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?

Insert your answer here There are 13583 cases.

```
dim(yrbss)
```

```
## [1] 13583    13
```

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?

Insert your answer here 1004 observations are missing weights from.

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"))
```

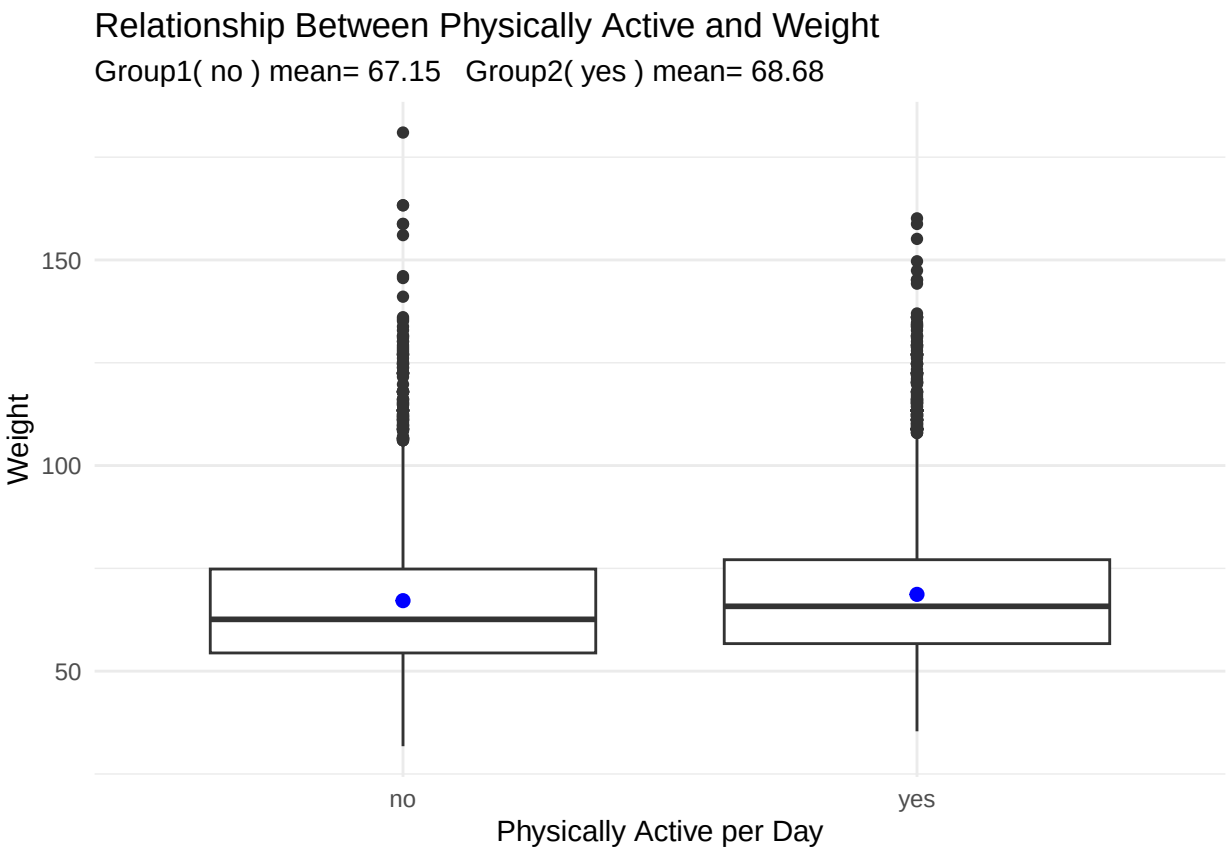
3. Make a side-by-side boxplot of `physical_3plus` and `weight`. Is there a relationship between these two variables? What did you expect and why?

Insert your answer here In the boxplot we can see that both means are higher than their medians and they are also pretty close to each other. Group 1 that didn't have physical activity at least 3 days got a mean weight of 67.15 and Group2 that have physical activity at least 3 days a week got a mean of 68.68.

```r
#line inside represent median
#circles inside mean
library(psych)
yrbss_c <- yrbss|>
  drop_na()

Summary_groups <- describeBy(yrbss_c$weight,
                  group=yrbss_c$physical_3plus, #same value as X aes
                  mat=TRUE, skew=FALSE)
Summary_groups
```

```
##     item group1 vars    n     mean       sd   min    max  range        se
## X11    1     no    1 2656 67.14974 17.99041 31.75 180.99 149.24 0.3490816
## X12    2    yes    1 5695 68.67747 16.41617 35.38 160.12 124.74 0.2175329
```

```r
ggplot(yrbss_c, aes(x = factor(physical_3plus), y = weight )) +
  geom_boxplot() +
  geom_point(data = Summary_groups, aes(x=group1, y=mean),
             color='blue', size=2)+
  labs(
    x = "Physically Active per Day",
    y = "Weight",
    title = "Relationship Between Physically Active and Weight",
    subtitle = paste("Group1(",Summary_groups[1,2],") mean=",round(Summary_groups[1,5],2)," ",
                     "Group2(",Summary_groups[2,2],") mean=",round(Summary_groups[2,5],2))
  )+
  theme_minimal()
```



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

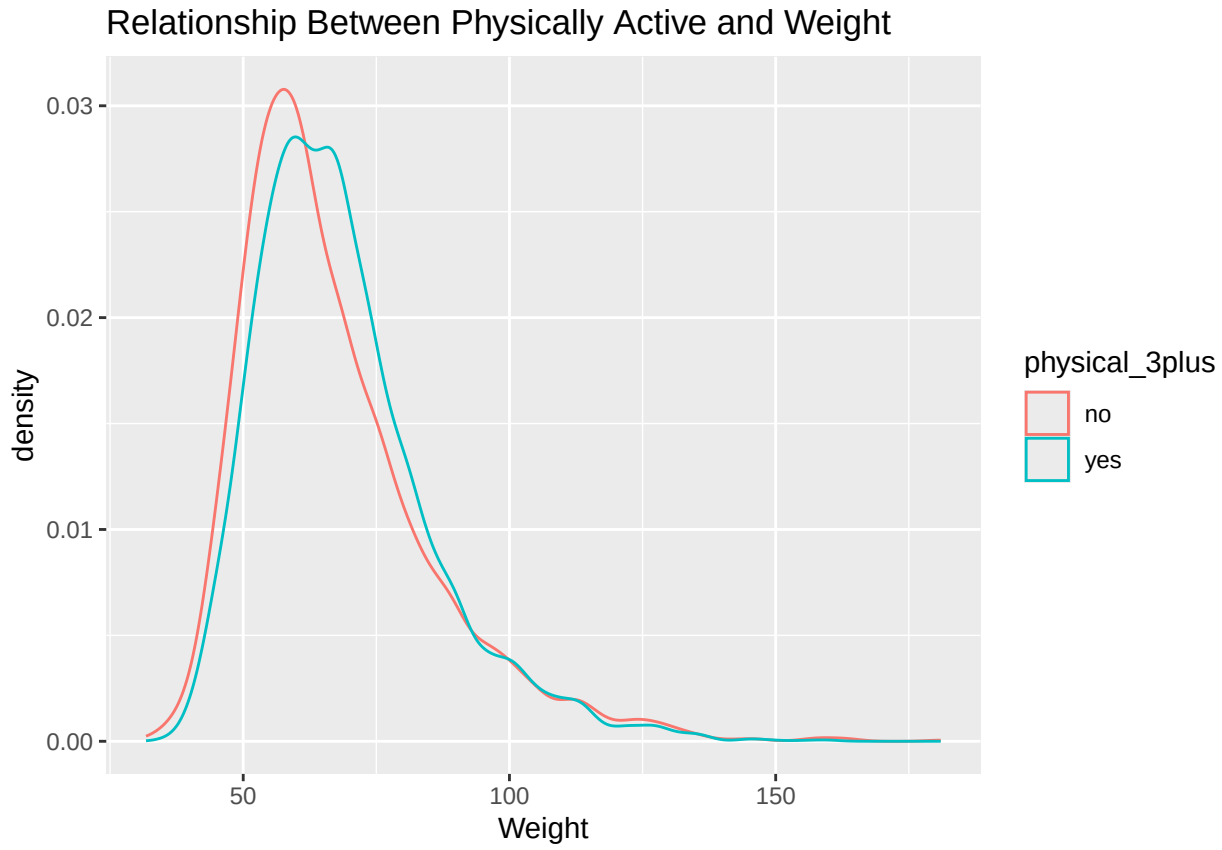
4. Are all conditions necessary for inference satisfied? Comment on each. You can compute the group sizes with the `summarize` command above by defining a new variable with the definition `n()`.

Insert your answer here Yes,it's randomized sample, the size of the sample is bigger than 30.

```
yrbss |>
  group_by(physical_3plus) |>
  summarise(mean_weight = mean(weight, na.rm = TRUE),
            sd_weight = sd(weight, na.rm = TRUE),
            n = n(),
            p_hat = n()/nrow(yrbss))|>
  drop_na()
```

```
## # A tibble: 2 x 5
##   physical_3plus mean_weight sd_weight     n p_hat
##   <chr>                <dbl>     <dbl> <int> <dbl>
## 1 no                    66.7      17.6  4404 0.324
## 2 yes                   68.4      16.5  8906 0.656
```

```
ggplot(yrbss_c, aes(x=weight, color = physical_3plus)) + geom_density()+
  labs(
    x = "Weight",
    title = "Relationship Between Physically Active and Weight")
```



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.

Insert your answer here

H0: The average weights for those who exercise at least 3 days a week are the same as those who don't udiff=0

H1: The average weights for those who exercise at least 3 days a week are different than those who don't udiff != 0

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"))

obs_diff
```

```
## Response: weight (numeric)
## Explanatory: physical_3plus (factor)
## # A tibble: 1 x 1
##    stat
##   <dbl>
## 1  1.77
```

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"))

null_dist
```

```
## Response: weight (numeric)
## Explanatory: physical_3plus (factor)
## Null Hypothesis: independence
## # A tibble: 1,000 x 2
##    replicate     stat
##        <int>    <dbl>
##  1         1 -0.294
##  2         2 -0.0734
##  3         3  0.104
##  4         4  0.163
##  5         5  0.00380
##  6         6  0.125
##  7         7 -0.241
##  8         8  0.327
```

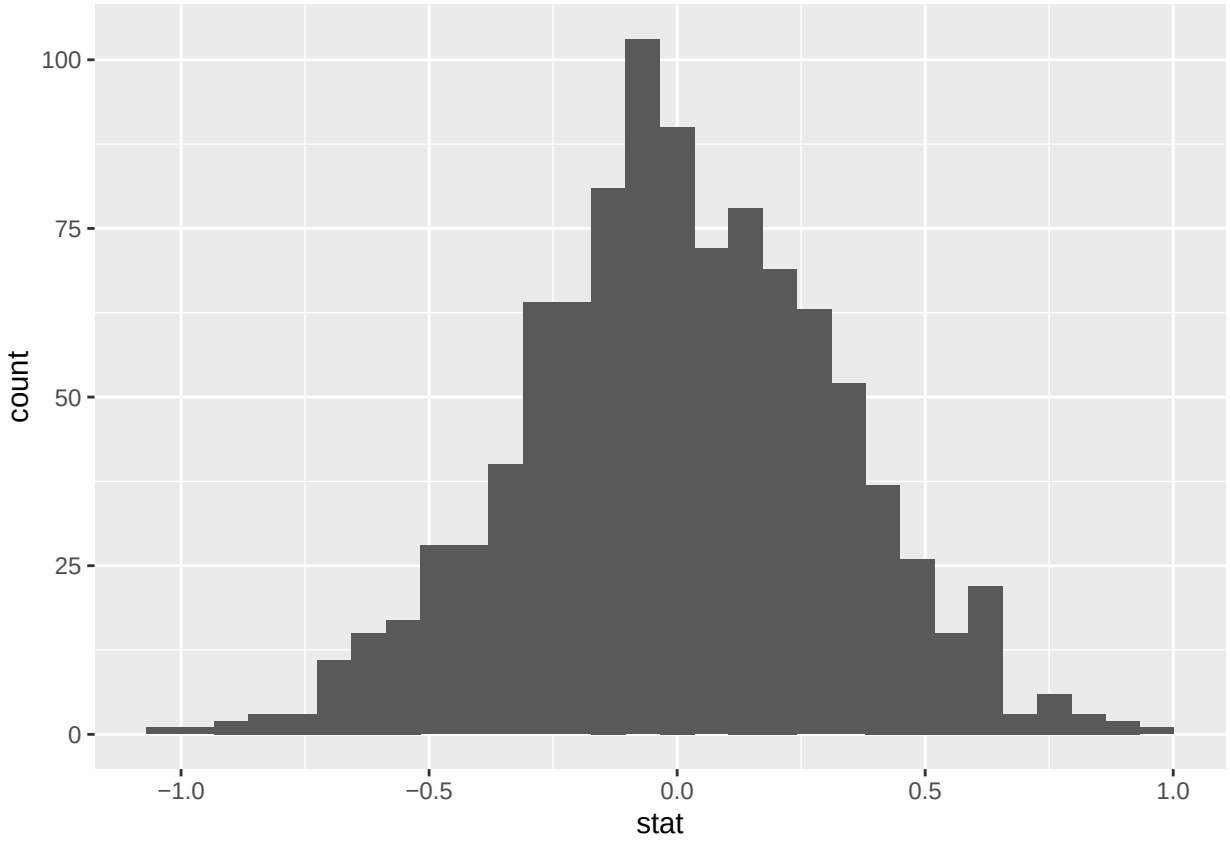
```
##  9         9  0.0141
## 10        10 -0.0539
## # i 990 more rows
```

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`?

Insert your answer here none

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.

---

there is no much difference between both means. The p_value is 0.05 so we will fail to request null hypotheses however based on the confidence interval :(0.659893763265534 , 2.39555821188704) we will reject H0 because we should get intervals close to 0.

```
library(psych)
yrbss_c <- yrbss|>
  drop_na()

Summary_groups <- describeBy(yrbss_c$weight,
                  group=yrbss_c$physical_3plus, #same value as X aes
                  mat=TRUE, skew=FALSE)
#Summary_groups

#activity-noweight
nact_wei_n <- Summary_groups[1,4]
nact_wei_mean <- Summary_groups[1,5]
nact_wei_sd <- Summary_groups[1,6]

#activity-weight
act_wei_n <- Summary_groups[2,4]
act_wei_mean <- Summary_groups[2,5]
act_wei_sd <- Summary_groups[2,6]


#95%confidence
#z_score <- 1.96

#activity-weight-intervals
#act_wei_upperinterval <- act_wei_mean + z_score*(act_wei_sd/sqrt(act_wei_n))
#act_wei_lowerinterval <- act_wei_mean - z_score*(act_wei_sd/sqrt(act_wei_n))
#paste("exercise intervals(",act_wei_lowerinterval,",",act_wei_upperinterval,")")


#no-activity-weight-intervals
#nact_wei_upperinterval <- nact_wei_mean + z_score*(nact_wei_sd/sqrt(nact_wei_n))
#nact_wei_lowerinterval <- nact_wei_mean - z_score*(nact_wei_sd/sqrt(nact_wei_n))
#paste("no exercise intervals(",nact_wei_lowerinterval,",",nact_wei_upperinterval,")")

#SE(Xa-Xna)
SE_Xa_Xn <-  sqrt((act_wei_sd^2/act_wei_n)+(nact_wei_sd^2/nact_wei_n))
SE_Xa_Xn
```

```
## [1] 0.4113132
```

```
Point_Estimate <- (act_wei_mean - nact_wei_mean)
#95%confidence interval:
#(Xa-Xna)+-196*SE_Xa_Xn
#ch <- Point_Estimate + z_score * SE_Xa_Xn
#cl <- Point_Estimate - z_score * SE_Xa_Xn
#point estimate +- margin of error,2 independent means
#paste("The confidence interval is ",cl," to ", ch)

#df
df <- min(act_wei_n - 1,nact_wei_sd - 1)
t_value <- qt(0.05/2, df, lower.tail = FALSE) # lower.tail = FALSE get positive value
cl <- Point_Estimate - t_value * SE_Xa_Xn
ch <- Point_Estimate + t_value * SE_Xa_Xn
paste("The confidence interval is ",cl," to ", ch)
```

```
## [1] "The confidence interval is  0.659893763265534  to  2.39555821188704"
```

```
p_value <- 2*pt(t_value, df, lower.tail = FALSE)
p_value
```

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

```
#H0 :act_wei_mean - nact_wei_mean =0
#HA :act_wei_mean - nact_wei_mean /=0

tail_area <- (Point_Estimate - 0 )/SE_Xa_Xn
```

More Practice

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

Insert your answer here There is 95% confidence that the height for the population of the sample is between the interval 1.6948 and 1.6993

```
yrbss_c <- yrbss|>
  drop_na()

yrbss_c_height <- yrbss_c %>%
  summarise(mean_height = mean(height),sd_height = sd(height),n = n())

z_score <- 1.96

height_upper <- yrbss_c_height$mean_height + z_score*(yrbss_c_height$sd_height/sqrt(yrbss_c_height$n))
height_lower <- yrbss_c_height$mean_height - z_score*(yrbss_c_height$sd_height/sqrt(yrbss_c_height$n))


paste("The confidence interval is ",height_lower," to ", height_upper)
```

```
## [1] "The confidence interval is  1.69481054772742  to  1.69929794227378"
```

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.

Insert your answer here There is 90% confidence that the height for the population of the sample is between the interval 1.69517 and 1.69894

```
yrbss_c <- yrbss|>
  drop_na()

yrbss_c_height <- yrbss_c %>%
  summarise(mean_height = mean(height),sd_height = sd(height),n = n())

z_score <- 1.645

height_upper <- yrbss_c_height$mean_height + z_score*(yrbss_c_height$sd_height/sqrt(yrbss_c_height$n))
height_lower <- yrbss_c_height$mean_height - z_score*(yrbss_c_height$sd_height/sqrt(yrbss_c_height$n))


paste("The confidence interval is ",height_lower," to ", height_upper)
```

```
## [1] "The confidence interval is  1.69517114193204  to  1.69893734806916"
```

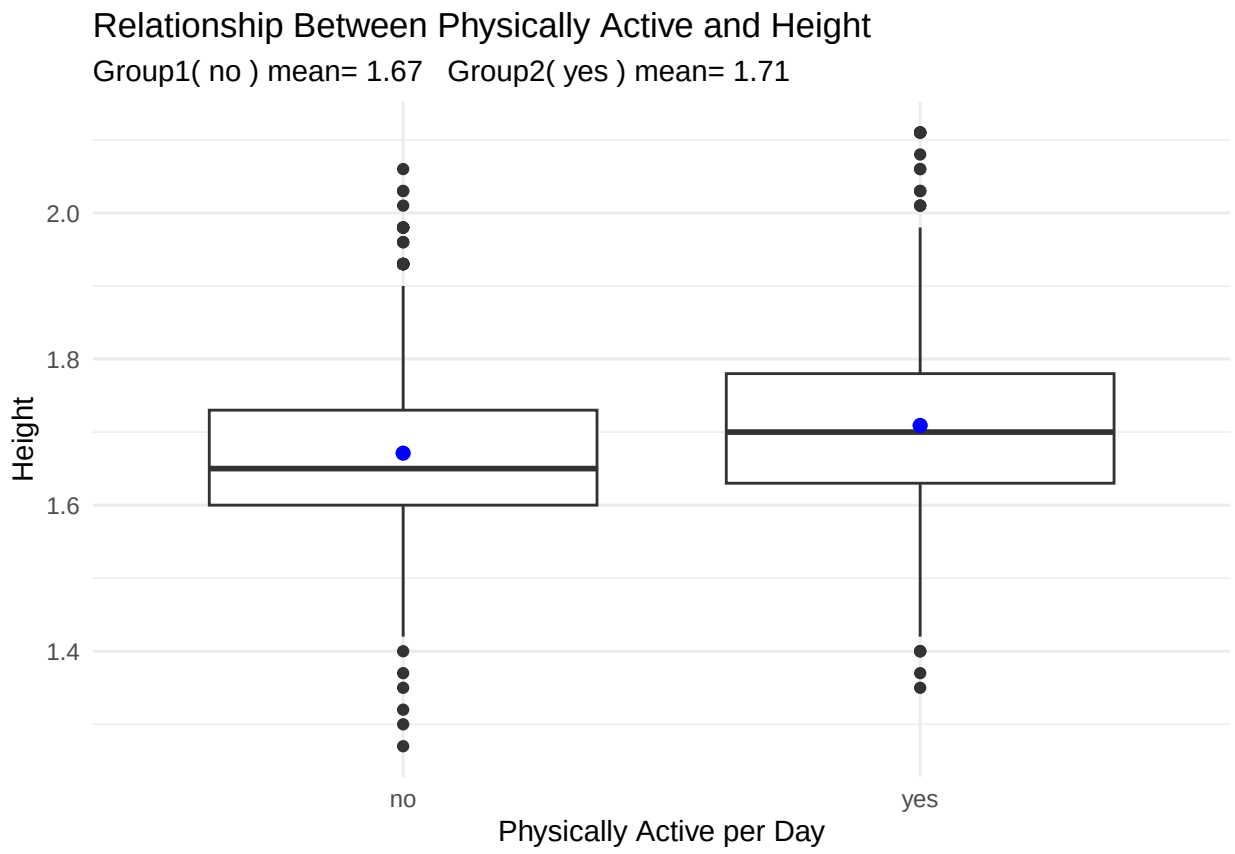
10. Conduct a hypothesis test evaluating whether the average height is different for those who exercise at least three times a week and those who don't.

Insert your answer here The box_plot can display the difference in the means Group1 got 0.03 and Group2 0.04. The p-value is 0.05 meaning we can reject the null hypothesis and accept the alternative hypothesis.

H0: The average height for those who exercise at least 3 days a week are the same as those who don't udiff=0

H1: The average height for those who exercise at least 3 days a week are different than those who don't udiff != 0

```
yrbss_c <- yrbss|>
  drop_na()

Summary_groups <- describeBy(yrbss_c$height,
                  group=yrbss_c$physical_3plus, #same value as X aes
                  mat=TRUE, skew=FALSE)

ggplot(yrbss_c, aes(x = factor(physical_3plus), y = height )) +
  geom_boxplot() +
  geom_point(data = Summary_groups, aes(x=group1, y=mean),
             color='blue', size=2)+
  labs(
    x = "Physically Active per Day",
    y = "Height",
    title = "Relationship Between Physically Active and Height",
    subtitle = paste("Group1(",Summary_groups[1,2],") mean=",round(Summary_groups[1,5],2)," ",
                     "Group2(",Summary_groups[2,2],") mean=",round(Summary_groups[2,5],2))
  )+
  theme_minimal()
```




```
#Summary_groups

#activity-noweight
nact_height_n <- Summary_groups[1,4]
nact_height_mean <- Summary_groups[1,5]
nact_height_sd <- Summary_groups[1,6]

#activity-weight
act_height_n <- Summary_groups[2,4]
act_height_mean <- Summary_groups[2,5]
act_height_sd <- Summary_groups[2,6]


#SE(Xa-Xna)
SE_Xa_Xn <-  sqrt((act_height_sd^2/act_height_n)+(nact_height_sd^2/nact_height_n))
SE_Xa_Xn
```

```
## [1] 0.002411465
```

```
mean_difference <- (act_height_mean - nact_height_mean)

#df
df <- min(act_height_n - 1,nact_height_n - 1)
t_value <- qt(0.05/2, df, lower.tail = FALSE) # lower.tail = FALSE get positive value
cl <- mean_difference - t_value * SE_Xa_Xn
```

```r
ch <- mean_difference + t_value * SE_Xa_Xn
paste("The confidence interval is ",round(cl,2)," to ", round(ch,2))
```

```
## [1] "The confidence interval is  0.03  to  0.04"
```

```r
Summary_groups
```

```
##     item group1 vars    n     mean        sd  min  max range          se
## X11    1     no    1 2656 1.671126 0.1022101 1.27 2.06  0.79 0.001983261
## X12    2    yes    1 5695 1.709147 0.1035233 1.35 2.11  0.76 0.001371802
```

```r
p_value <- 2*pt(t_value, df, lower.tail = FALSE)
p_value
```

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

```r
#H0 :act_wei_mean - nact_wei_mean =0
#HA :act_wei_mean - nact_wei_mean /=0

#tail_area <- (mean_difference - 0 )/SE_Xa_Xn
```

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.

Insert your answer here There are 7 different options

```r
table(yrbss$hours_tv_per_school_day)
```

```
##
##           <1            1            2            3            4           5+
##         2168         1750         2705         2139         1048         1595
## 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.

Insert your answer here Both means are pretty close ,the group of people that sleep 8 hours got a mean of 1.6896 with a standard deviation of 0.1045 and the group that sleep different time than 8 hours got a mean of 1.6947 with a standard deviation of 0.1049. We got p_value = 0.05 however the confidence of interval is from 0 to 0.01, which is close to 0 so we will fail to reject Null hypothesis.

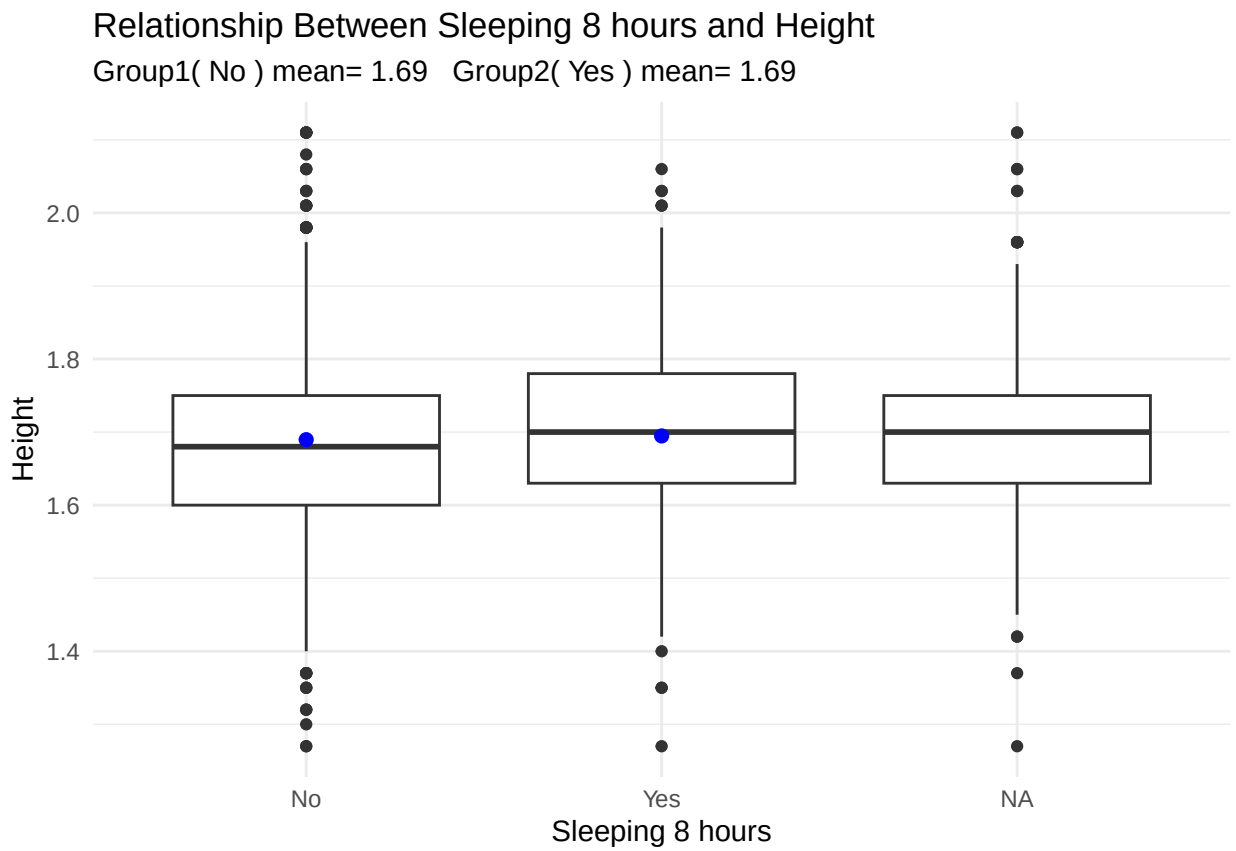
H0: There is no difference in the average height and people who sleep 8 hours
udiff=0

Ha: There is difference in the average height and people who sleep 8 hours udiff!=0

```r
library(psych)


table(yrbss$school_night_hours_sleep)
```

```
##
##   <5  10+    5    6    7    8    9
##  965  316 1480 2658 3461 2692  763
```

```r
yrbss <- yrbss %>%
  mutate(sleep_8hr = ifelse(yrbss$school_night_hours_sleep == 8, "Yes", "No"))


Summarysleep_8hr <-  describeBy(yrbss$height,
                  group=yrbss$sleep_8hr, #same value as X aes
                  mat=TRUE, skew=FALSE)


ggplot(yrbss, aes(x = sleep_8hr, y = height )) +
  geom_boxplot() +
  geom_point(data = Summarysleep_8hr, aes(x=group1, y=mean),
             color='blue', size=2)+
  labs(
    x = "Sleeping 8 hours",
    y = "Height",
    title = "Relationship Between Sleeping 8 hours and Height",
    subtitle = paste("Group1(",Summarysleep_8hr[1,2],") mean=",round(Summarysleep_8hr[1,5],2)," ",
                     "Group2(",Summarysleep_8hr[2,2],") mean=",round(Summarysleep_8hr[2,5],2))
  )+
  theme_minimal()
```




```
#activity-noweight
nact_sleep_8hr_n <- Summarysleep_8hr[1,4]
nact_sleep_8hr_mean <- Summarysleep_8hr[1,5]
nact_sleep_8hr_sd <- Summarysleep_8hr[1,6]

#activity-weight
act_sleep_8hr_n <- Summarysleep_8hr[2,4]
act_sleep_8hr_mean <- Summarysleep_8hr[2,5]
act_sleep_8hr_sd <- Summarysleep_8hr[2,6]


#SE(Xa-Xna)
SE_Xa_Xn <-  sqrt((act_sleep_8hr_sd^2/act_sleep_8hr_n)+(nact_sleep_8hr_sd^2/nact_sleep_8hr_n))
SE_Xa_Xn
```

```
## [1] 0.002367986
```

```
mean_difference <- (act_sleep_8hr_mean - nact_sleep_8hr_mean)

#df
df <- min(act_sleep_8hr_n - 1,nact_sleep_8hr_n - 1)
t_value <- qt(0.05/2, df, lower.tail = FALSE) # lower.tail = FALSE get positive value
cl <- mean_difference - t_value * SE_Xa_Xn
ch <- mean_difference + t_value * SE_Xa_Xn

paste("The confidence interval is ",round(cl,2)," to ", round(ch,2))
```

```
## [1] "The confidence interval is  0  to  0.01"
```

```
Summarysleep_8hr
```

```
##     item group1 vars    n     mean        sd  min  max range          se
## X11    1     No    1 8976 1.689609 0.1045306 1.27 2.11  0.84 0.001103322
## X12    2    Yes    1 2505 1.694707 0.1048668 1.27 2.06  0.79 0.002095242
```

```
p_value <- 2*pt(t_value, df, lower.tail = FALSE)
p_value
```

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

```
(mean_difference - 0 )/SE_Xa_Xn
```

```
## [1] 2.152728
```