Data Exploration

The objective of those exercises is to get started with the python, pandas, seaborn/plotly frawork. I recommend to work with notebooks.
You can use LLM (AI), but at the condition that you understand each and every line of code written in your file.

1 Fundamentals

1. Toy dataset: Correlation
   When analyzing data, a first important aspect to understand it is to analyze the relation between variables. For instance, if I want to predict if chocolate can cure cancer, I can start by checking if people eating more chocolate tend to have less cancer. We can use correlation coefficients to do so naively. However, we will see that these coefficients should be analyzed with caution
   (a) Load the dataset `coffee_effects.csv` found on the class website
   (b) Plot the first few lines to check the content (`.head(2)`), or simply write the name of the dataframe in a notebook)
   (c) Check the Pearson correlation between variables. You can directely use `df.corr()` for instance. You can plot the correlation matrix for instance with `seaborn` library with a command such as `sns.heatmap(correlation, annot=True, cmap='coolwarm', fmt=".2f")`
   (d) Do the same with Spearman (option `method='spearman'`)
   (e) Observe the correlation values and try to interprete the relation between the variables
   (f) Draw a scatterplot with the caffeine consumtion on the x-axis, and another variable in the y-axis. Repeat for all 3 variables. Use `plotly` library to have an interactive plot, for instance with the command the command
   `px.scatter(df, x='caffeine_consumption_mg', y='productivity')`
   (g) For each relation, you should observe something unusual. Ask yourself questions such as: is the relation linear? Is it monotonous? How strong is the relation between changes in the two variables?

2. Car Dataset: Exploration and Cleaning
   (a) Load the dataset `cars_synthtic.csv` .
   (b) Compute the classic descriptors of the `price` column using pandas' `describe` function. Check the mean, std, percentiles, and extreme values...
   (c) Plot the distribution of the `price` variable using a histogram. You can directly use pandas plotting tools (`df[col].plot.hist()`). Vary the number of bins using the `bin` parameter to see if the distribution seems to follow a bell curve. Use a `kde` plot instead of a `hist` .
   (d) Let's use bootstrapping to check manually if this distribution can be normal. Using `np.random.normal` , generate 1000 values with the same mean and standard deviation than the price. Plot a similar histogram. Observe that the original data was significantly different from the generated data, and thus not accurately described by the mean and std. Observe in particular the differences with min and max values.

3. Data Cleaning
   (a) Describe the variable `length` in the same way as above. Observe that you encounter difficulties, and try to find the cause. Search for abnormal values...
   (b) Fix the problem temporarily by replacing erroneous values by `np.nan`. You can for instance use dataframe indexing, like: `df.loc[df['B'] < threshold, 'B'] = np.nan`
   (c) Try to do the same process with the `weight` column. You should encounter yet another problem. Try to understand what is going on, and then to fix it. You can use for instance `df.info` to find the type assigned to columns, and `pd.to_numeric` with option `errors="coerce"` to ignore errors (nb.: you'll certainly introduce new errors doing so, but let's start with a *quick and dirty* approach).
   (d) For columns with few missing values, remove the corresponding rows. You can use the `dropna()` function. It has a `subset` parameter to take only some columns into account. For columns with many missing values, keep them for now.
4. Visualizing
   (a) To quickly visualize various information about your dataframe, you can use a dedicated tool. For instance, install the `ydata-profiling` package (import with `ydata_profiling`, and apply it to your dataframe using the `ProfileReport` function. You could also have used packages `DataPrep`, `SweetViz`, `AutoViz`
   (b) To really understand your data, you will however often have to spend time designing your own plots. In this example, use plotly's `px.scatter` function to design a plot in which: x is the `year`, y is the `price`, the symbol shape depends on the `type`, the symbol color corresponds to car's `color` and the symbol size corresponds to the car's `weight`. Try to check if you see some patterns in it. For instance, does it seem that the color or the type has an influence on the price?

2 Going Further

5. Mastering the scores

   To be sure to understand correctly what is going on, we will write code to compute manually simple scores. For each question, use the native function in python corresponding to those scores to check that your function is correct
   (a) Write a function to compute the variance. Compute the variance of a column.
   (b) Write a function co compute the MAD (mean absolute distance to the mean or median).
   (c) Write a function to compute the covariance between two variables
   (d) Using the covariance, write a function to compute the Pearson (linear) correlation coefficient
   (e) Using the Pearson CC function, write a function that compute the Spearman CC (you can use for instance `scipy.stats.rankdata`).
6. Statistical significance
   (a) Plot the cleaned distribution of the `length`. Does it look like a normal distribution? What about the distribution of length for the SUV only? Standard cars only?
   (b) To know if a variable follows or not a given distribution, the best is to use a *statistical test*. The Shapiro-Wilk test is a classic method to check normality for a variable. Check the Wikipedia page to see how to interpret it, then see how to run it in python (`scipy.stats.shapiro`).
   (c) Evaluate if the variable `length` follows a normal distribution, for instance considering a p-value of 0.05, or 0.01 ?

7. Real data
   (a) On the class page, you can find a dataset corresponding to real data about used cars, for one brand. Download it (you can also find the reference to the original dataset, containing other brands, if you prefer).
   (b) Apply a similar analysis on this real data.