Grade analysis

Fabrice Rossi

This lab is dedicated to the analysis of a real world data set described below. Your work must be submitted as a github project and as a zip file. You must:

- fork the following github project: `https://github.com/fabrice-rossi/r-101-grade-analysis`. Your fork must be public.
- create a R project on your computer from the forked github project and make an initial commit with the classical R project configuration files;
- write all your answers in a quarto document: commit the initial version of this document but dot not include the rendering of the document in the repository;
- each time you are satisfied by your answer to a question, commit the modifications;
- push the commits on a regular basis;
- at the end of the session, make a final commit with a push and then prepare to upload a zip file on moodle with at least:
  - the R project file (ending with `.Rproj`);
  - the data files (csv or rds format);
  - the quarto document;
  - the result of rendering your document to html.

All graphical representations must be done with ggplot2 and all calculations must be done with dplyr and tidyr.

We study in this lab a data set of grades/marks which gives results of students during a semester. Each student is identified by their `Id`. Students belong to groups identify by the `Group` variable. During the semester, students received 10 grades given by the `MCQ_x` variables. They could also take online tests with grades recorded in the `Online_MCQ_xx` variables. The final evaluation of the semester is reported in the `Exam` column. All grades are between 0 (worst) and 20 (best).

Notice that many grades are missing either because the students were absent or because they chose not to participate to an online test (those tests were not mandatory). Missing grades are given by the `NA` special value. To avoid any problem in the aggregation functions, it is recommanded to use the parameter `na.rm=TRUE`. To detect missing values, one can use the `anyNA` function, an aggregate function that returns `TRUE` if and only if there is at least a missing value in the vector. For instance, if `grades` contains the grades,

```r
grades |> summarise(anyNA(Exam))
```

anyNA(Exam)
TRUE

The call `is.na(x)` returns a vector of `TRUE` and `FALSE` values, with a `TRUE` at a given position if and only if the value was missing at the same position in `x`. For instance

```r
is.na(c(1, 2, NA, 4, NA))

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

Question 1 Load the data set using a local file name (using here). Notice that the file is included in the forked repository.

Question 2 Compute the minimum, maximum, median and mean grade at the `Exam` and include the results as a table in the rendered document (using, e.g. `knitr::kable`).

Question 3 Counts the number of students who did not take the final exam (i.e. for whom `Exam` is `NA`) and include the results in a sentence in your rendered document.

Question 4 Represent graphically in an adapted way the distribution of grades at the `Exam`. Make sure to adapt the code to avoid any error or warning from ggplot2.

Question 5 Compute the number of students in each `Group` and include the results in your document as a table.

Question 6 Represent the same information as in the previous question in a graphical way.

Question 7 Represent graphically the distribution of the grades at the `Exam` conditioned by the group. Test at least two different representations.

Question 8 Compute the number of students who missed the `Exam` in each `Group` (i.e. students whose `Exam` is `NA`). Beware that this number can be zero in some groups and this can induce difficulties with `group_by`. A way to circumvent this problem is to note that the `sum` of a vector of `TRUE`/`FALSE` values is exactly the number of times `TRUE` values that appear in the vector. For instance

```r
sum(c(TRUE, FALSE, TRUE))

## [1] 2
```

Question 9 Represent graphically the results obtained in the previous question. Working directly with the table obtained before is possible using the `geom_col` function from ggplot2. Other approaches are possible with the help of `mutate` for instance. In any case, the graphical representation must include all groups, even those where all students took the final exam.

Question 10 Create a new data frame built from the grades data set reshaped to a long format. The new data frame should keep the `Id` and the `Group` as the orignal variables. The first lines of the data frame should have the following form (the actual values may be different)

Id	Group	name	value
561	grp_16	Exam	10.00
561	grp_16	MCQ_1	15.00
561	grp_16	MCQ_2	18.46
561	grp_16	MCQ_3	12.73
561	grp_16	MCQ_4	8.24

Do no include in your rendered document the table but keep it in a variable to be used in later questions.

Question 11 Using the long format, compute the number of missing grades in total for each student.

Question 12 Represent graphically the distribution of the number of missing grades per student.

Question 13 Using the long format, compute the same information as in question 8.

The stringr library proposes many convenient functions to deal with strings. In particular, the function `str_starts` is very convenient to select rows in which a variable contains a text that starts with a given sequence of characters. For instance, if `grades` contains the grades, then

```
grades |> filter(str_starts(Group, "grp_1"))
```

keeps only the students in groups that starts with `grp_1`. The function `str_ends` has a similar role but with respect to the end of the strings.

Question 14 Using the long format, compute the number of missing grades for the online tests for each student.

Question 15 Represent graphically the distribution of the grades at the Exam conditioned by the number of missing grades for the online tests. Notice that this cannot be done directly using the table obtained in the previous question. It is recommended to use the row-wise functionality of dplyr to perform the calculations needed for this question, starting with the original data frame (not the long format one) and leveraging the `c_across` function. In this approach, columns can be selected with the `starts_with` function. Another approach consists in joining the results obtained from the previous question with the original table.

Question 16 Create a table with two columns: `Id` to identify students and `Missed` with value `TRUE` when the student miss at least one `MCQ_xx` grade and `FALSE` when they miss no grade.

Question 17 Create a table with two columns: `Group` to identify groups and `P_missed` with the percentage of students in each group who missed at least one `MCQ_xx` grade.

Question 18 Represent graphically the average grade at the `Exam` per group as a function of the percentage of missed `MCQ_xx` grade as defined in the previous question. The simplest way to obtain the data is to use `inner_join` to merge two intermediate tables (the one obtained in the previous question and one giving the average grades per group).