Report

The first step in this project was to choose the dataset that will be used. In this case we decided to use the next dataset from Kaggle:

Dataset: High School Student Performance & Demographics

The main reasons to choose this dataset were:

- **Diverse types of Data:** the dataset selected has a reasonable amount of columns which resulted in a varied amount of Data types.
- **Size of the dataset:** the dataset has enough rows to be useful while being below the max size mentioned in the worksheet.
- **Meets the requirements:** A deciding factor when selecting the dataset was that the dataset fulfilled the requirements.
- **Personal interest:** Our group found the topic of the dataset to be of great interest when comparing it to other datasets of similar characteristics.

Luckily this dataset was one of the first considered by our team. Another strong competitor was a dataset about crypto currencies but it lacked variety when it came to data types since almost all of the columns were numeric values.

When it came to loading the dataset into the Rstudio workspace, it was rather simple. The CSV of the dataset could be loaded using the default values of read.csv() function. The file had a header, it was separated by commas and it contained no quotes, decimals nor missing values. After this, we started with the data preparation.

When it comes to data types, we ordered them in three categories:

- **Logical fields:** These fields contained “yes” or ”no” values that would become booleans.
- **Numeric fields:** In this case, the fields already were numbers so no transformation is required.
- **Factor fields:** The strings contained in these fields would be transformed into factors.

To achieve this, we made the necessary transformations to their ideal data type, removed unnecessary rows such as “student_id” and reordered them based on data types so that the dataset would be neat and tidy for its use.

As for identifying outliers, we decided to use all the numeric columns available in the dataset. After taking a look at the values of the outliers found, we chose not to remove any outliers, as we consider them to simply be less common values caused by natural variations, and not errors induced by measurement or handling oversights. Additionally, they were all in the range of values specified by the dataset provider.

Since the dataset had no missing values, we decided to change 2% of the values from the health column since health could be a sensitive topic to some students and they could refuse to give personal information regarding their health. To overcome this lack of data, we decided to replace all missing values by the rounded mean of all the values that are not missing. This last bit was easily achieved thanks to the implementation of “na.rm” in the mean function that ignores all missing values.

We also created the column named normalized_health which takes the values of the health column and normalizes them in the range of [0,1] using the Mix-max Scaling formula:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Another new column, this time numeric, was created by the name of “week_alcohol”, and is calculated by adding the values of the “weekday_alcohol” and “weekend_alcohol” and dividing the result by 2 to make the average.

The last added column, that is categorical, is called “final_grade_letter” and provides a letter based on the final grade obtained by the students. In this case we follow the next logic to assign a letter:

- Final grade > 90 -> A
- Final grade > 80 -> B
- Final grade > 70 -> C
- Final grade > 60 -> D
- Final grade < 60 -> F

Finally, we saved any changes to the dataset in a binary RData file called “datasets.RData” that would be later used in the Shiny app.

When it comes to ways in which we could enrich the data from the dataset, some of the more interesting ones would be:

- **Study Method:** Getting information about the study method used by both the teacher and students would certainly be useful for any use that might be given to the dataset.
- **More Data:** Having more data is always welcome since it helps getting better predictions and reduces the probability of miscalculating.
- **Teacher dataset:** Adding information about the teachers would make the dataset more complete and versatile. This way the dataset could be used in more applications.

Another concern are the potential bias-related issues in the dataset. In our case, the biggest one would be the fact that some students don't feel comfortable sharing their information. This would result in a lack of data from certain spectrums such as lower than average performing students. One way of addressing this issue would be to reassure the students that their data will be taken anonymously and that their data will not be used for anything apart from the specified uses that they have given consent to.

Starting with the Shiny application, we decided to approach it with a functionality point of view. We also tried to implement as many things as we could from what we learned in class. Because of this, the app features:

- **Title:** Displays the name of the dataset and is using HTML content.
- **2 Selection Boxes:** Here we enter the two columns we want to compare.
- **Tabs:** Using these tabs, we can change the dataset that will be used for all the operations.
- **Slider:** This slider lets us decide the amount of sample data that will be used to calculate the plots.

- **Button:** When pressed, this button draws the plots based on the choices selected.
- **Plot:** After selecting the wanted characteristics and pressing the button, a plot will be drawn.
- **Varios CheckBoxes:**
    - **K-means:** When checking this box, a slider will appear to select the amount of groups wanted. After selecting, the dots in the plot will be painted in different colors to represent the different groups.
    - **Correlation:** When checking this box, the correlation between the two variables selected will be displayed.
    - **Tendency line:** When checking this box, the tendency line will be calculated and drawn in the plot.

Another approach we considered was to include more than two columns and generate all possible plots between them. The code for this can be found in the last lines, which are commented out to avoid interference with the actual project. Ultimately, we decided against implementing this approach as it added complexity that we deemed unwarranted when compared to the value it provided to the project.