Solution Review: Group By Aggregations

This lesson provides the solution to the previous challenge.

WE'LL COVER THE FOLLOWING

- Group by aggregations

Group by aggregations

```
import pandas as pd

# Loading dataset
def read_csv():
    # Define the column names as a list
    names = ["mpg", "cylinders", "displacement", "horsepower", "weight", "acceleration", "mod
    # Read in the CSV file from the webpage using the defined column names
    df = pd.read_csv("auto-mpg.data", header=None, names=names, delim_whitespace=True)
    return df

# Describing data
def group_aggregation(df, group_var, agg_var):

    # Grouping the data and taking mean
    grouped_df = df.groupby([group_var])[agg_var].mean()
    return grouped_df

# Calling the function
print(group_aggregation(read_csv(),"cylinders","mpg"))
```

According to the problem statement, we need to group the **Auto MPG Dataset** on the basis of a column. Then we have to calculate the mean of the grouped data according to another column. Before doing it, we have to read the data first. There is no need to explain how to read the data, as we studied that in detail previously. Dataset is read from **line 4** to **line 9**.

Moving towards the main implementation, look at the header of the `group_aggregation(df, group_var, agg_var)` function. It takes *three* arguments

`group_aggregation(df,group_var,agg_var)` function. It takes *three* arguments as input:

- `df`: A dataframe containing the dataset in the form of a matrix.
- `group_var`: A variable that will group the data
- `agg_var`: A variable to aggregate or describe the grouped data with statistics

Line 15 is the most important line. We are using a built-in function `groupby()` on `df` which takes *one* arguments: `group_var`. It will return the data grouped according to `group_var` column. Next, we are calculating the `mean` of the grouped data with the `agg_var` column. Then at **line 16**, we are returning the result.

At **line 19** we are calling the function `group_aggregation(read_csv(),"cylinders","mpg"))`. First control will transfer to `read_csv()` at **line 4** and we'll get a dataframe.

The next argument is `cylinders`. According to the dataset we have *five* different values for `cylinders`: **3**,**4**,**5**,**6**, and **8**. So the dataframe `df` will be grouped into *five* groups, each group represented by a different cylinder count. In simple words, all cars having **3** cylinders will form one group. Similarly, all cars having **4** cylinders will form another group, and so on.

The last argument is `mpg`. According to the dataset, `mpg` holds continuous values. According to implementation, the *mean* of `mpg` for all the cars (of the same group) will be returned and printed.

According to the result, you will notice:

- The average `mpg` of cars having **3** `cylinders` in total is **20.550000**.
- The average `mpg` of cars having **4** `cylinders` in total is **29.286765**.
- The average `mpg` of cars having **5** `cylinders` in total is **27.366667**.
- The average `mpg` of cars having **6** `cylinders` in total is **19.985714**.
- The average `mpg` of cars having **8** `cylinders` in total is **14.963107**.

That's it about analyzing the dataset with different statistical techniques using Pandas. The next chapter explains how to clean a dataset.