Tree heights

39 marks

Students in two different senior statistics courses (DataViz having 107 students and EDA having 42 students) at the University of Waterloo were recently offered the opportunity to take a simple, time limited online quiz. Before beginning the quiz they had no idea whatsoever what questions would be asked of them, however they would each earn a single bonus mark for taking the quiz, whatever their answers.

They were asked two questions and given very little time to answer them.

The information presented in the quiz was as follows:

> *"The coast redwood is perhaps the tallest species of tree growing today.*
>
> - *Do you think the tallest tree of this species alive today is*
>
> *A. less than XXX metres tall? B. more than XXX metres tall?*
>
> *Answer A or B.*
>
> - *Write down your best guess (in metres) of how tall you think the tallest tree might be."*

In place of *XXX* above, about half of the students (randomly selected) had the number 50 appear and the others had the number 150 appear. The value of *XXX* presented to the students is called the *anchor* for that question.

For the record, and presumably unknown to the students taking the quiz, the tallest coast redwood tree so far found was discovered in 2006. It was named Hyperion after the Titan of Greek mythology of that name (meaning "the high one") and was measured to be 115.7 metres tall.

The student quiz results are given in the file `trees.csv`.
This may be read into `R` using `read.csv()` (assuming you have the csv file in a directory/folder given by `dataDirectory`) as

```
trees <- read.csv(file.path(dataDirectory, "trees.csv"))
```

Note that the answers to the first question are recorded as logical values in the variable `greater` indicating whether a student believed the tallest tree to be greater than the value of the `anchor` (or not).

Of interest are the heights the students guessed the tallest tree to be and how that compares to the true value as well as to each of the "anchor" values.

a. **(2 marks)** Describe the target population likely intended by the investigator.
   - All humans

b. **(2 marks)** Describe the study population.
   - All students in the two classes: data visualization and eda

c. **(2 marks)** Describe the sampling plan **and** the sample it produced.
   - The sampling plan is stratified sampling because there are two different values (XXX) that is randomly selected for each student. The sample produced is self-selected because not all students will write this question in the end of the test.

d. **(2 marks)** How many students claimed the tallest tree would be greater than the anchor value but then guessed a smaller value? How many claimed it would be less but then gave a greater value?

```
count1 = 0
for (row in 1:nrow(trees)){
   if((trees[row, "greater"] == TRUE) & (trees[row,"height"] < trees[row,"anchor"])){count1 = 1 + count1}}
count1
```

```
## [1] 2
```

```
count2 = 0
for (row in 1:nrow(trees)){
   if((trees[row, "greater"] == FALSE) & (trees[row,"height"] > trees[row,"anchor"])){count2 = 1 + count2}}
count2
```

```
## [1] 5
```

Show your code.

e. **(5 marks)** Consider the sample attribute: average `height`. This would be the average of the student guesses for the tallest tree. Determine the sample error of this attribute as estimate of the tallest tree for each of the following samples (show your code):

- Sample Error = a(S) - a(Pstudy)

i. *(1 mark)* All students of both classes.

```
mean(trees$height) - 115.7
```

```
## [1] 12.47829
```

ii. *(1 mark)* Students in "DataViz".

```
dv_trees <- trees[trees$class == "DataViz",]
mean(dv_trees$height)- 115.7
```

```
## [1] 15.59032
```

iii. *(1 mark)* Students in "EDA".

```
eda_trees <- trees[trees$class == "EDA",]
mean(eda_trees$height) - 115.7
```

```
## [1] 4.438889
```

iv. *(1 mark)* Students who were presented with the smaller anchor.

```
small_anchor_trees <- trees[trees$anchor == 50,]
mean(small_anchor_trees$height)- 115.7
```

```
## [1] -32.21563
```

v. *(1 mark)* Students who were presented with the larger anchor.

```
large_anchor_trees <- trees[trees$anchor == 150,]
mean(large_anchor_trees$height) - 115.7
```

```
## [1] 56.48462
```

f. **(12 marks)** In this question, you will look at producing numerical summaries for samples defined by every combination of `class` and `anchor`.

i. *(3 marks)* Using the `by()` function, determine the sample size and the median and inter-quartile range of the heights given by students in every combination of `class` and `anchor`.

Show your code and the result it produces.

```
classSummary <- with (trees,
                      by(height,
                         INDICES = list(class = trees$class, anchor = trees$anchor),
                         FUN = summary
                         ))
classSummary
```

```
## class: DataViz
## anchor: 50
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
##    3.00   56.25   72.50   78.96  100.00  200.00
## ------------------------------------------------------------
## class: EDA
## anchor: 50
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   65.00   70.00  100.00   99.64  100.00  200.00
## ------------------------------------------------------------
## class: DataViz
## anchor: 150
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    70.0   100.0   160.0   192.1   200.0  1000.0
## ------------------------------------------------------------
## class: EDA
## anchor: 150
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   60.00   86.25  120.00  133.18  187.50  300.00
```

ii. *(1 mark)* What is the `class`, if any, of the structure returned by `by()`? - the class returned by by() are the 4 combinations of the given anchor values and the class. These 4 classes are (DataViz(50)/EDA(50)/DataViz(150)/EDA(150))

iii. *(2 marks)* Using `cut()` add a new variable to `trees` called `sizegrp` that has four groups dividing the observations according to the quartiles of the variable `height`.

Note, `sizegrp` should contain **no** `NA` values.

Show your code and the factor levels for `trees$sizegrp`.

```
sizegrp <- with(trees,
                cut(height,
                    breaks = fivenum(height),
                    labels = c("0-25%","25-50%","50-75%","75-100%"),
                    include.lowest = TRUE, na.rm = TRUE))
trees$sizegrp <- sizegrp
levels(sizegrp)
```

```
## [1] "0-25%"   "25-50%"  "50-75%"  "75-100%"
```

iv. *(2 marks)* How large is each group in `sizegrp`? Explain why the groups are not equal sized. Show any code you use.

```
vapply(split(trees, f = sizegrp),
       FUN = nrow,
       FUN.VALUE = numeric(1L))
```

```
##   0-25%  25-50%  50-75% 75-100%
##      33      45      21      30
```

- The reason why the groups are not equal sized is because that there are many same height in the tree dataset. For example, there are many student that put 100 as their guess for the tallest tree height. Therefore, the groups size are not equal because there are too many equal numbers in the data.

v. *(4 marks)* Using `vapply()` **and** your new factor `sizegrp` produce a 3×4 matrix whose columns identify each of the 4 groups by the factor levels of `sizegrp` and whose rows are labelled `n`, `anchor_50`, and `anchor_150`.

The first row `n` contains the sample size for each group, the second row `anchor_50` contains the proportion of students in each group presented with the anchor value of 50, and the third row `anchor_150` the proportion in each group presented with anchor value 150.

Round each proportion to 2 decimal places.

Show your code.

Describe any patterns you find.

```
n <- vapply(split(trees, f = sizegrp),
       FUN = nrow,
       FUN.VALUE = numeric(1L))
anchor_50 <- vapply(split(small_anchor_trees, f = sizegrp),
       FUN = nrow,
       FUN.VALUE = numeric(1L))
anchor_150 <-vapply(split(large_anchor_trees, f = sizegrp),
       FUN = nrow,
       FUN.VALUE = numeric(1L))
rbind(n,anchor_50,anchor_150)
```

```
##             0-25% 25-50% 50-75% 75-100%
## n              33      45      21       30
## anchor_50      16      22       8       18
## anchor_150     16      22       8       19
```

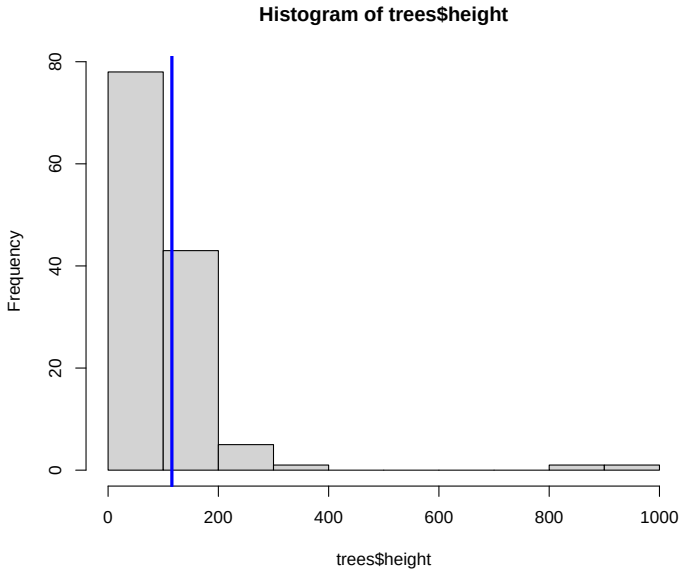
- The patterns I found is that in the size of the 25-50% quantile group is the highest in both anchor value 50 and 150. And the lowest size group both appears in the 50-75% quantile group.

g. **(11 marks)** Here we will look at the entire distribution of height provided by students in different groups.

i. *(2 marks)* Using bin widths of 50, draw a histogram of the heights provided by all students. Add a "blue" vertical dashed line of width 3 at the height obtained by Hyperion.

Show your code.

```
hist(trees$height, binwidth = 50)
abline(v = 115.7, col = "blue", lwd = 3)
```



ii. *(5 marks)* In this question, you are required to use `sapply()` with `split()` to draw a histogram for each `anchor` **and** to return (as the value of the `sapply()`) the average heights for each anchor.

The histograms are to be drawn subject to the following constraints:

- the two histograms appear side by side in the same display
- each histogram has the same x range on its horizontal axis
- each histogram uses bins of width 50
- each histogram has a vertical blue line at Hyperion's height
- each histogram has a vertical red line at the value of its anchor

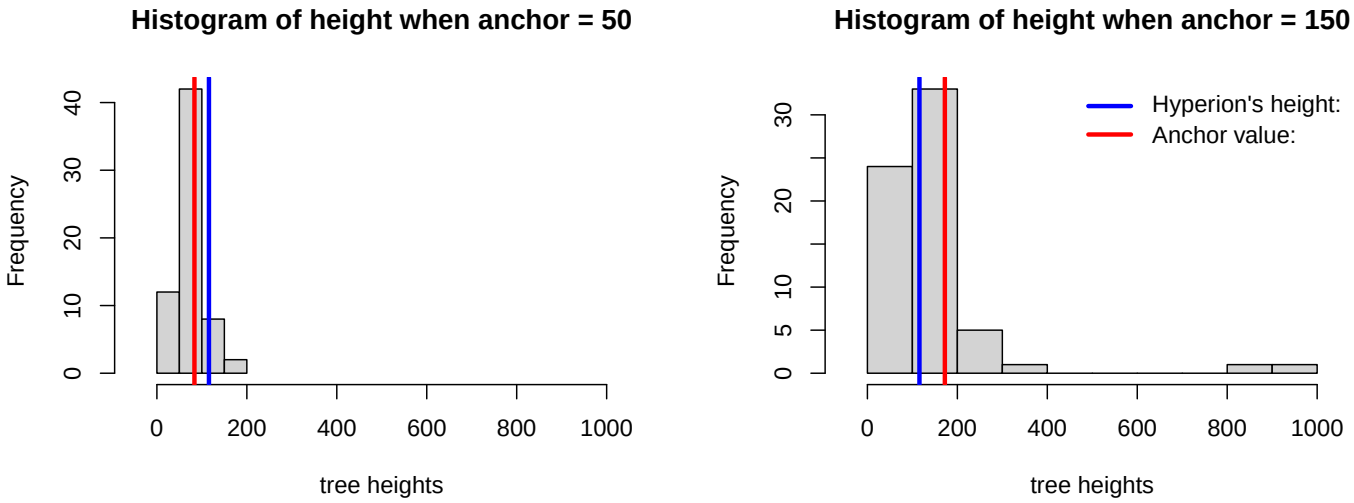
- the title of each histogram says the anchor for which it is displaying values
- label the axes as appropriate
- on the right hand plot **only**, add a legend to indicate which vertical line is the anchor value and which is Hyperion's height

Note: In your `RMarkdown` you might want to use the following header for the R chunk

`{r, out.width = "100%", fig.width = 10, fig.height = 4}`

Show your code.

```
anchor_mean <- sapply(split(trees, trees$anchor),
            FUN = function(x) mean(x$height))
par(mfrow=c(1,2))
xlim <- extendrange(c(0, trees$height))
hist(small_anchor_trees$height, binwidth = 50, xlim = xlim, xlab = "tree heights", main = "Histogram of hei
abline(v = 115.7, col = "blue", lwd = 3)
abline(v = anchor_mean[[1]], col = "red", lwd = 3)
hist(large_anchor_trees$height, binwidth = 50, xlim = xlim, xlab = "tree heights", main = "Histogram of hei
abline(v = 115.7, col = "blue", lwd = 3)
abline(v = anchor_mean[[2]], col = "red", lwd = 3)
legend("topright",  bty = "n",
       legend = c(expression("Hyperion's height: "),
                  expression("Anchor value: ")),
       col = c("blue", "red"), lwd = 3)
```



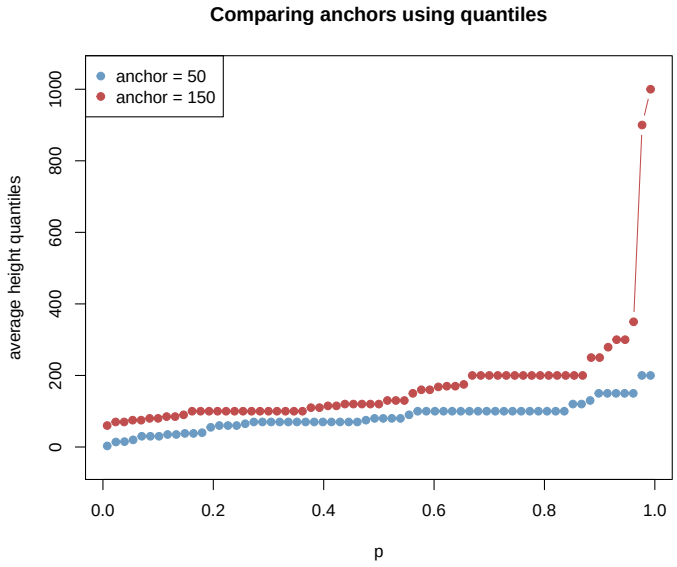
iii. *(4 marks)* **Overlaid quantiles plot** On a single plot, for each `anchor` draw overlaid the sample quantiles of the heights for that anchor. When drawing the plot,

- make sure all heights fit on the plot
- label the axes appropriately
- title the plot "Comparing anchors using quantiles"
- use arguments `pch = 19, type = "b"`
- distinguish the two anchor groups by colour
- add a legend to identify the two anchor groups

Show your code.

```
ylim <- extendrange(c(small_anchor_trees$height, large_anchor_trees$height))
plot(x = ppoints(length(small_anchor_trees$height)),
     y = sort(small_anchor_trees$height),
     main = "Comparing anchors using quantiles",
     ylab = "average height quantiles", xlab = "p",
```

```
     pch = 19, type = "b", ylim = ylim,
     col = adjustcolor("steelblue", alpha.f = 0.8))
points(x = ppoints(length(large_anchor_trees$height)),
       y = sort(large_anchor_trees$height),
       pch = 19, type = "b",
       col = adjustcolor("firebrick", alpha.f = 0.8))
legend("topleft", legend = c("anchor = 50", "anchor = 150"),
       col = adjustcolor(c("steelblue","firebrick"),alpha.f = 0.8), pch = 19)
```



h. **(3 marks) Conclusions**: Based on your entire analysis above, what do you conclude from this study?

Conclusions: I observed that there are around 5 percent, 7 out of 129 participants may not be clear of the intention of this study or did not pay attention while doing the test. Also, when an individual is given a anchor value of 50, the highest guess of the students is 200. However, when an individual is given an anchor of 150, it changes how students guess the highest tree in the world by alot. The average guesses are higher than the ones given anchor = 50. Therefore, the students guesses are influenced with the given anchor value, it would be possible that the average guess of the students will be higher if we are given another anchor value that is greater than 150.