END2971 – Advanced Computer Programming

Mehmet G. Güler - İsmail Sevim Due 04.01.2022@23.45 (Sharp)

Assignment 2

Explanation:

- *You have a dataset that has cars for sale. Each row is a car for sale. You will analyze this data. The data is here.*
- *You will submit your jupyter codes by **zipping** through online.yildiz.edu.tr.*
- *Submit a Jupyter file which has a cell that has the question and in the following cell there must be the answer. An example jupyter file can be found here:*
- *Keep the same group. If your group members decided not to work somehow or withdraw the lecture or any other unexpected thing happened , you can continue alone. But send me an email regarding your situation.*
- *This sheet is **3 pages.***

TASKS:

1. There are some repetitive rows, i.e. the same cars are given in different lines. If everything except the price is the same, then they are ads for the same car which means these rows are repetitive. Find and clear them. If two lines belong to the same car, then erase the data which has the larger price since it means that the car is not sold and the owner has reduced the price in a new ad to sell the car.

2. Find the average price of cars for each years and return it as a dictionary like:

```
{ "2005": 21900,
  "2006": 32120,
  "2007": 41100
…. }
```

3. Find the standard deviation of the prices of each brand. You can find the definition of the standard deviation from wikipedia or use the following formula where Xbar is the average of the data.

$$s = \sqrt{\frac{(x_i - \bar{x})^2}{n-1}}$$

The output should be dictionary like

```
{ "Audi": 2397,
  "BMW": 3160,
  "Mercedes": 1106
…. }
```

4. Prices of some cars are missing. Find them and replace them with the average prices of the years they belong to. For example if the car is from 2015, then assign the average price of the cars in 2015.

5. Find the average price of each city and return it as a dictionary like above.

6. Find the average price of each brand and return it as a dictionary like those above.

7. Find the average price of the Mercedes-Benz cars that has automatic transmission in İstanbul.

8. Find the number of cars whose price is less than 100K TL, between 100K(exclusive) and 200k (inclusive) and larger than 200k(exclusive). (K means thousand.) The output should be dictionary like this:

```
{ "0-100":20,
  "100-200":40,
  "200 and over"50}
```

9. Write a function which generates the above list automatically. That is, write a function which calculates the number of cars for a given list of prices. For example, for the above question, the input will be [100,200]. If you given [50, 100, 150, 200, 250], then the function will return a dictionary like this:

```
{ "0-50":20,
  "50-100":30,
  "100-150":40,
  "150-200":40,
  "200-250": 50}
  "250- and over": 120}
```

10. Find the number of cars that has an *automatic transmission* whose price is less than 100K TL, between 100K(exclusive) and 200k (inclusive) and larger than 200k(exclusive). (K means thousand.) Again, the output should be dictionary.

11. For every brand, find the difference for the average prices of automatic transmission and manual transmission and return it as a dictionary.

12. Create a new dataframe which is ordered in terms of the prices of the cars.

13. Create a new dataframe which is ordered in terms of the cities first and then prices of the cars.

14. Divide the price of the cars by its power (Cost/power) and add it as a new column to the datafame.

15. How many different series each brand have? The output should be a dictionary like above.

16. Write a function called analyze_car_data that has at least two parameters as follows:

```
def analyze_car_data(count_type, property_type):
```

- Count can be price or count or power
- Property can be City, Brand, Fuel, Transmission, Chassis or Color.

If you call the function as follows then it will return the average car price of each city.

```
analyze_car_data("Price", "City")
```

For example, if you call the function as follows then it will return the number of cars for each transmission type.

```
analyze_car_data("Count", "Transmission")
```

If you call the function as follows then it will return the average power of cars for each Brand.

```
analyze_car_data("Power", "Brand")
```