Inference for categorical data

Getting Started

Load packages

In this lab, we will explore and visualize the data using the **tidyverse** suite of packages, and perform statistical inference using **infer**. The data can be found in the companion package for OpenIntro resources, **openintro**.

Let's load the packages.

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

The data

You will be analyzing the same dataset as in the previous lab, where you delved into a sample from the Youth Risk Behavior Surveillance System (YRBSS) survey, which uses data from high schoolers to help discover health patterns. The dataset is called `yrbss`.

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

A total of 4,792 respondents reported not texting while driving at all. Among those surveyed, 925 indicated that they texted while driving for 1 to 2 days. Additionally, 493 respondents admitted to texting while driving between 3 to 5 days, while 311 reported doing so for 6 to 9 days. Furthermore, 373 individuals mentioned texting while driving for a duration of 10 to 19 days, and 298 indicated they did so for 20 to 29 days.

```
# Loading the dplyr package
library(dplyr)

# Counting occurrences of each unique value in the text_while_driving_30d column
count_each <- yrbss %>%
group_by(text_while_driving_30d) %>%
summarise(count = n())

# Displaying the result
print(count_each)
```

```
## # A tibble: 9 x 2
##   text_while_driving_30d count
##   <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
```

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

Approximately 3.40% of individuals reported texting while driving daily over the past 30 days and stated that they never wear helmets.

```
# Calculating the proportion of respondents who texted while driving for 30 days and never wore a helme
proportion <- yrbss %>%
filter(text_while_driving_30d == "30" & helmet_12m == "never") %>%
nrow() / nrow(yrbss)

# Displaying the proportion
print(proportion)
```

```
## [1] 0.03408673
```

Remember that you can use `filter` to limit the dataset to just non-helmet wearers. Here, we will name the dataset `no_helmet`.

```
data('yrbss', package='openintro')
no_helmet <- yrbss %>%
  filter(helmet_12m == "never")
```

Also, it may be easier to calculate the proportion if you create a new variable that specifies whether the individual has texted every day while driving over the past 30 days or not. We will call this variable `text_ind`.

```
no_helmet <- no_helmet %>%
  mutate(text_ind = ifelse(text_while_driving_30d == "30", "yes", "no"))
```

Inference on proportions

When summarizing the YRBSS, the Centers for Disease Control and Prevention seeks insight into the population *parameters*. To do this, you can answer the question, “What proportion of people in your sample reported that they have texted while driving each day for the past 30 days?” with a statistic; while the question “What proportion of people on earth have texted while driving each day for the past 30 days?” is answered with an estimate of the parameter.

The inferential tools for estimating population proportion are analogous to those used for means in the last chapter: the confidence interval and the hypothesis test.

```
# Remove missing values
no_helmet_clean <- no_helmet %>%
  filter(!is.na(text_ind))
```

```
no_helmet_clean %>%
  specify(response = text_ind, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1   0.0649   0.0774
```

Note that since the goal is to construct an interval estimate for a proportion, it's necessary to both include the `success` argument within `specify`, which accounts for the proportion of non-helmet wearers than have consistently texted while driving the past 30 days, in this example, and that `stat` within `calculate` is here "prop", signaling that you are trying to do some sort of inference on a proportion.

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?

Margin Error: 0.004

```
# Calculating the proportion of individuals who texted while driving every day for the last 30 days and
p <- nrow(yrbss[yrbss$text_while_driving_30d == "30" & yrbss$helmet_12m == "never", ]) / nrow(yrbss)

# Calculating the margin of error
margin_of_error <- 1.96 * sqrt(p * (1 - p) / nrow(yrbss))

# Displaying the margin of error
margin_of_error
```

```
## [1] 0.004351235
```

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.

We are 95% confident that the proportion of students who reported sleeping well lies between 76.8% and 78.3%.

```
# Creating a new column indicating whether students slept well
yrbss <- yrbss %>%
  mutate(slept_well = ifelse(school_night_hours_sleep > 5, "yes", "no"))

# Removing rows with NA values in 'slept_well'
yrbss <- yrbss %>%
  filter(!is.na(slept_well))

# Perform bootstrapping to calculate the confidence interval for the proportion of students who slept w
good_sleep_results <- yrbss %>%
  specify(response = slept_well, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)

# Displaying the confidence interval results
print(good_sleep_results)
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1    0.769    0.784
```

We are 95% confident that the proportion of students who either watch TV or do not watch TV falls between 13.1% and 14.3%.

```
# Creating a new column indicating whether students do not watch TV
yrbss <- yrbss %>%
  mutate(did_not_watch_tv = ifelse(hours_tv_per_school_day == "do not watch", "yes", "no"))

# Filtering out any rows with missing values in the new column
yrbss <- yrbss %>%
  filter(!is.na(did_not_watch_tv))

# Conducting bootstrapping to compute the confidence interval for the proportion of students who do not
tv_results <- yrbss %>%
  specify(response = did_not_watch_tv, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)

# Displaying the confidence interval results
print(tv_results)
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1    0.131    0.143
```

How does the proportion affect the margin of error?

Imagine you've set out to survey 1000 people on two questions: are you at least 6-feet tall? and are you left-handed? Since both of these sample proportions were calculated from the same sample size, they should have the same margin of error, right? Wrong! While the margin of error does change with sample size, it is also affected by the proportion.

Think back to the formula for the standard error: $SE = \sqrt{p(1-p)/n}$. This is then used in the formula for the margin of error for a 95% confidence interval:

$$ME = 1.96 \times SE = 1.96 \times \sqrt{p(1-p)/n}\,.$$

Since the population proportion p is in this ME formula, it should make sense that the margin of error is in some way dependent on the population proportion. We can visualize this relationship by creating a plot of ME vs. p.

Since sample size is irrelevant to this discussion, let's just set it to some value ($n = 1000$) and use this value in the following calculations:

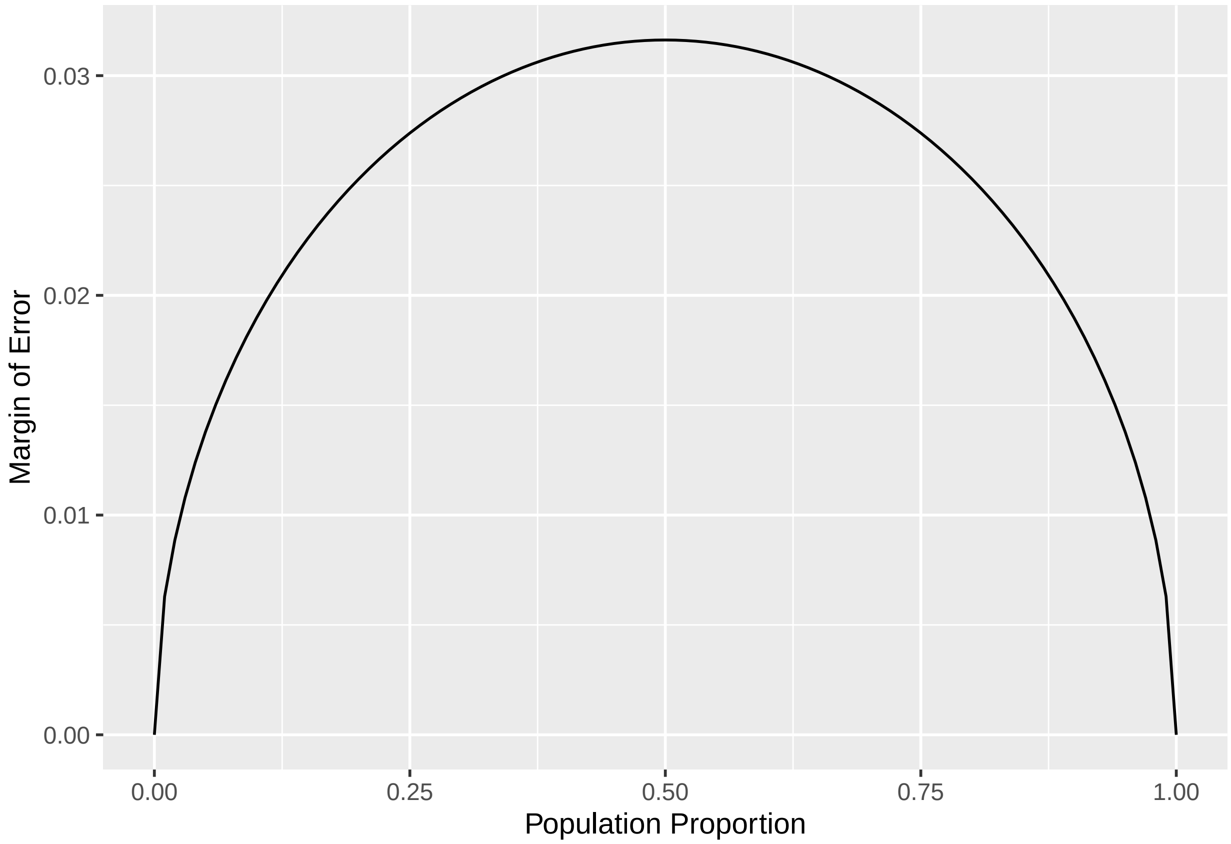
```
n <- 1000
```

The first step is to make a variable `p` that is a sequence from 0 to 1 with each number incremented by 0.01. You can then create a variable of the margin of error (`me`) associated with each of these values of `p` using the familiar approximate formula ($ME = 2 \times SE$).

```
p <- seq(from = 0, to = 1, by = 0.01)
me <- 2 * sqrt(p * (1 - p)/n)
```

Lastly, you can plot the two variables against each other to reveal their relationship. To do so, we need to first put these variables in a data frame that you can call in the `ggplot` function.

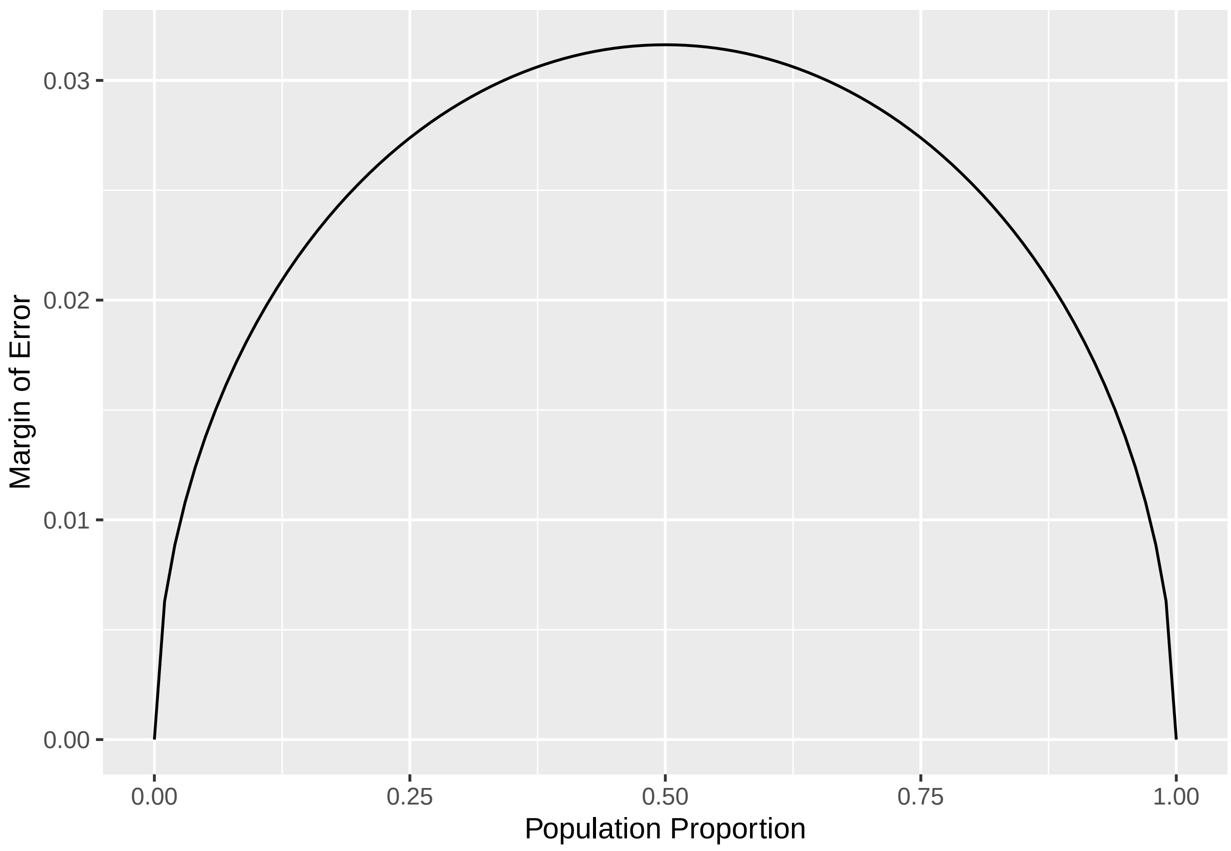
```
dd <- data.frame(p = p, me = me)
ggplot(data = dd, aes(x = p, y = me)) +
  geom_line() +
  labs(x = "Population Proportion", y = "Margin of Error")
```



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?

The distribution is both symmetric and unimodal, showing that the largest margin of error happens when the population proportion is 0.5. As the population proportion moves closer to 0 or 1, the margin of error becomes minimal. Additionally, the margin of error shrinks as the sample size grows, due to its inverse relationship with the square root of the sample size.

```r
# Setting the sample size
sample_size <- 1000

# Creating a sequence of population proportions from 0 to 1 in increments of 0.01
population_proportion <- seq(0, 1, by = 0.01)

# Calculating the margin of error for each proportion
margin_of_error <- 2 * sqrt(population_proportion * (1 - population_proportion) / sample_size)

# Creating a data frame with population proportions and corresponding margins of error
error_data <- data.frame(population_proportion = population_proportion, margin_of_error = margin_of_erro

# Generating the plot
ggplot(data = error_data, aes(x = population_proportion, y = margin_of_error)) +
  geom_line() + labs(x = "Population Proportion", y = "Margin of Error")
```



Success-failure condition

We have emphasized that you must always check conditions before making inference. For inference on proportions, the sample proportion can be assumed to be nearly normal if it is based upon a random sample of independent observations and if both $np \geq 10$ and $n(1-p) \geq 10$. This rule of thumb is easy enough to follow, but it makes you wonder: what's so special about the number 10?

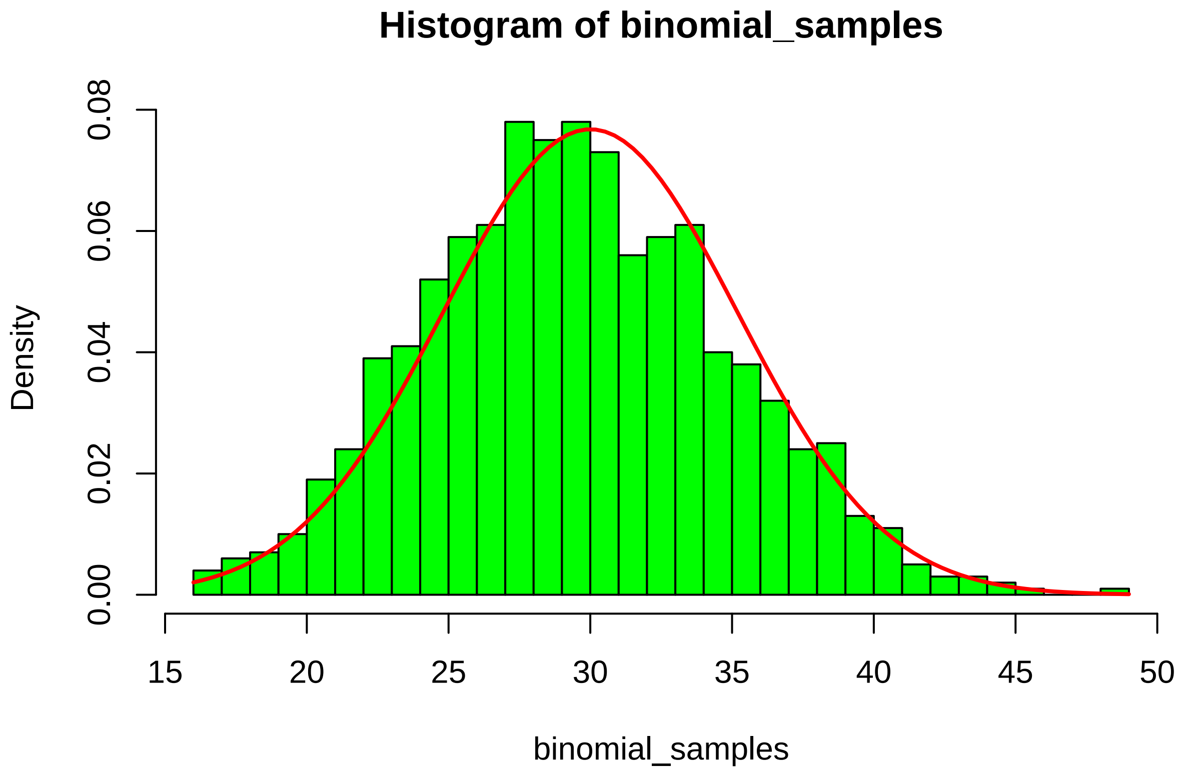
The short answer is: nothing. You could argue that you would be fine with 9 or that you really should be using 11. What is the "best" value for such a rule of thumb is, at least to some degree, arbitrary. However, when np and $n(1-p)$ reaches 10 the sampling distribution is sufficiently normal to use confidence intervals and hypothesis tests that are based on that approximation.

You can investigate the interplay between n and p and the shape of the sampling distribution by using simulations. Play around with the following app to investigate how the shape, center, and spread of the distribution of $\hat{p}$ changes as n and p changes.

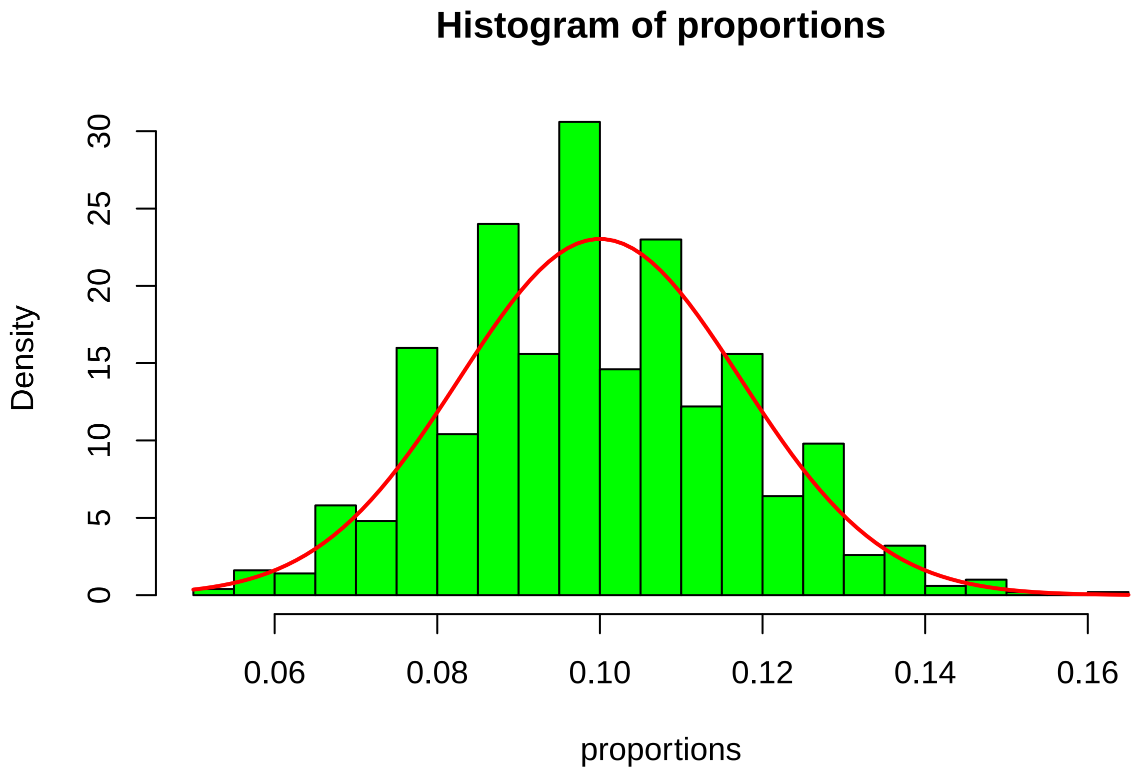
6. Describe the sampling distribution of sample proportions at $n = 300$ and $p = 0.1$. Be sure to note the center, spread, and shape.

The distribution appears to follow a normal pattern, with its center located at 0.1 and a spread ranging from 0.04 to 0.16.

```
# Setting the seed for random number generation to ensure consistent results
set.seed(100)

# Specifying the number of trials and the probability of success
sample_size <- 300
success_prob <- 0.1
num_samples <- 1000

# Generating a set of binomial random samples
binomial_samples <- replicate(num_samples, rbinom(1, sample_size, success_prob))

# Plotting a histogram of the binomial samples
hist(binomial_samples,
     #ylim = c(0, 1.4),
     col = "green",
     probability = TRUE,  # Use probability for y-axis
     breaks = 25)

# Adding a normal distribution curve based on the samples
curve(dnorm(x, mean = sample_size * success_prob, sd = sqrt(success_prob * (1 - success_prob) * sample_
      col = "red",
      lwd = 2,
      add = TRUE)
```



```r
# Calculate the proportions from the binomial samples
proportions <- binomial_samples / sample_size

# Plot a histogram of the sample proportions
hist(proportions,
     #ylim = c(0, 1.4),
     col = "green",
     probability = TRUE,  # Use probability for y-axis
     breaks = 25)

# Overlay the normal distribution curve for the sample proportions
curve(dnorm(x, mean = success_prob, sd = sqrt(success_prob * (1 - success_prob) / sample_size)),
      col = "red",
      lwd = 2,
      add = TRUE)
```



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.

The distribution appears to follow a normal pattern, with a mean located at 0.1 and a range extending from 0.05 to 0.17.

8. Now also change n. How does n appear to affect the distribution of $\hat{p}$?

When n decreases, the spread increases, while increasing n causes the spread to decrease. With {reps} = 1000, a larger n results in a more bell-shaped histogram. The center remains at 0.1, which is the p-value. A larger sample size smooths out the distribution of p hat's.

More Practice

For some of the exercises below, you will conduct inference comparing two proportions. In such cases, you have a response variable that is categorical, and an explanatory variable that is also categorical, and you are comparing the proportions of success of the response variable across the levels of the explanatory variable. This means that when using `infer`, you need to include both variables within `specify`.

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 (H0): Individuals who sleep 10 or more hours each day are more likely to engage in strength training every day of the week.

Alternative Hypothesis (HA): Individuals who sleep 10 or more hours each day are not more likely to engage in strength training every day of the week.

Given a p-value of 0.1531929, we would not reject the null hypothesis since it exceeds the significance level of 0.05. Thus, we conclude that those who sleep 10 or more hours per day are not significantly more likely to participate in daily strength training.

```
# Calculating the proportion of students who sleep 10 or more hours on school nights
# and ones who are engaged in strength training more than 6 days in the past week
total_students <- nrow(yrbss)
selected_students <- yrbss %>%
  filter(school_night_hours_sleep >= 10, strength_training_7d > 6)

proportion <- nrow(selected_students) / total_students
proportion
```

```
## [1] 0.1531929
```

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.

There is a 5% probability of observing a change when there is none. A Type I error refers to a false positive, which occurs when researchers mistakenly reject a true null hypothesis. In such cases, they may present their results as significant, despite the fact that the findings are actually not significant.

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.

Sample size of at least: 9604.

```
# Defining parameters for the calculation
probability <- 0.5
z_value <- 1.96
margin_of_error <- 0.01

# Calculating the required sample size
sample_size <- (z_value^2 * probability * (1 - probability)) / (margin_of_error^2)
sample_size
```

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