Inference for numerical data

Josh Iden

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

Every two years, the Centers for Disease Control and Prevention conduct the Youth Risk Behavior Surveillance System (YRBSS) survey, where it takes data from high schoolers (9th through 12th grade), to analyze health patterns. You will work with a selected group of variables from a random sample of observations during one of the years the YRBSS was conducted.

Load the `yrbss` data set into your workspace.

```
data('yrbss', package='openintro')
```

There are observations on 13 different variables, some categorical and some numerical. The meaning of each variable can be found by bringing up the help file:

```
?yrbss
```

1. What are the cases in this data set? How many cases are there in our sample?

Remember that you can answer this question by viewing the data in the data viewer or by using the following command:

```
glimpse(yrbss)
```

```
## Rows: 13,583
## Columns: 13
## $ age                       <int> 14, 14, 15, 15, 15, 15, 15, 14, 15, 15, 15, 1~
## $ gender                    <chr> "female", "female", "female", "female", "fema~
## $ grade                     <chr> "9", "9", "9", "9", "9", "9", "9", "9", "9", ~
```

```
## $ hispanic                 <chr> "not", "not", "hispanic", "not", "not", "not"~
## $ race                     <chr> "Black or African American", "Black or Africa~
## $ height                   <dbl> NA, NA, 1.73, 1.60, 1.50, 1.57, 1.65, 1.88, 1~
## $ weight                   <dbl> NA, NA, 84.37, 55.79, 46.72, 67.13, 131.54, 7~
## $ helmet_12m               <chr> "never", "never", "never", "never", "did not ~
## $ text_while_driving_30d   <chr> "0", NA, "30", "0", "did not drive", "did not~
## $ physically_active_7d     <int> 4, 2, 7, 0, 2, 1, 4, 4, 5, 0, 0, 0, 4, 7, 7, ~
## $ hours_tv_per_school_day  <chr> "5+", "5+", "5+", "2", "3", "5+", "5+", "5+",~
## $ strength_training_7d     <int> 0, 0, 0, 0, 1, 0, 2, 0, 3, 0, 3, 0, 0, 7, 7, ~
## $ school_night_hours_sleep <chr> "8", "6", "<5", "6", "9", "8", "9", "6", "<5"~
```

The cases are the observations (rows) in the dataframe. There are 13,583 cases ## Exploratory data analysis

You will first start with analyzing the weight of the participants in kilograms: `weight`.

Using visualization and summary statistics, describe the distribution of weights. The `summary` function can be useful.

```
summary(yrbss$weight)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   29.94   56.25   64.41   67.91   76.20  180.99    1004
```

2. How many observations are we missing weights from?

This information is visible by the number of NAs = 1004

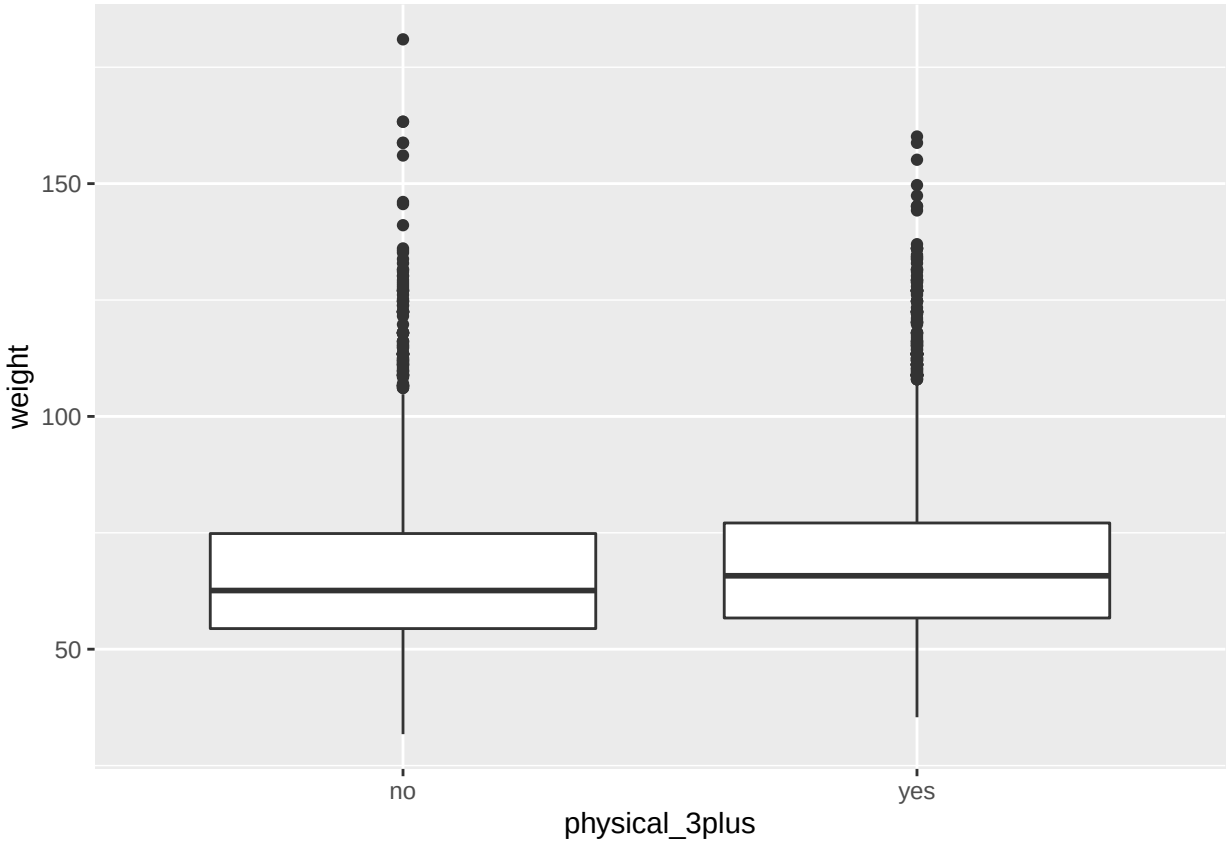
Next, consider the possible relationship between a high schooler's weight and their physical activity. Plotting the data is a useful first step because it helps us quickly visualize trends, identify strong associations, and develop research questions.

First, let's create a new variable `physical_3plus`, which will be coded as either "yes" if they are physically active for at least 3 days a week, and "no" if not.

```
yrbss <- yrbss %>%
  mutate(physical_3plus = ifelse(yrbss$physically_active_7d > 2, "yes", "no"))
```

3. Make a side-by-side boxplot of `physical_3plus` and `weight`. Is there a relationship between these two variables? What did you expect and why?

```
ggplot(na.omit(yrbss), aes(x=physical_3plus, y=weight)) +
  geom_boxplot()
```



I expect people who are more physically active to be in better fitness and therefore in better shape – however, I did not expect this relationship to be particularly noticeable given the basic criteria provided which doesn't account for other factors such as age, height, or gender, or number of observations between groups.

The box plots show how the medians of the two distributions compare, but we can also compare the means of the distributions using the following to first group the data by the `physical_3plus` variable, and then calculate the mean `weight` in these groups using the `mean` function while ignoring missing values by setting the `na.rm` argument to `TRUE`.

```
yrbss %>%
  group_by(physical_3plus) %>%
  summarise(mean_weight = mean(weight, na.rm = TRUE))
```

```
## # A tibble: 3 x 2
##   physical_3plus mean_weight
##   <chr>                <dbl>
## 1 no                    66.7
## 2 yes                   68.4
## 3 <NA>                  69.9
```

There is an observed difference, but is this difference statistically significant? In order to answer this question we will conduct a hypothesis test.

Inference

4. Are all conditions necessary for inference satisfied? Comment on each. You can compute the group sizes with the `summarize` command above by defining a new variable with the definition `n()`.

```
yrbss |> na.omit(yrbss) |>
  group_by(physical_3plus) |>
  summarize(count = n()) |>
  mutate(pct = count/sum(count))
```

```
## # A tibble: 2 x 3
##   physical_3plus count   pct
##   <chr>          <int> <dbl>
## 1 no              2656 0.318
## 2 yes             5695 0.682
```

The data is independent since it is a random sample, but it is not satisfy normality as the distribution of physical activity is skewed

5. Write the hypotheses for testing if the average weights are different for those who exercise at least times a week and those who don't.

H_0 (Null Hypothesis): Exercise has no effect on average weight when frequency of physical activity is normally distributed H_A (Alternative Hypothesis): Exercise has an effect on average weight when frequency of physical activity is normally distributed

Next, we will introduce a new function, `hypothesize`, that falls into the `infer` workflow. You will use this method for conducting hypothesis tests.

But first, we need to initialize the test, which we will save as `obs_diff`.

```
obs_diff <- na.omit(yrbss) %>%
  specify(weight ~ physical_3plus) %>%
  calculate(stat = "diff in means", order = c("yes", "no"))
```

Notice how you can use the functions `specify` and `calculate` again like you did for calculating confidence intervals. Here, though, the statistic you are searching for is the difference in means, with the order being `yes - no != 0`.

After you have initialized the test, you need to simulate the test on the null distribution, which we will save as `null`.

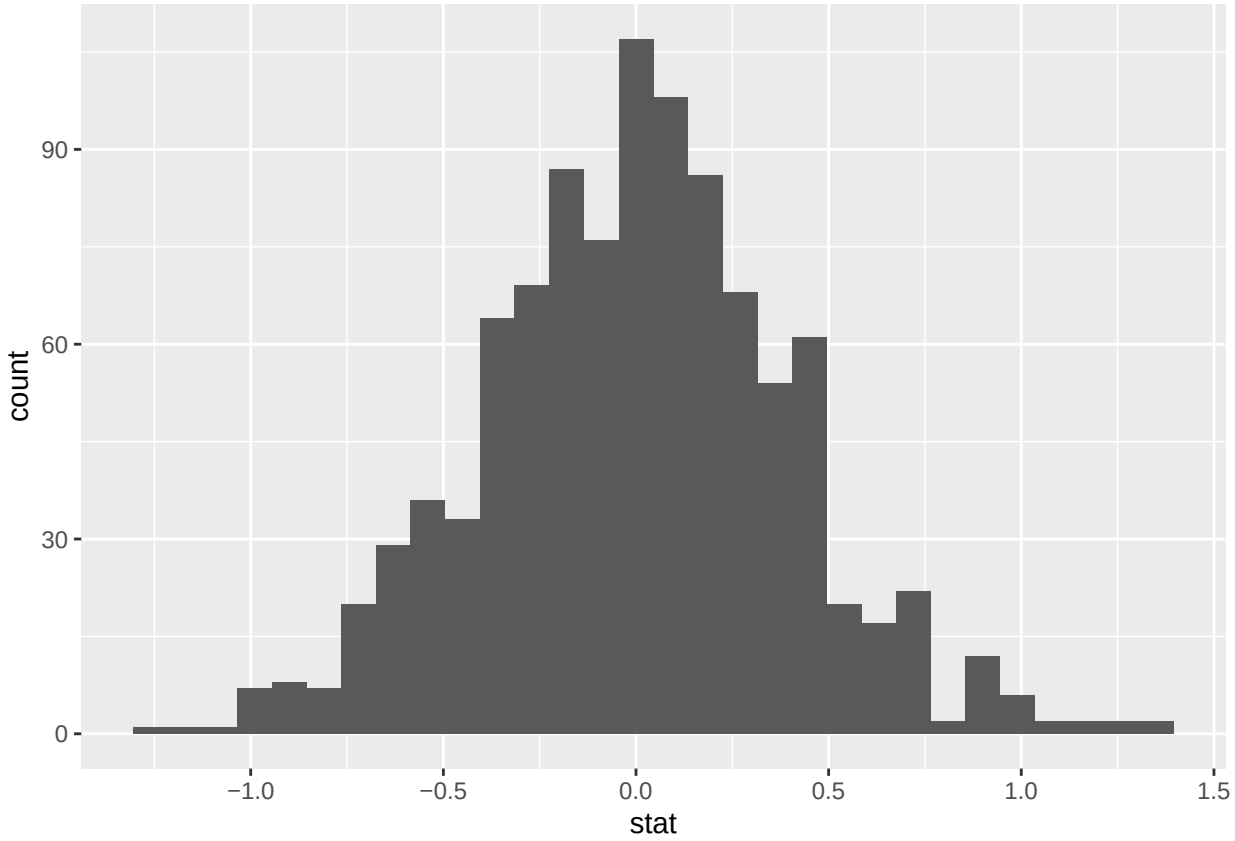
```
null_dist <- na.omit(yrbss) %>%
  specify(weight ~ physical_3plus) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "diff in means", order = c("yes", "no"))
```

Here, `hypothesize` is used to set the null hypothesis as a test for independence. In one sample cases, the `null` argument can be set to "point" to test a hypothesis relative to a point estimate.

Also, note that the `type` argument within `generate` is set to `permute`, whichis the argument when generating a null distribution for a hypothesis test.

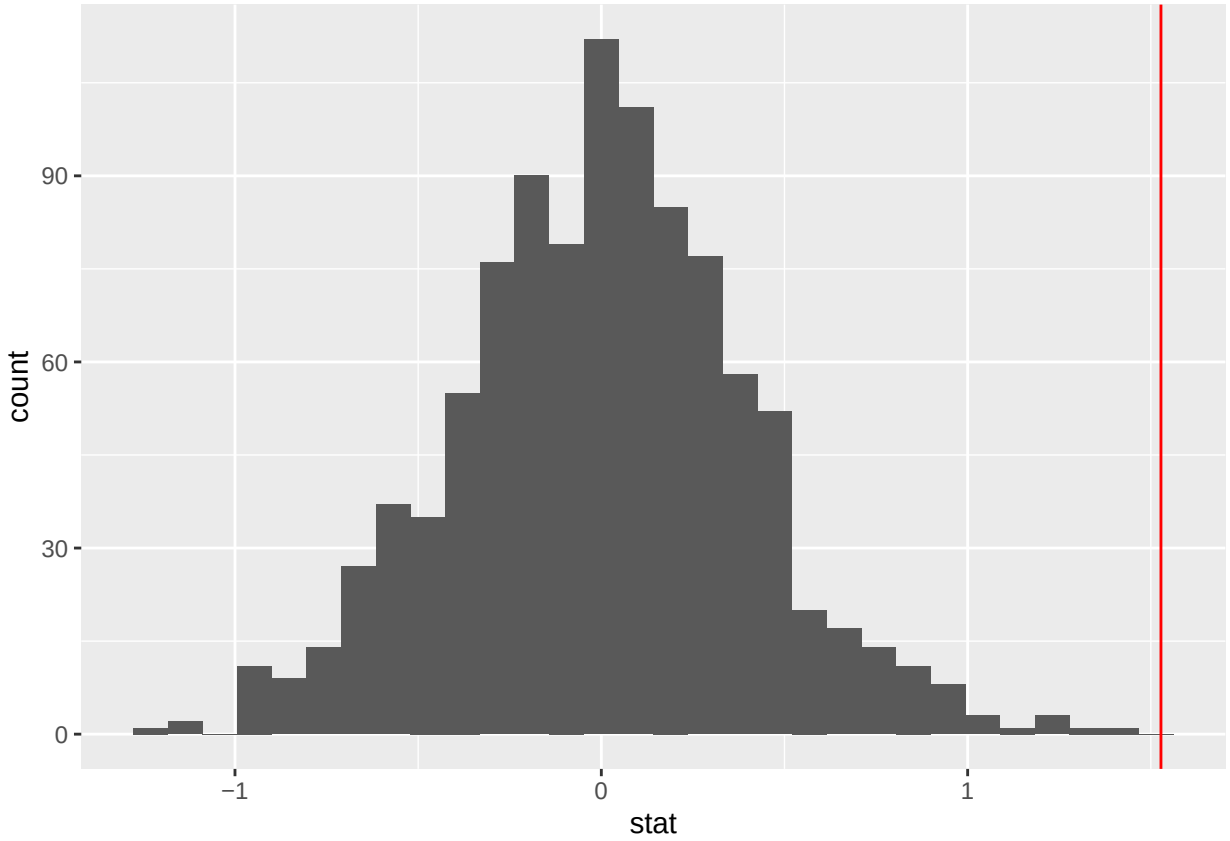
We can visualize this null distribution with the following code:

```
ggplot(data = null_dist, aes(x = stat)) +
  geom_histogram()
```



6. How many of these `null` permutations have a difference of at least `obs_stat`?

```
ggplot(data = null_dist, aes(x = stat)) +
  geom_histogram() +
  geom_vline(xintercept = obs_diff$stat, color="red")
```




```
null_dist |>
  count(null_dist$stat > obs_diff$stat)
```

```
## Response: weight (numeric)
## Explanatory: physical_3plus (factor)
## Null Hypothesis: independence
## # A tibble: 1 x 2
##   `null_dist$stat > obs_diff$stat`     n
##   <lgl>                            <int>
## 1 FALSE                             1000
```

No null permutations have a difference of at least obs_stat

Now that the test is initialized and the null distribution formed, you can calculate the p-value for your hypothesis test using the function `get_p_value`.

```
null_dist %>%
  get_p_value(obs_stat = obs_diff, direction = "two_sided")
```

```
## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1       0
```

This the standard workflow for performing hypothesis tests.

7. Construct and record a confidence interval for the difference between the weights of those who exercise at least three times a week and those who don't, and interpret this interval in context of the data.

```
yrbss |> na.omit(yrbss) |>
  specify(weight ~ physical_3plus) |>
  hypothesize(null = "independence") |>
  generate(reps = 1000, type = "permute") |>
  calculate(stat = "diff in means", order = c("yes", "no")) |>
  get_ci(level = 0.95)
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1   -0.763    0.719
```

Since the observed mean of 1.53 is above the confidence interval, we reject the null hypothesis

---

More Practice

8. Calculate a 95% confidence interval for the average height in meters (`height`) and interpret it in context.

```
z <- 1.96 #z-score for 95% CI
yrbss |> na.omit(yrbss$height) |>
  summarise(mean = mean(height), sd = sd(height),
            n = n()) |>
  mutate(cu_lower = mean-(z*(sd/sqrt(n))),
         cu_upper = mean+(z*(sd/sqrt(n))))
```

```
## # A tibble: 1 x 5
##    mean    sd     n cu_lower cu_upper
##   <dbl> <dbl> <int>    <dbl>    <dbl>
## 1  1.70 0.105  8351     1.69     1.70
```

The confidence interval makes sense, 95% of the means will be between 1.69 and 1.70

9. Calculate a new confidence interval for the same parameter at the 90% confidence level. Comment on the width of this interval versus the one obtained in the previous exercise.

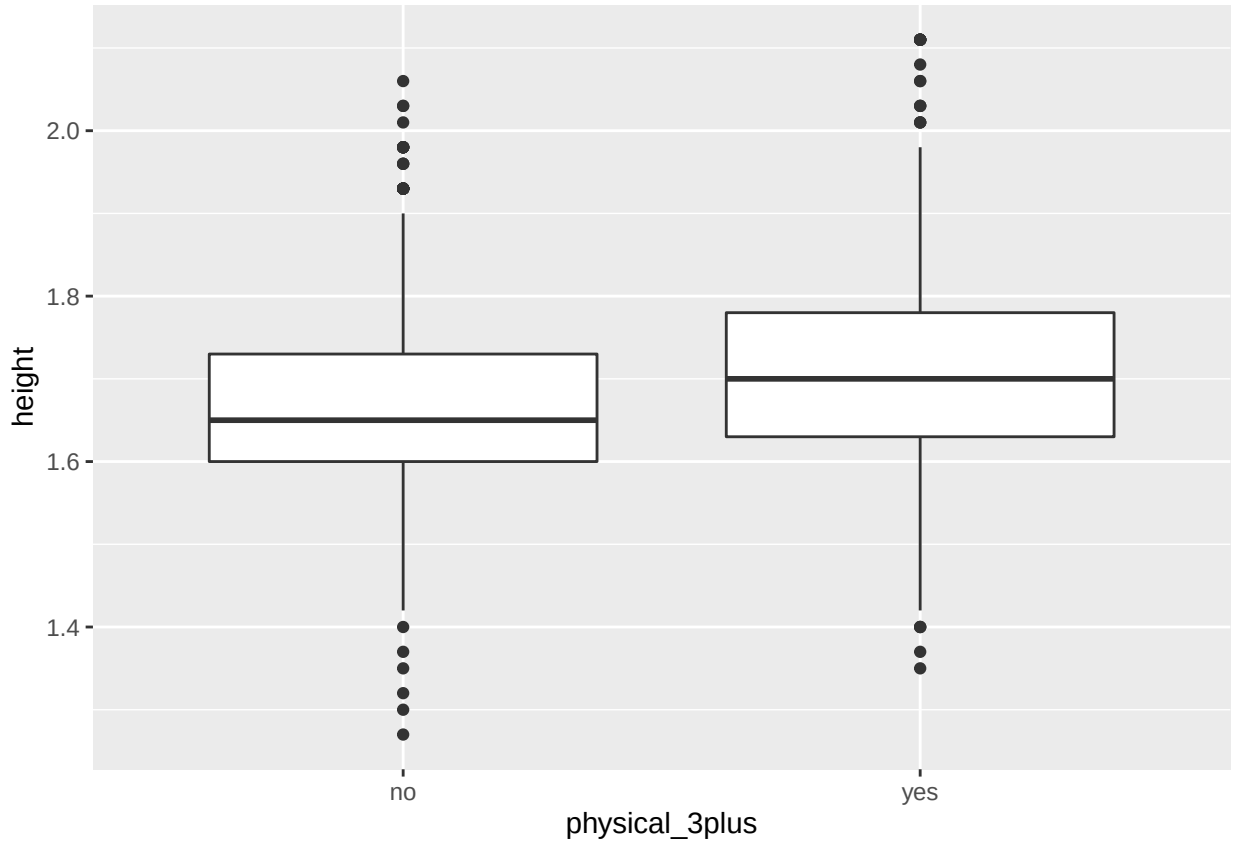
```
z <- 1.645 #z-score for 90% CI
yrbss |> na.omit(yrbss$height) |>
  summarise(mean = mean(height), sd = sd(height),
            n = n()) |>
  mutate(cu_lower = mean-(z*(sd/sqrt(n))),
         cu_upper = mean+(z*(sd/sqrt(n))))
```

```
## # A tibble: 1 x 5
##    mean    sd     n cu_lower cu_upper
##   <dbl> <dbl> <int>    <dbl>    <dbl>
## 1  1.70 0.105  8351     1.70     1.70
```

The confidence interval is narrower, almost indistinguishable. A lower degree of confidence is less specific

10. Conduct a hypothesis test evaluating whether the average height is different for those who exercise at least three times a week and those who don't.

```
library(ggformula)
gf_boxplot(gformula = height ~ physical_3plus, na.omit(yrbss))
```




```
yrbss |> na.omit(yrbss) |>
  specify(height ~ physical_3plus) |>
  hypothesize(null = "independence") |>
  generate(reps = 1000, type = "permute") |>
  calculate(stat = "diff in means", order = c("yes", "no")) |>
  get_ci(level = 0.95)
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1 -0.00495  0.00489
```

11. Now, a non-inference task: Determine the number of different options there are in the dataset for the `hours_tv_per_school_day` there are.

```
yrbss |>
    group_by(hours_tv_per_school_day) |>
    summarise(count = n()) |>
    na.omit()
```

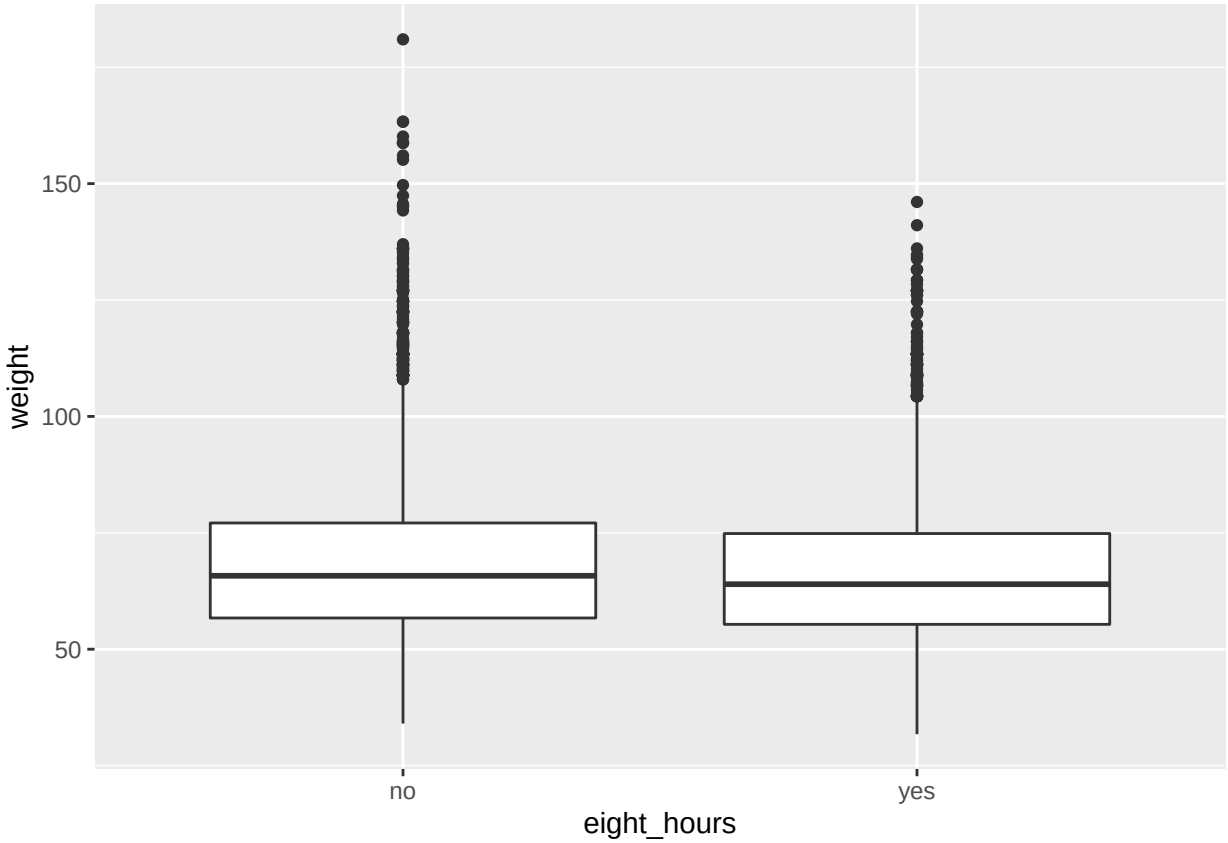
```
## # A tibble: 7 x 2
##   hours_tv_per_school_day count
##   <chr>                   <int>
## 1 <1                       2168
## 2 1                        1750
## 3 2                        2705
## 4 3                        2139
## 5 4                        1048
## 6 5+                       1595
## 7 do not watch             1840
```

12. Come up with a research question evaluating the relationship between height or weight and sleep. Formulate the question in a way that it can be answered using a hypothesis test and/or a confidence interval. Report the statistical results, and also provide an explanation in plain language. Be sure to check all assumptions, state your α level, and conclude in context.

H_O (Null hypothesis): There is no relationship between weight and sleeping 8 hours or more H_A (Alternative hypothesis): There is a relationship between weight and sleeping 8 hours or more

```
hour_vals <- c("8","9","10+")
sleep <- yrbss |>
  mutate(eight_hours = ifelse(school_night_hours_sleep %in% hour_vals, "yes", "no")) |>
  na.omit(select(c("weight","eight_hours")))
```

```
gf_boxplot(gformula = weight ~ eight_hours, sleep)
```




```
sleep |>
  group_by(eight_hours) |>
  summarize(avg_weight = mean(weight))
```

```
## # A tibble: 2 x 2
##   eight_hours avg_weight
##   <chr>            <dbl>
## 1 no                68.6
## 2 yes               67.2
```

Calculate observed difference: obs_diff

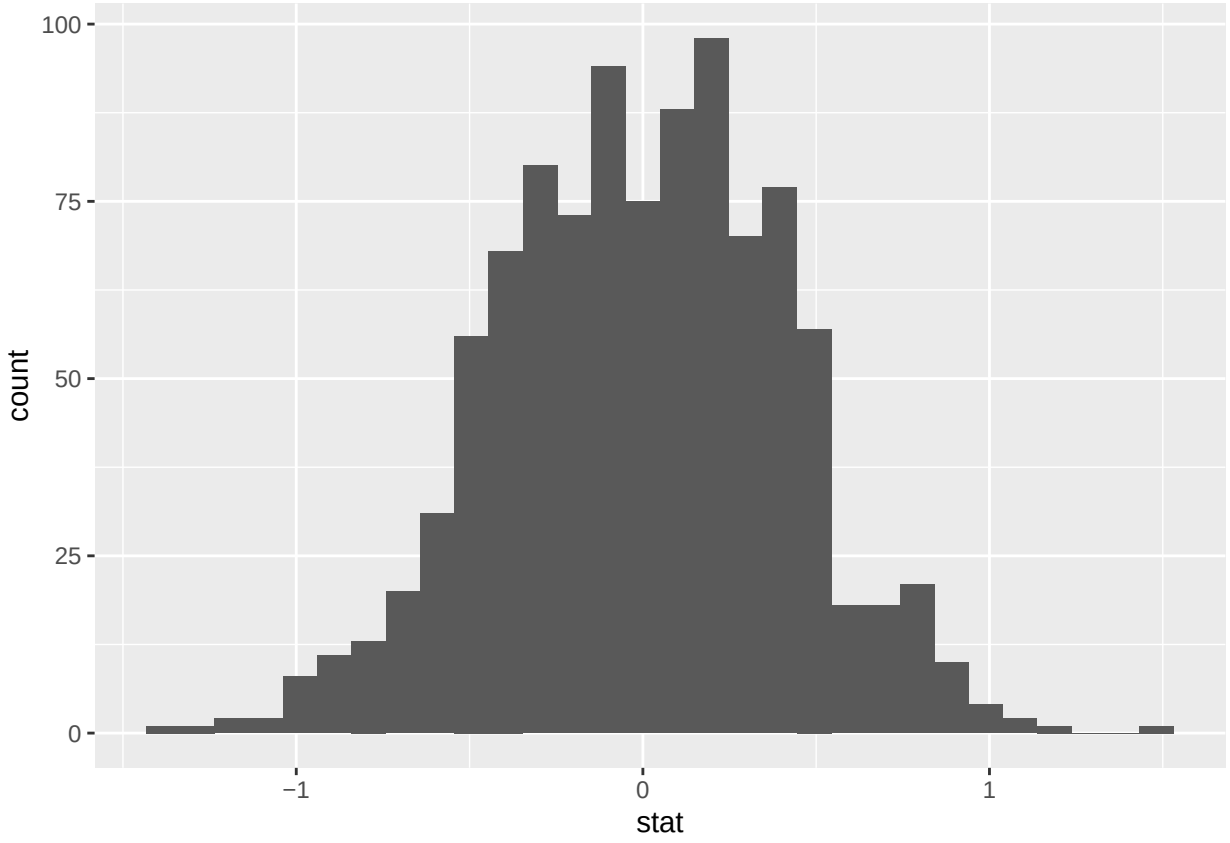
```
obs_diff <- sleep |>
  specify(weight ~ eight_hours) |>
  calculate(stat = "diff in means", order=c("yes","no"))
obs_diff
```

```
## Response: weight (numeric)
## Explanatory: eight_hours (factor)
## # A tibble: 1 x 1
##    stat
##   <dbl>
## 1 -1.41
```

Calculate null distribution: null_dist

```
null_dist <- sleep |>
  specify(weight ~ eight_hours) |>
  hypothesize(null = "independence") |>
  generate(reps = 1000, type = "permute") |>
  calculate(stat = "diff in means", order=c("yes","no"))

ggplot(null_dist, aes(x = stat)) +
  geom_histogram()
```



Get p-value

```
p_val <- null_dist |>
  get_p_value(obs_stat = obs_diff, direction = "two_sided")

p_val
```

```
## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1       0
```

```
mean(sleep$weight)
```

```
## [1] 68.19158
```

```r
sleep |>
  specify(weight ~ eight_hours) |>
  hypothesize(null = "independence") |>
  generate(reps = 1000, type = "permute") |>
  calculate(stat = "diff in means", order=c("yes","no")) |>
  get_ci(level = 0.95)
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1   -0.738    0.785
```
