# DATA606 Chap7 Lab7 Inference for numerical data

Alexis Mekueko

10/16/2020

Github link: https://github.com/asmozo24/DATA606_Lab7

Web Link
: file:///C:/Users/Petit%20Mandela/Documents/R/DATA606_Homework7/DATA606_Homework7/DATA606_Homework7.html

```
library(tidyverse) #loading all library needed for this assignment
library(openintro)
library(infer)
library(gplots)

library(httr)
library(rvest)
library(xml2)
#head(fastfood)
#library(readxl)
#library(data.table)
#library(readr)
#library(plyr)
#library(dplyr)
#library(dice)
# #library(VennDiagram)
# #library(help = "dice")
#library(DBI)
#library(dbplyr)

#library(rstudioapi)
#library(RJDBC)
#library(odbc)
#library(RSQLite)
#library(rvest)
#library(stringr)
#library(readtext)
#library(ggpubr)
#library(fitdistrplus)
#library(ggplot2)
#library(moments)
#library(qualityTools)
library(normalp)
#library(utils)
#library(MASS)
#library(qqplotr)
library(DATA606)
```

```
##
## Welcome to CUNY DATA606 Statistics and Probability for Data Analytics
## This package is designed to support this course. The text book used
## is OpenIntro Statistics, 3rd Edition. You can read this by typing
## vignette('os3') or visit www.OpenIntro.org.
##
## The getLabs() function will return a list of the labs available.
##
## The demo(package='DATA606') will list the demos that are available.
```

```
getLabs()
```

```
##  [1] "Lab1"  "Lab2"  "Lab3"  "Lab4"  "Lab5a" "Lab5b" "Lab6"  "Lab7"  "Lab8"
## [10] "Lab9"
```

```
library(knitr)
#library(StMoSim)
```

##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)
yrbss
```

```
## # A tibble: 13,583 x 13
##      age gender grade hispanic race  height weight helmet_12m text_while_driv~
##    <int> <chr>  <chr> <chr>    <chr>  <dbl>  <dbl> <chr>      <chr>
##  1    14 female 9     not      Blac~  NA      NA   never      0
##  2    14 female 9     not      Blac~  NA      NA   never      <NA>
##  3    15 female 9     hispanic Nati~   1.73   84.4 never      30
##  4    15 female 9     not      Blac~   1.6    55.8 never      0
##  5    15 female 9     not      Blac~   1.5    46.7 did not r~ did not drive
##  6    15 female 9     not      Blac~   1.57   67.1 did not r~ did not drive
##  7    15 female 9     not      Blac~   1.65  132.  did not r~ <NA>
##  8    14 male   9     not      Blac~   1.88   71.2 never      <NA>
##  9    15 male   9     not      Blac~   1.75   63.5 never      <NA>
## 10    15 male   10    not      Blac~   1.37   97.1 did not r~ <NA>
## # ... with 13,573 more rows, and 4 more variables: physically_active_7d <int>,
## #   hours_tv_per_school_day <chr>, strength_training_7d <int>,
## #   school_night_hours_sleep <chr>
```

```
?yrbss
```

```
## starting httpd help server ... done
```

```
#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:
```

# Exercice1

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

# There are observations on 13 different variables in this 13583 sample (after cleanup)…the cases are about students male and female from ages 14 to 19 on various activities.

```
view(yrbss)
glimpse(yrbss)
```

```
## Rows: 13,583
## Columns: 13
## $ age                     <int> 14, 14, 15, 15, 15, 15, 15, 14, 15, 15, 15...
## $ gender                  <chr> "female", "female", "female", "female", "f...
## $ grade                   <chr> "9", "9", "9", "9", "9", "9", "9", "9", "9...
## $ hispanic                <chr> "not", "not", "hispanic", "not", "not", "n...
## $ race                    <chr> "Black or African American", "Black or Afr...
## $ height                  <dbl> NA, NA, 1.73, 1.60, 1.50, 1.57, 1.65, 1.88...
## $ weight                  <dbl> NA, NA, 84.37, 55.79, 46.72, 67.13, 131.54...
## $ helmet_12m              <chr> "never", "never", "never", "never", "did n...
## $ text_while_driving_30d  <chr> "0", NA, "30", "0", "did not drive", "did ...
## $ physically_active_7d    <int> 4, 2, 7, 0, 2, 1, 4, 4, 5, 0, 0, 0, 4, 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, ...
## $ school_night_hours_sleep <chr> "8", "6", "<5", "6", "9", "8", "9", "6", "...
```

```
#is.na(airlines_arrival2) # checking if there is a missing data in the dataset, return is yes
#sum(is.na(airlines_arrival2)) # file to big, checking the sum of all missing data (return is 09 missing data)
yrbss1 <- na.omit(yrbss) # delete/remove the missings data because it is an imcomplete observation
```

## 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.

# Exercie 2

How many observations are we missing weights from? I already cleaned up all the misses

```
summary(yrbss$weight1)
```

```
## Warning: Unknown or uninitialised column: `weight1`.
```

```
## Length  Class   Mode
##      0   NULL   NULL
```

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

```
summary(yrbss$weight1)
```

```
## Warning: Unknown or uninitialised column: `weight1`.
```

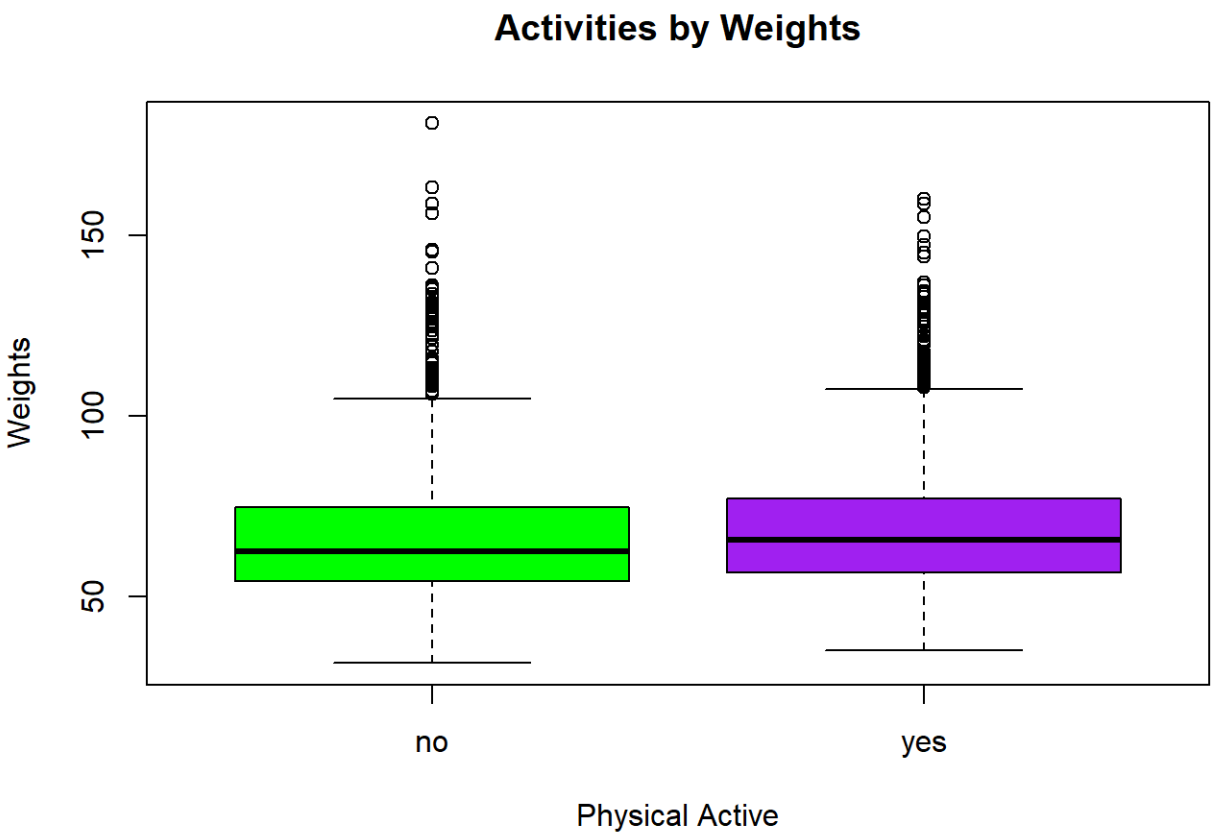
```
## Length  Class   Mode
##      0   NULL   NULL
```

# Exercice 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? # it is hard to say if there is a relationship between weight and physical activity. Thought, people who exercise more lose weight Vs. those who don't.

```
yrbss1 <- yrbss1 %>%
  mutate(physical_3plus = ifelse(yrbss1$physically_active_7d > 2, "yes", "no"))


boxplot(weight ~ physical_3plus, data = yrbss1, xlab = "Physical Active",
   ylab = "Weights", main = "Activities by Weights" , col = c("green","purple"))
```




# Exercice 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(). Answer: Yes, conditions for inference are satisfied. we assume independent sample since the the problem statement did not let's to think otherwise Large sample Normal distribution (male/female…age spread from 14 to 18 etc…) The mean_weight : "no" physical_3plus = 67.15; "yes" physical_3plus = 68.68

```
# The box plots show how the medians of the two distributions compare, but we can also compare the mea
ns of the distributions using the following to first group the data by the physical_3plus variable, an
d then calculate the mean weight in these groups using the mean function while ignoring missing values
by setting the na.rm argument to TRUE.
#
# There is an observed difference, but is this difference statistically significant? In order to answe
r this question we will conduct a hypothesis test.

yrbss1 %>%
  group_by(physical_3plus) %>%
  summarise(mean_weight = mean(weight, na.rm = TRUE))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 2
##   physical_3plus mean_weight
##   <chr>                <dbl>
## 1 no                    67.1
## 2 yes                   68.7
```

#Exercice 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. Answer: H_nul = there is no difference in the average weights are different for those who exercise at least 03 times a week and those who don't. H_alt = there is difference in the average weights are different for those who exercise at least 03 times a week and those who don't.

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 <- yrbss1 %>%
  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 <- yrbss1 %>%
  specify(weight ~ physical_3plus) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "diff in means", order = c("yes", "no"))
```

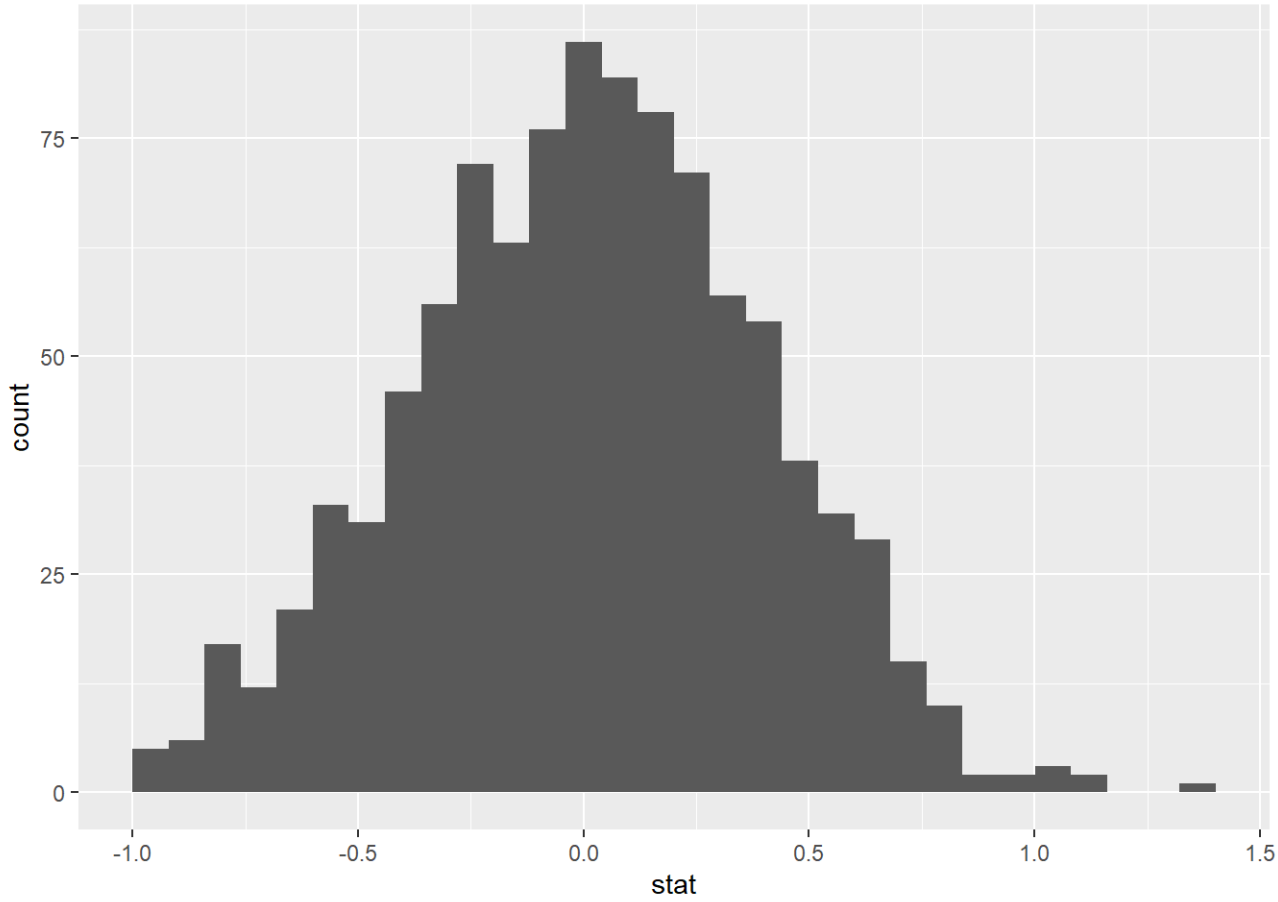
Here, hypothesize is used to set the null hypothesis as a test for independence. In one sample cases, the null argument can be set to "point" to test a hypothesis relative to a point estimate.

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()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




# Exercice 6

How many of these null permutations have a difference of at least obs_stat? Answer: 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")
```

```
## Warning: Please be cautious in reporting a p-value of 0. This result is an
## approximation based on the number of `reps` chosen in the `generate()` step. See
## `?get_p_value()` for more information.
```

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

```
## Warning: Please be cautious in reporting a p-value of 0. This result is an
## approximation based on the number of `reps` chosen in the `generate()` step. See
## `?get_p_value()` for more information.
```

#Exercice 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 a probability that difference between the weights of those who exercise at least three times a week and those who don't lies on (0.72, 2.33)

```
 yrbss2 <- filter( yrbss1, physical_3plus == "no" & weight != 0)
 physical_no <- yrbss2$weight

physical_yes <- (filter( yrbss1, physical_3plus == "yes" & weight != 0))$weight
glimpse (physical_no)
```

```
##  num [1:2656] 55.8 46.7 67.1 69.8 66.7 ...
```

```
n_yes <- nrow(physical_yes)
n_no <- nrow(physical_no)
df <- n_yes - 1
mean_no <- mean(physical_no)
mean_yes <- mean(physical_yes)
sd_no <- sd(physical_no)
sd_yes <- sd(physical_yes)
SE <- sqrt( (sd_yes^2)/n_yes + (sd_no^2)/n_no)
t_value <- qt(0.05/2, df, lower.tail = FALSE)
point_estimate <-  mean_yes - mean_no
lower_CI <- point_estimate - t_value * SE
upper_CI <- point_estimate + t_value * SE
lower_CI
```

```
## numeric(0)
```

```
upper_CI
```

```
## numeric(0)
```

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

#Exercice 8 Calculate a 95% confidence interval for the average height in meters (height) and interpret it in context. we are 95% confidence interval that the average height in meters (height) is (1.69, 1.7)

```
n_yes <- 8351
#n_no <- nrow(physical_no)
df <- n_yes - 1
mean_h <- mean(yrbss1$height)
#mean_yes <- mean(physical_yes)
sd_h <- sd(yrbss1$height)
#sd_yes <- sd(physical_yes)
SE <- sd_h/sqrt(n_yes)

t_value <- qt(0.05/2, df, lower.tail = FALSE)
point_estimate <-  mean_h
lower_CI <- point_estimate - t_value * SE
upper_CI <- point_estimate + t_value * SE
lower_CI
```

```
## [1] 1.69481
```

```
upper_CI
```

```
## [1] 1.699298
```

#Exercice 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. Answer: we are 90% confidence interval that the average height in meters (height) is (1.695, 1.699) The confidence interval becomes more narrowed.

```
t_value <- qt(0.1/2, df, lower.tail = FALSE)
point_estimate <-  mean_h
lower_CI <- point_estimate - t_value * SE
upper_CI <- point_estimate + t_value * SE
lower_CI
```

```
## [1] 1.695171
```

```
upper_CI
```

```
## [1] 1.698937
```

#Exercice 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. H_null: there is no difference in the average height for those who exercise at least three times a week and those who don't. H_alt: there is difference in the average height for those who exercise at least three times a week and those who don't.

Answer: p_value = 0.05 < alpha (0.1), thus reject the H_null. There is a difference in the average height for those who exercise at least three times a week and those who don't. strange finding

```
yrbss3 <- filter( yrbss1, physical_3plus == "no" & height != 0)
height_no <- yrbss3$height

height_yes <- (filter( yrbss1, physical_3plus == "yes" & weight != 0))$height
#glimpse (height_yes)

n_yes <- 5695
n_no <- 2656
df <- n_yes - 1
mean_no <- mean(height_no)
mean_yes <- mean(height_yes)
sd_no <- sd(height_no)
sd_yes <- sd(height_yes)
SE <- sqrt( (sd_yes^2)/n_yes + (sd_no^2)/n_no)
t_value <- qt(0.05/2, df, lower.tail = FALSE)
point_estimate <-  mean_yes - mean_no
lower_CI <- point_estimate - t_value * SE
upper_CI <- point_estimate + t_value * SE
lower_CI
```

```
## [1] 0.03329348
```

```
upper_CI
```

```
## [1] 0.04274826
```

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

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

# Exercice 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. Answer: There are 07 different options there are in the dataset for the hours_tv_per_school_day.

```
glimpse(yrbss1$hours_tv_per_school_day)
```

```
##  chr [1:8351] "5+" "2" "3" "5+" "do not watch" "<1" "4" "5+" "5+" "5+" "5+" ...
```

# Exercice 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.

Let’s find out the correlation between sleep and weight.

asumming a 95% CI , Conduct a hypothesis test evaluating whether the average weight is different for those who sleep at least 8hrs a day and those who don’t. H_null: there is no difference in the average weight for those who sleep at least 8hrs a day and those who don’t. H_alt: there is difference in the average weight for those who sleep at least 8hrs a day and those who don’t. p_value = 0.05 = alpha, which mean we are at the boundary which mean the research question evaluation need to be adjust or the data does not provide convincing about the relationship between weight and sleeping at least 8hrs.

```
yrbss1 <- yrbss1 %>%
  mutate(sleep_8hrs = ifelse(yrbss1$school_night_hours_sleep > 7, "yes", "no"))


yrbss4 <- filter( yrbss1, sleep_8hrs == "no" & weight != 0)
weight_no <- yrbss4$weight

weight_yes <- (filter( yrbss1, sleep_8hrs == "yes" & weight != 0))$weight
#glimpse (weight_yes)

n_yes <- 2295
n_no <- 6056
df <- n_yes - 1
mean_no <- mean(weight_no)
mean_yes <- mean(weight_yes)
sd_no <- sd(weight_no)
sd_yes <- sd(weight_yes)
SE <- sqrt( (sd_yes^2)/n_yes + (sd_no^2)/n_no)
t_value <- qt(0.05/2, df, lower.tail = FALSE)
point_estimate <-  mean_yes - mean_no
lower_CI <- point_estimate - t_value * SE
upper_CI <- point_estimate + t_value * SE
lower_CI
```

```
## [1] -2.374375
```

```
upper_CI
```

```
## [1] -0.7877884
```

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

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