                                      CBS News/New York Times 2016-11-06 0.4500
## 8                                 NBC News/Wall Street Journal 2016-11-05 0.4400
## 9                                                    IBD/TIPP 2016-11-07 0.4120
## 10                                           Selzer & Company 2016-11-06 0.4400
## 11                                          Angus Reid Global 2016-11-04 0.4800
## 12                                        Monmouth University 2016-11-06 0.5000
## 13                                             Marist College 2016-11-03 0.4400
## 14                                   The Times-Picayune/Lucid 2016-11-07 0.4500
## 15                                       USC Dornsife/LA Times 2016-11-07 0.4361
## 16                        RKM Research and Communications, Inc. 2016-11-05 0.4760
## 17                                         CVOTER International 2016-11-06 0.4891
## 18                                            Morning Consult 2016-11-05 0.4500
## 19                                               SurveyMonkey 2016-11-06 0.4700
## 20                       Rasmussen Reports/Pulse Opinion Research 2016-11-06 0.4500
## 21                                              Insights West 2016-11-07 0.4900
## 22                                  RAND (American Life Panel) 2016-11-01 0.4370
## 23 Fox News/Anderson Robbins Research/Shaw & Company Research 2016-11-03 0.4550
## 24                                     CBS News/New York Times 2016-11-01 0.4500
## 25                                    ABC News/Washington Post 2016-11-05 0.4700
## 26                                                       Ipsos 2016-11-04 0.4300
## 27                                    ABC News/Washington Post 2016-11-04 0.4800
## 28                                                     YouGov 2016-11-06 0.4290
## 29                                                   IBD/TIPP 2016-11-06 0.4070
## 30                                    ABC News/Washington Post 2016-11-03 0.4700
## 31                                             IBD/TIPP 2016-11-03 0.4440
## 32                                             IBD/TIPP 2016-11-05 0.4300
## 33                              ABC News/Washington Post 2016-11-02 0.4700
## 34                              ABC News/Washington Post 2016-11-01 0.4700
## 35                              ABC News/Washington Post 2016-10-31 0.4600
## 36                                                Ipsos 2016-11-03 0.4320
## 37                                             IBD/TIPP 2016-11-04 0.4420
## 38                                               YouGov 2016-11-01 0.4600
## 39                                             IBD/TIPP 2016-10-31 0.4460
## 40                                                Ipsos 2016-11-02 0.4550
## 41                   Rasmussen Reports/Pulse Opinion Research 2016-11-03 0.4400
## 42                              The Times-Picayune/Lucid 2016-11-06 0.4500
## 43                                                Ipsos 2016-11-01 0.4470
## 44                                             IBD/TIPP 2016-11-02 0.4400
## 45                                             IBD/TIPP 2016-11-01 0.4400
## 46                   Rasmussen Reports/Pulse Opinion Research 2016-11-02 0.4200
## 47                                                Ipsos 2016-10-31 0.4400
## 48                              The Times-Picayune/Lucid 2016-11-05 0.4500
## 49                   Rasmussen Reports/Pulse Opinion Research 2016-10-31 0.4400
## 50                               Google Consumer Surveys 2016-10-31 0.3769
## 51                                  CVOTER International 2016-11-05 0.4925
## 52                   Rasmussen Reports/Pulse Opinion Research 2016-11-01 0.4400
## 53                                  CVOTER International 2016-11-04 0.4906
## 54                              The Times-Picayune/Lucid 2016-11-04 0.4500
## 55                                 USC Dornsife/LA Times 2016-11-06 0.4323
## 56                                  CVOTER International 2016-11-03 0.4853
## 57                              The Times-Picayune/Lucid 2016-11-03 0.4400
## 58                                 USC Dornsife/LA Times 2016-11-05 0.4263
## 59                                  CVOTER International 2016-11-02 0.4878
## 60                                 USC Dornsife/LA Times 2016-11-04 0.4256
## 61                                  CVOTER International 2016-11-01 0.4881
## 62                              The Times-Picayune/Lucid 2016-11-02 0.4400
## 63                                      Gravis Marketing 2016-10-31 0.4600
## 64                                 USC Dornsife/LA Times 2016-11-03 0.4338
## 65                              The Times-Picayune/Lucid 2016-11-01 0.4300
## 66                                 USC Dornsife/LA Times 2016-11-02 0.4247
## 67                                      Gravis Marketing 2016-11-02 0.4700
## 68                                 USC Dornsife/LA Times 2016-11-01 0.4236
## 69                              The Times-Picayune/Lucid 2016-10-31 0.4200
## 70                                 USC Dornsife/LA Times 2016-10-31 0.4328
##         se_hat     lower     upper
## 1  0.010592790 0.4492385 0.4907615
## 2  0.002978005 0.3744632 0.3861368
## 3  0.010534681 0.3993524 0.4406476
## 4  0.008204286 0.4339199 0.4660801
## 5  0.003869218 0.4624165 0.4775835
## 6  0.013883131 0.4527896 0.5072104
## 7  0.013174309 0.4241788 0.4758212
## 8  0.013863610 0.4128278 0.4671722
## 9  0.014793245 0.3830058 0.4409942
## 10 0.017560908 0.4055813 0.4744187
## 11 0.014725994 0.4511376 0.5088624
## 12 0.018281811 0.4641683 0.5358317
## 13 0.016190357 0.4082675 0.4717325
## 14 0.009908346 0.4305800 0.4694200
## 15 0.009096403 0.4182714 0.4539286
## 16 0.015722570 0.4451843 0.5068157
## 17 0.012400526 0.4647954 0.5134046
## 18 0.012923005 0.4246714 0.4753286
## 19 0.001883809 0.4663078 0.4736922
## 20 0.012845233 0.4248238 0.4751762
## 21 0.016304940 0.4580429 0.5219571
## 22 0.010413043 0.4165908 0.4574092
## 23 0.014966841 0.4256655 0.4843345
## 24 0.016339838 0.4179745 0.4820255
## 25 0.011340235 0.4477735 0.4922265
## 26 0.010451057 0.4095163 0.4504837
## 27 0.012170890 0.4561455 0.5038545
## 28 0.001704722 0.4256588 0.4323412
## 29 0.015337369 0.3769393 0.4370607
## 30 0.013249383 0.4440317 0.4959683
## 31 0.016580236 0.4115033 0.4764967
## 32 0.016475089 0.3977094 0.4622906
## 33 0.014711237 0.4411665 0.4988335
## 34 0.014610041 0.4413648 0.4986352
## 35 0.014496630 0.4315871 0.4884129
## 36 0.011018764 0.4104036 0.4535964
## 37 0.017514599 0.4076720 0.4763280
## 38 0.014193655 0.4321809 0.4878191
## 39 0.015579317 0.4154651 0.4765349
## 40 0.011358664 0.4327374 0.4772626
## 41 0.012816656 0.4148798 0.4651202
## 42 0.009786814 0.4308182 0.4691818
## 43 0.011810940 0.4238510 0.4701490
## 44 0.016858185 0.4069586 0.4730414
## 45 0.016907006 0.4068629 0.4731371
## 46 0.012743626 0.3950230 0.4449770
## 47 0.012973281 0.4145728 0.4654272
## 48 0.009898535 0.4305992 0.4694008
## 49 0.012816656 0.4148798 0.4651202
## 50 0.003107749 0.3708089 0.3829911
## 51 0.012609413 0.4677860 0.5172140
## 52 0.012816656 0.4148798 0.4651202
## 53 0.012998976 0.4651225 0.5160775
## 54 0.009830663 0.4307323 0.4692677
## 55 0.009144248 0.4143776 0.4502224
## 56 0.013381202 0.4590733 0.5115267
## 57 0.009684792 0.4210182 0.4589818
## 58 0.009047108 0.4085680 0.4440320
## 59 0.013711286 0.4609264 0.5146736
## 60 0.009046705 0.4078688 0.4433312
## 61 0.013441133 0.4617559 0.5144441
## 62 0.009697721 0.4209928 0.4590072
## 63 0.006807590 0.4466574 0.4733426
## 64 0.009106200 0.4159522 0.4516478
## 65 0.009677647 0.4110322 0.4489678
## 66 0.009119319 0.4068265 0.4425735
## 67 0.010114336 0.4501763 0.4898237
## 68 0.009015504 0.4059299 0.4412701
## 69 0.009679479 0.4010286 0.4389714
## 70 0.008834895 0.4154839 0.4501161
```

3. The final tally for the popular vote was Clinton 48.2% and Trump 46.1%. Add a column called `hit` to `pollster_results` that states if the confidence interval included the true proportion $p = 0.482$ or not. What proportion of confidence intervals included p?

```
# The `pollster_results` object has already been loaded. Examine it using the `head` function.
head(pollster_results)
```

```
##                                                     pollster    enddate  X_hat
## 1                                   ABC News/Washington Post 2016-11-06 0.4700
## 2                                    Google Consumer Surveys 2016-11-07 0.3803
## 3                                                      Ipsos 2016-11-06 0.4200
## 4                                                     YouGov 2016-11-07 0.4500
## 5                                           Gravis Marketing 2016-11-06 0.4700
## 6 Fox News/Anderson Robbins Research/Shaw & Company Research 2016-11-06 0.4800
##        se_hat     lower     upper
## 1 0.010592790 0.4492385 0.4907615
## 2 0.002978005 0.3744632 0.3861368
## 3 0.010534681 0.3993524 0.4406476
## 4 0.008204286 0.4339199 0.4660801
## 5 0.003869218 0.4624165 0.4775835
## 6 0.013883131 0.4527896 0.5072104
```

```
# Add a logical variable called `hit` that indicates whether the actual value exists within the confide
avg_hit <- pollster_results %>% mutate(hit=(lower<0.482 & upper>0.482)) %>% summarize(mean(hit))
avg_hit
```

```
##   mean(hit)
## 1 0.3142857
```

4. If these confidence intervals are constructed correctly, and the theory holds up, what proportion of confidence intervals should include p?

- ☐ A. 0.05
- ☐ B. 0.31
- ☐ C. 0.50
- [✓ D. 0.95]

5. A much smaller proportion of the polls than expected produce confidence intervals containing p.

Notice that most polls that fail to include p are underestimating. The rationale for this is that undecided voters historically divide evenly between the two main candidates on election day.

In this case, it is more informative to estimate the spread or the difference between the proportion of two candidates d, or $0.482 - 0.461 = 0.021$ for this election.

Assume that there are only two parties and that $d = 2p - 1$. Construct a 95% confidence interval for difference in proportions on election night.

```r
# Add a statement to this line of code that will add a new column named `d_hat` to `polls`. The new col
polls <- polls_us_election_2016 %>% filter(enddate >= "2016-10-31" & state == "U.S.")  %>%
  mutate(d_hat = rawpoll_clinton/100 - rawpoll_trump/100)

# Assign the sample size of the first poll in `polls` to a variable called `N`. Print this value to the
N <- polls$samplesize[1]
N
```

```
## [1] 2220
```

```r
# Assign the difference `d_hat` of the first poll in `polls` to a variable called `d_hat`. Print this v
d_hat <- polls$d_hat[1]
d_hat
```

```
## [1] 0.04
```

```r
# Assign proportion of votes for Clinton to the variable `X_hat`.
X_hat <- (d_hat+1)/2
X_hat
```

```
## [1] 0.52
```

```r
# Calculate the standard error of the spread and save it to a variable called `se_hat`. Print this valu
se_hat <- 2*sqrt(X_hat*(1-X_hat)/N)
se_hat
```

```
## [1] 0.02120683
```

```r
# Use `qnorm` to calculate the 95% confidence interval for the difference in the proportions of voters.
ci <- c(d_hat - qnorm(0.975)*se_hat, d_hat + qnorm(0.975)*se_hat)
ci
```

```
## [1] -0.001564627  0.081564627
```

6. Create a new object called `pollster_results` that contains the pollster's name, the end date of the poll, the difference in the proportion of voters who declared a vote either, and the lower and upper bounds of the confidence interval for the estimate.

```r
# The subset `polls` data with 'd_hat' already calculated has been loaded. Examine it using the `head`
head(polls)
```

```
##   state  startdate    enddate
## 1  U.S. 2016-11-03 2016-11-06
## 2  U.S. 2016-11-01 2016-11-07
## 3  U.S. 2016-11-02 2016-11-06
## 4  U.S. 2016-11-04 2016-11-07
## 5  U.S. 2016-11-03 2016-11-06
## 6  U.S. 2016-11-03 2016-11-06
##                                                     pollster grade samplesize
## 1                                   ABC News/Washington Post    A+       2220
## 2                                      Google Consumer Surveys     B      26574
## 3                                                        Ipsos    A-       2195
## 4                                                       YouGov     B       3677
## 5                                             Gravis Marketing    B-      16639
## 6 Fox News/Anderson Robbins Research/Shaw & Company Research     A       1295
##   population rawpoll_clinton rawpoll_trump rawpoll_johnson rawpoll_mcmullin
## 1         lv           47.00         43.00            4.00               NA
## 2         lv           38.03         35.69            5.46               NA
## 3         lv           42.00         39.00            6.00               NA
## 4         lv           45.00         41.00            5.00               NA
## 5         rv           47.00         43.00            3.00               NA
## 6         lv           48.00         44.00            3.00               NA
##   adjpoll_clinton adjpoll_trump adjpoll_johnson adjpoll_mcmullin  d_hat
## 1        45.20163      41.72430        4.626221               NA 0.0400
## 2        43.34557      41.21439        5.175792               NA 0.0234
## 3        42.02638      38.81620        6.844734               NA 0.0300
## 4        45.65676      40.92004        6.069454               NA 0.0400
## 5        46.84089      42.33184        3.726098               NA 0.0400
## 6        49.02208      43.95631        3.057876               NA 0.0400
```

```r
# Create a new object called `pollster_results` that contains columns for pollster name, end date, d_ha
d_hat = polls$rawpoll_clinton/100 - polls$rawpoll_trump/100
X_hat = (d_hat + 1) / 2
polls <- mutate(polls, X_hat, se_hat = 2 * sqrt(X_hat * (1 - X_hat) / samplesize), lower = d_hat - qnor
pollster_results <- select(polls, pollster, enddate, d_hat, lower, upper)
pollster_results
```

```
##                                                      pollster    enddate
## 1                                    ABC News/Washington Post 2016-11-06
## 2                                     Google Consumer Surveys 2016-11-07
## 3                                                       Ipsos 2016-11-06
## 4                                                      YouGov 2016-11-07
## 5                                            Gravis Marketing 2016-11-06
## 6  Fox News/Anderson Robbins Research/Shaw & Company Research 2016-11-06
## 7                                     CBS News/New York Times 2016-11-06
## 8                                NBC News/Wall Street Journal 2016-11-05
## 9                                                    IBD/TIPP 2016-11-07
## 10                                           Selzer & Company 2016-11-06
## 11                                          Angus Reid Global 2016-11-04
## 12                                        Monmouth University 2016-11-06
## 13                                             Marist College 2016-11-03
## 14                                 The Times-Picayune/Lucid 2016-11-07
## 15                                    USC Dornsife/LA Times 2016-11-07
## 16                      RKM Research and Communications, Inc. 2016-11-05
## 17                                      CVOTER International 2016-11-06
## 18                                           Morning Consult 2016-11-05
## 19                                               SurveyMonkey 2016-11-06
## 20                   Rasmussen Reports/Pulse Opinion Research 2016-11-06
## 21                                             Insights West 2016-11-07
## 22                                RAND (American Life Panel) 2016-11-01
## 23 Fox News/Anderson Robbins Research/Shaw & Company Research 2016-11-03
## 24                                    CBS News/New York Times 2016-11-01
## 25                                   ABC News/Washington Post 2016-11-05
## 26                                                      Ipsos 2016-11-04
## 27                                  ABC News/Washington Post 2016-11-04
## 28                                                    YouGov 2016-11-06
## 29                                                  IBD/TIPP 2016-11-06
## 30                                  ABC News/Washington Post 2016-11-03
## 31                                                  IBD/TIPP 2016-11-03
## 32                                                  IBD/TIPP 2016-11-05
## 33                                  ABC News/Washington Post 2016-11-02
## 34                                  ABC News/Washington Post 2016-11-01
## 35                                  ABC News/Washington Post 2016-10-31
## 36                                                     Ipsos 2016-11-03
## 37                                                  IBD/TIPP 2016-11-04
## 38                                                    YouGov 2016-11-01
## 39                                                  IBD/TIPP 2016-10-31
## 40                                                     Ipsos 2016-11-02
## 41                Rasmussen Reports/Pulse Opinion Research 2016-11-03
## 42                                The Times-Picayune/Lucid 2016-11-06
## 43                                                     Ipsos 2016-11-01
## 44                                                  IBD/TIPP 2016-11-02
## 45                                                  IBD/TIPP 2016-11-01
## 46                Rasmussen Reports/Pulse Opinion Research 2016-11-02
## 47                                                     Ipsos 2016-10-31
## 48                                The Times-Picayune/Lucid 2016-11-05
## 49                Rasmussen Reports/Pulse Opinion Research 2016-10-31
## 50                                 Google Consumer Surveys 2016-10-31
## 51                                     CVOTER International 2016-11-05
## 52                Rasmussen Reports/Pulse Opinion Research 2016-11-01
## 53                                     CVOTER International 2016-11-04
## 54                                The Times-Picayune/Lucid 2016-11-04
## 55                                    USC Dornsife/LA Times 2016-11-06
## 56                                     CVOTER International 2016-11-03
## 57                                The Times-Picayune/Lucid 2016-11-03
## 58                                    USC Dornsife/LA Times 2016-11-05
## 59                                     CVOTER International 2016-11-02
## 60                                    USC Dornsife/LA Times 2016-11-04
## 61                                     CVOTER International 2016-11-01
## 62                                The Times-Picayune/Lucid 2016-11-02
## 63                                         Gravis Marketing 2016-10-31
## 64                                    USC Dornsife/LA Times 2016-11-03
## 65                                The Times-Picayune/Lucid 2016-11-01
## 66                                    USC Dornsife/LA Times 2016-11-02
## 67                                         Gravis Marketing 2016-11-02
## 68                                    USC Dornsife/LA Times 2016-11-01
## 69                                The Times-Picayune/Lucid 2016-10-31
## 70                                    USC Dornsife/LA Times 2016-10-31
##      d_hat        lower         upper
## 1   0.0400 -0.001564627  0.0815646272
## 2   0.0234  0.011380104  0.0354198955
## 3   0.0300 -0.011815309  0.0718153088
## 4   0.0400  0.007703641  0.0722963589
## 5   0.0400  0.024817728  0.0551822719
## 6   0.0400 -0.014420872  0.0944208716
## 7   0.0400 -0.011860967  0.0918609675
## 8   0.0400 -0.014696100  0.0946961005
## 9  -0.0150 -0.073901373  0.0439013728
## 10  0.0300 -0.039307332  0.0993073320
## 11  0.0400 -0.017724837  0.0977248370
## 12  0.0600 -0.011534270  0.1315342703
## 13  0.0100 -0.053923780  0.0739237800
## 14  0.0500  0.011013152  0.0889868476
## 15 -0.0323 -0.068233293  0.0036332933
## 16  0.0320 -0.029670864  0.0936708643
## 17  0.0278 -0.020801931  0.0764019309
## 18  0.0300 -0.020889533  0.0808895334
## 19  0.0600  0.052615604  0.0673843956
## 20  0.0200 -0.030595930  0.0705959303
## 21  0.0400 -0.023875814  0.1038758144
## 22  0.0910  0.050024415  0.1319755848
## 23  0.0150 -0.043901373  0.0739013728
## 24  0.0300 -0.034344689  0.0943446886
## 25  0.0400 -0.004497495  0.0844974954
## 26  0.0400 -0.001341759  0.0813417590
## 27  0.0500  0.002312496  0.0976875039
## 28  0.0390  0.032254341  0.0457456589
## 29 -0.0240 -0.085171524  0.0371715236
## 30  0.0400 -0.011988727  0.0919887265
## 31  0.0050 -0.060404028  0.0704040277
## 32 -0.0100 -0.075220256  0.0552202561
## 33  0.0300 -0.027745070  0.0877450695
## 34  0.0200 -0.037362198  0.0773621983
## 35  0.0000 -0.057008466  0.0570084657
## 36  0.0490  0.005454535  0.0925454654
## 37  0.0050 -0.064121736  0.0741217361
## 38  0.0300 -0.025791885  0.0857918847
## 39  0.0090 -0.052426619  0.0704266191
## 40  0.0820  0.037443982  0.1265560180
## 41  0.0000 -0.050606052  0.0506060525
## 42  0.0500  0.011491350  0.0885086495
## 43  0.0730  0.026563876  0.1194361238
## 44 -0.0010 -0.067563833  0.0655638334
## 45 -0.0040 -0.070756104  0.0627561042
## 46 -0.0300 -0.080583275  0.0205832746
## 47  0.0680  0.016894089  0.1191059111
## 48  0.0500  0.011051757  0.0889482429
## 49  0.0000 -0.050606052  0.0506060525
## 50  0.0262  0.013635277  0.0387647229
## 51  0.0333 -0.016106137  0.0827061365
## 52  0.0000 -0.050606052  0.0506060525
## 53  0.0124 -0.038560140  0.0633601401
## 54  0.0600  0.021340150  0.0986598497
## 55 -0.0475 -0.083637121 -0.0113628793
## 56  0.0009 -0.051576011  0.0533760106
## 57  0.0500  0.011807815  0.0881921851
## 58 -0.0553 -0.091100799 -0.0194992008
## 59  0.0056 -0.048162418  0.0593624177
## 60 -0.0540 -0.089809343 -0.0181906570
## 61  0.0067 -0.046002019  0.0594020190
## 62  0.0500  0.011756829  0.0882431712
## 63  0.0100 -0.016769729  0.0367697294
## 64 -0.0351 -0.071090499  0.0008904993
## 65  0.0300 -0.008295761  0.0682957614
## 66 -0.0503 -0.086413709 -0.0141862908
## 67  0.0200 -0.019711083  0.0597110831
## 68 -0.0547 -0.090406512 -0.0189934879
## 69  0.0200 -0.018430368  0.0584303678
## 70 -0.0362 -0.071126335 -0.0012736645
```

7. What proportion of confidence intervals for the difference between the proportion of voters included d, the actual difference in election day?

```r
# The `pollster_results` object has already been loaded. Examine it using the `head` function.
head(pollster_results)
```

```
##                                                     pollster    enddate  d_hat
## 1                                   ABC News/Washington Post 2016-11-06 0.0400
## 2                                    Google Consumer Surveys 2016-11-07 0.0234
## 3                                                      Ipsos 2016-11-06 0.0300
## 4                                                     YouGov 2016-11-07 0.0400
## 5                                            Gravis Marketing 2016-11-06 0.0400
## 6 Fox News/Anderson Robbins Research/Shaw & Company Research 2016-11-06 0.0400
##          lower      upper
## 1 -0.001564627 0.08156463
## 2  0.011380104 0.03541990
## 3 -0.011815309 0.07181531
## 4  0.007703641 0.07229636
## 5  0.024817728 0.05518227
## 6 -0.014420872 0.09442087
```

```r
# Add a logical variable called `hit` that indicates whether the actual value (0.021) exists within the
avg_hit <- pollster_results %>% mutate(hit=(lower<0.021 & upper>0.021)) %>% summarize(mean(hit))
avg_hit
```

```
##   mean(hit)
## 1 0.7714286
```

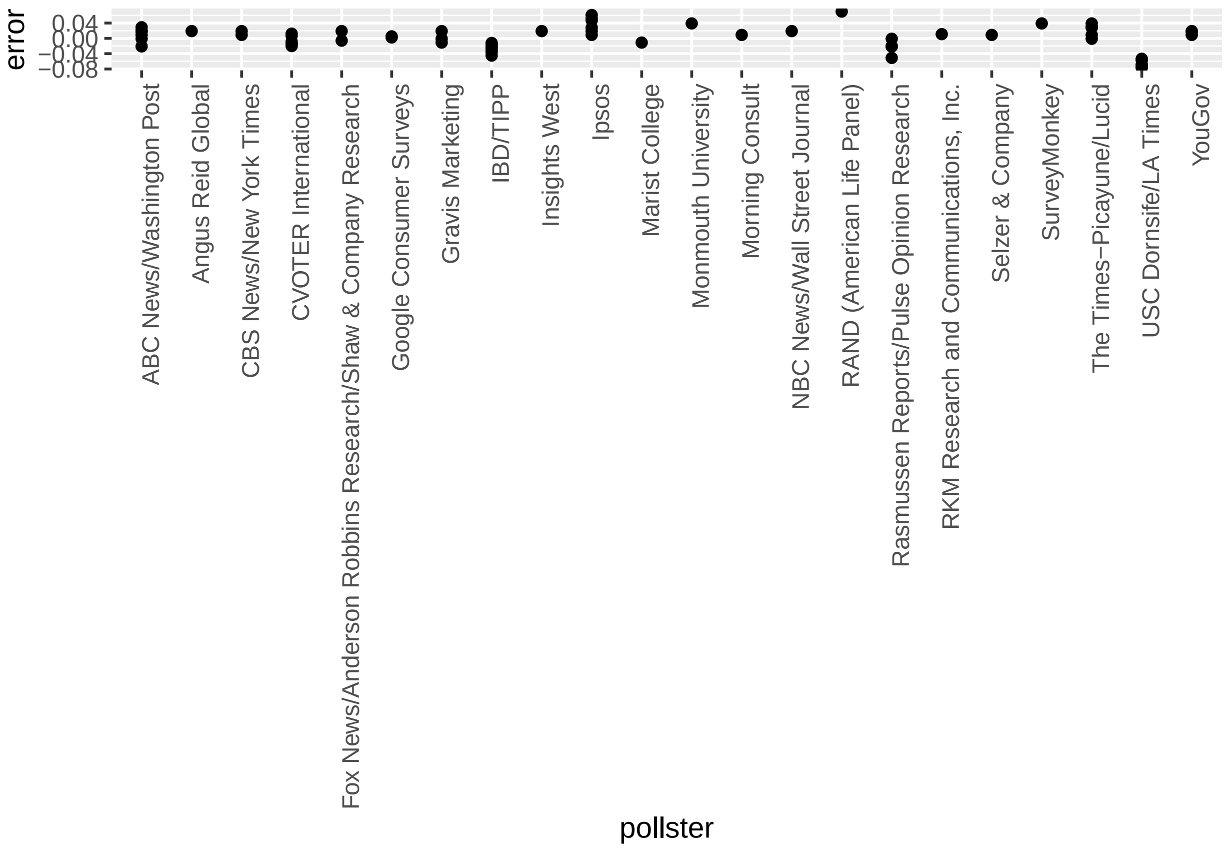
8. Although the proportion of confidence intervals that include the actual difference between the proportion of voters increases substantially, it is still lower that 0.95.

In the next chapter, we learn the reason for this. To motivate our next exercises, calculate the difference between each poll's estimate $\overline{d}$ and the actual $d = 0.021$. Stratify this difference, or error, by pollster in a plot.

```r
# The `polls` object has already been loaded. Examine it using the `head` function.
head(polls)
```

```
##   state  startdate    enddate
## 1  U.S. 2016-11-03 2016-11-06
## 2  U.S. 2016-11-01 2016-11-07
## 3  U.S. 2016-11-02 2016-11-06
## 4  U.S. 2016-11-04 2016-11-07
## 5  U.S. 2016-11-03 2016-11-06
## 6  U.S. 2016-11-03 2016-11-06
##                                                    pollster grade samplesize
## 1                                   ABC News/Washington Post    A+       2220
## 2                                    Google Consumer Surveys     B      26574
## 3                                                      Ipsos    A-       2195
## 4                                                     YouGov     B       3677
## 5                                           Gravis Marketing    B-      16639
## 6 Fox News/Anderson Robbins Research/Shaw & Company Research     A       1295
##   population rawpoll_clinton rawpoll_trump rawpoll_johnson rawpoll_mcmullin
## 1         lv           47.00         43.00            4.00               NA
## 2         lv           38.03         35.69            5.46               NA
## 3         lv           42.00         39.00            6.00               NA
## 4         lv           45.00         41.00            5.00               NA
## 5         rv           47.00         43.00            3.00               NA
## 6         lv           48.00         44.00            3.00               NA
##   adjpoll_clinton adjpoll_trump adjpoll_johnson adjpoll_mcmullin  d_hat  X_hat
## 1        45.20163      41.72430        4.626221               NA 0.0400 0.5200
## 2        43.34557      41.21439        5.175792               NA 0.0234 0.5117
## 3        42.02638      38.81620        6.844734               NA 0.0300 0.5150
## 4        45.65676      40.92004        6.069454               NA 0.0400 0.5200
## 5        46.84089      42.33184        3.726098               NA 0.0400 0.5200
## 6        49.02208      43.95631        3.057876               NA 0.0400 0.5200
##        se_hat        lower      upper
## 1 0.021206832 -0.001564627 0.08156463
## 2 0.006132712  0.011380104 0.03541990
## 3 0.021334733 -0.011815309 0.07181531
## 4 0.016478037  0.007703641 0.07229636
## 5 0.007746199  0.024817728 0.05518227
## 6 0.027766261 -0.014420872 0.09442087
```

```
# Add variable called `error` to the object `polls` that contains the difference between d_hat and the
error <- polls$d_hat - 0.021
polls <- mutate(polls, error)
polls %>% ggplot(aes(x = pollster, y = error)) +
  geom_point() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



9. Remake the plot you made for the previous exercise, but only for pollsters that took five or more polls.

You can use dplyr tools `group_by` and `n` to group data by a variable of interest and then count the number of observations in the groups. The function `filter` filters data piped into it by your specified condition.

For example:

```
data %>% group_by(variable_for_grouping)
    %>% filter(n() >= 5)
```

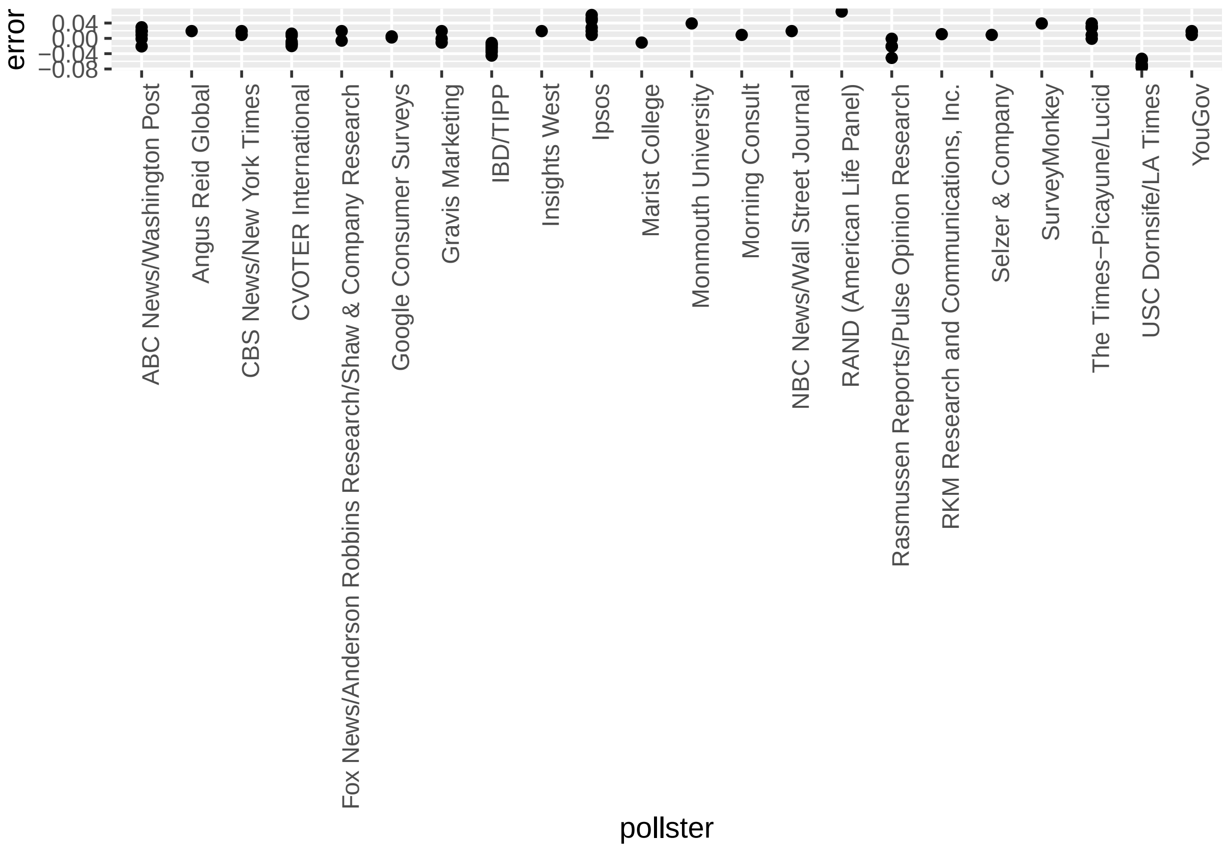
```
# The `polls` object has already been loaded. Examine it using the `head` function.
head(polls)
```

```
##   state  startdate    enddate
## 1  U.S. 2016-11-03 2016-11-06
## 2  U.S. 2016-11-01 2016-11-07
## 3  U.S. 2016-11-02 2016-11-06
## 4  U.S. 2016-11-04 2016-11-07
## 5  U.S. 2016-11-03 2016-11-06
## 6  U.S. 2016-11-03 2016-11-06
##                                                      pollster grade samplesize
## 1                                    ABC News/Washington Post    A+       2220
## 2                                     Google Consumer Surveys     B      26574
## 3                                                       Ipsos    A-       2195
## 4                                                      YouGov     B       3677
## 5                                            Gravis Marketing    B-      16639
## 6 Fox News/Anderson Robbins Research/Shaw & Company Research     A       1295
##   population rawpoll_clinton rawpoll_trump rawpoll_johnson rawpoll_mcmullin
## 1         lv           47.00         43.00            4.00               NA
## 2         lv           38.03         35.69            5.46               NA
## 3         lv           42.00         39.00            6.00               NA
## 4         lv           45.00         41.00            5.00               NA
## 5         rv           47.00         43.00            3.00               NA
## 6         lv           48.00         44.00            3.00               NA
##   adjpoll_clinton adjpoll_trump adjpoll_johnson adjpoll_mcmullin  d_hat  X_hat
## 1        45.20163      41.72430        4.626221               NA 0.0400 0.5200
## 2        43.34557      41.21439        5.175792               NA 0.0234 0.5117
## 3        42.02638      38.81620        6.844734               NA 0.0300 0.5150
## 4        45.65676      40.92004        6.069454               NA 0.0400 0.5200
## 5        46.84089      42.33184        3.726098               NA 0.0400 0.5200
## 6        49.02208      43.95631        3.057876               NA 0.0400 0.5200
##        se_hat         lower      upper  error
## 1 0.021206832 -0.001564627 0.08156463 0.0190
## 2 0.006132712  0.011380104 0.03541990 0.0024
## 3 0.021334733 -0.011815309 0.07181531 0.0090
## 4 0.016478037  0.007703641 0.07229636 0.0190
## 5 0.007746199  0.024817728 0.05518227 0.0190
## 6 0.027766261 -0.014420872 0.09442087 0.0190
```

```
# Add variable called `error` to the object `polls` that contains the difference between d_hat and the
error <- polls$d_hat - 0.021
polls <- mutate(polls, error)
polls %>% group_by(pollster) %>% filter(n() >= 5)
```

```
## # A tibble: 48 x 21
## # Groups:   pollster [7]
##    state startdate  enddate    pollster grade samplesize population
##    <fct> <date>     <date>     <fct>    <fct>      <int> <chr>
##  1 U.S.  2016-11-03 2016-11-06 ABC New~ A+          2220 lv
##  2 U.S.  2016-11-02 2016-11-06 Ipsos    A-          2195 lv
##  3 U.S.  2016-11-04 2016-11-07 IBD/TIPP A-          1107 lv
##  4 U.S.  2016-11-05 2016-11-07 The Tim~ <NA>        2521 lv
##  5 U.S.  2016-11-01 2016-11-07 USC Dor~ <NA>        2972 lv
##  6 U.S.  2016-10-31 2016-11-06 CVOTER ~ C+          1625 lv
##  7 U.S.  2016-11-02 2016-11-06 Rasmuss~ C+          1500 lv
##  8 U.S.  2016-11-02 2016-11-05 ABC New~ A+          1937 lv
##  9 U.S.  2016-10-31 2016-11-04 Ipsos    A-          2244 lv
## 10 U.S.  2016-11-01 2016-11-04 ABC New~ A+          1685 lv
## # ... with 38 more rows, and 14 more variables: rawpoll_clinton <dbl>,
## #   rawpoll_trump <dbl>, rawpoll_johnson <dbl>, rawpoll_mcmullin <dbl>,
## #   adjpoll_clinton <dbl>, adjpoll_trump <dbl>, adjpoll_johnson <dbl>,
## #   adjpoll_mcmullin <dbl>, d_hat <dbl>, X_hat <dbl>, se_hat <dbl>,
## #   lower <dbl>, upper <dbl>, error <dbl>
```

```
polls %>% ggplot(aes(x = pollster, y = error)) +
  geom_point() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



Section 4 Overview

In Section 4, you will look at statistical models in the context of election polling and forecasting.

After completing Section 4, you will be able to:

- Understand how aggregating data from different sources, as poll aggregators do for poll data, can improve the precision of a prediction.
- Understand how to fit a multilevel model to the data to forecast, for example, election results.
- Explain why a simple aggregation of data is insufficient to combine results because of factors such as pollster bias.
- Use a data-driven model to account for additional types of sampling variability such as pollster-to-pollster variability.

Poll Aggregators

The textbook for this section is available here and here

Key points

- Poll aggregators combine the results of many polls to simulate polls with a large sample size and therefore generate more precise estimates than individual polls.
- Polls can be simulated with a Monte Carlo simulation and used to construct an estimate of the spread and confidence intervals.
- The actual data science exercise of forecasting elections involves more complex statistical modeling, but these underlying ideas still apply.

Code: Simulating polls

Note that to compute the exact 95% confidence interval, we would use `qnorm(.975)*SE_hat` instead of `2*SE_hat`.

```
d <- 0.039
Ns <- c(1298, 533, 1342, 897, 774, 254, 812, 324, 1291, 1056, 2172, 516)
p <- (d+1)/2

# calculate confidence intervals of the spread
confidence_intervals <- sapply(Ns, function(N){
    X <- sample(c(0,1), size=N, replace=TRUE, prob = c(1-p, p))
    X_hat <- mean(X)
    SE_hat <- sqrt(X_hat*(1-X_hat)/N)
    2*c(X_hat, X_hat - 2*SE_hat, X_hat + 2*SE_hat) - 1
})

# generate a data frame storing results
polls <- data.frame(poll = 1:ncol(confidence_intervals),
                    t(confidence_intervals), sample_size = Ns)
names(polls) <- c("poll", "estimate", "low", "high", "sample_size")
polls
```

```
##    poll    estimate           low       high sample_size
## 1     1 0.020030817 -0.0354707836 0.07553242        1298
## 2     2 0.009380863 -0.0772449415 0.09600667         533
## 3     3 0.005961252 -0.0486328871 0.06055539        1342
## 4     4 0.030100334 -0.0366474636 0.09684813         897
## 5     5 0.085271318  0.0136446370 0.15689800         774
## 6     6 0.070866142 -0.0543095137 0.19604180         254
## 7     7 0.029556650 -0.0405989265 0.09971223         812
## 8     8 0.086419753 -0.0242756708 0.19711518         324
## 9     9 0.027110767 -0.0285318074 0.08275334        1291
## 10   10 0.022727273 -0.0388025756 0.08425712        1056
## 11   11 0.042357274 -0.0005183189 0.08523287        2172
## 12   12 0.112403101  0.0249159791 0.19989022         516
```

Code: Calculating the spread of combined polls

Note that to compute the exact 95% confidence interval, we would use `qnorm(.975)` instead of `1.96`.

```
d_hat <- polls %>%
    summarize(avg = sum(estimate*sample_size) / sum(sample_size)) %>%
    .$avg

p_hat <- (1+d_hat)/2
moe <- 2*1.96*sqrt(p_hat*(1-p_hat)/sum(polls$sample_size))
round(d_hat*100,1)
```

```
## [1] 3.6
```

```
round(moe*100, 1)
```

```
## [1] 1.8
```

Assessment 4.1: Statistical Models

1. Heights Revisited

We have been using urn models to motivate the use of probability models. However, most data science applications are not related to data obtained from urns. More common are data that come from individuals. Probability plays a role because the data come from a random sample. The random sample is taken from a population and the urn serves as an analogy for the population.

Let's revisit the heights dataset. For now, consider x to be the heights of all males in the data set. Mathematically speaking, x is our population. Using the urn analogy, we have an urn with the values of x in it.

What are the population average and standard deviation of our population?

Instructions - Execute the lines of code that create a vector x that contains heights for all males in the population. - Calculate the average of x. - Calculate the standard deviation of x.

```
# Load the 'dslabs' package and data contained in 'heights'
library(dslabs)
library(dplyr)
data(heights)

# Make a vector of heights from all males in the population
x <- heights %>% filter(sex == "Male") %>%
  .$height

# Calculate the population average. Print this value to the console.
mean(x)
```

```
## [1] 69.31475
```

```
# Calculate the population standard deviation. Print this value to the console.
sd(x)
```

```
## [1] 3.611024
```

2. Sample the population of heights

Call the population average computed above and the standard deviation . Now take a sample of size 50, with replacement, and construct an estimate for and .

Instructions - Use the sample function to sample N values from x. - Calculate the mean of the sampled heights. - Calculate the standard deviation of the sampled heights.

```
# The vector of all male heights in our population `x` has already been loaded for you. You can examine
head(x)
```

```
## [1] 75 70 68 74 61 67
```

```
# Use the `set.seed` function to make sure your answer matches the expected result after random samplin
set.seed(1)
```

```
# Define `N` as the number of people measured
N <- 50

# Define `X` as a random sample from our population `x`
X <- sample(x, N, replace = TRUE)

# Calculate the sample average. Print this value to the console.
mean(X)

## [1] 70.47293

# Calculate the sample standard deviation. Print this value to the console.
sd(X)

## [1] 3.426742
```

3. Sample and Population Averages

What does the central limit theory tell us about the sample average and how it is related to , the population average?

Possible Answers - [ ] A. It is identical to . - [X] B. It is a random variable with expected value   and standard error /√N. - [ ] C. It is a random variable with expected value   and standard error . - [ ] D. It underestimates .

4. Confidence Interval Calculation

We will use X̄ as our estimate of the heights in the population from our sample size N. We know from previous exercises that the standard estimate of our error X̄− is /√N.

Construct a 95% confidence interval for .

Instructions - Use the sd and sqrt functions to define the standard error se - Calculate the 95% confidence intervals using the qnorm function. Save the lower then the upper confidence interval to a variable called ci.

```
# The vector of all male heights in our population `x` has already been loaded for you. You can examine
head(x)

## [1] 75 70 68 74 61 67

# Use the `set.seed` function to make sure your answer matches the expected result after random samplin
set.seed(1)

# Define `N` as the number of people measured
N <- 50

# Define `X` as a random sample from our population `x`
X <- sample(x, N, replace = TRUE)

# Define `se` as the standard error of the estimate. Print this value to the console.
X_hat <- mean(X)
se_hat <- sd(X)
se <- se_hat / sqrt(N)
se
```

```
## [1] 0.4846145
```

```
# Construct a 95% confidence interval for the population average based on our sample. Save the lower an
ci <- c(qnorm(0.025, mean(X), se), qnorm(0.975, mean(X), se))
```

5. Monte Carlo Simulation for Heights

Now run a Monte Carlo simulation in which you compute 10,000 confidence intervals as you have just done. What proportion of these intervals include ?

Instructions - Use the replicate function to replicate the sample code for B <- 10000 simulations. Save the results of the replicated code to a variable called res. The replicated code should complete the following steps: -1. Use the sample function to sample N values from x. Save the sampled heights as a vector called X. -2. Create an object called interval that contains the 95% confidence interval for each of the samples. Use the same formula you used in the previous exercise to calculate this interval. -3. Use the between function to determine if is contained within the confidence interval of that simulation. - Finally, use the mean function to determine the proportion of results in res that contain mu.

```
# Define `mu` as the population average
mu <- mean(x)

# Use the `set.seed` function to make sure your answer matches the expected result after random samplin
set.seed(1)

# Define `N` as the number of people measured
N <- 50

# Define `B` as the number of times to run the model
B <- 10000

# Define an object `res` that contains a logical vector for simulated intervals that contain mu
res <- replicate(B, {
  X <- sample(x, N, replace = TRUE)
  X_hat <- mean(X)
  se_hat <- sd(X)
  se <- se_hat / sqrt(N)
  interval <- c(qnorm(0.025, mean(X), se) , qnorm(0.975, mean(X), se))
  between(mu, interval[1], interval[2])
})

# Calculate the proportion of results in `res` that include mu. Print this value to the console.
mean(res)
```

```
## [1] 0.9479
```

6. Visualizing Polling Bias

In this section, we used visualization to motivate the presence of pollster bias in election polls. Here we will examine that bias more rigorously. Lets consider two pollsters that conducted daily polls and look at national polls for the month before the election.

Is there a poll bias? Make a plot of the spreads for each poll.

Instructions - Use ggplot to plot the spread for each of the two pollsters. - Define the x- and y-axes usingusing aes() within the ggplot function. - Use geom_boxplot to make a boxplot of the data. - Use geom_point to add data points to the plot.