Ironhack Student Portal

Lesson Goals

In this lesson we will learn:

- Grouping data in Pandas
- Using the aggregation functions to summarize grouped data

Introduction

Aggregating and summarizing are essential tools in data analysis. They allow us to perform computations on our data or look at descriptive statistics for subsets of the data. These calculations can help us make meaningful inference regarding our data.

We will use the `vehicles.csv` data set you used in Module 1. In case you don't have the data set handy, download it again from here. Extract the content of the downloaded file to your machine. `vehicles.csv` is contained in the extracted folder.

Grouping

We have looked at the group by clause in SQL in previous lessons. Pandas has a similar function that enables us to perform aggregations - the `groupby` function.

Applying the `groupby` function to a DataFrame will return a DataFrameGroupBy object. We then specify the columns that we intend to group on.

Recall the vehicles dataset from previous lessons:

```
import numpy as np
import pandas as pd

vehicles = pd.read_csv('vehicles.csv')
vehicles.groupby(['Transmission'])
<pandas.core.groupby.groupby.DataFrameGroupBy object at 0x00000177FBAB0F98>
```

This object contains information that can be "unleashed" when an aggregation is applied to this object.

Aggregations

We can apply different aggregation functions to our grouped data. We can use some standard functions or define our own functions and then apply them to the aggregated data using the `agg` function.

Some standard aggregation functions are: `mean` , `sum` , `count` , `median` , `min` , `max` , `std` .

We can also use the `agg` function to apply multiple aggregations at once to all columns specified.

After aggregating, we can subset the data to only apply the aggregation to the columns that we choose.

Here are some examples of standard aggregation functions:

```
vehicles.groupby(['Transmission'])['Highway MPG', 'City MPG', 'Combined MPG'].mean()
               Highway MPG      City MPG        Combined MPG
Transmission
Auto (AV)       40.000000       35.000000       37.000000
Auto (AV-S6)    25.000000       22.000000       23.000000
Auto (AV-S8)    22.000000       20.000000       21.000000
Auto(A1)        37.000000       41.000000       39.000000
Auto(AM-S6)     32.978261       24.315217       27.554348
...     ...     ...     ...
Manual 5 spd    14.000000       14.000000       14.000000
Manual 5-spd    25.664312       19.242327       21.634391
Manual 6-spd    26.202229       18.306232       21.153941
Manual 7-spd    26.205882       18.220588       21.117647
Manual(M7)      22.333333       14.000000       17.000000
45 rows Г— 3 columns

vehicles.groupby(['Fuel Type', 'Cylinders'])['CO2 Emission Grams/Mile'].median()
Fuel Type                    Cylinders
CNG                          4.0          253.197321
                             6.0          417.030882
                             8.0          568.070913
Diesel                       4.0          308.484848
                             5.0          391.538462
                                             ...
Regular                      8.0          634.785714
                             10.0         776.500000
                             12.0         683.615385
Regular Gas and Electricity  4.0          129.000000
Regular Gas or Electricity   4.0           51.000000
Name: CO2 Emission Grams/Mile, Length: 48, dtype: float64

vehicles.groupby(['Fuel Type'])['Combined MPG'].agg(['mean', 'median', 'std'])
                                mean    median  std
Fuel Type
CNG                         18.133333   14.5    7.436663
Diesel                      23.488474   21.0    7.054702
Gasoline or E85             17.572385   17.0    3.822538
Gasoline or natural gas     15.350000   12.0    5.343712
Gasoline or propane         13.500000   13.5    1.603567
...     ...     ...     ...
Premium and Electricity     26.300000   25.5    5.141165
Premium or E85              20.090909   20.0    3.676502
Regular                     20.144698   20.0    5.317500
Regular Gas and Electricity 41.937500   38.5    5.246824
Regular Gas or Electricity  42.000000   42.0    0.000000

13 rows Г— 3 columns
```

Custom Aggregation Functions

We do not have to be limited by the range of standard aggregation functions. If the need arises, we can write our own aggregation function.

For example, in our vehicle dataset, we might want to find out for each level of transmission, what is the most common vehicle class. In other words, we would like to find the mode.

We can write our own implementation of the mode function, but it would be more efficient to use the scipy implementation of this function. Scipy is a Python package for scientific computing.

Let us first define our custom function using the scipy mode function. We create a custom function since the mode function returns a tuple with the mode and the frequency of the mode. We are only interested in the first part of the tuple.

```
from scipy import stats

def agg_mode(x):
    return(stats.mode(x)[0])
```

Now we can use our custom aggregation function using the `agg` function:

```
vehicles.groupby("Transmission")["Vehicle Class"].agg(agg_mode)
Transmission
Auto (AV)           Compact Cars
Auto (AV-S6)        Compact Cars
Auto (AV-S8)        Midsize Cars
Auto(A1)         Subcompact Cars
Auto(AM-S6)         Compact Cars
                      ...
Manual 5 spd                Vans
Manual 5-spd        Compact Cars
Manual 6-spd        Compact Cars
Manual 7-spd    Minicompact Cars
Manual(M7)           Two Seaters
Name: Vehicle Class, Length: 45, dtype: object
```

Summary

In this lesson we learned how to summarization and aggregation with DataFrames. We learned to use the standard aggregation functions and how to make custom aggregation functions.