Inference for numerical 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

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?

This dataset contains 13,583 rows, which corresponds to the total number of cases in the sample. Out of these, there are 13 unique types of cases that we can see by executing the below query.

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.

```
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?

IThere are 9476 missing values in total.

```
sum(is.na(yrbss))
```

```
## [1] 9476
```

There are 1004 missing entries in the weight observations (Execute below query).

```
library(dplyr)
yrbss %>% summarise_all(~ sum(is.na(.)))
```

```
## # A tibble: 1 x 13
##     age gender grade hispanic  race height weight helmet_12m
##   <int>  <int> <int>    <int> <int>  <int>  <int>      <int>
## 1    77     12    79      231  2805   1004   1004        311
## # i 5 more variables: text_while_driving_30d <int>, physically_active_7d <int>,
## #   hours_tv_per_school_day <int>, strength_training_7d <int>,
## #   school_night_hours_sleep <int>
```

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.

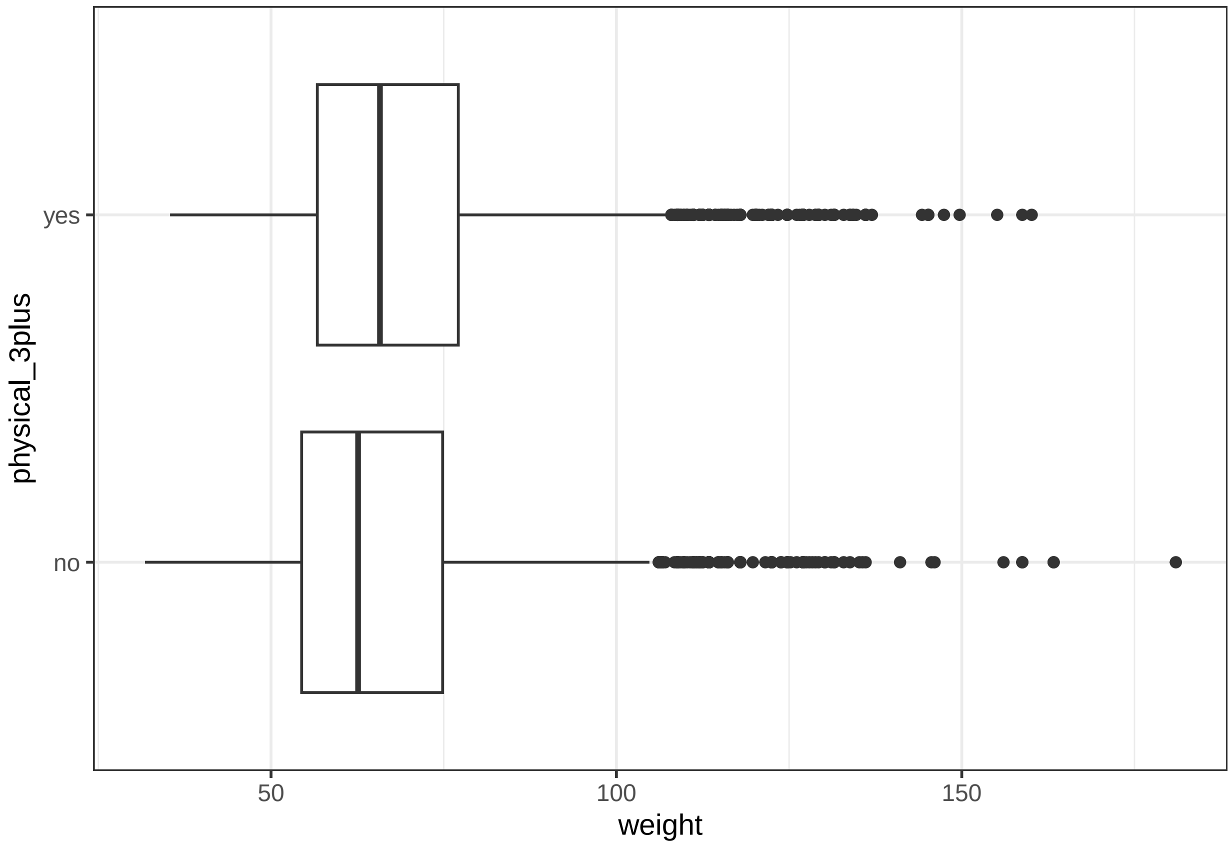
First, let's create a new variable `physical_3plus`, which will be coded as either "yes" if they are physically active for at least 3 days a week, and "no" if not.

```
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?

```
yrbss2 <- yrbss %>%
mutate(physical_3plus = if_else(physically_active_7d > 2, "yes", "no")) %>%
drop_na()

ggplot(data = yrbss2, aes(x = weight, y = physical_3plus)) +
geom_boxplot() +
theme_bw()
```



The boxplot show's the connection between a student's weight and their physical activity, specifically if they are active at least 3 times a week, shows that students who are less active actually weigh less than those who are active at least 3 times a week. This is surprising for us because we usually expect that people who exercise more would weigh less.

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(mean_weight = mean(weight, na.rm = TRUE), count = sum(!is.na(weight)))
```

```
## # A tibble: 3 x 3
##   physical_3plus mean_weight count
##   <chr>                <dbl> <int>
## 1 no                    66.7  4022
## 2 yes                   68.4  8342
## 3 <NA>                  69.9   215
```

We can have two cases here:-

1. Independent sample (These are assumed to be true, though not completely certain or correct) 2. Normality - with at least 30 samples (This can be confirmed)

Yes, the conditions are met. The 'count' variable has been added in the above code to confirm the success-failure condition.

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 (H0): Students who are physically active at least 3 days a week have the same average weight as those who are not physically active at least 3 days a week.

Alternative Hypothesis (HA): Students who are physically active at least 3 days a week have a different average weight compared to those who are not.

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`.

```r
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`.

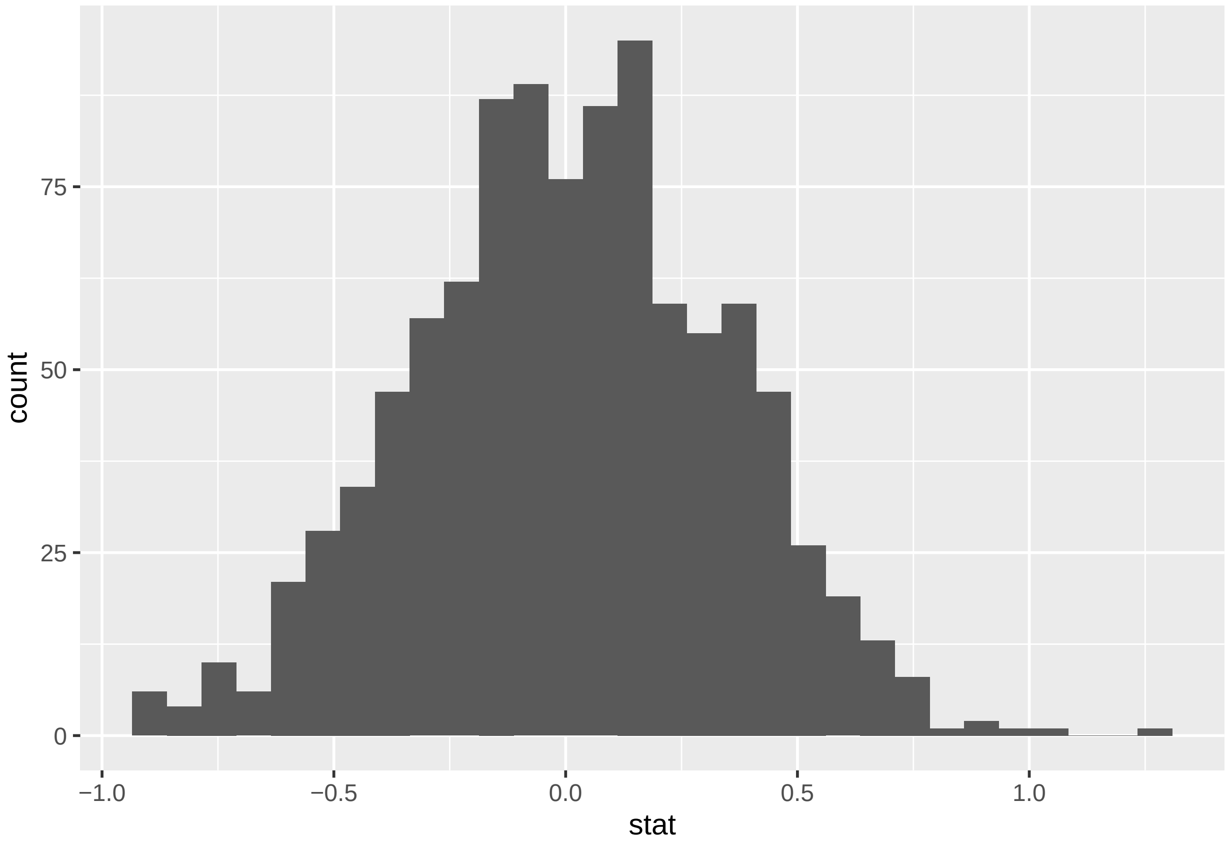
```r
null_dist <- yrbss %>%
  drop_na(physical_3plus) %>%
  specify(weight ~ physical_3plus) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "diff in means", order = c("yes", "no"))
```

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.

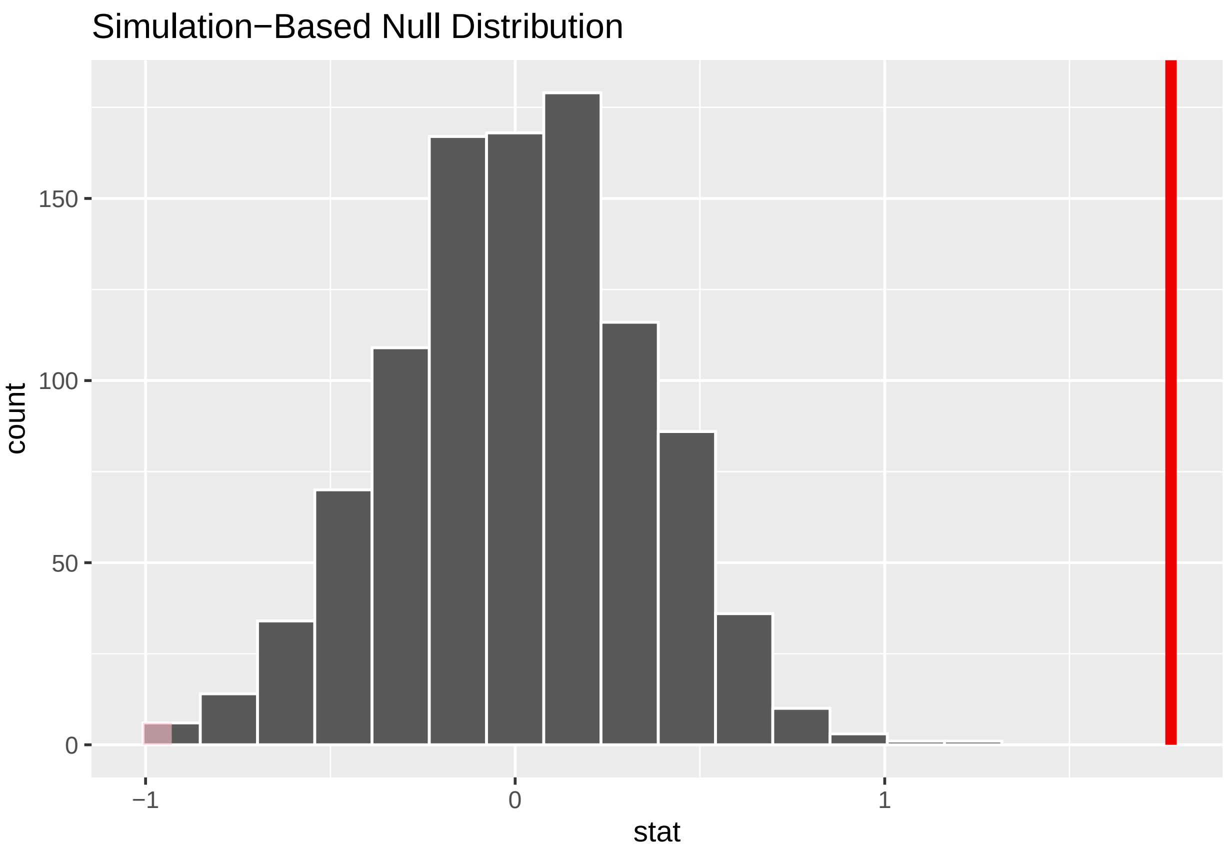
We can visualize this null distribution with the following code:

```r
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 result is a tiny value, nearly zero.

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 = sqrt(var(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
```

The standard deviation is 17.6 for individuals who are not physically active at least 3 days a week, while it is 16.5 for those who are physically active.

Now we will calculate the average weights (Mean).

```
# Calculating the average weights
yrbss %>%
  group_by(physical_3plus) %>%
  summarise(average_weight = mean(weight, na.rm = TRUE))
```

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

Calculating the sample size for each category.

```
# Sample size for each category
yrbss %>%
  group_by(physical_3plus) %>%
  summarise(n = n())
```

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

Values that are given and calculating the upper/lower confidence intervals.

```
# Given values
xnot3 <- 66.67389
nnot3 <- 4404
snot3 <- 17.6
z <- 1.96  # This is the Z-value for 95% confidence interval

# Calculating the upper and lower confidence intervals
standard_error_not <- snot3 / sqrt(nnot3)
uci_not <- xnot3 + z * standard_error_not
lci_not <- xnot3 - z * standard_error_not

uci_not
```

```
## [1] 67.1937
```

```
lci_not
```

```
## [1] 66.15408
```

Now, calculating the upper and lower confidence intervals for the second group.

```
# Calculating the upper and lower confidence intervals for the second group
x3 <- 68.4
n3 <- 8906
s3 <- 16.5

standard_error_3 <- s3 / sqrt(n3)
u_ci <- x3 + z * standard_error_3
l_ci <- x3 - z * standard_error_3
u_ci
```

```
## [1] 68.74269
```

Lower confidence interval.

```
l_ci
```

```
## [1] 68.05731
```

As from above calculations, we are 95% confident that students who exercise at least three times a week have an average weight ranging from 68.05 kg to 68.74 kg. Similarly, students who do not exercise at least three times a week have an average weight between 66.15 kg and 67.19 kg, also with 95% confidence.

More Practice

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

First of all, we will extract height data and remove missing values. After that we will calculate the mean, standard deviation, maximum and standard error. Post that we will check the extreme outliers.

```
# Extracting height data and removing missing values
height_data <- yrbss %>%
  select(height) %>%
  drop_na()

# Now, calculating mean, standard deviation, maximum, and standard error
mean_height <- mean(height_data$height)
sd_height <- sd(height_data$height)
max_height <- max(height_data$height)
standard_error_height <- sd_height / sqrt(nrow(height_data))

# Checking for extreme outliers
cat("The maximum height in the data is", max_height, "and the mean plus 2.5 times the standard deviatio
## The maximum height in the data is 2.11 and the mean plus 2.5 times the standard deviation is 1.95298
```

The maximum value in the height data is 2.11, and the mean plus 2.5 times the standard deviation is 1.952984.

Now, we will calculate the t-value for the confidence interval and compute the upper and lower bounds of the confidence interval.

```r
# Calculating the t-value for the confidence interval
t_value_height <- qt(0.05 / 2, df = nrow(height_data) - 1, lower.tail = FALSE)

# Computing the upper and lower bounds of the confidence interval
upper_bound_height <- mean_height + t_value_height * standard_error_height
lower_bound_height <- mean_height - t_value_height * standard_error_height

cat("The 95% confidence interval ranges from", round(lower_bound_height, 3), "meters to", round(upper_b
```

```
## The 95% confidence interval ranges from 1.689 meters to 1.693 meters.
```

The 95% confidence interval ranges between 1.689 meters and 1.693 meters.

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.

```r
# Calculating the t-value for the 90% confidence interval
t_value_height_90 <- qt(0.1 / 2, df = nrow(height_data) - 1, lower.tail = FALSE)

# Determining the upper and lower limits of the 90% confidence interval
upper_limit_height_90 <- mean_height + t_value_height_90 * standard_error_height
lower_limit_height_90 <- mean_height - t_value_height_90 * standard_error_height

cat("The 90% confidence interval extends from", round(lower_limit_height_90, 3), "meters to", round(uppe
```

```
## The 90% confidence interval extends from 1.69 meters to 1.693 meters.
```

The 90% confidence interval ranges from 1.69 meters to 1.693 meters.

There is a small difference between the two confidence intervals in 8th and 9th question. Additionally, we see the 95% confidence interval has a slightly wider range compared to the 90% confidence interval.

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 height between individuals who exercise at least three times a week and those who do not.

Conditions for inference regarding the difference in means:

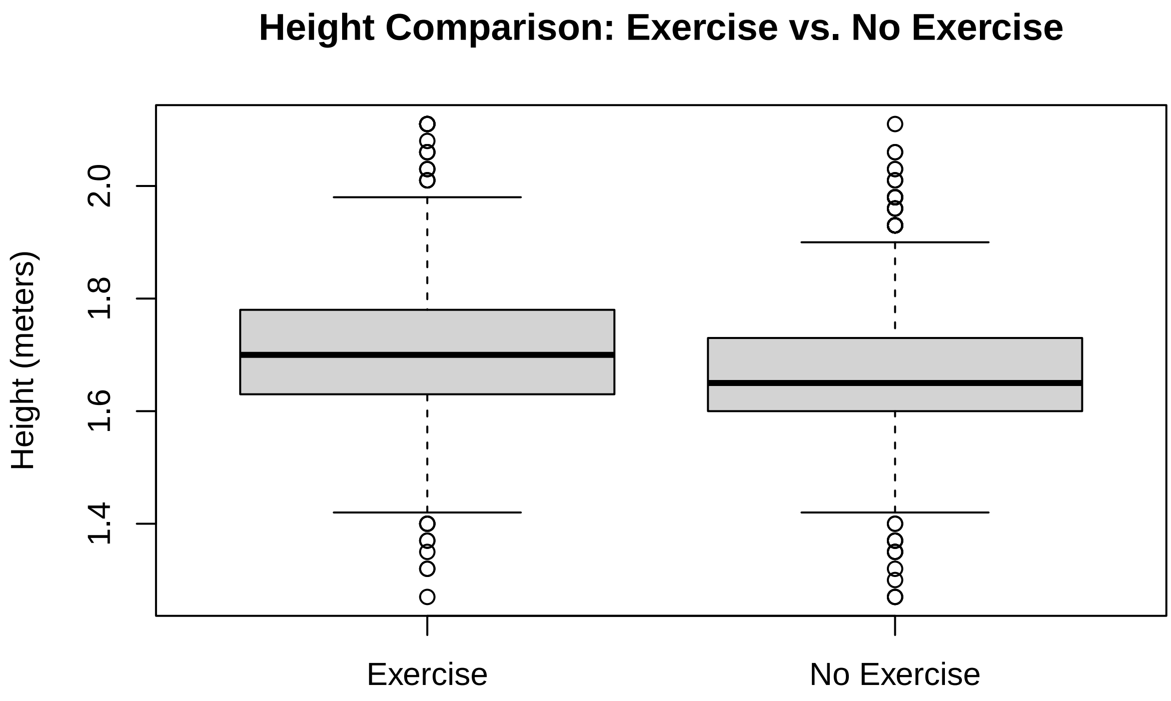
1. Independent samples (This can be assumed to be yes, but it's not entirely certain or correct)
2. Normality – while there are a few outliers, they are not excessively extreme.

Firstly, we will filter the height data for those who exercise and those who do not.

```r
# Filtering height data for those who exercise and those who do not
height_exercise <- yrbss %>%
  filter(physical_3plus == "yes") %>%
  pull(height) %>%
  na.omit()

height_noexercise <- yrbss %>%
  filter(physical_3plus == "no") %>%
  pull(height) %>%
  na.omit()

# Creating a box plot for comparison
boxplot(height_exercise, height_noexercise,
        names = c("Exercise", "No Exercise"),
        main = "Height Comparison: Exercise vs. No Exercise",
        ylab = "Height (meters)")
```



Now, we will calculate the statistics for the group who does not exercise and who does.

```r
# Calculating statistics for the group that does not exercise
mean_noexercise <- mean(height_noexercise)
sd_noexercise <- sd(height_noexercise)
max_noexercise <- max(height_noexercise)

# Calculating statistics for the group that exercises
mean_exercise <- mean(height_exercise)
```

```r
sd_exercise <- sd(height_exercise)
max_exercise <- max(height_exercise)

# Displaying the maximum height and mean plus 2.5 standard deviations for the non-exercise group
cat("The maximum height for those not exercising is", max_noexercise, "and the mean plus 2.5 times the
```

```
## The maximum height for those not exercising is 2.11 and the mean plus 2.5 times the standard deviati
```

```r
# Displaying the maximum height and mean plus 2.5 times the standard deviation for the exercise group
cat("The maximum height for those who exercise is", max_exercise, "and the mean plus 2.5 times the stan
```

```
## The maximum height for those who exercise is 2.11 and the mean plus 2.5 times the standard deviation
```

Calculating the mean difference, standard error, degrees of freedom, t-value and mainly the confidence interval

```r
# Calculate the mean difference
mean_difference <- mean_exercise - mean_noexercise

# Calculate the standard error
standard_error <- sqrt(
  (sd_exercise^2 / length(height_exercise)) +
  (sd_noexercise^2 / length(height_noexercise))
)

# Calculate the degrees of freedom and T-value for the confidence interval
degrees_freedom <- length(height_noexercise) + length(height_exercise) - 2
t_value <- qt(0.05 / 2, degrees_freedom, lower.tail = FALSE)

# Calculate the confidence interval
upper_limit <- mean_difference + t_value * standard_error
lower_limit <- mean_difference - t_value * standard_error

# Displaying the confidence interval
cat("The 95% confidence interval ranges from", round(lower_limit, 2), "to", round(upper_limit, 2), "\n"
```

```
## The 95% confidence interval ranges from 0.03 to 0.04
```

Calculating the P-value.

```r
# Calculate the P-value
p_value <- 2 * pt(t_value, degrees_freedom, lower.tail = FALSE)

# Display the P-value and the conclusion about the null hypothesis
cat("The P-value is", p_value, ", which indicates that the null hypothesis can be rejected.\n")
```

```
## The P-value is 0.05 , which indicates that the null hypothesis can be rejected.
```

According to the calculation done above, the P-value is 0.05, indicating that we are good to reject the null hypothesis. However, this result is unexpected, and seems like it may lead to a Type I error.

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.

```
# Counting observations for each category of hours spent watching TV per school day
tv_counts <- yrbss %>%
  count(hours_tv_per_school_day)

# Displaying the results
print(tv_counts)
```

```
## # 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
```

There are 7 distinct options, in addition to 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.

Research Question: Is there significant evidence to suggest that students with heights above the average sleep more similar than those with heights below the average?

Null Hypothesis: There is no association between students' heights relative to the average and the quality of their sleep.

Alpha Level is: 0.05 (indicating 95% confidence).

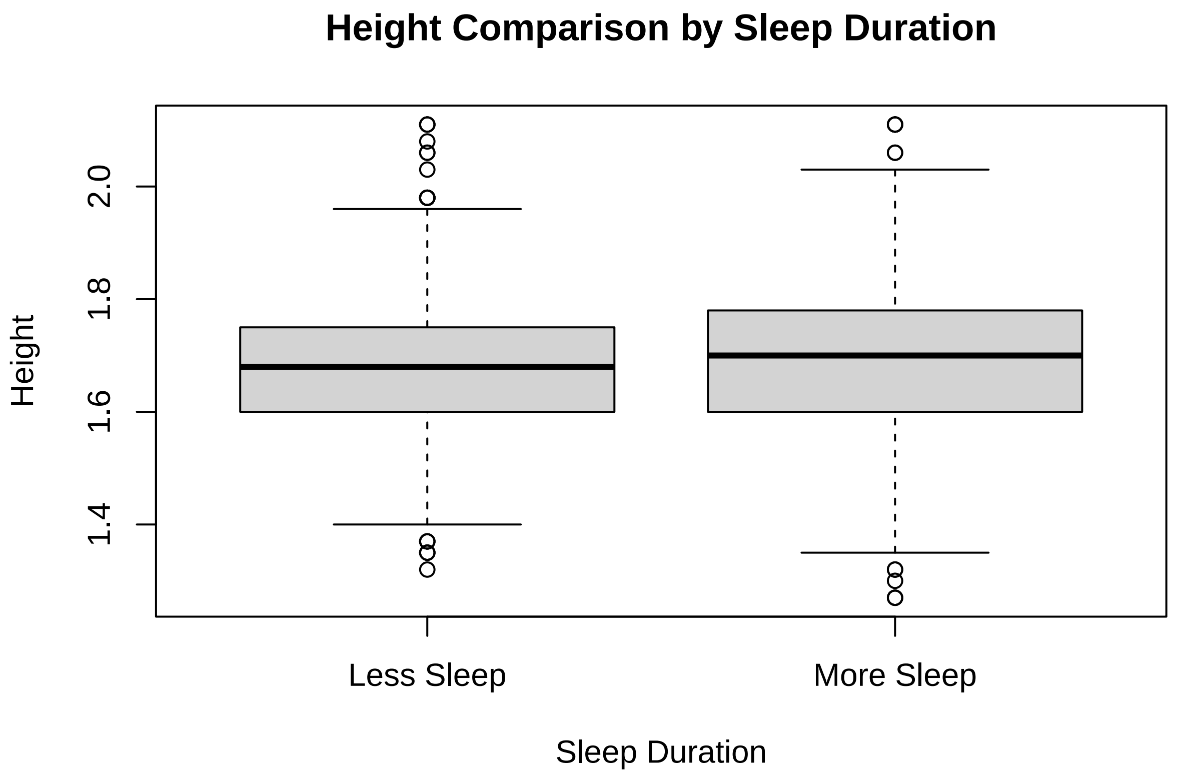
Creating a new column, filtering the dataset and creating a boxplot.

```
# Creating a new column to indicate whether students sleep more than 6 hours on school nights
yrbss <- yrbss %>%
  mutate(sleep_6plus = ifelse(school_night_hours_sleep > 5, "yes", "no"))

# Filtering the dataset for students who sleep less than 6 hours
height_less_sleep <- yrbss %>%
  filter(sleep_6plus == "no") %>%
  select(height, sleep_6plus) %>%
  na.omit()

# Filtering the dataset for students who sleep 6 or more hours
height_more_sleep <- yrbss %>%
  filter(sleep_6plus == "yes") %>%
  select(height, sleep_6plus) %>%
  na.omit()
```

```r
# Creating a box plot to visualize height differences based on sleep duration
boxplot(height_less_sleep$height, height_more_sleep$height,
        names = c("Less Sleep", "More Sleep"),
        main = "Height Comparison by Sleep Duration",
        ylab = "Height",
        xlab = "Sleep Duration")
```



Calculating statistics.

```r
# Calculating statistics for students who sleep less than 6 hours
mean_less_sleep <- mean(height_less_sleep$height)
sd_less_sleep <- sd(height_less_sleep$height)
max_less_sleep <- max(height_less_sleep$height)

# Calculating statistics for students who sleep 6 or more hours
mean_more_sleep <- mean(height_more_sleep$height)
sd_more_sleep <- sd(height_more_sleep$height)
max_more_sleep <- max(height_more_sleep$height)

# Result for the maximum height of students who sleep less than 6 hours and the mean plus 2.5 standard
cat("The maximum height of students who sleep less than 6 hours is", max_less_sleep,
    "and the mean plus 2.5 standard deviations is", (mean_less_sleep + (2.5 * sd_less_sleep)), "\n")
```

```
## The maximum height of students who sleep less than 6 hours is 2.11 and the mean plus 2.5 standard de
```

Result

```r
# Result for the maximum height of students who sleep 6 or more hours and the mean plus 2.5 standard de
cat("The maximum height of students who sleep 6 or more hours is", max_more_sleep,
    "and the mean plus 2.5 standard deviations is", (mean_more_sleep + (2.5 * sd_more_sleep)), "\n")
```

```
## The maximum height of students who sleep 6 or more hours is 2.11 and the mean plus 2.5 standard devia
```

Calculating the mean difference, standard error, degree of freedom, t-value and confidence interval

```r
# Calculating the mean difference in height between the two groups
mean_diff_height_sleep <- mean_more_sleep - mean_less_sleep

# Calculating the standard error
standard_error_height_sleep <-
  sqrt(
    (mean_more_sleep^2 / nrow(height_more_sleep)) +
    (mean_less_sleep^2 / nrow(height_less_sleep))
  )

# Calculating the degrees of freedom and the T-value
degrees_of_freedom_height_sleep <- nrow(height_more_sleep) + nrow(height_less_sleep) - 2
t_value_height_sleep <- qt(0.05 / 2, degrees_of_freedom_height_sleep, lower.tail = FALSE)

# Calculating the confidence interval
right_interval_height_sleep <- mean_diff_height_sleep + t_value_height_sleep * standard_error_height_sle
left_interval_height_sleep <- mean_diff_height_sleep - t_value_height_sleep * standard_error_height_slee

# Confidence interval result
cat("The 95% confidence interval is from", round(left_interval_height_sleep, 2),
    "meters to", round(right_interval_height_sleep, 2), "meters\n")
```

```
## The 95% confidence interval is from -0.07 meters to 0.08 meters
```

Calculating the p-value.

```r
# Calculating the P-value
p_value_height_sleep <- 2 * pt(t_value_height_sleep, degrees_of_freedom_height_sleep, lower.tail = FALSE

# The P-value result
cat("The P-value is", p_value_height_sleep, "and thus the null hypothesis can be rejected.\n")
```

```
## The P-value is 0.05 and thus the null hypothesis can be rejected.
```

The P-value is 0.05, indicating that we can reject the null hypothesis.
