Mini Project 2

PSTAT100: Data Science Concepts and Analysis

Instructor: Ali Abuzaid

STUDENT NAME

- Kai Barker (kaibarker)
- Luke Drushell (drushell)

💡 Instructions

- This mini project is designed to give you practical experience with real-world data using R and Shiny. You'll create an interactive web application that allows users to explore and visualize a dataset.
- Work in groups of **2 students** from the same discussion section.
- Individual submissions will not be accepted.
-
  - Please use the provided `MP 2.qmd` file to type your Documentation and Presentation and submit it as a PDF file. You can utilize `RStudio`for this purpose. For guidance, refer to the Tutorial: Hello, Quarto).
- Please submit a `.zip` file that includes:

Your `app.R` file (fully working Shiny app).
A short project report (PDF).
Reminder: If your app fails to open or the .zip is incorrect, you will receive a score of **ZERO**. Test everything before submission.

🔥 Due Date

Due Date: Sunday, June 1, 2025, 11:59 PM

1 Documentation and Presentation

1.1 App Purpose

The purpose of our app is to allow users to explore the “Diamonds” dataset. This dataset, sourced from kaggle (https://www.kaggle.com/datasets/shivam2503/diamonds), contains diamond characteristics such as carat, cut, clarity, etc., with each row representing a diamond. Our app allows users to choose from the available variables and select a plotting method in order better visualize that variable, as well as the feature for users to create their own diamond. Users can see their own diamond graphed among the ones present in the dataset, and can choose any of the variables to plot, where the custom diamond data will appear red (for scatter plots). We hope to offer users an interactive platform to explore the data, as well as provide some insight to how diamonds may be valued, or how certain characteristics of diamonds tend to look, given real world data.

1.2 How it Works

The user interface consists of three main components. The first component is the variable and graph selector. In the upper left corner of the application, users can select a variable from the dataset that they would like to visualize. Below the variable selector, in the same box, users can choose a plot for their selected variable. Their options are a scatter plot, histogram, and a box plot. The second component is the graph itself. Displayed on the right hand side of the screen, users will see their selected variable and plot projected, with appropriate axis and title labeling, as well as a summary of the variable below the plot. The last component is the option for the user to create their own diamond. Users can input/select values for each of the diamonds characteristics in order to create their own diamond. Observe that there is no option for the user to input a price. This is because a diamonds price is relative to its characteristics. The price of the diamond is generated based on the users inputs and a multiple regression model. If a custom diamond is made, it will be added to the data and will appear on the graph displayed, for a scatter plot, the new data will appear red.

1.3 Insights

Working on the shiny application, we had several important takeaways about the diamonds data and the design of our app. Initially, we noticed that the data had already been cleaned, as there are no NA values, and all variables and data are relevant to diamonds and their characteristics. However, the variables “depth”, “x”, “y”, and “z” were a little confusing so they needed to be renamed. The “depth” variable was changed to “depth_pct”, as it represents the total depth percentage, while “x”, “y”, “z” were changed to length(mm), width(mm), and depth(mm) respectively, as they represent the diamonds physical dimensions. There weren’t any observed outliers in the data, nor did anything in particular stand out. However, it seems that one of the most important parameters contributing to the price of the diamond is the amount of carats. This was important for estimating the price of diamonds (for the custom diamond feature) based on the users inputs. Based on previous work (PSTAT 126) on this data, it was found that price and carat needed to be log transformed in the model in order for it to maintain linearity. Ensuring that the model is linear is important for ensuring a more accurate prediction of diamond price. One limitation of the current app design is that users can only visualize one variable at a time, and can’t directly compare two variables of their choosing. Allowing the user to select the individual x and y axes on the graph, (a user might want to plot carat against price or depth against clarity) could provide more useful exploration of the data.

1.4 Reflections

Building this app was an opportunity to get familiar with Shiny, and to strengthen our skills in R, data analysis, and data visualization. While the finished product turned out well, some features could be changed or tweaked if this project were to be repeated. The main issue with the current version is that adding a new custom diamond will show up on the scatter plot quite well, but not on the box plot or histogram. While this is simply due to the nature of the plots, where a single (highlighted red) point on a scatter plot is apparent, while a single value in a box plot or histogram will get grouped with existing data, it makes the connection between being able to create your own diamond and visualizing the data with the plots less practical. If there are no apparent changes to the latter two plots, at the very least we can observe the summary statistics changing as custom diamonds are added to the dataset. Another feature that could be improved is the plot selection logic. While the current version of the app allows users to select whichever variable or plot they want, this is not always appropriate as certain variable and plot pairs won’t always make sense. For example, placing a factor variable in a scatter plot, or trying to use one of the non-numeric variables in a histogram. While the app currently notifies users that the histogram must take numeric data, it might be a better alternative to default certain variables to specific graph types, to not show incompatible plots when certain variables are selected, or offer a wider selection of visualization options. Despite these minor flaws, our team collaborated well in a timely manner, and the desired outcome and functionality of the app was implemented as planned.
