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"

In [6]:
```
import numpy as np
import pandas as pd
```

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

In [8]:
```
df = pd.read_csv("Cars.csv")
df.head()
```

Out[8]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	name
0	18.0	8	307.0	130.0	3504	12.0	70	usa	chevrolet chevelle malibu
1	15.0	8	350.0	165.0	3693	11.5	70	usa	buick skylark 320
2	18.0	8	318.0	150.0	3436	11.0	70	usa	plymouth satellite
3	16.0	8	304.0	150.0	3433	12.0	70	usa	amc rebel sst
4	17.0	8	302.0	140.0	3449	10.5	70	usa	ford torino

3) Display shape of dataframe

Expected Output - (398, 9)

In [10]:
```
df.shape
```

Out[10]:
```
(398, 9)
```

4) Print all columns of dataframe

Return an array containing names of all the columns.

In [12]:
```
print(df.columns)
list(df.columns)
```

```
Index(['mpg', 'cylinders', 'displacement', 'horsepower', 'weight',
       'acceleration', 'model_year', 'origin', 'name'],
      dtype='object')
```

Out[12]:
```
['mpg',
 'cylinders',
 'displacement',
 'horsepower',
 'weight',
 'acceleration',
 'model_year',
 'origin',
 'name']
```

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

In [14]:
```
df.set_index('name', inplace=True)
df.head()
```

Out[14]:

name	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin
chevrolet chevelle malibu	18.0	8	307.0	130.0	3504	12.0	70	usa
buick skylark 320	15.0	8	350.0	165.0	3693	11.5	70	usa
plymouth satellite	18.0	8	318.0	150.0	3436	11.0	70	usa
amc rebel sst	16.0	8	304.0	150.0	3433	12.0	70	usa
ford torino	17.0	8	302.0	140.0	3449	10.5	70	usa

7) Print a list of all the unique mpg values

In [16]:
```
import pandas as pd
df = pd.read_csv('Cars.csv')
unique_mpg = df['mpg'].drop_duplicates()
print(list(unique_mpg))
```

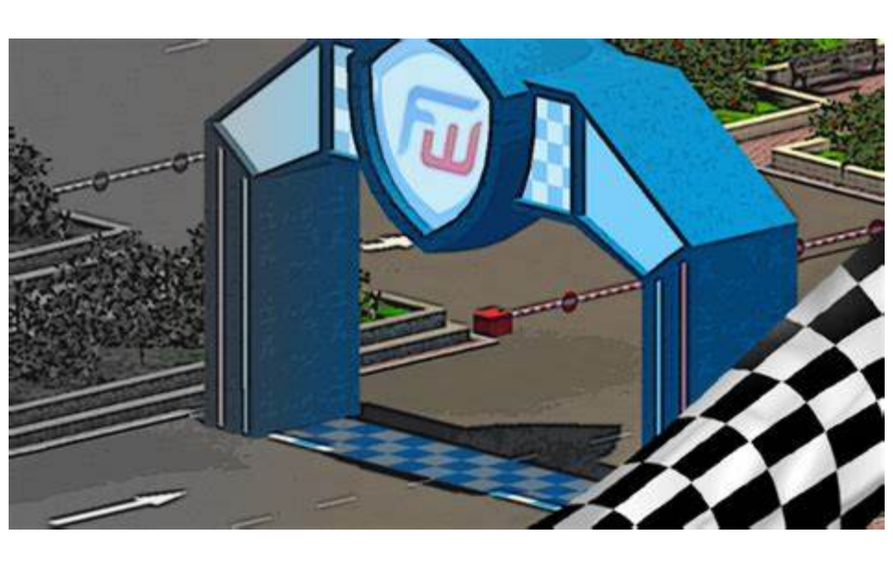
```
[18.0, 15.0, 16.0, 17.0, 14.0, 24.0, 22.0, 21.0, 27.0, 26.0, 25.0, 10.0, 11.0, 9.0, 28.0, 19.0, 12.0, 13.0, 23.0, 30.0, 31.0, 35.0, 20.0, 29.0, 32.0, 33.0, 17.5, 15.5, 14.5, 22.5, 24.5, 18.5, 29.5, 26.5, 16.5, 31.5, 36.0, 25.5, 33.5, 20.5, 30.5, 21.5, 43.1, 36.1,
32.8, 39.4, 19.9, 19.4, 20.2, 19.2, 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, 37.0, 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, 39.0, 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, 34.0, 38.0, 44.0]
```

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

In [18]:
```
df['hp_per_weight'] = df['horsepower'] / df['weight']
df.set_index('hp_per_weight', inplace=True)
df.head()
```

Out[18]:

hp_per_weight	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	name
0.037100	18.0	8	307.0	130.0	3504	12.0	70	usa	chevrolet chevelle malibu
0.044679	15.0	8	350.0	165.0	3693	11.5	70	usa	buick skylark 320
0.043655	18.0	8	318.0	150.0	3436	11.0	70	usa	plymouth satellite
0.043694	16.0	8	304.0	150.0	3433	12.0	70	usa	amc rebel sst
0.040591	17.0	8	302.0	140.0	3449	10.5	70	usa	ford torino



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?

In [21]:
```
df.reset_index(inplace=True)

df['horsepower'] = pd.to_numeric(df['horsepower'], errors='coerce')

max_hp_row = df.loc[df['horsepower'].idxmax()]

print("Car with highest horsepower:", max_hp_row['name'])
```

```
Car with highest horsepower: pontiac grand prix
```

2. How many cars have mpg ≥ 35?

In [23]:
```
count = df[df['mpg'] >= 35].shape[0]

print("Number of cars with mpg ≥ 35:", count)
```

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

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

In [25]:
```
df['horsepower'] = pd.to_numeric(df['horsepower'], errors='coerce')
filtered_df = df[(df['horsepower'] > 100) & (df['weight'] < 3000)]
filtered_df = df[(df['horsepower'] > 100) & (df['weight'] < 3000)]
most_common_origin = filtered_df['origin'].value_counts().idxmax()
print("Most common origin for cars with horsepower > 100 and weight < 3000:", most_common_origin)
```

```
Most common origin for cars with horsepower > 100 and weight < 3000: usa
```

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

In [27]:
```
japan_cars = df[df['origin'] == 'japan']
mean_acceleration = round(japan_cars['acceleration'].mean(), 2)
print("Mean acceleration of cars from Japan:", mean_acceleration)
```

```
Mean acceleration of cars from Japan: 16.17
```

5. Which year had the highest average mpg?

In [37]:
```
print(df.columns)
avg_mpg_by_year = df.groupby('model_year')['mpg'].mean()
year_with_highest_mpg = avg_mpg_by_year.idxmax()
print("Year with the highest average mpg:", year_with_highest_mpg)
```

```
Index(['hp_per_weight', 'mpg', 'cylinders', 'displacement', 'horsepower',
       'weight', 'acceleration', 'model_year', 'origin', 'name'],
      dtype='object')
Year with the 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.

In [39]:
```
median_mpg = df['mpg'].median()

high_mpg_cars = df[df['mpg'] > median_mpg].copy()

high_mpg_cars['hp_to_weight'] = high_mpg_cars['horsepower'] / high_mpg_cars['weight']

max_ratio = high_mpg_cars['hp_to_weight'].max()

best_cars = high_mpg_cars[high_mpg_cars['hp_to_weight'] == max_ratio]

print(best_cars[['horsepower', 'weight', 'hp_to_weight']])
```

```
    horsepower  weight  hp_to_weight
23       113.0    2234      0.050582
```

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

In [7]:
```
import pandas as pd

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

In [11]:
```
df = pd.read_csv('Cars.csv', index_col='name')
```

In [19]:
```
print(df.columns)
import matplotlib.pyplot as plt
import seaborn as sns


avg_mpg = df.groupby(['model_year', 'origin'])['mpg'].mean().reset_index()
```

```
Index(['mpg', 'cylinders', 'displacement', 'horsepower', 'weight',
       'acceleration', 'model_year', 'origin'],
      dtype='object')
```

In [15]:
```
print(df.columns)
```

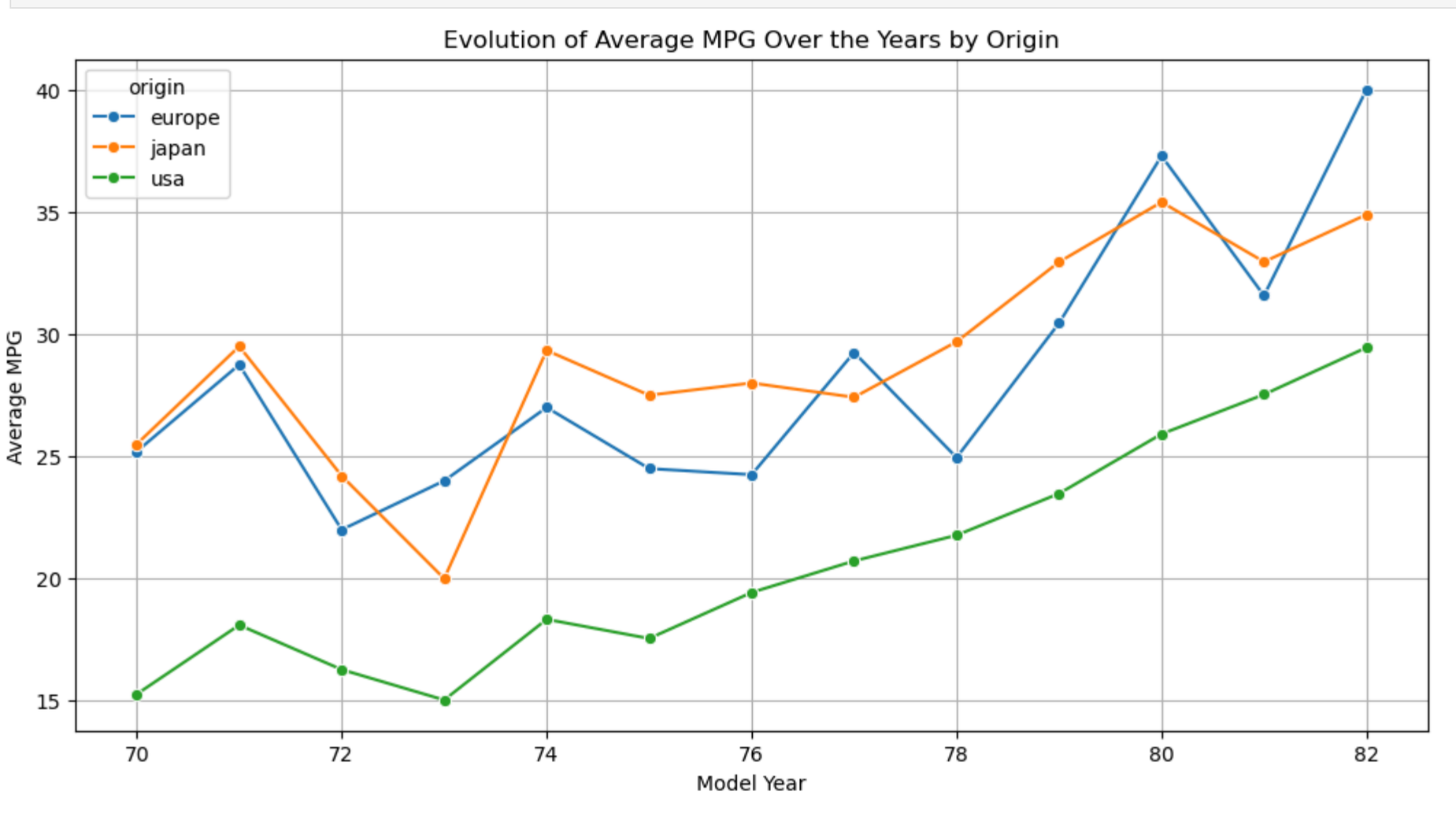
```
Index(['mpg', 'cylinders', 'displacement', 'horsepower', 'weight',
       'acceleration', 'model_year', 'origin'],
      dtype='object')
```

In [17]:
```
avg_mpg = df.groupby(['model_year', 'origin'])['mpg'].mean().reset_index()

import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(12, 6))
sns.lineplot(data=avg_mpg, x='model_year', y='mpg', hue='origin', marker='o')

plt.title('Evolution of Average MPG Over the Years by Origin')
plt.xlabel('Model Year')
plt.ylabel('Average MPG')
plt.grid(True)
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.

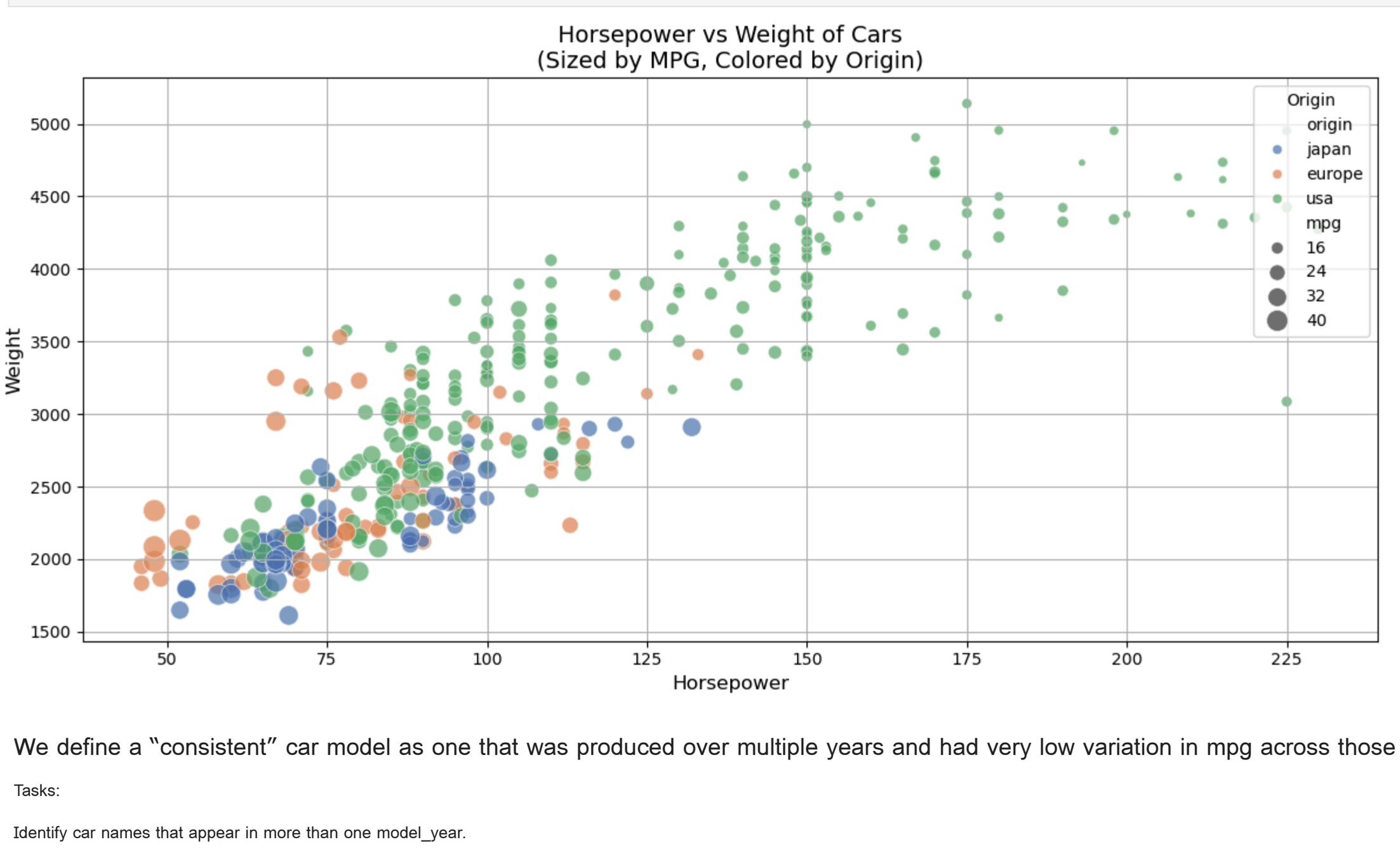
In [5]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [7]:
```
df = pd.read_csv('Cars.csv')
```

In [9]:
```
# Convert origin values to lowercase (just in case)
df['origin'] = df['origin'].str.lower()

# Create scatter plot
plt.figure(figsize=(12, 6))
sns.scatterplot(
    data=df,
    x='horsepower',
    y='weight',
    hue='origin',
    size='mpg',
    palette='deep',
    hue_order=['japan', 'europe', 'usa'],
    sizes=(20, 200),
    alpha=0.7
)

# Add titles
plt.title('Horsepower vs Weight of Cars\n(Sized by MPG, Colored by Origin)', fontsize=14)
plt.xlabel('Horsepower', fontsize=12)
plt.ylabel('Weight', fontsize=12)
plt.grid(True)
plt.tight_layout()
plt.legend(title='Origin', loc='upper right')
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).

In [15]:
```
import pandas as pd

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

print(df.columns.tolist())
```

```
['mpg', 'cylinders', 'displacement', 'horsepower', 'weight', 'acceleration', 'model_year', 'origin', 'name']
```

In [19]:
```
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_')
```

In [21]:
```
df['model_year']
```

Out[21]:
```
0      70
1      70
2      70
3      70
4      70
       ..
393    82
394    82
395    82
396    82
397    82
Name: model_year, Length: 398, dtype: int64
```

In [23]:
```
import pandas as pd

# Load data
df = pd.read_csv('Cars.csv')

# Normalize column names
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_')

# Step 1: Find car names appearing in multiple model years
name_years = df.groupby('name')['model_year'].nunique()
repeated_names = name_years[name_years > 1].index

# Step 2: Filter only those cars
filtered_df = df[df['name'].isin(repeated_names)]

# Step 3: Compute statistics
stats = filtered_df.groupby('name').agg(
    mpg_std=('mpg', 'std'),
    appearances=('mpg', 'count'),
    avg_mpg=('mpg', 'mean')
).reset_index()

# Step 4: Filter those with very low variation
consistent_cars = stats[stats['mpg_std'] < 1.0]

# Step 5: Sort by appearances (desc), then avg_mpg (desc)
consistent_sorted = consistent_cars.sort_values(
    by=['appearances', 'avg_mpg'], ascending=[False, False]
)

# Display final result
print(consistent_sorted[['name', 'appearances', 'avg_mpg']])
```

```
                         name  appearances    avg_mpg
26           ford galaxie 500            3  14.333333
41           plymouth fury iii           3  14.333333
49         toyota corolla 1200           2  31.500000
35                  mazda 626            2  31.450000
53            volkswagen rabbit          2  29.250000
21                datsun pl510           2  27.000000
45                  saab 99le            2  24.500000
51              toyota mark ii           2  19.500000
22                dodge aspen            2  18.850000
10  chevrolet chevelle malibu            2  17.500000
4             amc matador (sw)           2  14.500000
28        ford gran torino (sw)          2  13.500000
29                   ford ltd            2  13.500000
```

Congratulations on completing the race. Kudos to you. Looking forward to meet you in real week.