SUMMER ANALYTICS 2024

Week-1 Assignment

Data Grand Prix!

Welcome to your first assignment of Summer Analytics 2025! We hope you are excited to implement and test everything you have learnt up until now. The dataset which you'll use includes information about cars.

We've got an interesting set of questions for you to get a basic understanding of pandas and data visualization libraries. GOOD LUCK!

Let's get started with importing numpy, pandas, seaborn and matplotlib!

Note - matplotlib should be imported with the command :

```
import matplotlib.pyplot as plt
```

So lets get started!! Buckle up your belts for this exciting ride!!

1) Start by importing all important libraries

For eg, "import numpy as np"

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

2) Read the csv file and assign it to a variable .

```
df = pd.read_csv('Cars.csv')
```

3) Display shape of dataframe

Expected Output - (398, 9)

```
print("Shape of dataframe:", df.shape)
```

```
Shape of dataframe: (398, 9)
```

4) Print all columns of dataframe

Return an array containing names of all the columns.

```
print("Columns in dataframe:", df.columns.values)
```

```
Columns in dataframe: ['mpg' 'cylinders' 'displacement' 'horsepower' 'weight' 'acceleration'
 'model_year' 'origin' 'name']
```

6) Set the 'name' column as the index of dataframe

```
df.set_index('name', inplace=True)
```

7) Print a list of all the unique mpg values

```
print("Unique MPG values:", df['mpg'].unique())
```

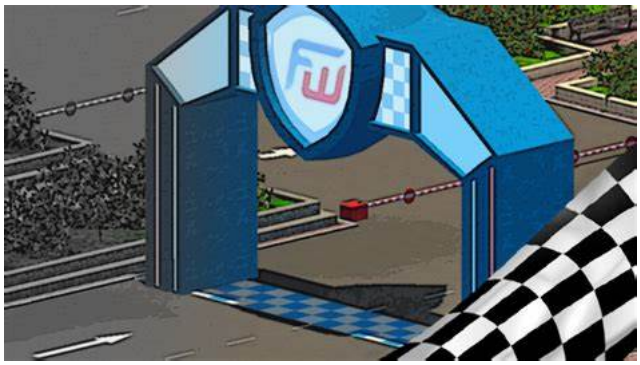
```
Unique MPG values: [18.  15.  16.  17.  14.  24.  22.  21.  27.  26.  25.  10.  11.   9.
 28.  19.  12.  13.  23.  30.  31.  35.  20.  29.  32.  33.  17.5 15.5
 14.5 22.5 24.5 18.5 29.5 26.5 16.5 31.5 36.  25.5 33.5 20.5 30.5 21.5
 43.1 36.1 32.8 39.4 19.9 19.4 20.2 19.2 25.1 20.6 20.8 18.6 18.1 17.7
 27.5 27.2 30.9 21.1 23.2 23.8 23.9 20.3 21.6 16.2 19.8 22.3 17.6 18.2
 16.9 31.9 34.1 35.7 27.4 25.4 34.2 34.5 31.8 37.3 28.4 28.8 26.8 41.5
 38.1 32.1 37.2 26.4 24.3 19.1 34.3 29.8 31.3 37.  32.2 46.6 27.9 40.8
 44.3 43.4 36.4 44.6 40.9 33.8 32.7 23.7 23.6 32.4 26.6 25.8 23.5 39.1
 39.  35.1 32.3 37.7 34.7 34.4 29.9 33.7 32.9 31.6 28.1 30.7 24.2 22.4
 34.  38.  44. ]
```

8) Create a column which contains the horsepower divided by weightas its metric and make this new column the index.

```
df['hp_per_weight'] = df['horsepower'] / df['weight']
df.set_index('hp_per_weight', inplace=True)
print("\nPreview of the updated dataframe:")
print(df.head())
```

```
Preview of the updated dataframe:
                mpg  cylinders  displacement  horsepower  weight  \
hp_per_weight
0.037100       18.0          8         307.0       130.0    3504
0.044679       15.0          8         350.0       165.0    3693
0.043655       18.0          8         318.0       150.0    3436
0.043694       16.0          8         304.0       150.0    3433
0.040591       17.0          8         302.0       140.0    3449

                acceleration  model_year origin
hp_per_weight
0.037100                12.0          70    usa
0.044679                11.5          70    usa
0.043655                11.0          70    usa
0.043694                12.0          70    usa
0.040591                10.5          70    usa
```



Checkpoint!! Congratulations on making it this far. You are really keeping up in Data Grand Prix. Now starts the real race i.e. graded questions of the quiz.

GRADED Questions (To be answered in the quiz)

Try to retrieve some information from the data and answer the questions below . BEST OF LUCK !!

1. What is name of car that has the highest horsepower?

```
import pandas as pd
import numpy as np
df = pd.read_csv('Cars.csv')
df['horsepower'] = pd.to_numeric(df['horsepower'], errors='coerce')
max_hp = df['horsepower'].max()
```

```
car_with_max_hp = df[df['horsepower'] == max_hp]['name'].values
print("Car(s) with highest horsepower:", car_with_max_hp)
```

```
Car(s) with highest horsepower: ['pontiac grand prix']
```

2. How many cars have mpg ≥ 35?

```
num_cars_mpg_35 = (df['mpg'] >= 35).sum()
print("Number of cars with mpg ≥ 35:", num_cars_mpg_35)
```

```
Number of cars with mpg ≥ 35: 36
```

3. What is the most common origin for cars with horsepower > 100 and weight < 3000?

```
filtered = df[(df['horsepower'] > 100) & (df['weight'] < 3000)]
most_common_origin = filtered['origin'].mode().iloc[0]
print("Most common origin:", most_common_origin)
```

```
Most common origin: usa
```

4. What is the mean acceleration of cars from Japan? (rounded to 2 decimals)

```
mean_acc_japan = round(df[df['origin'] == 'japan']['acceleration'].mean(), 2)
print("Mean acceleration (Japan):", mean_acc_japan)
```

```
Mean acceleration (Japan): 16.17
```

5. Which year had the highest average mpg?

```
year_highest_avg_mpg = df.groupby('model_year')['mpg'].mean().idxmax()
print("Year with highest average mpg:", year_highest_avg_mpg)
```

```
Year with highest average mpg: 80
```

Congratulations on coming this far! Since we were having so much fun playing with this dataset, let's move towards finish line by attempting some Ungraded questions!

Note: These questions are UNGRADED, and are given as an extra exercise.

Find the car (or cars) with the best ratio of horsepower to weight among all cars that also have above-median mpg.

```
median_mpg = df['mpg'].median()
above_median = df[df['mpg'] > median_mpg].copy()
above_median['hp_to_weight'] = above_median['horsepower'] / above_median['weight']
best_ratio = above_median['hp_to_weight'].max()
best_cars = above_median[above_median['hp_to_weight'] == best_ratio]['name'].values
print("Car(s) with best hp/weight ratio (above-median mpg):", best_cars)
```

```
Car(s) with best hp/weight ratio (above-median mpg): ['bmw 2002']
```

Design a multi-line plot using Matplotlib or Seaborn that shows the evolution of average mpg over the years, separately for each origin

```
import matplotlib.pyplot as plt
import seaborn as sns
avg_mpg = df.groupby(['model_year', 'origin'])['mpg'].mean().reset_index()
plt.figure(figsize=(10,6))
sns.lineplot(data=avg_mpg, x='model_year', y='mpg', hue='origin', marker='o')
plt.title('Average MPG over Years by Origin')
plt.xlabel('Model Year')
plt.ylabel('Average MPG')
plt.legend(title='Origin')
plt.show()
```

Create a Seaborn scatterplot (or PairGrid) where:

X = horsepower

Y = weight

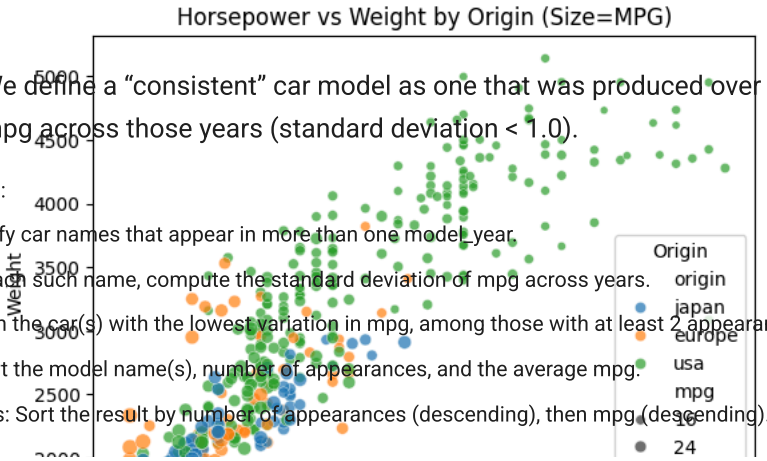
Color by: origin

Size by: mpg

Hue order = ['japan', 'europe', 'usa']

Add meaningful plot titles and axis titles.

```
sns.scatterplot(
    data=df,
    x='horsepower',
    y='weight',
    hue='origin',
    size='mpg',
    hue_order=['japan', 'europe', 'usa'],
    alpha=0.7
)
plt.title('Horsepower vs Weight by Origin (Size=MPG)')
plt.xlabel('Horsepower')
plt.ylabel('Weight')
plt.legend(title='Origin')
plt.show()
```



We define a “consistent” car model as one that was produced over multiple years and had very low variation in mpg across those years (standard deviation < 1.0).

Tasks:

Identify car names that appear in more than one model_year.

For each such name, compute the standard deviation of mpg across years.

Return the car(s) with the lowest variation in mpg, among those with at least 2 appearances and std(mpg) < 1.0.

Report the model name(s), number of appearances, and the average mpg.

Bonus: Sort the result by number of appearances (descending), then mpg (descending).

```
grouped = df.groupby('name').agg(
    num_years=('model_year', 'nunique'),
    mpg_std=('mpg', 'std'),
    mpg_mean=('mpg', 'mean'),
    count=('name', 'count')
).reset_index()
consistent = grouped[(grouped['num_years'] > 1) & (grouped['mpg_std'] < 1.0)]
consistent_sorted = consistent.sort_values(['num_years', 'mpg_mean'], ascending=[False, False])
print(consistent_sorted[['name', 'num_years', 'mpg_mean']])
```

```
                          name  num_years   mpg_mean
141           ford galaxie 500          3  14.333333
223          plymouth fury iii          3  14.333333
267        toyota corolla 1200          2  31.500000
175                  mazda 626          2  31.450000
287          volkswagen rabbit          2  29.250000
95                datsun pl510          2  27.000000
260                  saab 99le          2  24.500000
276             toyota mark ii          2  19.500000
98                 dodge aspen          2  18.850000
49   chevrolet chevelle malibu          2  17.500000
11             amc matador (sw)          2  14.500000
143      ford gran torino (sw)          2  13.500000
148                   ford ltd          2  13.500000
```