APSTA-GE 2003 Intermediate Quantitative Methods

Summer Assignment (Assignment 0)

A3SR Welcome Packet Team

04 September 2020

Introduction

This document serves as a practical exercise of the 2003: Intermediate Quantitative Methods course. It helps students recap some key concepts in mathematics and statistics, including: **T-test** and **Linear Regression**. The practical exercise simulates an **Ordinary Least Squares (OLS) Linear Regression** model with pooled sample variance.

This document is carefully scaffolded (with lots of comments) to help students get familiar with R scripts and R Markdown.

Submission of this assignment is not required.

Part 1. Recap

One Sample T-test

In a one-sample t-test, we want to test whether the mean of a sample collected from a population is statistically different from the hypothesized population mean, μ_0. The hypothesis testing problem can be formulated as:

The null hypothesis:

$$H_0 : \mu = \mu_0$$

The alternative hypothesis:

$$H_a : \mu \neq \mu_0$$

Given the sample mean, $\bar{x}$, the hypothesized population mean, μ_0, and the standard error (SE) of the sample mean, $SE(\bar{x})$, we can construct a T-test:

$$T = \frac{\bar{x} - \mu_0}{SE(\bar{x})}$$

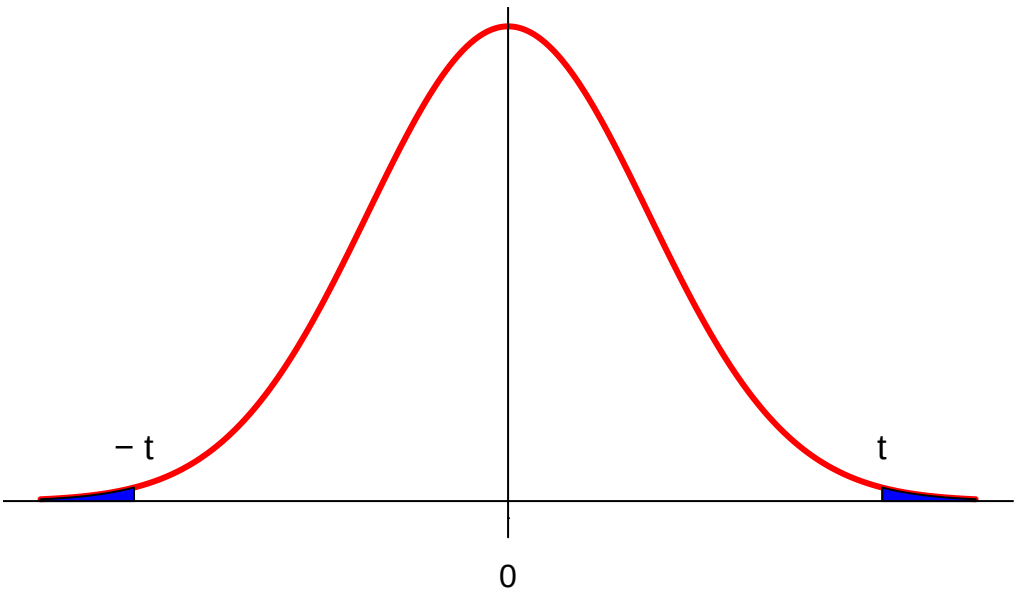
The null distribution of T is a t distribution with $n - 1$ degrees of freedom .

If we set a significance level of 0.05 (for a 2-tailed test), then:

If the absolute value of T, $abs(T)$, is greater than the 97.5th quantile of a T distribution (d.f. = n - 1), we reject the null hypothesis in favor of the alternative that the population mean is different from μ_0.

Note that for a sample size of n: $SE(\bar{x}) = \frac{s}{\sqrt{n}}$ where s is the sample standard deviation.

Here is a graph showing the p-value areas (in blue) under a normal distribution curve (in red). Note that p-value is the sum of areas of both sides.



Two (Independent) Sample T-test

In a two sample T-test problem, we want to test whether the mean of population 1 is the same as the mean of population 2. The hypothesis testing problem can be formulated as:

The null hypothesis:

$$H_0 : \mu_1 = \mu_2$$

The alternative hypothesis:

$$H_a : \mu_1 \neq \mu_2$$

We can construct a T-test to test H_0 versus H_a:

$$T = \frac{(\bar{x_1} - \bar{x_2})}{SE(\bar{x_1} - \bar{x_2})}$$

Here, the $\bar{x_1}$ and $\bar{x_2}$ are the means of population 1 and 2.

We can estimate the standard error of $\bar{x_1} - \bar{x_2}$ using the following formula:

$$SE(\bar{x_1} - \bar{x_2}) = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$$

In this formula, s_1 and s_2 represent the sample standard deviations for sample group 1 and 2, respectively.

If we believe that the two populations have **unequal** variances, then we can estimate s_1 and s_2 separately.

If we believe that the two populations have **equal** variances , then we can say $s_1 = s_2 = s_{pooled}$ where:

$$s_{pooled} = \sqrt{\frac{s_1^2(n_1 - 1) + s_2^2(n_2 - 1)}{n_1 + n_2 - 2}}$$

Then, the null distribution of T is a T distribution with degrees of freedom as follows:

In general, we use the smaller of $n_1 - 1$ and $n_2 - 1$ as the degrees of freedom for the T distribution.

If assuming equal variances and using s_{pooled}, we can use $n_1 + n_2 - 2$ degrees of freedom.

Linear Regression with a Dummy Variable Predictor

First, let's read in some data.

In order to read in the file, you will need to download the `height_sex.csv` data set to your computer. To import the data set to R, you can use two methods:

1. Include the absolute file path in your `read.csv()` function, as shown below:

```
height <- read.csv("/Users/temp/Downloads/height_sex.csv")
```

Interpretation: Read the CSV file at the "Downloads" folder under temp and assign the data set to a variable called `height`.

2. Set up your working directory" in R. Here is a reference article.

To check your current working directory, you can use the `getwd()` command:

```
getwd()
```

To set up your working directory, you can use the `setwd(dir = "")` command. Remember to include the absolute (or relative) path of the location where you would like to keep files. Then, you can put the data files in your working directory and directly call them.

```
# Load data under the root of your working directory and assign to `height`
height <- read.csv("height_sex.csv")
```

After importing the data, we can examine the first few rows by calling the `head()` function.

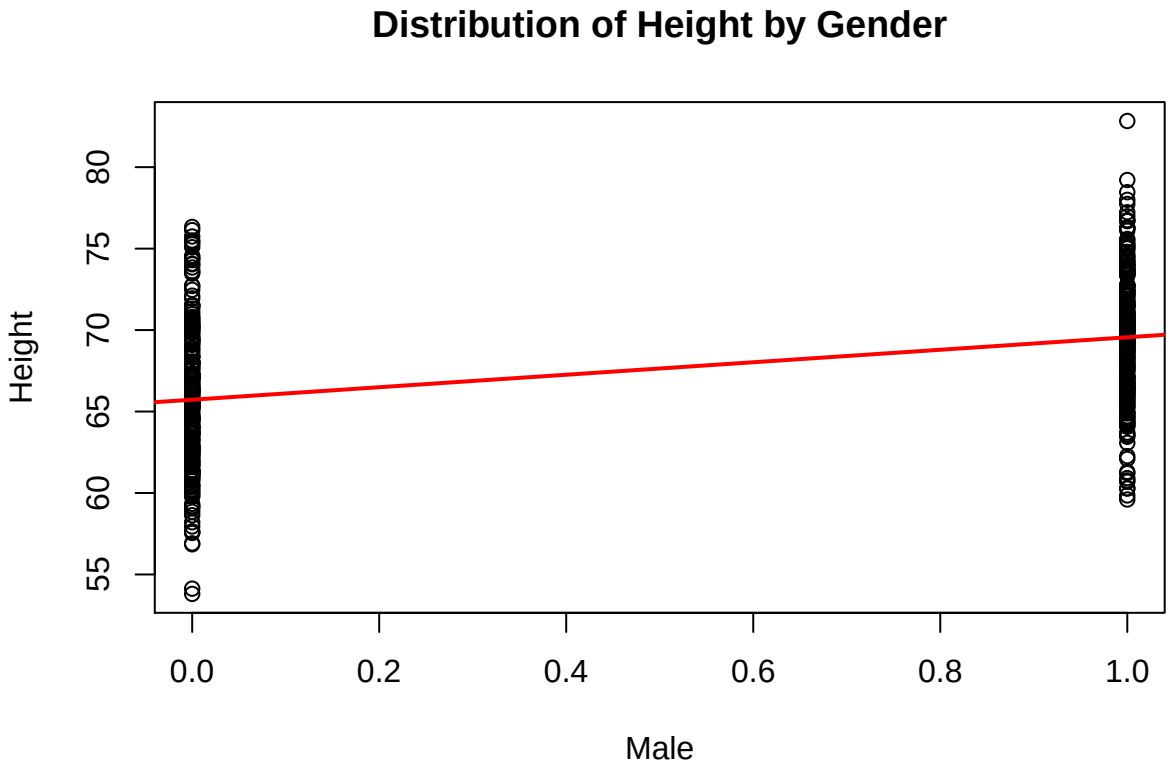
```
# Inspect the first few rows
head(height)
```

```
##   X id Gender Birth.Order   Height    Weight Male
## 1 1  1      M           1 65.44441 166.14611    1
## 2 2  2      F           2 65.44836  76.74407    0
## 3 3  3      M           2 68.39149 140.14253    1
## 4 4  4      M           1 73.92210 165.30409    1
## 5 5  5      F           1 70.98392 117.61932    0
## 6 6  6      F           1 74.07854 114.30186    0
```

For this assignment, we are going to focus on the `Height` and `Male` variables. `Height` is given in inches and `Male` is an indicator variable (equal to 1 if the person is male and equal to 0 if the person is female). If we plot height against the gender indicator variable, we will get:

```
# Plot height vs. gender
plot(
  x = height$Male,
  y = height$Height,
  xlab = "Male",                          # x-axis label
  ylab = "Height",                        # y-axis label
  main = "Distribution of Height by Gender" # Plot title
)
```

```r
# Add a reference line to reveal the association using a linear regression model
abline(
  reg = lm(height$Height~height$Male),      # reg: an object with a coef. method
  col = "red",                              # Change the line color to red
  lwd = 2                                   # Change the line width to 200%
)
```



Minimizing Squared Differences

- Suppose for a moment that you only have one set of heights, say, $\{60, 63, 63, 70, 72\}$, and you want to find some value, M, such that the sum of the squared differences between M and each value in your set of heights is minimized. It turns out that the value of M which satisfies this minimization problem is the mean of the data set (you can prove this with some relatively simple calculus). Note: minimizing squared differences is akin to minimizing absolute distances (the mean is the solution to both minimization problems); however, we generally choose to minimize squared differences instead of absolute differences to make the calculus easier. The proof is left to the interested reader, but hopefully this claim already feels somewhat intuitive.

Now, suppose we try to fit an Ordinary Least Squares (OLS) regression line to this data. This would involve finding values of β_0 (the intercept) and β_1 (the slope) such that the (squared) vertical distance between each point (in the above graph) and the line $Height = \beta_0 + \beta_1 * Male$ is minimized.

There are only two possible values of `Male` in this data set: 0 and 1. Therefore, one way to think about minimizing the vertical distances between each data point and the line is this:

Find the mean height of female (i.e., everyone with Male = 0), find the mean height of male (i.e., everyone with Male = 1), and connect the two points (0, mean female height) and (1, mean male height).

Now we can try to write the equation for this line in the form $Height = \beta_0 + \beta_1 * Male$. Using the strategy described above, we know that (0, mean female height) is the y-intercept of the line. Therefore β_0 = mean female height. We also know that β_1 is the slope of the line. Since we know two points on the line, we can calculate the slope as:

$$\frac{y_2 - y_1}{x_2 - x_1} = \frac{(\text{mean male height} - \text{mean female height})}{(1-0)} = \text{mean male height} - \text{mean female height}$$

Therefore,

$$\beta_1 = \text{mean male height} - \text{mean female height}$$

A question to think about: is testing $\beta_1 = 0$ the same as testing the mean male height and the mean female height to be the same (this is the two sample t-test problem)?

Part 2. Practical Exercise

1. The mean male height is calculated below and saved as `avg_m_height`. Fill in the rest of the code to:

a) save the mean female height as `avg_f_height` and

b) print the mean female height.

```
# Create a variable and store mean male height as `avg_m_height`.
avg_m_height <- mean(height$Height[height$Male == 1])

# Print mean male height.
avg_m_height
```

```
## [1] 69.55706
```

```
# Save mean female height as the variable`avg_f_height`.
### INSERT CODE HERE ###

# Print mean female height.
### INSERT CODE HERE ###
```

2. Calculate the mean difference of the two groups and plot the regression line:

a) Save and print: `diff_avg = avg_m_height - avg_f_height`

b) Add (0, avg_f_height) and (1, avg_m_height) to the plot from above (in yellow)

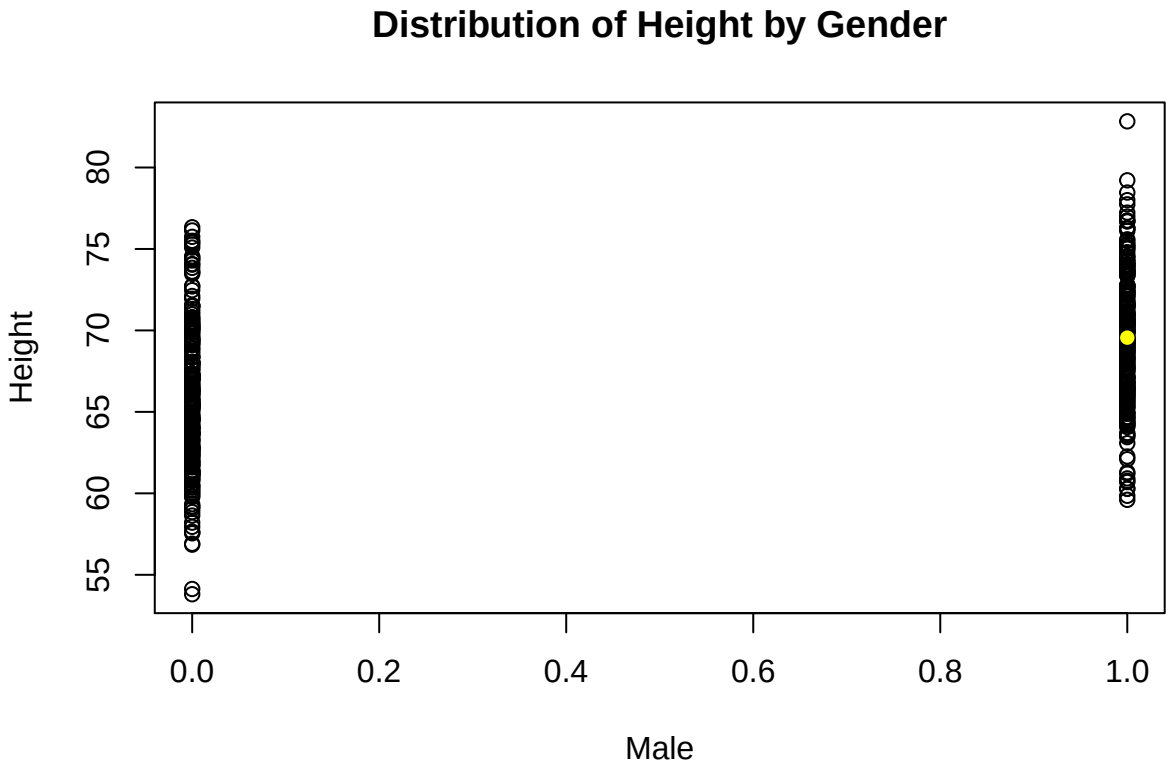
c) Draw a line connecting (0, avg_f_height) and (1, avg_m_height) using the `abline()` function.

```
# Save the difference in means as the variable `diff_avg`.
### INSERT CODE HERE ###

# Print `diff_avg`
### INSERT CODE HERE ###

# Re-plot the data
plot(
  x = height$Male,
  y = height$Height,
  xlab = "Male",
  ylab = "Height",
  main = "Distribution of Height by Gender"
)

# Add (0, avg_f_height) to the plot (in yellow).
# Note: (1, avg_m_height) has been added for you.
# Note: `pch = 16` makes the dots filled in, which makes them easier to see
points(x = 1, y = avg_m_height, col = "yellow", pch = 16)
```




```
### INSERT CODE HERE ###

# Draw the line: Height = b_0 + b_1*Male
# where b_0 = avg_f_height and b_1 = diff_avg.
# Use the abline() function which takes parameters a and b where
# a = y - intercept and
# b = slope
# Syntax is: abline(a, b)
### INSERT CODE HERE ###
```

3. Estimate the standard error for an independent sample T-test by hand (note: assume pooled standard deviation)

a) Calculate `n_m` and `n_f`, the number of males and females in the sample, respectively

b) Calculate `sd_m` and `sd_f`, the sample standard deviations for the males' heights and females' heights, respectively

c) Calculate the pooled standard deviation for the two groups using

$$s_{pooled} = \sqrt{\frac{s_1^2(n_1 - 1) + s_2^2(n_2 - 1)}{n_1 + n_2 - 2}}$$

and save the result as `sd_pooled`.

d) Use the following equation: $SE(\bar{x_1} - \bar{x_2}) = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$ to estimate $SE(\bar{x_1} - \bar{x_2})$ where $s_1 = s_2 = s_{pooled}$ and $n_1 =$ `n_f` $n_2 =$ `n_m`. Save the result as `se_avg_diff` and print the result.

```
# Add code to calculate the number of females and save the result as `n_f`.
n_m <- length(height$Height[height$Male==1])
### INSERT CODE HERE ###
```

```
# Add code to calculate `sd_f`, the sample standard deviation for females' heights
sd_m <- sd(height$Height[height$Male==1])
### INSERT CODE HERE ###

# Calculate pooled standard deviation and save as `sd_pooled`
# Note: sqrt() calculates square root
### INSERT CODE HERE ###

# Print `sd_pooled`
### INSERT CODE HERE ###

# Estimate the standard error of the difference in means and
# save it as `se_avg_diff`.
### INSERT CODE HERE ###

# Print `se_avg_diff`
### INSERT CODE HERE ###
```

4. Calculate a T-Statistic by hand to compare the null hypothesis that `avg_m - avg_f` $= 0$ versus the alternative hypothesis that `avg_m - avg_f` $\neq 0$.

a) Calculate $T = \frac{\text{(Mean Diff)}-0}{\text{SE Mean Diff}}$ and save the result as `t_manual`.

b) Use the `pt` function to calculate a p-value for t-test (pooled degrees of freedom = n_f + n_m - 2). *Hint:* You will want to use the parameter `lower.tail = FALSE` because t score should be > 0 for this example. Then, multiply by 2 to get a 2-sided test.

```
# Calculate a test statistic and save it as `t_manual`.
### INSERT CODE HERE ###

# Print `t_manual`
### INSERT CODE HERE ###

# Use `pt` to calculate a p-value.
### INSERT CODE HERE ###
```

5. Do the same thing using the `t.test()` function in R. Compare the value of `t` and p-value computed by the t.test() function to the values you calculated above. Note that setting `var.equal = TRUE` causes `t.test()` to use pooled variance. Do you get the same value (to two decimal places) for the T-Statistic?

```
# Use the `t.test()` function to compare the males' heights and females' heights.
# Note: the code has already been written for you.
# Just change the if argument from FALSE to TRUE to run the code.

if (FALSE) {  # change to `TRUE` to run the code

  t.test(
    x = height$Height[height$Male == 1],  # Set of male heights
    y = height$Height[height$Male == 0],  # Set of female heights
    var.equal = TRUE
  )

}
```

6. Use the `lm()` function to implement ordinary least squares regression to estimate β_0 and β_1 in the equation $Height = \beta_0 + \beta_1 * Male$.

```r
# Use `lm()` to estimate b_0 and b_1 in the equation.
# Save the result as `lin_mod`.
# Note: this has been done for you.
lin_mod <- lm(formula = Height ~ Male, data = height)

# Inspect a summary of the results using the `summary()` function on `lin_mod`.
### INSERT CODE HERE ###
```

7. Reprint the following values that you saved in previous problems:

- mean females height (`avg_f_height`)
- difference in mean height for males and females (`diff_avg`)
- standard error of this mean difference (`se_avg_diff`)
- the T-Statistic for the difference in means (`t_manual`).

Which of these values can you find in the `lm()` output above?

```r
# Re-print `avg_f_height`
### INSERT CODE HERE ###

# Re-print `diff_avg`
### INSERT CODE HERE ###

# Re-print `se_avg_diff`
### INSERT CODE HERE ###

# Re-print `t_manual`
### INSERT CODE HERE ###
```

8. Note that the t-test for the intercept in the `lm()` output above tests the null hypothesis: $\beta_0 = 0$ against the alternative hypothesis: $\beta_0 \neq 0$. In the space below, please show how you could calculate the standard error (.2576) and the t value (255.14) of the intercept without using `lm()` or `t.test()`.

Hint: `lm()` uses s_{pooled} to estimate the standard deviation of heights for both groups.

```r
# Estimate the standard error of b_0
# Hint: remember that b_0 = avg_f_height and you can estimate the SE of the mean
# using the central limit theorem.
### INSERT CODE HERE ###

# Use the standard error of b_0 to calculate the t value
### INSERT CODE HERE ###
```