IMT 573: Problem Set 6

Statistical Learning

Jenny Skytta

Due: May 8, 2022

Collaborators: Jennifer Thao, Shuyin Liu, Jenny Ha

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 Cloud.

1. Download the `06_ps_statlearn.Rmd` file from Canvas or save a copy to your local directory on RStudio Cloud. Supply your solutions to the assignment by editing `06_ps_statlearn.Rmd`.
2. Replace the "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 *strongly* 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 without errors you can do so with the `eval=FALSE` option as follows:

```
a + b # these object don't exist
# if you run this on its own it with give an error
```

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`, download and rename the knitted PDF file to `ps6_YourLastName_YourFirstName.pdf`, and submit the PDF file on Canvas.

Setup: In this problem set you will need, at minimum, the following R packages.

```
# Load standard libraries
install.packages("reshape")
library(tidyverse)
```

```
library(reshape)
library(gridExtra)
library(knitr) # this will keep code on the page!
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
```

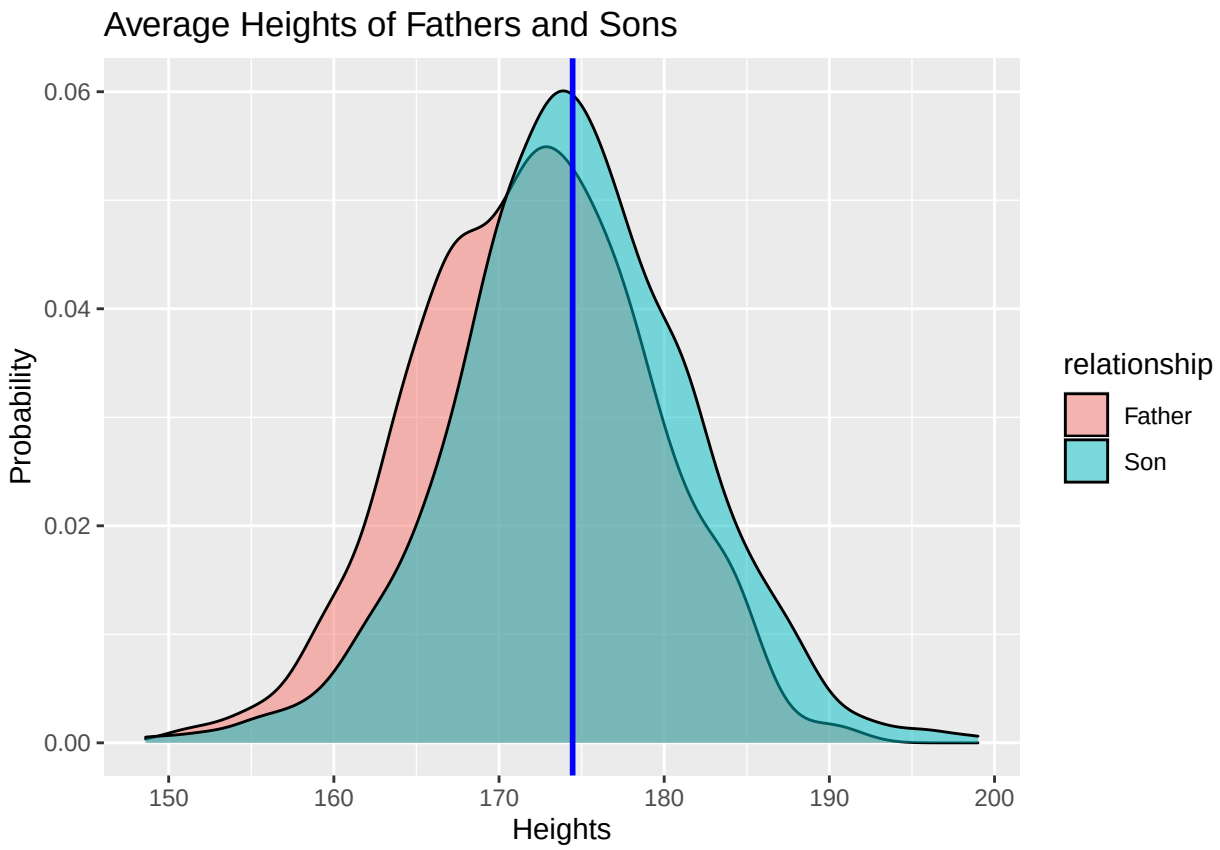
Problem 1: Are sons taller than fathers?

Here we analyze the dataset of fathers' and sons' height, used by Pearson and which we saw in the last problem set. It contains two variables, fathers' height and sons' height. If you take a simple mean, you see that in average sons are taller than fathers. But can this difference just be due to chance? Let's find out.

(a) To begin load the `fatherson.csv.bz2` data. Create density plots of both heights on the same figure. Comment the plots. What do they look like? What do they suggest in terms of fathers' and sons' relative height? The simple means of heights of father (~171.9) compared to that of the means of sons (~174.4) suggests that on average, sons are 3 inches taller than their fathers. In viewing the density plot, if I'm reading correctly, it appears there is only a 6% probability that sons are taller than their fathers. That seems considerably low overall. There's some things that aren't entirely clear from the dataset overall. Are these heights paired between father's and son's directly? Assuming this, I took the difference between the two heights with sons subtracting father's height (assumption of taller sons) and added a column with these values. I then added some boolean logic to sum the frequency that son's are taller than fathers. With a sample size of 1078 pairs of fathers and sons, this isn't generalizable data overall, as we haven't factored in how the sample was generated. Within this sample size, however, the son's do appear to be taller than their fathers at a ratio of 1.73.

```
father_son <- as_tibble(read.csv("data/fatherson.csv.bz2", sep = ""))

#reshape the data so that relationships and heights are parsed out
 father_son_new <-  gather(father_son, key="relationship", value="heights", 1:2) %>%
   mutate("dev_mean" = (heights)-(mean(heights))) %>%
    mutate("sqr_dev" = dev_mean^2)

height_diff <- father_son %>% # creating a stats tibble
  mutate("difference" = sheight-fheight) %>%  #difference in heights
  mutate("son_taller" = if_else(difference > 0, true = "Son_Taller", false = "Dad_Taller")) %>%
  mutate("sqr_diff" = (difference)^2) %>%  #squared difference
  mutate("dev_mean" = (sheight+fheight)-(mean(sheight+fheight))) %>% #mean of the difference
  mutate("sqr_dev" = dev_mean^2) #squared deviation

avg_dad_ht <- mean(height_diff$fheight) # mean of father ~ 171.9
avg_son_ht <- mean(height_diff$sheight) # mean of son ~ 174.4

combo_mean <-mean(father_son_new$heights) # mean of sons/fathers heights
combo_SD <- sd(father_son_new$heights) #stardard deviation of combined heights

DF <- (length(height_diff$fheight) - 1) # degrees of freedom 1077
N = length(height_diff$fheight) # sample pop 1078

 #create stacked density plot
height_plot<- ggplot(father_son_new, aes(heights, fill = relationship)) +
  geom_density(alpha = 0.5) +
  geom_vline(xintercept=avg_son_ht, size=1, color="blue") +
  scale_fill_discrete(labels = c("Father", "Son")) +
```

```
   labs(
    x = "Heights",
    y = "Probability",
    title = "Average Heights of Fathers and Sons"
  )

ratio_son_taller = 684/394   # ratio of son to father average heights
variance <- (sum(height_diff$sqr_dev))/DF #calculating variance
SD <- sqrt(variance) #standard deviation

show(height_plot)
```




```
# we'll need cum. prob threshold
# which tail we are comparing ( 1 or 2)
# degrees of freedom
# sample size n = 1078
# average squared difference of means


var_stp_1 <- sum(height_diff$difference) #sum of difference 2729.5
var_stp_2<- sum(height_diff$sqr_diff) # sum of sqrd difference  7450170.25
var_stp_3 <- var_stp_2^2
```

```
numerator <- var_stp_1/N # creating formula by scratch
denominator <- var_stp_2 - (var_stp_2/N) # building denominator
denominator2 <- sqrt(denominator/ (DF * N)) #denominator
mean_diff <- abs(mean(height_diff$sheight)- mean(height_diff$fheight))
T <- numerator/denominator2 # final T formula to calculate t score
RT <- t.test(father_son$fheight, father_son$sheight, paired = TRUE)
# using R to check the work I completed
```

(b) But is this difference statistically significant? Let's do a t-test. Here I ask you to *compute yourself the t-value*, do not use any pre-existing functions! What do you find? Why did you use/did you not use pooled standard deviations? Explain! Hint: read OIS 7.3

(c) Look up the t-distribution table. (Or compute the relevant quantiles). What is the likelihood that such a t value happens just by random chance? Hint: be sure to consider the *degrees of freedom* in current case carefully! The null hypothesis is that the average heights will be 175 and that there is no significant difference in height between sons and fathers and the alternate hypothesis is that sons are taller than fathers. The alpha level is .05 for a normal distribution, and my degrees of freedom are DF. I spent an inordinate amount of time working on this problem; likely far TOO much time. I manually attempted to calculate the t test which was agonizing because I didn't really trust what I was calculating. To check my work, I then ran the test using the built in R function and the results appear similar. I think overall though I'm running this incorrectly because the numbers seem 'off'.

(d) Based on your above analysis, state clearly your conclusion to the question - are sons taller than fathers? The results suggest there is statistical significance, I believe but I just don't trust these results nor do I know if I'm interpreting this correctly. The p value is less than 2.2 which provides 95% confidence in the results that son's are taller than fathers.

Problem 2: Fathers and Sons - the Monte Carlo approach

Next, let's re-visit the fathers and sons height, but this time by doing Monte Carlo analysis on a computer. You will proceed as follows: create two samples of random normals, similar to the data above, using the mean and standard deviation over both fathers and sons. Call one of these samples `fathers' and the other`sons' '. What is the difference in their means? And now you repeat this exercise many times and see if you can get as big a difference as what you saw above in the data.

```
rdad_mean <-mean(height_diff$fheight) # dads means of height differences
rson_mean <-mean(height_diff$sheight) # sons means of height differences
rdad_SD <- sd(height_diff$fheight) # dad's SD of heights
rson_SD <- sd(height_diff$sheight) # son's SD of heights

#creating a random set of data using the SD and mean from previous data for dads
fathers <- as.tibble(rnorm(1000, mean = rdad_mean, sd = rdad_SD)) %>%
  mutate("fheight" = value) %>%
  select(fheight)
```

(a) First, compute the overall mean and standard deviation of combined fathers' and sons' heights. Now create two sets of normal random variables, both with the same mean and standard deviation that you just computed above. Call one of these `fathers` and the other `sons`. What is the father-son mean difference? Compare the result with that you found in the previous problem.

```
## Warning: `as.tibble()` was deprecated in tibble 2.0.0.
```

```
## Please use `as_tibble()` instead.
## The signature and semantics have changed, see `?as_tibble`.
```

```
#creating a random set of data using the SD and mean from previous data for sons
sons <- as.tibble(rnorm(1000, mean = rson_mean, sd = rson_SD)) %>%
  mutate("sheight" = value) %>%
  select(sheight)
#combining sets and recalculating same metrics as original set
father_son_join <- cbind(fathers, sons) %>%
  mutate("difference" = sheight-fheight) %>%
  mutate("son_taller" = if_else(difference > 0, true = "Son_Taller", false = "Dad_Taller")) %>%
  mutate("sqr_diff" = (difference)^2) %>%
  mutate("dev_mean" = (sheight+fheight)-(mean(sheight+fheight))) %>%
  mutate("sqr_dev" = dev_mean^2)

combo_var_stp_1 <- sum(father_son_join$difference) #sum of difference ~ -2462
combo_var_stp_2<- sum(father_son_join$sqr_diff) # sum of sqrd difference ~106213
combo_var_stp_3 <- combo_var_stp_2^2 # ~11281263529
combo_var <- combo_var_stp_2/(length(father_son_join$fheight-1))
combo_mean_diff <- abs(mean(father_son_join$sheight)- mean(father_son_join$fheight))
```

The mean difference in this exercise (2a) is ~2.46 while the result in the previous exercise was ~2.53 so they appear to be somewhat aligned overall. I still feel like my attempts at manually calculating the data rather than using formulas has yielded a lot of distrust overall in what I am doing.

```
#creating a random set of data 1000 x using the SD and mean from previous data for dads
sims <- replicate(n=1000, rnorm(1000, mean = rdad_mean, sd = rdad_SD))
#creating a random set of data 1000 x using the SD and mean from previous data for dads
sims2 <- replicate(n=1000, rnorm(1000, mean = rson_mean, sd = rson_SD))

# wrangling data to same format - melting into long tibble
mean_of_diff <- as.tibble(melt(sims, id = sweights))
mean_of_diff2 <- as.tibble(melt(sims2, id = fweights))

#adding column with same labels as other datasets
mean_of_diff <- mean_of_diff %>%
  mutate("sheight" = value) %>%
  select(sheight) # keeping single column to combine
#adding column with same labels as other datasets
mean_of_diff2 <- mean_of_diff2 %>%
  mutate("fheight" = value) %>%
  select(fheight) # keeping single column to combine


mean_of_diff_final <- cbind(mean_of_diff, mean_of_diff2) %>%
  mutate("difference" = sheight-fheight) %>%
  mutate("son_taller" = if_else(difference > 0, true = "Son_Taller", false = "Dad_Taller")) %>%
  mutate("sqr_diff" = (difference)^2) %>%
  mutate("dev_mean" = (sheight+fheight)-(mean(sheight+fheight))) %>%
  mutate("sqr_dev" = dev_mean^2)
#combining sets and recalculating same metrics as original set
mc_mean_of_diff <- sum(mean_of_diff_final$difference) #sum of difference ~ -258231
```

```
mc_final_var_stp_2<- sum(mean_of_diff_final$sqr_diff) # sum of sqrd difference ~106215136
mc_final_var_stp_3 <- mc_final_var_stp_2^2 # 11281655278794372
mc_final_var <- mc_final_var_stp_2/(length(mean_of_diff_final$fheight-1))
mc_final_mean_diff <- abs(mean(mean_of_diff_final$sheight)- mean(mean_of_diff_final$fheight))
mc_final_rdad_SD <- sd(mean_of_diff_final$fheight) # ~7.15
mc_final_rson_SD <- sd(mean_of_diff_final$sheight) # ~6.96
```

(b) Now repeat the previous question a large number of times R (1000 or more). Each time store the difference, so you end up with R different values for the difference. What is the mean of the difference values? Explain what do you get. What is it standard deviation? Compare it to that you computed in the previous problem for the difference in data (when doing t-test). What is the largest difference (in absolute value)? The final mean of difference between the randomly generated heights was ~0.01 with standard deviations for fathers at ~6.96 and sons at ~6.97. In the previous trial, I had standard deviations for fathers at ~ 6.97 and sons at ~ 7.15 respectively.

```
Quantile_final <- quantile(abs(mean_of_diff_final$difference), prob=0.95)
Quantile_OG_data <- quantile(abs(height_diff$difference), probs = 0.95)
OG_set <- qt(0.05, df=1077, lower.tail = FALSE) # tscore ~1.64
Ranset <- qt(0.05, df=99999, lower.tail = FALSE) # tscore ~1.64
```

(c) Find the 95% quantile of (the absolute value) your difference. Compare this number to the actual father-son difference you found in the data. For the original set of father / son heights, I calculated the 95% quantile absolute value as 14.9. With this randomly generated set, I calculated the 95% quantile absolute value as 20.2.

Hint: use the R function `quantile` for this.

Citations

Steps for calculating the standard deviation https://www.scribbr.com/statistics/standard-deviation/

Code written above is from the previous course IMT 511 which used the below text to support class scripts. https://www.google.com/books/edition/Programming_Skills_for_Data_Science/BnB6DwAAQBAJ?hl=en&gbpv=1&printsec=frontcover