Inference for numerical data

Victor H Torres

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?

```
ls(yrbss)
```

```
##  [1] "age"                    "gender"
##  [3] "grade"                  "height"
##  [5] "helmet_12m"             "hispanic"
##  [7] "hours_tv_per_school_day" "physically_active_7d"
##  [9] "race"                   "school_night_hours_sleep"
## [11] "strength_training_7d"   "text_while_driving_30d"
## [13] "weight"
```

```r
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"~
```

There are 13,583 cases in our sample with 13 variables, to do that, I used the ls function which returns a vector of character strings containing all the variables and functions that are defined in the current working directory

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

```r
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"~
```

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.

```r
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?

```
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
```

There are 1004(NA's) missing observations in the weight column

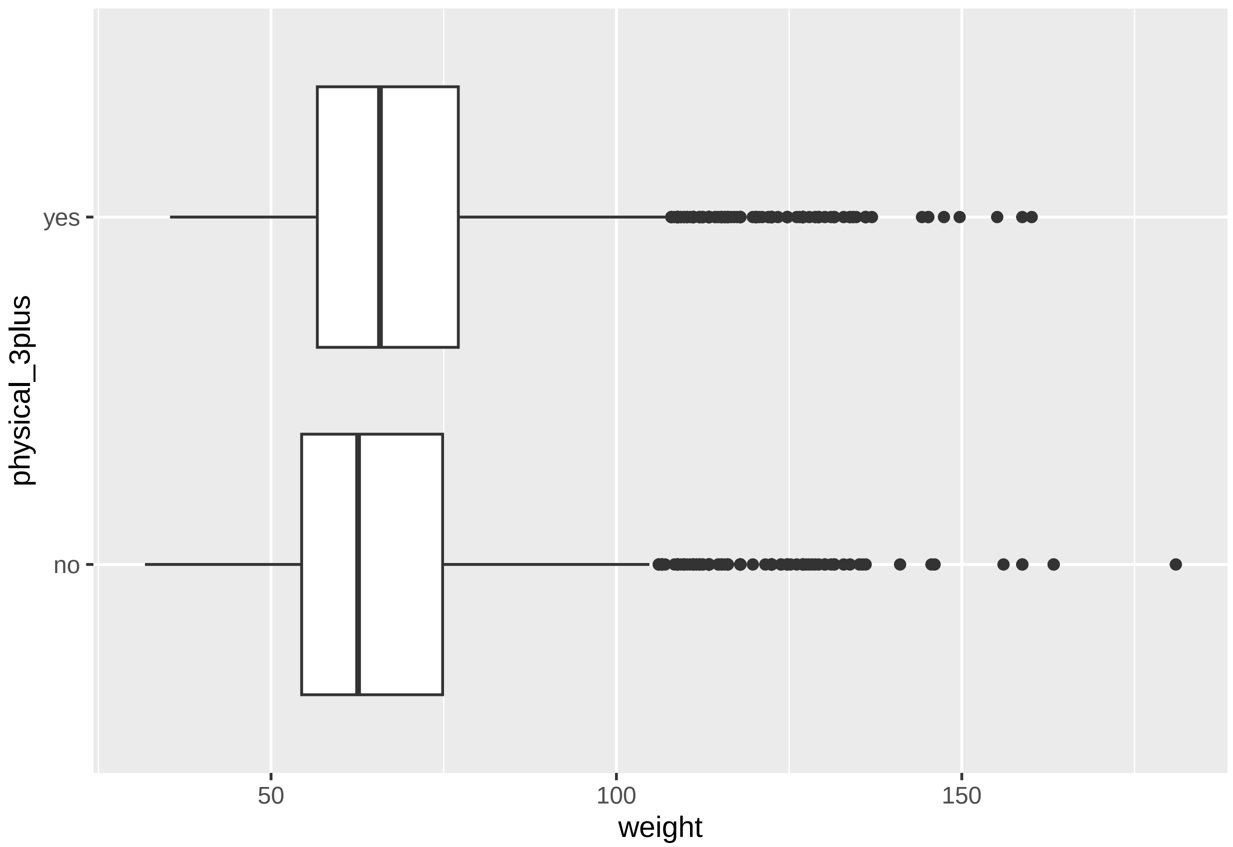
Next, consider the possible relationship between a high schooler's weight and their physical activity. Plotting the data is a useful first step because it helps us quickly visualize trends, identify strong associations, and develop research questions.

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?

```
yrbss_boxplot <- yrbss %>%
  mutate(physical_3plus = ifelse(yrbss$physically_active_7d > 2, "yes", "no")) %>%
  na.exclude()
ggplot(yrbss_boxplot, aes(x=weight, y=physical_3plus)) + geom_boxplot() + theme_gray()
```



There is a slightly difference between the two variables(no:66.7,yes:68.4), to determine the actual number I can calculate the mean in "weight" and ignoring the missing values(NAs)

```
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
```

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%>%
  group_by(physical_3plus) %>%
  summarise(n=n()) %>%
  mutate(freq=n/sum(n), np=n*freq)
```

```
## # A tibble: 3 x 4
##   physical_3plus     n   freq      np
##   <chr>          <int>  <dbl>   <dbl>
## 1 no              4404 0.324  1428.
## 2 yes             8906 0.656  5839.
## 3 <NA>             273 0.0201    5.49
```

All the conditions are necessary for inference, as we can see and the table above, N, and NP are large enough to meet the conditions

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.

HO: Students who are physically active 3+ days per week have the same average weight as those who are not physically active 3+ days per week.

HA: Students who are physically active 3+ days per week have a different average weight than those who don't exercise 3+ days per week. 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 <- yrbss %>%
  drop_na(physical_3plus) %>%
  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`.

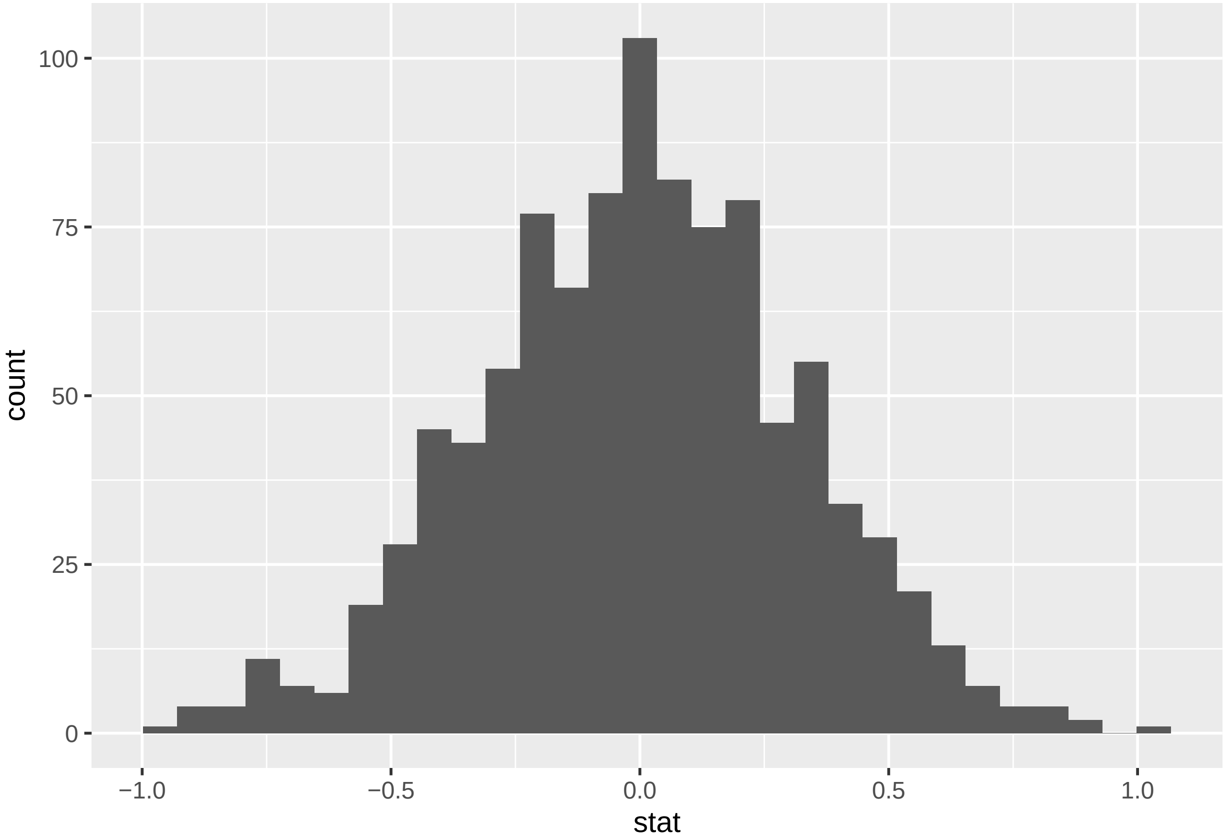
```
null_dist <- yrbss %>%
  drop_na(physical_3plus) %>%
  specify(weight ~ physical_3plus) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "diff in means", order = c("yes", "no"))
```

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.

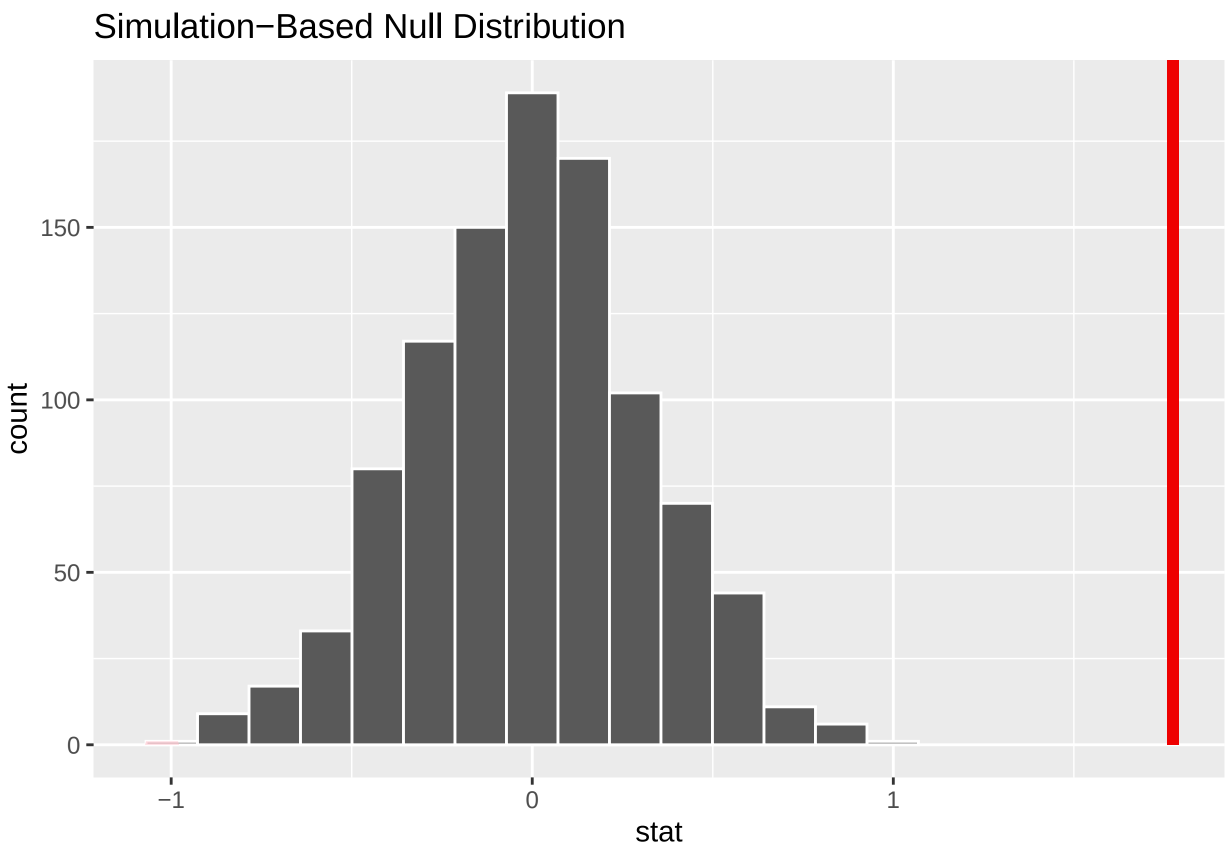
We can visualize this null distribution with the following code:

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



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

```
visualize(null_dist) +
  shade_p_value(obs_stat = obs_diff, direction = "two_sided")
```



The red line in the plot indicates the obs_stat value meaning that the permutations have a difference by a little over the 1

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 %>%
  group_by(physical_3plus) %>%
  summarise(sd_weight = sd(weight, na.rm = TRUE))
```

```
## # A tibble: 3 x 2
##   physical_3plus sd_weight
##   <chr>              <dbl>
## 1 no                  17.6
## 2 yes                 16.5
## 3 <NA>                17.6
```

```
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
```

```
yrbss %>%
  group_by(physical_3plus) %>%
  summarise(freq = table(weight)) %>%
  summarise(n = sum(freq))
```

```
## # A tibble: 3 x 2
##   physical_3plus     n
##   <chr>          <int>
## 1 no              4022
## 2 yes             8342
## 3 <NA>             215
```

```
# not Active
not_active_mean <- 66.7
not_active_sd <- 17.6
not_active_n <- 4022

# active
active_mean <- 68.4
active_sd <- 16.5
active_n <- 8342

z <- 1.96
upper_not_active <- not_active_mean + z * (not_active_sd / sqrt(not_active_n))
upper_not_active
```

```
## [1] 67.24394
```

```
lower_not_active <- not_active_mean - z * (not_active_sd / sqrt(not_active_n))
lower_not_active
```

```
## [1] 66.15606
```

```
upper_active <- active_mean + z * (active_sd / sqrt(active_n))
upper_active
```

```
## [1] 68.75408
```

```
lower_active <- active_mean - z * (active_sd / sqrt(active_n))
lower_active
```

```
## [1] 68.04592
```

Data analysis show that the CI interval is between -0.6 and 0.6, same values for the true mean with a 95% of confidence(-0.6 and 0.6) * * *

More Practice

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

```
height_table <- as.data.frame(table(yrbss$height))
height_freq <- sum(height_table$Freq)

# mean, standard deviation and sample size
height_mean <- mean(yrbss$height, na.rm = TRUE)
height_mean
```

```
## [1] 1.691241
```

```
height_sd <- sd(yrbss$height, na.rm = TRUE)
height_sd
```

```
## [1] 0.1046973
```

```
height_n <- yrbss %>%
  summarise(freq = table(height)) %>%
  summarise(n = sum(freq, na.rm = TRUE))
height_n
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1 12579
```

```
z_height <- 1.96

# confidence interval for height
upper_height<- height_mean + z_height * (height_sd / sqrt(height_n))
upper_height
```

```
##          n
## 1 1.693071
```

```
lower_height <- height_mean - z_height * (height_sd / sqrt(height_n))
lower_height
```

```
##           n
## 1 1.689411
```

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_height <- 1.65
upper_height_90<- height_mean + z_height * (height_sd / sqrt(height_n))
upper_height_90
```

```
##           n
## 1 1.692781
```

```
lower_height_90 <- height_mean - z_height * (height_sd / sqrt(height_n))
lower_height_90
```

```
##           n
## 1 1.689701
```

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.

The average heights of students who are not physically active 3+ days per week is between 1.66m and 1.67m,with a 95% confidence interval. While for those who are physically active is between 1.701m and 1.705m. The P-value is below 0.5 so we should reject the null hypothesis.

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(n())
```

```
## # A tibble: 8 x 2
##   hours_tv_per_school_day ‘n()‘
##   <chr>                   <int>
## 1 1                        1750
## 2 2                        2705
## 3 3                        2139
## 4 4                        1048
## 5 5+                       1595
## 6 <1                       2168
## 7 do not watch             1840
## 8 <NA>                      338
```

It will be 7 options, besides the NA option, total 8

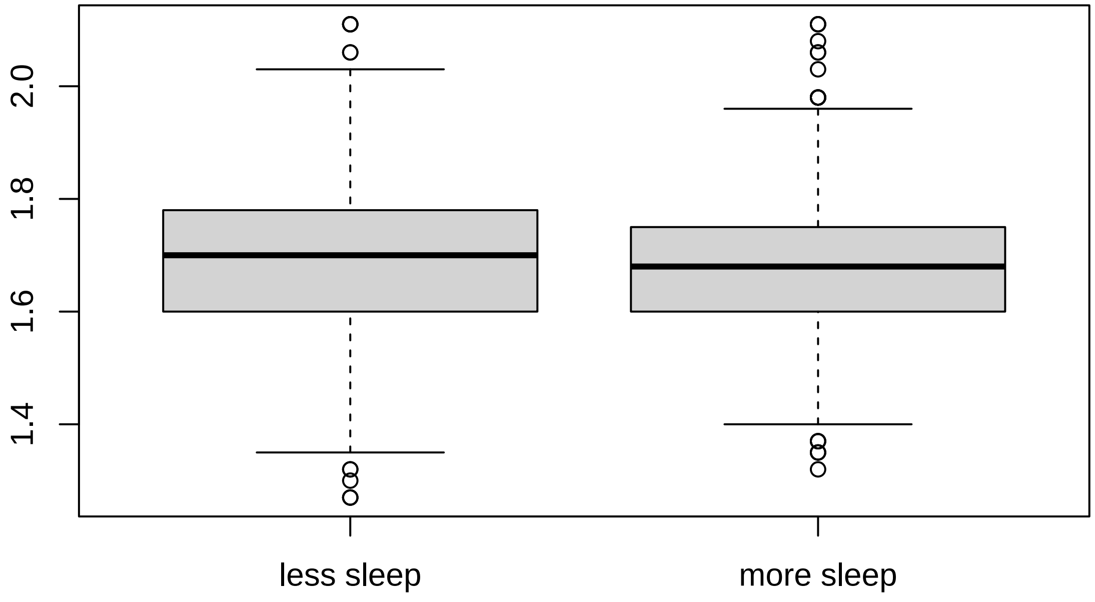
12. Come up with a research question evaluating the relationship between height or weight and sleep. Formulate the question in a way that it can be answered using a hypothesis test and/or a confidence interval. Report the statistical results, and also provide an explanation in plain language. Be sure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

```
yrbss <- yrbss %>%
  mutate(sleep_less = ifelse(yrbss$school_night_hours_sleep < 6, "yes", "no"))

height_less <- yrbss %>%
  select(height, sleep_less) %>%
  filter(sleep_less == "no") %>%
  na.omit()

height_more <- yrbss %>%
  select(height, sleep_less) %>%
  filter(sleep_less == "yes") %>%
  na.omit()
```

```
boxplot(height_less$height, height_more$height,
        names = c("less sleep", "more sleep"))
```



```
less_sleep_mean <- mean(height_less$height)
less_sleep_mean
```

```
## [1] 1.692256
```

```
less_sleep_sd <- sd(height_less$height)
less_sleep_sd
```

```
## [1] 0.1042161
```

```
more_sleep_mean <- mean(height_more$height)
more_sleep_mean
```

```
## [1] 1.685185
```

```
more_sleep_sd <- sd(height_more$height)
more_sleep_sd
```

```
## [1] 0.1059036
```

```
diff_mean <- more_sleep_mean - less_sleep_mean
diff_mean
```

```
## [1] -0.0070715
```

```
diff_sd <- sqrt(((more_sleep_mean^2) / nrow(height_more)) + ((less_sleep_mean^2) / nrow(height_less)))
diff_sd
```

```
## [1] 0.03818596
```

```
sleep_df <- 2492-1
t_sleep <- qt(.05/2, sleep_df, lower.tail = FALSE)

# confidence interval
upper_sleep<- diff_mean + t_sleep * diff_sd
upper_sleep
```

```
## [1] 0.06780798
```

```
lower_sleep<- diff_mean - t_sleep * diff_sd
lower_sleep
```

```
## [1] -0.08195098
```

```
p_value_sleep <- 2 * pt(t_sleep, sleep_df, lower.tail = FALSE)
p_value_sleep
```

```
## [1] 0.05
```

P value is 0.05
