INFO2000

Group Exercise 8

Exploratory Data Analysis (EDA) with Pandas

Due: Thursday 2024 November 14 10:50 am (end of class)

Overview

The goal of this exercise is for you to practice ingesting data, cleaning it, and visualizing it using Pandas. In addition, you will be grouping the data into forms that allow you to draw valuable insights or conclusions.

NOTE: If you get a **SettingWithCopyWarning** warning, it means that you are trying to change a view of a dataframe, and not a copy. This usually happens when you assign the result of an operation on an original dataframe to a new dataframe and then try to change the new dataframe. e.g.

```
clean_df = df.dropna() # assignment
clean_df.Length = pd.to_numeric(clean_df.Length) # cast length to float. This throws a warning
```

Fix it by explicitly creating a copy:

```
clean_df = df.dropna().copy() # assignment with copy
clean_df.Length = pd.to_numeric(clean_df.Length) # no warning
```

1 Data cleaning

You have been provided with Car_sales.csv which contains data about car sales, on ELC.

1. Download the csv file from ELC and create a python notebook where all the work from today's exercise will go.
2. Import all the necessary libraries e.g. pandas, numpy, seaborn, matplotlib etc., and read in the data into a pandas dataframe.
3. print the first 5 items and the last 3 items of the dataframe.
4. print out the information about the dataframe
5. print the description of the dataframe.
6. print the shape of the dataframe.
7. print/display the count of missing/null values in each column
8. Drop all columns with more than 20% missing/NA values
9. Drop all rows with NA or missing values.
10. Print the shape, info, and description of the dataframe. Print the sum of null values in each column (confirm that there are none).
11. confirm that all datatypes are what you expect them to be. e.g. are all dates of type DateTime? are all numbers represented as floats and ints? if not, make the necessary conversions.

2 Drawing Insights

1. Plot a bar graph of the total sales based on manufacturer in descending order.
2. Plot a scatter plot pitting the fuel efficiency (y-axis) against the engine size (x-axis). Differentiate the plot based on the vehicle type i.e. 'passenger' and 'car' types should have different hues. **Hint:** Use a seaborn scatterplot
3. Create a new column labelled "origin" which represents the continent of origin for each of the cars .e.g. North America for Ford, Europe for Audi, Asia for Toyota etc. **Hint:** df['column_name'].unique() will give you the unique items in a column.

4. Compute revenue = sales * price, convert it into millions, and put it in a new column labeled "revenue"
5. Plot a bar graph of revenue vs origin.
6. Create new columns labeled 'Launch_year' and 'Launch_month' based on the Latest_Launch column. Example:

```
df['Latest_Launch'] = pd.to_datetime(df['Latest_Launch'])
# Extract year since launch
df['Launch_Year'] = df['Latest_Launch'].dt.year
```

7. plot a line plot of the launch year vs. revenue
8. plot a line plot of the launch month vs. revenue

NOTE: One group member should push the completed work to GitHub under a new folder named GroupExercise8.

Demonstrate your work to the instructor or TA for grading in class.