Inference for numerical data

Kevin Havis

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?

Each case, or row, represents a high school student.There are 13,583 observations in this data set (including any nulls)

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"~
```

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?

We are missing weights from 1004 observations.

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.

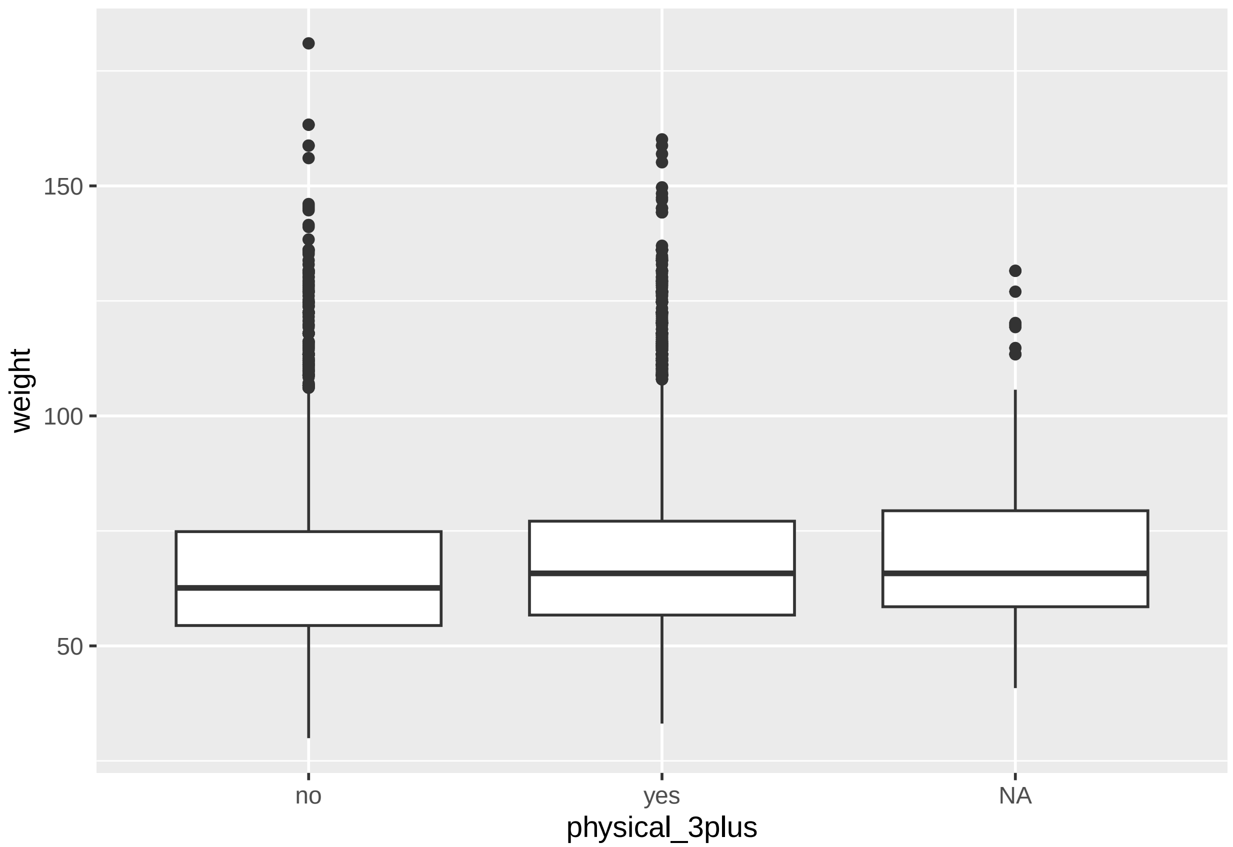
First, let's create a new variable `physical_3plus`, which will be coded as either "yes" if they are physically active for at least 3 days a week, and "no" if not.

```r
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?

From the plot there seems to be a slight relationship between weight and physical activity; the IQR and median weight for students who do engage in 3+ days of physical activity are slightly higher, although there are more more extreme outliers for students who are not active.

```r
ggplot(yrbss, aes(x = physical_3plus, y = weight)) +
  geom_boxplot()
```



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().

Assuming these students were sampled at random, the conditions needed for inference are satisfied as n is greater than 30 for both groups

```
yrbss |>
  group_by(physical_3plus) |>
  summarize(count = n())
```

```
## # A tibble: 3 x 2
##   physical_3plus count
##   <chr>          <int>
## 1 no              4404
## 2 yes             8906
## 3 <NA>             273
```

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.

Null hypothesis: there is no difference between the average weights of those who exercise and those who don't $(H_0 : \bar{x}_y = \bar{x}_n)$

Alternative hypothesis: there is difference between the average weights of those who exercise and those who don't $(H_0 : \bar{x}_y \neq \bar{x}_n)$

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`.

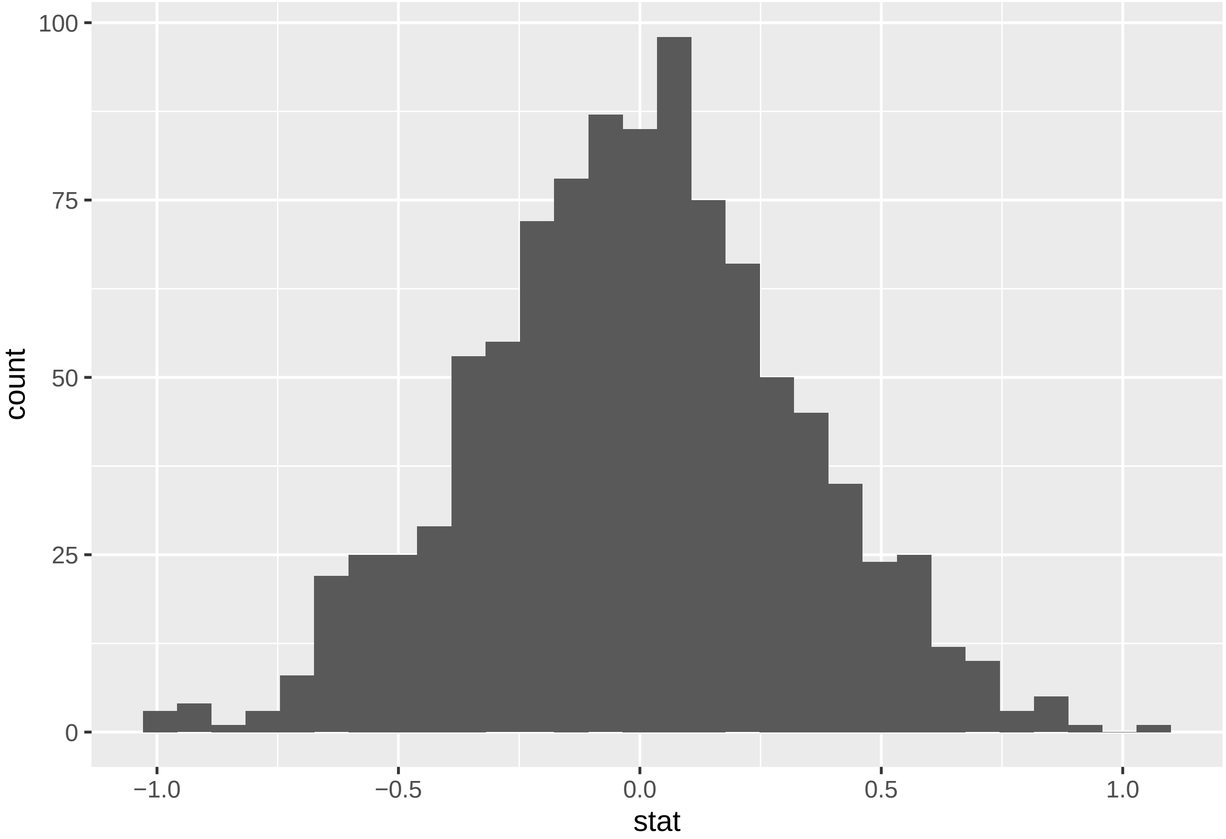
```
null_dist <- yrbss %>%
  drop_na(physical_3plus) %>%
  specify(weight ~ physical_3plus) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "diff in means", order = c("yes", "no"))
```

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`, which is 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`?

Assuming `obs_stat` is supposed to reference the observed difference in means, none of the records from the null distribution are at least obs_stat.

```
obs_stat = 1.77

null_dist |>
  filter(stat >= obs_stat)
```

```
## Response: weight (numeric)
## Explanatory: physical_3plus (factor)
## Null Hypothesis: independence
## # A tibble: 0 x 2
## # i 2 variables: replicate <int>, stat <dbl>
```

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.

The confidence interval at 95% confidence level is [1.10, 2.44] for the observed difference. We would therefore fail to reject the null hypothesis as our point estimate is within this range.

```
set.seed(42)

ci_dist <- yrbss %>%
  drop_na(physical_3plus) %>%
  specify(weight ~ physical_3plus) %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "diff in means", order = c("yes", "no")) |>
  get_confidence_interval(point_estimate = obs_stat, level = .95, type = "se")

print(ci_dist)
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1     1.10     2.44
```

More Practice

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

My point estimate is 1.69, which falls between my 95% confidence interval for 1.688 and 1.692, and as such we would fail to reject the null hypothesis

```
yrbss %>%
  summarise(mean_height = mean(height, na.rm = TRUE))
```

```
## # A tibble: 1 x 1
##   mean_height
##         <dbl>
## 1        1.69
```

```
height_obs_mean <- 1.69

height_dist <- yrbss %>%
  drop_na(height) %>%
  specify(response = height) %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "mean") |>
  get_confidence_interval(point_estimate = height_obs_mean, level = .95, type = "se")
```

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.

The confidence interval does not change materially; our confidence interval at 90% level is 1.688 to 1.691, which is slightly narrower. We would still fail to reject the null hypothesis.

```
height_dist_90 <- yrbss %>%
  drop_na(height) %>%
  specify(response = height) %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "mean") |>
  get_confidence_interval(point_estimate = height_obs_mean, level = .90, type = "se")
```

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.

Null hypothesis: There is no difference in average height for those who exercise at least 3x a week

Alternative hypothesis: There is a difference in average height for those who exercise at least 3x a week

Our observed mean difference in height for those who exercise 3+ days a week versus those who do not is 0.0376. Our confidence interval falls at (0.337, 0.0416) which captures our observed mean difference, so we would fail to reject the null hypothesis

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

```
## # A tibble: 3 x 2
##   physical_3plus mean_height
##   <chr>                <dbl>
## 1 no                    1.67
## 2 yes                   1.70
## 3 <NA>                  1.71
```

```
obs_diff_height <- yrbss %>%
  drop_na(physical_3plus) %>%
  specify(height ~ physical_3plus) %>%
  calculate(stat = "diff in means", order = c("yes", "no"))
```

```
null_dist_height <- yrbss %>%
  drop_na(physical_3plus) %>%
  specify(height ~ physical_3plus) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "diff in means", order = c("yes", "no")) |>
  get_confidence_interval(point_estimate = obs_diff_height, level = .95, type = "se")
```

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.

There are 7 unique values of `hours_tv_per_school_day`, excluding nulls.

```
unique(yrbss$hours_tv_per_school_day)
```

```
## [1] "5+"           "2"            "3"            "do not watch" "<1"
## [6] "4"            "1"            NA
```

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.

Question: Does sleeping for less than 8 hours a day affect weight in high school students?

Null hypothesis: There is no relationship between weight and sleep, when sleep is less than 8 hours a day

Alternative hypothesis: There is a relationship between weight and sleep, when sleep is less than 8 hours a day

Conditions for independence and sample size are satisfied. We calculate our observed difference in means to be 0.5. We proceed to calculate a confidence interval for alpha = 0.05 and find it to be (-0.107, 1,11). Given our point estimate falls within this range, we fail to reject the null hypothesis.

To summarize, we are 95% confident there is not a significant difference between students who sleep more or less than 8 hours, and their weight.

```
unique(yrbss$school_night_hours_sleep)
```

```
## [1] "8"   "6"   "<5"  "9"   "10+" "7"   "5"   NA
```

```
yrbss <- yrbss %>%
  mutate(sleep_less8 = ifelse(yrbss$school_night_hours_sleep %in%(c("6", "<5", "7")), "yes", "no"))
```

```
yrbss |>
  group_by(sleep_less8) |>
  summarize(count = n())
```

```
## # A tibble: 2 x 2
##   sleep_less8 count
##   <chr>       <int>
## 1 no           6499
## 2 yes          7084
```

```
obs_diff_sleep8 <- yrbss %>%
  drop_na(sleep_less8) %>%
  specify(weight ~ sleep_less8) %>%
  calculate(stat = "diff in means", order = c("yes", "no"))
```

```
null_dist_sleep8 <- yrbss %>%
  drop_na(sleep_less8) %>%
  specify(weight ~ sleep_less8) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "diff in means", order = c("yes", "no")) |>
  get_confidence_interval(level = 0.95, point_estimate = obs_diff_sleep8, type="se")
```
