Trees and class intersections

QUESTIONS

```r
library(tidyverse)
```

28 marks

In the previous question “Trees and pipelines” you constructed two data sets, `trees_DV` and `trees_EDA`, containing the results from students in the courses “Data Visualization” and “Exploratory Data Analysis”. These were the results from the quiz described in the file “TreesContext”.

Some students were in both courses. To explore this a little more, analogous data sets to `trees_DV` and `trees_EDA` were constructed but with the (fake) student identification numbers attached.

This data is contained in the files “trees_DV_id.csv” and “trees_EDA_id.csv”.
Read these data in and assign the values to `trees_DV-id` and `trees_EDA_id` respectively.

In this question, the results of students in both course (who therefore took the quiz twice) will be explored.

In all parts, show your code unless told otherwise.

a. **(2 marks)** Read the data in (with `read_csv()`) and assign the values to `trees_DV-id` and `trees_EDA_id` respectively. Show the first two rows of each data set.

```r
(trees_DV_id <- read_csv(file.path(dataDirectory, "trees_DV_id.csv")))
(trees_EDA_id <- read_csv(file.path(dataDirectory, "trees_EDA_id.csv")))
```

```r
head(trees_DV_id, 2)
```

```
## # A tibble: 2 x 5
##       id class   anchor greater height
##    <dbl> <chr>    <dbl> <lgl>    <dbl>
## 1 210823 DataViz    150 FALSE      110
## 2 191321 DataViz    150 FALSE      100
```

```r
head(trees_EDA_id, 2)
```

```
## # A tibble: 2 x 5
##       id class anchor greater height
##    <dbl> <chr>  <dbl> <lgl>    <dbl>
## 1 276025 EDA      150 FALSE       75
## 2 253384 EDA       50 TRUE        70
```

b. **(9 marks)** A focus on those students in both courses.

   i. *(2 marks)* Determine which students took the quiz twice.

      Show the number of such students and their ids.

```
intersect(trees_DV_id$id, trees_EDA_id$id)
```

```
## [1] 276547 193079 277079 276628 271952  80685 282166
```

i. *(2 marks)* Use an appropriate `"join"` function to construct a single data set called `trees_2` which contains **only** those students who answered the quiz for **both** courses.

`trees_2` should contain `id` as well as all variables associated with each course (a total of 9 variables).

Assign the result to `trees_2` and print the result.

```
trees_2 <- inner_join(trees_DV_id, trees_EDA_id, by = "id")
trees_2
```

```
## # A tibble: 7 x 9
##       id class.x anchor.x greater.x height.x class.y anchor.y greater.y height.y
##    <dbl> <chr>      <dbl> <lgl>        <dbl> <chr>      <dbl> <lgl>        <dbl>
## 1 276547 DataViz       50 TRUE            80 EDA          150 FALSE           85
## 2 193079 DataViz       50 TRUE           100 EDA           50 TRUE            70
## 3 277079 DataViz       50 TRUE           150 EDA          150 TRUE           150
## 4 276628 DataViz       50 TRUE            70 EDA           50 TRUE            70
## 5 271952 DataViz      150 FALSE          100 EDA           50 TRUE           100
## 6  80685 DataViz      150 FALSE          115 EDA          150 FALSE          115
## 7 282166 DataViz       50 TRUE            60 EDA          150 FALSE           70
```

i. *(5 marks)* Determine whether these students were asked the same question or not and whether they gave the same answers or not.

Effect this by using a pipeline beginning with `trees_2`.

It should return a `tibble`

- having only three variates: `id`, `same_question`, and `same_answer`
  - `same_question` is a logical vector indicating whether that student was presented with the same question in both courses or not
  - `same_answer` is a logical indicating for those who had the same question whether they provided identical answers or not. For those who had different questions, this should be missing.
- having rows ordered by ascending `id`

Assign the result to `trees_common` and print the result.

```r
# id, same_question, same_answer
trees_2 %>%
transmute(id = id,
          same_question = (if_else(anchor.x == anchor.y, TRUE, FALSE)),
          same_answer = (if_else(same_question == FALSE, NA, if_else(height.x == height.y, TRUE, FALSE
arrange(id) ->
trees_common
trees_common
```

```
## # A tibble: 7 x 3
##       id same_question same_answer
##    <dbl> <lgl>         <lgl>
## 1  80685 TRUE          TRUE
## 2 193079 TRUE          FALSE
## 3 271952 FALSE         NA
## 4 276547 FALSE         NA
## 5 276628 TRUE          TRUE
## 6 277079 FALSE         NA
## 7 282166 FALSE         NA
```

c. **(8 marks)** Putting all students together.

   i. *(2 marks)* Combine `trees_DV_id` and `trees_EDA_id` so that no new columns are added, none are deleted, and the answers from the `class` `"DataViz"` appear above those from the class `"EDA"`.

      Save the combined data as `trees_all` and evaluate the following:

      ```
      dim(trees_all)
      head(trees_all, 2)
      tail(trees_all, 2)
      ```

```
trees_all <- full_join(trees_DV_id, trees_EDA_id)
dim(trees_all)
```

```
## [1] 129   5
```

```
head(trees_all, 2)
```

```
## # A tibble: 2 x 5
##       id class   anchor greater height
##    <dbl> <chr>    <dbl> <lgl>    <dbl>
## 1 210823 DataViz    150 FALSE      110
## 2 191321 DataViz    150 FALSE      100
```

```
tail(trees_all, 2)
```

```
## # A tibble: 2 x 5
##       id class anchor greater height
##    <dbl> <chr>  <dbl> <lgl>    <dbl>
## 1 270550 EDA       50 TRUE        70
## 2 176205 EDA      150 TRUE       200
```

i. *(6 marks)* Using only `trees_all` and `trees_common`, construct a data set called `trees_work` such that it

- has only variables `id`, `class`, `anchor`, `greater`, and `height` (in that order)
- has rows sorted in ascending order of `id` number, then `class`
- contains no data that was duplicated (i.e. same student, question, and answers)
- if a student answered twice, with the same questions and the same answer, only one answer is used and the `class` is changed to `"BOTH"`
- contains each set of answers from a student, provided they are different answers if to the same questions

The data set `trees_work` **must** be constructed in a **single** pipeline with **no** intermediate assignment

When done, evaluate the following:

```
# To test your answer the following should return TRUE
(nrow(trees_work) +
   sum(trees_common$same_answer, na.rm = TRUE)  == nrow(trees_all))
#
dim(trees_work)
head(trees_work, 2)
tail(trees_work, 2)
```

```
full_join(trees_all, trees_common, by = "id") %>%
arrange(id, class) %>%
mutate(class = if_else(!is.na(same_question) | !is.na(same_answer), if_else(same_question==TRUE & same
unique() %>%
select(-same_question, -same_answer) ->
trees_work
(nrow(trees_work) +
      sum(trees_common$same_answer, na.rm = TRUE)  == nrow(trees_all))
```

```
## [1] TRUE
```

```
dim(trees_work)
```

```
## [1] 127   5
```

```
head(trees_work, 2)
```

```
## # A tibble: 2 x 5
##      id class   anchor greater height
##   <dbl> <chr>    <dbl> <lgl>    <dbl>
## 1 54196 DataViz     50 FALSE       30
## 2 80685 BOTH       150 FALSE      115
```

```
tail(trees_work, 2)
```

```
## # A tibble: 2 x 5
##       id class   anchor greater height
##    <dbl> <chr>    <dbl> <lgl>    <dbl>
## 1 282166 DataViz     50 TRUE        60
## 2 282166 EDA        150 FALSE       70
```

d. **(9 marks)** Analysis of the `trees_work` data just constructed.

   i. *(2 marks)* Are any of the answers provided by students contradictory? If so, which student(s) and what is contradictory.

      If not, why not?

      Show code and results to support your answer.

- Yes, there are answers that are provided by students contradictory.

```
trees_work %>%
filter(anchor > height & greater == TRUE | anchor < height & greater == FALSE)
```

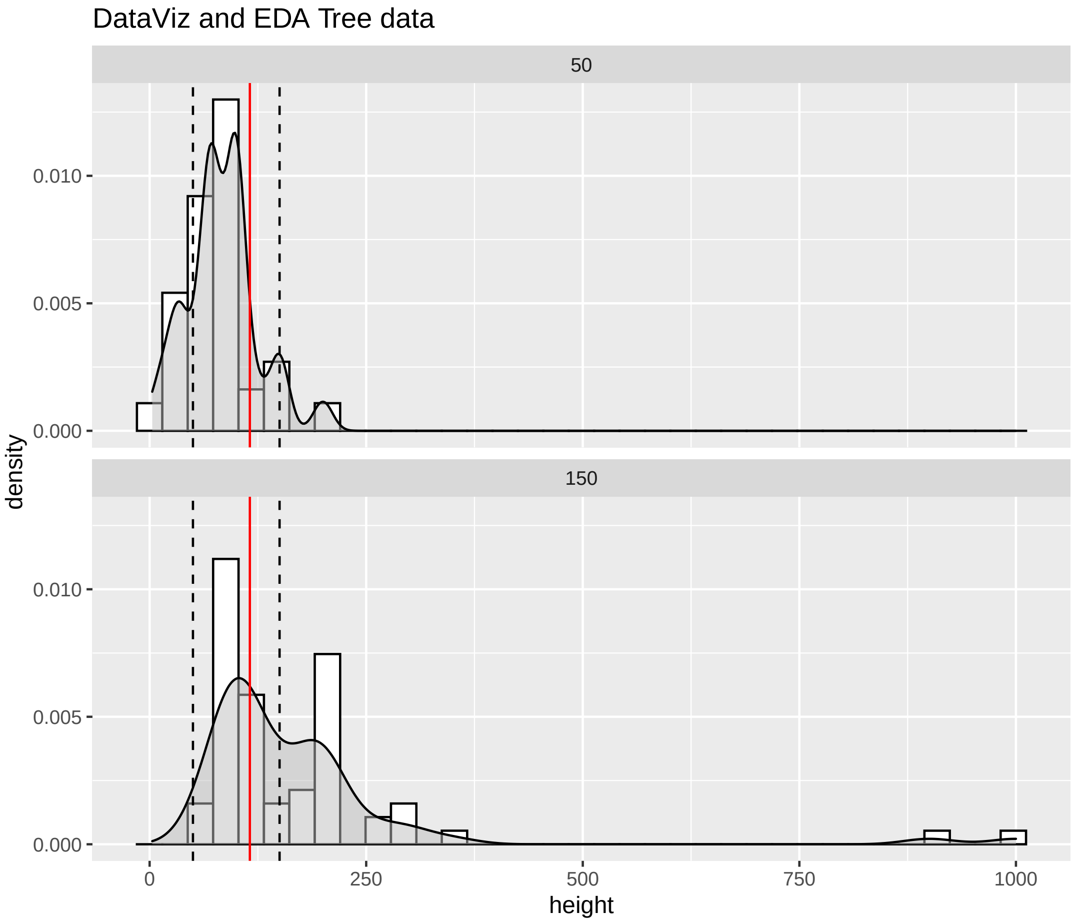
```
## # A tibble: 7 x 5
##       id class   anchor greater height
##    <dbl> <chr>    <dbl> <lgl>    <dbl>
## 1 159770 EDA        150 FALSE      200
## 2 185045 DataViz    150 FALSE      200
## 3 201444 DataViz    150 FALSE      200
## 4 225936 DataViz     50 TRUE        15
## 5 226687 DataViz    150 FALSE      200
## 6 228849 DataViz     50 TRUE        35
## 7 235612 DataViz    150 FALSE      168
```

\newpage

   i. *(7 marks)* Using the `trees_work` data with `ggplot2`, draw a **single** histogram of the estimated heights. The plot should have

      - ggplot be fed the data via a pipe
      - have a title
      - have fill colour "white" with "black" border
      - have bins = 35
      - have a 50% transparent "grey" filled density estimate overlaid
      - have a vertical "black" line dashed line at each anchor point
      - a vertical "red" solid line at the actual height of the tallest tree ("Hyperion" at 115.7 metres)
      - the plots facetted by anchor and stacked vertically

```
trees_work %>%
ggplot(data = ., mapping = aes(x = height)) + aes(y = ..density..) +
geom_histogram(bins = 35, fill = "white", col = "black") +
geom_density(fill = "grey", alpha = 0.5) +
geom_vline(xintercept = 50, col = "black", linetype="dashed") +
geom_vline(xintercept = 150, col = "black", linetype="dashed") +
geom_vline(xintercept = 115.7, col = "red") +
facet_wrap(~ anchor, ncol = 1) +
ggtitle("DataViz and EDA Tree data")
```


DataViz and EDA Tree data
50
150
density
0.000
0.005
0.010
0
250
500
750
1000
height