After exploring Lab L02, I became more confident in manipulating datasets in Google Colab using the provided tools. At the beginning, it seemed challenging and confusing, seeing the many instructions and pre-written code blocks. However, after skimming through the lab, I came across a few syntaxes that I was already familiar with. The import statements that called libraries like matplotlib, pandas, and numpy immediately caught my interest.

I also noticed that sklearn has precompiled datasets, which were imported using: from sklearn.datasets import load_iris. Another statement preceding this line was: from sklearn import datasets. I was initially confused as I tried to understand why the entire dataset was imported first, and then specific parts were imported again using another statement. I haven't found a clear reason for this yet.

Although there were only a few features in the dataset, the assignment was based on multiclass classification, whereas I expected something binary. That was a surprise. What made it more interesting was how the target variable was specified, like: df['species'] = iris.target_names[iris.target] I wished I could manually specify the target variables instead of using the list that was already provided. Viewing the data with methods like .head() and .tail() on the DataFrame was familiar to me, and it reminded me of different ways to explore the data. I followed the steps to create a plot and found them interesting to memorize. The pattern is as follows: first, use plt to create an empty figure window with a specified size, for example: plt.figure(figsize=(13, 8)) Next, specify the type of figure to be created. Examples include bar for a bar chart and scatter for a scatter plot: plt.bar(x, y, options) This method takes in the x values, y values, and other options such as colors. Then we can add a title with plt.title("Title"), labels for the axes with plt.xlabel() and plt.ylabel(), and finally display the plot with plt.show(). This was fun, as I was able to visualize the data effectively and gain a better understanding of the relationships within it.

species_data = df[df['species'] == species] There was one line I couldn't fully understand because of my limited knowledge of Pandas, although I notices it is a filtering technique, I have to explore how it truly works.