IMT 573: Problem Set 6 - Inference and Monte Carlo

Vighnesh Misal

Due: Tuesday, November 12, 2019

Collaborators: Ashish Anand

Instructions:

Before beginning this assignment, please ensure you have access to R and RStudio; this can be on your own personal computer or on the IMT 573 R Studio Server.

1. Download the `problemset4.rmd` file from Canvas or save a copy to your local directory on RStudio Server. Open `problemset4.rmd` in RStudio and supply your solutions to the assignment by editing `problemset4.rmd`.

2. Replace the "Insert Your Name Here" text in the `author:` field with your own full name. Any collaborators must be listed on the top of your assignment.

3. Be sure to include well-documented (e.g. commented) code chucks, figures, and clearly written text chunk explanations as necessary. Any figures should be clearly labeled and appropriately referenced within the text. Be sure that each visualization adds value to your written explanation; avoid redundancy – you do no need four different visualizations of the same pattern.

4. Collaboration on problem sets is fun and useful, and we encourage it, but each student must turn in an individual write-up in their own words as well as code/work that is their own. Regardless of whether you work with others, what you turn in must be your own work; this includes code and interpretation of results. The names of all collaborators must be listed on each assignment. Do not copy-and-paste from other students' responses or code.

5. All materials and resources that you use (with the exception of lecture slides) must be appropriately referenced within your assignment.

6. Remember partial credit will be awarded for each question for which a serious attempt at finding an answer has been shown. Students are encouraged to attempt each question and to document their reasoning process even if they cannot find the correct answer. If you would like to include R code to show this process, but it does not run withouth errors you can do so with the `eval=FALSE` option.

7. When you have completed the assignment and have **checked** that your code both runs in the Console and knits correctly when you click `Knit PDF`, rename the knitted PDF file to `ps4_ourLastName_YourFirstName.pdf`, and submit the PDF file on Canvas.

Setup

Load any R packages of interest here.

```
library(dplyr)
library(ggplot2)
library(tidyverse)
library(tidyr)
```

NOTE: You do not need to perform all calculations in R. Writing them in LaTeX and/or plain text is completely fine. However, be sure your work is readable and understandable. If you do solve problems programmatically, clearly describe your approach and what you are doing.

Problem 1: Fathers and Sons

We will examine the heights of fathers and sons (from the previous problem set). If we look at sample means, we see that sons are taller than their fathers. But could this difference be due to chance?

(a) Load the data and examine it. are fathers' heights and are sons' heights. How many observations are there? Are there missing values?

```
fsdataset <- read.csv("C:/Users/iGuest/Downloads/fatherson.csv", sep = "")

nrow(fsdataset)

## [1] 1078

unique(is.na(fsdataset$fheight))

## [1] FALSE

unique(is.na(fsdataset$sheight))

## [1] FALSE
```

There are no missing values

There are 1078 observations

(b) What is an appropriate measurement type/scale for these variables? Are the values discrete or continuous?

Ratio scaled

The variables are continuous

(c) Describe the fathers' and sons' heights. What do the descriptive statistics look like? Are there any unexpected values? In general, who tends to be taller: fathers or sons?

```
summary(fsdataset)

##     fheight          sheight     
##  Min.   :149.9   Min.   :148.6  
##  1st Qu.:167.1   1st Qu.:170.0  
##  Median :172.1   Median :174.3  
##  Mean   :171.9   Mean   :174.5  
##  3rd Qu.:176.8   3rd Qu.:179.0  
##  Max.   :191.6   Max.   :199.0

var <- tbl_df(fsdataset$sheight - fsdataset$fheight)

son_taller_than_father <- var %>% filter(var$value >= 12)
father_taller_than_son <- var %>% filter(var$value <= -12)

nrow(son_taller_than_father)

## [1] 102

nrow(father_taller_than_son)

## [1] 29
```

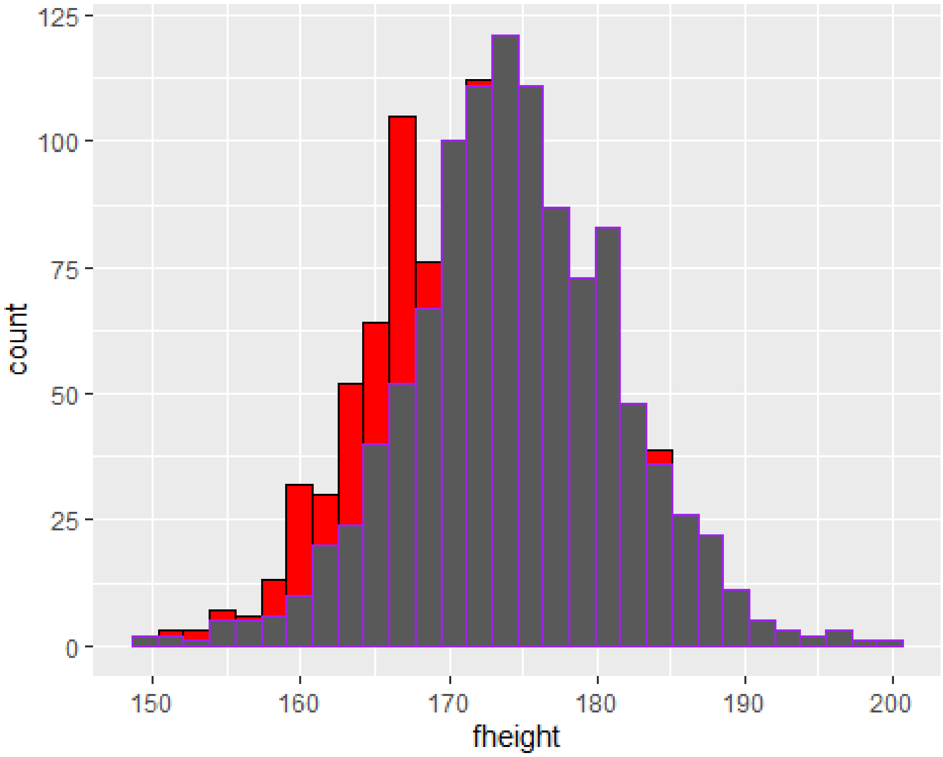
Based on summary data, sons' mean and median heights are greater than the fathers'

On further inspection, there are 102 instances in which the son's are 12 cm (1 foot) or more, taller than their father and 29 instances in which the father's are 12 cm (1 foot) or more taller than their son's

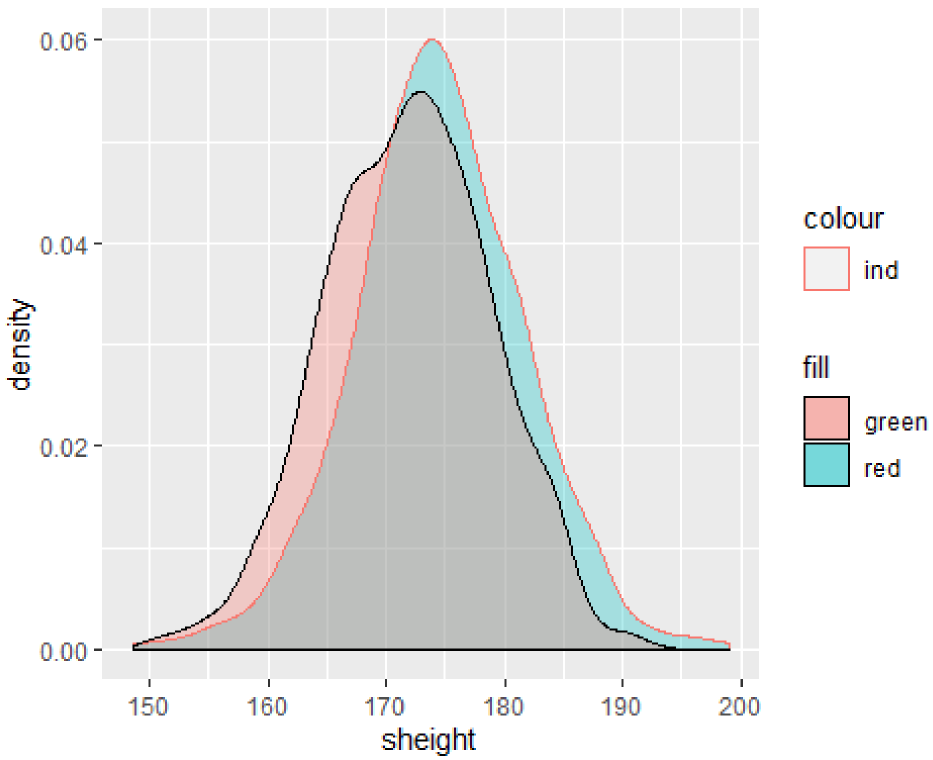
(d) Create a density plot with both sets of heights overlayed on the same figure. How do these plots look? What do they suggest in terms of fathers' and sons' relative heights?

```
ggplot(data = fsdataset) + geom_histogram(mapping = aes(x = fheight), fill =
"red", color = "black") + geom_histogram(mapping = aes(x = sheight), color =
"purple")
```

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



```
ggplot(data = fsdataset) + geom_density(aes(x = sheight,group='ind',
colour='ind',fill='red'),alpha=0.3)+ geom_density(aes(x =
fheight,group='ind',fill='green'),alpha=0.3)
```



A lot of father’s tend to have height less than 170 cm as observed from the following plots.

(e) Let’s do a t-test to determine if the differences we observe are statistically significant. Compute the t-statistic yourself (i.e. do NOT use any pre-existing functions that perform the test). We want to perform what is called a two-sample t-test (we are not going to assume the fathers and sons are paired in some way) and we want to test whether there is a difference in the means.

```
mean_fheight <- mean(fsdataset$fheight)
mean_sheight <- mean(fsdataset$sheight)

sd_fheight <- sd(fsdataset$fheight)
sd_sheight <- sd(fsdataset$sheigh)

sd_fheight <- sd_fheight*sd_fheight
sd_sheight <- sd_sheight*sd_sheight

denom <- (sd_fheight/1078) + (sd_sheight/1078)
t_val <- (mean_fheight - mean_sheight)/sqrt(denom)
t_val

## [1] -8.32387
```

```
mean_fheight <- mean(fsdataset$fheight)
mean_sheight <- mean(fsdataset$sheight)

sd_fheight <- sd(fsdataset$fheight)
sd_sheight <- sd(fsdataset$sheigh)

sd_pooled <- sqrt(((1078-1)*(sd_fheight)^2 + (1078-
1)*(sd_sheight)^2)/(1078+1078-2))

t_val_pooled <- (mean_fheight - mean_sheight)/(sd_pooled*sqrt(1/1078 +
1/1078))

t_val_pooled

## [1] -8.32387
```

(f) Did you use pooled or unpooled standard errors in your calculations? Why or why not? (Hint: see OpenIntro Stats 7.3.4)

I performed the t-test using both pooled and unpooled standard variations. The t_val obtained from both the methods was the same (-8.32387). Ideally the pooled t-test should be used as the means are quite close to one another.

(g) Using a t-table, what is the likelihood that the t-statistic you calculated occurs just by random chance? (Hint: be sure you have the appropriate degrees of freedom)

We obtain a value of 1.960 from the t table

```
t.test(fsdataset$fheight,fsdataset$sheight)

##
##  Welch Two Sample t-test
##
## data:  fsdataset$fheight and fsdataset$sheight
## t = -8.3239, df = 2152.6, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.128532 -1.935475
## sample estimates:
## mean of x mean of y
##  171.9252  174.4572
```

We observe that if the sample data were to equal the null hypothesis, the t-test will produce a value of 0. But, since the absolute t-value that we observed was 8.3, it means the data is quite dissimilar from the null-hypothesis as evidenced by the increase in t-value. Hence, its highly unlikely that it occurs by random chance.

(h) What do you find when performing the t-test? Are the differences statistically significant? Interpret your results.

#Since the absolute t-value is less than t-critical we can reject the null-hypothesis. When the null-hypothesis is rejected we can say that the differences are statistically significant.

Problem 2: A Monte Carlo Approach

Now, let’s examine the same data but using a what’s called a Monte Carlo approach. In essence, we’re going to leverage repeated (re-)sampling of our data (something we’ll discuss more in a few weeks when talking about bootstrapping).

(a) What is the overall mean and standard deviation for all heights? (i.e. when examining fathers’ and sons’ heights together)

```
fheight <- fsdataset$fheight
sheight <- fsdataset$sheight
combined_df <- c(fheight,sheight)
combined_df <- data.frame(combined_df)

combined_df <- as.numeric(unlist(combined_df))

mean(combined_df)

## [1] 173.1912

sd(combined_df)

## [1] 7.173111
```

(b) Create two samples of data pulled from random normals. For both of these distributions, let the size of the sample equal that of the fathers’ (or sons’) heights. Let the mean and standard deviation be those that you calculated in 2-a. Note that you want two samples pulled from the same distribution - one of these we’ll call “fathers” and the other we’ll call “sons.” What scenario are we simulating here with respect to the differences in fathers and sons heights? (Hint: think about a null hypothesis)

```
sample_fathers <- rnorm(1078, mean = mean(combined_df), sd = sd(combined_df))
sample_sons <- rnorm(1078, mean = mean(combined_df), sd = sd(combined_df))
```

We are simulating a scenario in which we try to bring the difference in means between the two sample closer to 0.

(c) What is the difference in means between the fathers' and sons' heights based using the simulated data? How does this compare to the difference in means for the dataset we read in?

```
mean(sample_fathers)-mean(sample_sons)

## [1] -0.1248088

mean(fsdataset$fheight) - mean(fsdataset$sheight)

## [1] -2.532004
```

The difference of the mean of two samples is much lower than the difference in mean for the unsimulated data.

(d) Now, repeat problem 2-b a large number of times (S; with S > 1000). At each iteration, store the difference in means of the fathers' and sons' heights so you ultimately end up with S different values for the difference in means.

```
result_df <- data.frame(mean_diff = as.numeric())



i =0
for (i in 1:3000){
sample_fathers <- rnorm(1078, mean = mean(combined_df), sd = sd(combined_df))
sample_sons <- rnorm(1078, mean = mean(combined_df), sd = sd(combined_df))

result_df[i,] <- mean(sample_fathers) - mean(sample_sons)

}
```

(d) What is the mean of the differences? Explain why you see the result that you do.

```
mean(result_df$mean_diff)

## [1] 0.001611539
```

As we perform multiple iterations we observe that the mean of differences decreases. The sample follows the central limit theorem.

(e) What is the standard error of the differences? How do these compare to the values we saw with the non-simulated data when computing the t-statistic?

```
standard_error <- sd(result_df$mean_diff)/sqrt(nrow(result_df))
standard_error

## [1] 0.005796953
```

The standard error is much lower in this case when compared to the standard error of unsimulated data.

(f) What is the largest difference we encounter (in terms of absolute value)? How does this compare to the difference in means that we saw with the non-simulated data?

```
max(abs(result_df$mean_diff))

## [1] 1.209023

abs(mean(fsdataset$fheight)-mean(fsdataset$sheight))

## [1] 2.532004
```

The max difference in means is much lower for the simulated data than unsimulated data.

(g) What is the 5th and 95th percentile of differences?

```
quantile(result_df$mean_diff, c(.05, .95))

##         5%        95%
## -0.5162904  0.5302514
```

(h) Now, increase S to increasingly large numbers and note the maximum difference in means that you see for each S. Do you see a maximum difference that is comparable to the actual difference in means that we encountered with the non-simulated data? If so, how often? Is this expected?

```
result_df_new <- data.frame(mean_diff_new = as.numeric())



i =0
for (i in 1:50000){
sample_fathers_new <- rnorm(1078, mean = mean(combined_df), sd =
sd(combined_df))
sample_sons_new <- rnorm(1078, mean = mean(combined_df), sd =
sd(combined_df))

result_df_new[i,] <- mean(sample_fathers) - mean(sample_sons)

}


mean(result_df_new$mean_diff_new)

## [1] 0.2882892

sd(result_df_new$mean_diff_new)/sqrt(nrow(result_df_new))

## [1] 0
```

```
max(abs(result_df_new$mean_diff_new))

## [1] 0.2882892

quantile(result_df_new$mean_diff_new, c(.05, .95))

##        5%       95% 
## 0.2882892 0.2882892
```

As we increase the number of iterations, the standard deviation between the difference of means decreases and tends to 0 which means that the sample mean approaches the population mean as we perform more iterations. This means that it follows the central limit theorem.