Sports_car_prices:data_analysis

April 10, 2023

1 Project Title - Sports_cars Data Analysis

This project is a data analysis of Sports_cars_price_df. Using Data Analysis concept to anylsis this dataset, Data Preparation and Cleaning.

I'm very Gratitude to Data Analysis with Python: Zero to Pandas. I have learnt this cousre from this website.

[1]:
```
!pip install jovian opendatasets --upgrade --quiet
```

Let's begin by downloading the data, and listing the files within the dataset.

[2]:
```
# Change this
dataset_url = 'https://www.kaggle.com/datasets/rkiattisak/
 ↪sports-car-prices-dataset'
```

[3]:
```
import opendatasets as od
od.download(dataset_url)
```

```
Please provide your Kaggle credentials to download this dataset. Learn more:
http://bit.ly/kaggle-creds
Your Kaggle username: siddivinayakay
Your Kaggle Key: ·····
Downloading sports-car-prices-dataset.zip to ./sports-car-prices-dataset
```

```
100%|      | 8.36k/8.36k [00:00<00:00, 5.82MB/s]
```

The dataset has been downloaded and extracted.

[4]:
```
# Change this
data_dir = './sports-car-prices-dataset/'
```

[5]:
```
import os
os.listdir(data_dir)
```

[5]:
```
['Sport car price.csv']
```

Let us save and upload our work to Jovian before continuing.

[6]:
```
project_name = "sport-cars-prices-data-analysis-project"
```

[7]:
```
!pip install jovian --upgrade -q
```

[8]:
```
import jovian
```

[9]:
```
jovian.commit(project=project_name)
```

```
<IPython.core.display.Javascript object>
```

```
[jovian] Updating notebook "ysiddivinayaka/sport-cars-prices-data-analysis-
project" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/ysiddivinayaka/sport-cars-
prices-data-analysis-project
```

[9]:
```
'https://jovian.com/ysiddivinayaka/sport-cars-prices-data-analysis-project'
```

1.1 Data Preparation and Cleaning

TODO - Data preparation and cleaning are important steps in the data analysis process that involve transforming raw data into a clean and organized format that can be used for analysis. The goal of data preparation and cleaning is to ensure that the data is accurate, complete, consistent, and in a format that can be easily analyzed.

Here are some common steps involved in data preparation and cleaning: * Data Collection * Data Cleaning * Data Transformation * Data Integration * Data Reduction * Data Normalization

[10]:
```
import pandas as pd
```

[11]:
```
sport_cars_price_df = pd.read_csv('sports-car-prices-dataset/Sport car price.
↪csv')
```

[12]:
```
sport_cars_price_df
```

[12]:

	Car Make	Car Model	Year	Engine Size (L)	Horsepower	Torque (lb-ft)	0-60 MPH Time (seconds)	Price (in USD)
0	Porsche	911	2022	3	379	331		
1	Lamborghini	Huracan	2021	5.2	630	443		
2	Ferrari	488 GTB	2022	3.9	661	561		
3	Audi	R8	2022	5.2	562	406		
4	McLaren	720S	2021	4	710	568		
…	…	…	…	…	…	…		
1002	Koenigsegg	Jesko	2022	5	1280	1106		
1003	Lotus	Evija	2021	Electric Motor	1972	1254		
1004	McLaren	Senna	2021	4	789	590		
1005	Pagani	Huayra	2021	6	764	738		
1006	Rimac	Nevera	2021	Electric Motor	1888	1696		

```
0                          4        101,200
1                        2.8        274,390
2                          3        333,750
3                        3.2        142,700
4                        2.7        298,000
…                        …            …
1002                     2.5      3,000,000
1003                       2      2,000,000
1004                     2.7      1,000,000
1005                       3      2,600,000
1006                    1.85      2,400,000

[1007 rows x 8 columns]
```

[13]:
```
sport_cars_price_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1007 entries, 0 to 1006
Data columns (total 8 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Car Make                 1007 non-null   object
 1   Car Model                1007 non-null   object
 2   Year                     1007 non-null   int64
 3   Engine Size (L)          997 non-null    object
 4   Horsepower               1007 non-null   object
 5   Torque (lb-ft)           1004 non-null   object
 6   0-60 MPH Time (seconds)  1007 non-null   object
 7   Price (in USD)           1007 non-null   object
dtypes: int64(1), object(7)
memory usage: 63.1+ KB
```

[14]:
```
type(sport_cars_price_df)
```

[14]:
```
pandas.core.frame.DataFrame
```

[15]:
```
sport_cars_price_df.describe()
```

[15]:
```
              Year
count  1007.000000
mean   2021.201589
std       2.019802
min    1965.000000
25%    2021.000000
50%    2021.000000
75%    2022.000000
max    2023.000000
```

[16]:
```
sport_cars_price_df.shape
```

[16]:
```
(1007, 8)
```

[17]:
```
print(sport_cars_price_df.isnull().sum())
```

```
Car Make                   0
Car Model                  0
Year                       0
Engine Size (L)           10
Horsepower                 0
Torque (lb-ft)             3
0-60 MPH Time (seconds)    0
Price (in USD)             0
dtype: int64
```

[18]:
```
sport_cars_price_df.columns
```

[18]:
```
Index(['Car Make', 'Car Model', 'Year', 'Engine Size (L)', 'Horsepower',
       'Torque (lb-ft)', '0-60 MPH Time (seconds)', 'Price (in USD)'],
      dtype='object')
```

[19]:
```
sport_cars_price_df.loc[844]
```

[19]:
```
Car Make                  Lamborghini
Car Model                   Aventador
Year                             2021
Engine Size (L)                   6.5
Horsepower                        730
Torque (lb-ft)                    509
0-60 MPH Time (seconds)           2.9
Price (in USD)                517,000
Name: 844, dtype: object
```

[20]:
```
sport_cars_price_df.at[247, 'Car Make']
```

[20]:
```
'Tesla'
```

[21]:
```
sport_cars_price_df.sample(20)
```

[21]:

	Car Make	Car Model	Year	Engine Size (L)	Horsepower	\
431	Ford	Mustang Mach 1	2021	5	480	
479	Jaguar	F-Type	2022	3	380	
308	Audi	RS 7	2022	4	591	
737	McLaren	GT	2022	4	620	
471	Jaguar	F-Type R	2021	5	575	
218	Mercedes-Benz	S63 AMG	2021	4	603	

868	Mercedes-Benz	AMG GT R	2021	4	577
33	Mercedes-Benz	SLS AMG	2015	6.2	622
106	McLaren	GT	2021	4	612
109	Chevrolet	Camaro ZL1	2021	6.2	650
37	Porsche	Taycan 4S	2022	Electric Motor	562
542	McLaren	Senna	2020	4	789
235	Porsche	Cayman GT4	2021	4	414
823	Bugatti	Chiron Super Sport 300+	2021	8	1578
305	Porsche	Taycan Turbo S	2021	Electric	750
665	Lamborghini	Aventador	2021	6.5	770
968	McLaren	Artura	2022	3	671
934	Maserati	GranTurismo	2021	4.7	454
89	Lexus	LC 500	2022	5	471
512	Toyota	GR Supra	2022	3	382

	Torque (lb-ft)	0-60 MPH Time (seconds)	Price (in USD)
431	420	4	51,000
479	339	4.9	70,900
308	590	3.5	114,000
737	465	3.1	211,300
471	516	3.5	105,900
218	664	3.4	152,500
868	516	3.5	183,000
33	468	3.2	222,000
106	465	3.1	212,500
109	650	3.5	64,000
37	479	3.8	104,000
542	590	2.7	1,500,000
235	309	4.2	100,200
823	1180	2.4	5,200,000
305	774	2.6	185,000
665	531	2.8	417,826
968	531	3	225,000
934	384	4.7	150,000
89	398	4.4	92,950
512	368	3.9	43,090

[22]:
```
sport_cars_price_df['Year']
```

[22]:
```
0       2022
1       2021
2       2022
3       2022
4       2021
       ...
1002    2022
1003    2021
```

```
1004    2021
1005    2021
1006    2021
Name: Year, Length: 1007, dtype: int64
```

[23]:
```
sport_cars_price_df.sort_values('Year', ascending = False).head(25)
```

[23]:

	Car Make	Car Model	Year	Engine Size (L)	Horsepower	Torque (lb-ft)	0-60 MPH Time (seconds)	Price (in USD)
567	Chevrolet	Corvette Z06	2023	5.5	625	650	2.6	85,000
638	Nissan	400Z	2023	3	400	350	4	40,000
364	Tesla	Roadster	2023	Electric	1,000+	737	< 1.9	200,000
0	Porsche	911	2022	3	379	331	4	101,200
359	Jaguar	F-Type R	2022	5	575	516	3.5	103,200
770	Porsche	718 Boxster	2022	2	300	280	4.9	63,000
376	Bugatti	Chiron	2022	8	1500	1180	2.4	3,000,000
772	Ferrari	SF90 Stradale	2022	4	986	590	2.5	625,000
773	Audi	TT RS Coupe	2022	2.5	394	354	3.5	68,000
775	BMW	Z4 Roadster	2022	2	255	295	5.2	50,000
777	Chevrolet	Camaro SS Convertible	2022	6.2	455	455	4	49,000
371	Jaguar	F-Type	2022	3	296	295	5.4	61,600
781	Bentley	Continental GT	2022	6	626			
783	Dodge	Challenger SRT Hellcat	2022	6.2	717			
784	Jaguar	F-Type	2022	2	296			
786	Maserati	GranTurismo	2022	4.7	454			
355	Acura	NSX	2022	3.5	573			
379	Koenigsegg	Jesko	2022	5	1280			
354	Tesla	Roadster	2022	Electric	1000+			
792	Audi	RS7	2022	4	591			
352	Rimac	Nevera	2022	Electric	1914			
351	Porsche	Cayman GT4	2022	4	414			
349	Mercedes-Benz	AMG C63	2022	4	503			
797	Porsche	718 Cayman GT4	2022	4	414			
344	Koenigsegg	Jesko	2022	5	1280			

```
781            664                    3.3        225,000
783            656                    3.5         68,000
784            295                    5.4         63,700
786            384                    4.7        150,980
355            476                    2.7        157,500
379           1015                    2.5      2,800,000
354         10,000+                    1.9        200,000
792            590                    3.5        117,000
352           1696                   1.85      2,400,000
351            309                    4.2        102,900
349            516                    3.8         69,900
797            309                    4.2        102,000
344           1106                    2.5      2,800,000
```

[50]:
```
cars_2021_df = sport_cars_price_df[sport_cars_price_df.Year == 2021]

cars_df = cars_2021_df[['Car Make',  'Price (in USD)']]
```

[51]:
```
cars_df
```

[51]:
```
           Car Make Price (in USD)
1       Lamborghini        274,390
4           McLaren        298,000
6     Mercedes-Benz        118,500
7         Chevrolet         59,900
9            Nissan        212,000
…               …              …
1001        Bugatti      3,000,000
1003          Lotus      2,000,000
1004        McLaren      1,000,000
1005         Pagani      2,600,000
1006          Rimac      2,400,000

[576 rows x 2 columns]
```

[28]:
```
import jovian
```

[29]:
```
jovian.commit()
```

```
<IPython.core.display.Javascript object>

[jovian] Updating notebook "ysiddivinayaka/sport-cars-prices-data-analysis-
project" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/ysiddivinayaka/sport-cars-
prices-data-analysis-project
```

[29]:
```
'https://jovian.com/ysiddivinayaka/sport-cars-prices-data-analysis-project'
```

1.2 Exploratory Analysis and Visualization

TODO - Exploratory data analysis (EDA) is an approach to analyzing and summarizing data sets in order to gain insights into the data and formulate hypotheses. Data visualization is a key component of EDA, as it allows analysts to see patterns and relationships in the data that may not be immediately obvious from numerical summaries or statistical models.

Data visualization is the graphic representation of data. It involves producing images that communicate relationships among the represented data to viewers. Visualizing data is an essential part of data analysis and machine learning. We'll use Python libraries Matplotlib and Seaborn to learn and apply some popular data visualization techniques. We'll use the words chart, plot, and graph interchangeably in this tutorial.

To begin, let's install and import the libraries. We'll use the matplotlib.pyplot module for basic plots like line & bar charts. It is often imported with the alias plt. We'll use the seaborn module for more advanced plots. It is commonly imported with the alias sns.

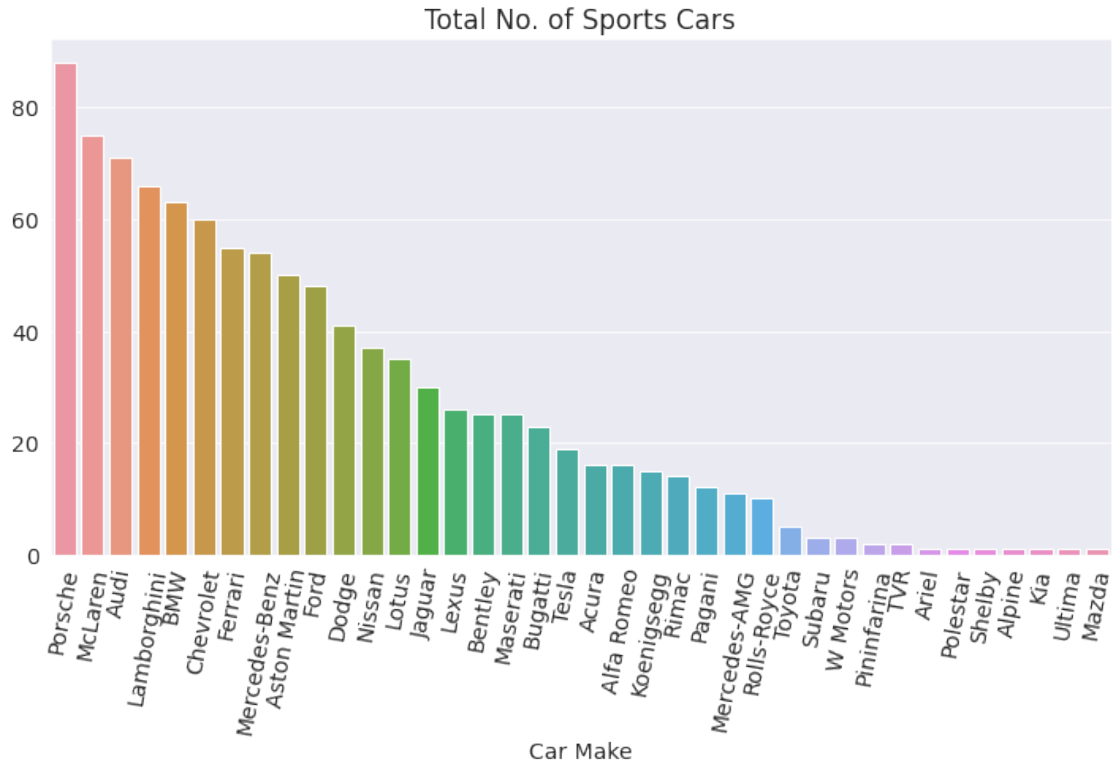
Let's begin by importing`matplotlib.pyplot` and `seaborn`.

[30]:
```
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline

sns.set_style('darkgrid')
matplotlib.rcParams['font.size'] = 14
matplotlib.rcParams['figure.figsize'] = (9, 5)
matplotlib.rcParams['figure.facecolor'] = '#00000000'
```

TODO - Explore one or more columns by plotting a graph below, and add some explanation about it

Using 'value_count' to get specific column series in a dataset. Using barplot plot graph to get 'Total No. of Sports Cars'.

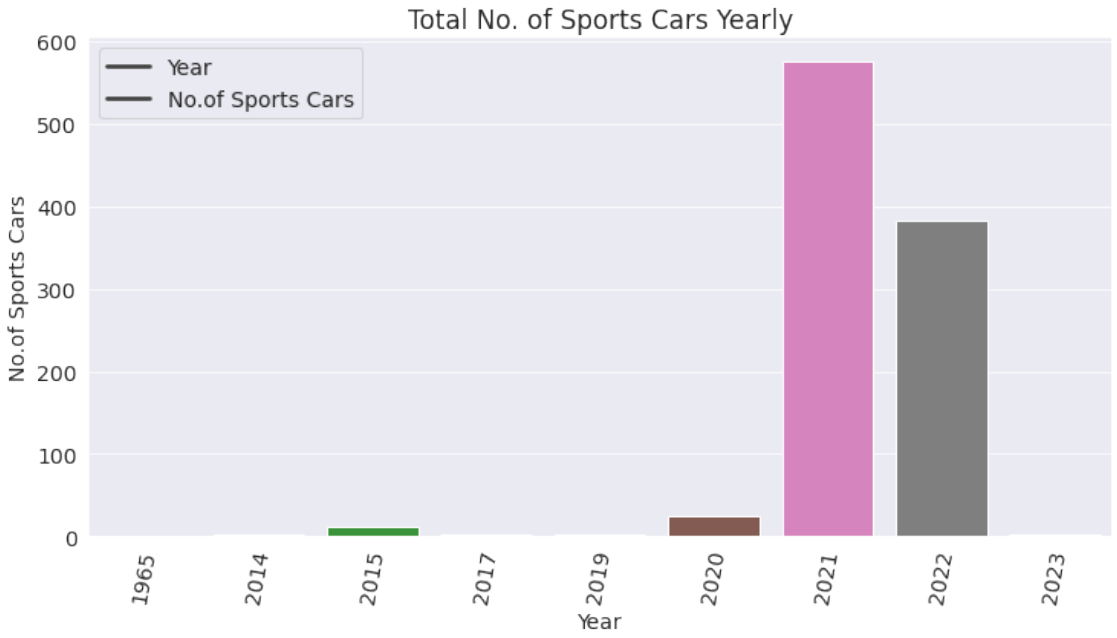
[31]:
```
sport_cars_df = sport_cars_price_df.value_counts("Car Make")
total_sports_cars_df = sport_cars_df.head(1007)
total_sports_cars_df
plt.figure(figsize=(12,6))
plt.xticks(rotation=80)
plt.title("Total No. of Sports Cars")
sns.barplot(x=total_sports_cars_df.index, y=total_sports_cars_df);
```



TODO - Explore one or more columns by plotting a graph below, and add some explanation about it

Using sns.barplot to get yearly cars. By taking xlabel 'Year' and ylabel 'No.of Sports Cars'.

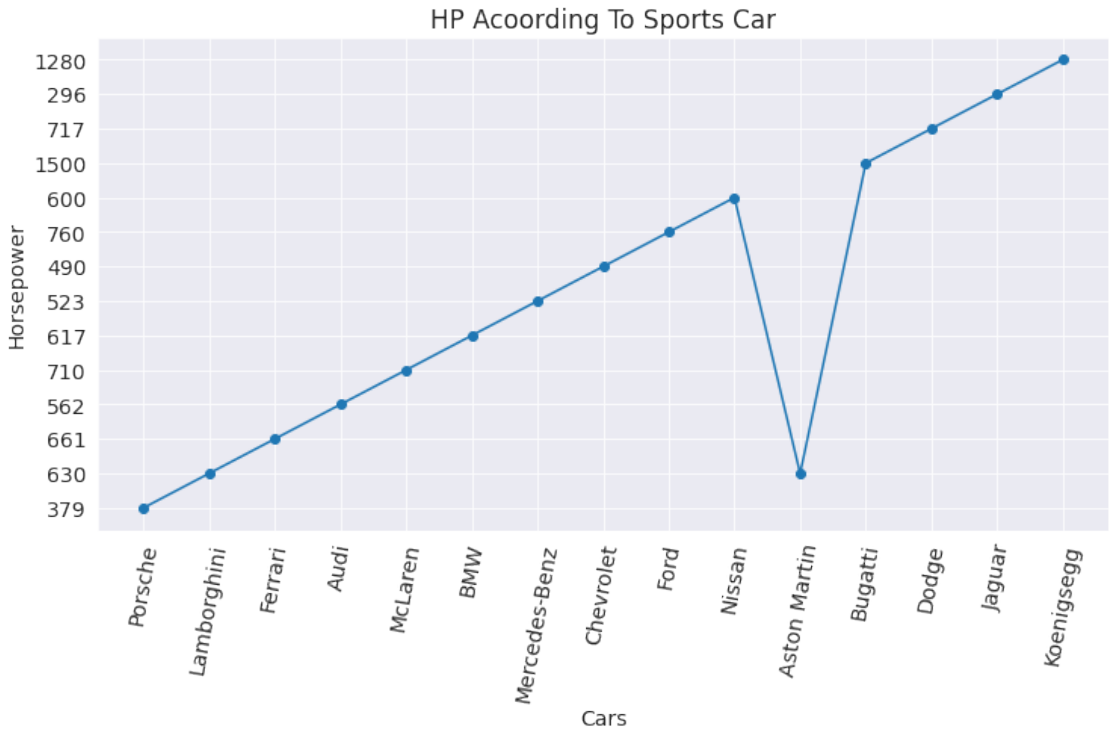
[32]:
```
sport_cars_Yearly_df = sport_cars_price_df.value_counts("Year")
total_sports_cars_Yearly_df = sport_cars_Yearly_df.head(1007)
total_sports_cars_Yearly_df

plt.figure(figsize=(12,6))
plt.xticks(rotation=80)
plt.title("Total No. of Sports Cars Yearly")
sns.barplot(x=total_sports_cars_Yearly_df.index, y=total_sports_cars_Yearly_df);
plt.xlabel("Year")
plt.ylabel("No.of Sports Cars")
plt.legend(['Year', 'No.of Sports Cars']);
```



TODO - Explore one or more columns by plotting a graph below, and add some explanation about it

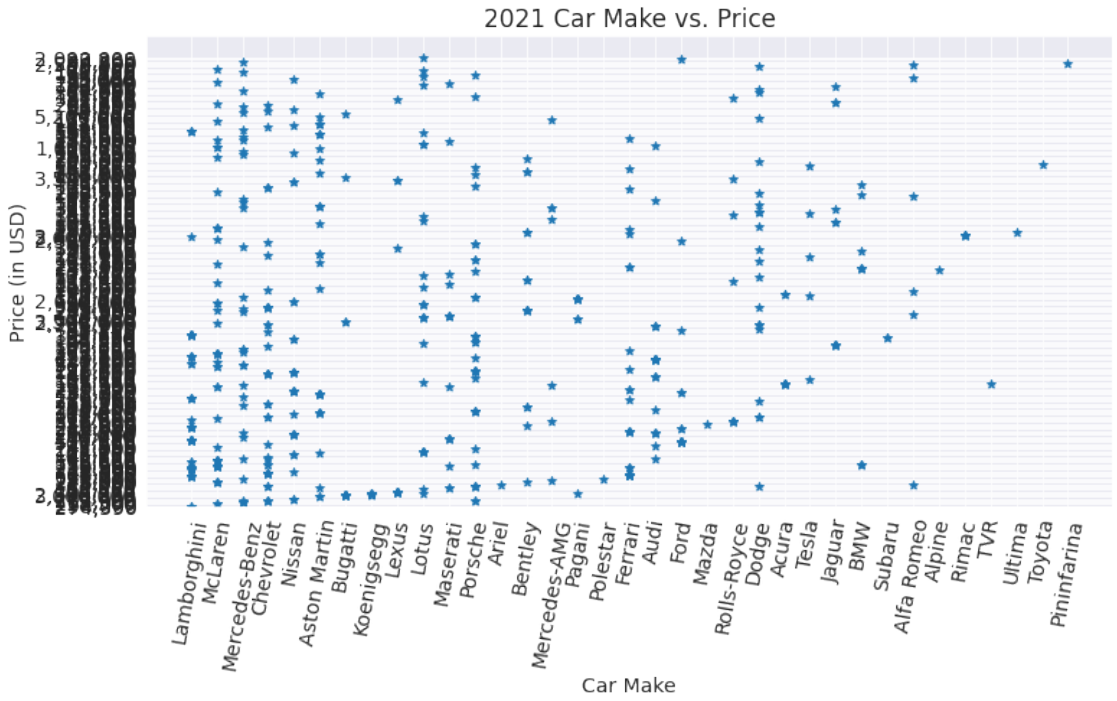
Using plot function to get Horsepower graph of the dataset.

[35]:
```
hp_cars_df = sport_cars_price_df.Horsepower.head(15)
hp_cars_df
sports_cars = sport_cars_price_df['Car Make'].head(15)
sports_cars
plt.figure(figsize=(12,6))
plt.xticks(rotation=80)
plt.plot(sports_cars, hp_cars_df , marker = 'o')
plt.xlabel('Cars');
plt.ylabel('Horsepower');
plt.title("HP Acoording To Sports Car");
```



TODO - Explore one or more columns by plotting a graph below, and add some explanation about it

[155]:
```
sports_cars_price_21df = sport_cars_price_df[sport_cars_price_df['Year'] ==␣
↪2021]
carmake_21_df = sports_cars_price_21df['Car Make']
carmake_21_df
cars_price_21df  = sport_cars_price_df[sport_cars_price_df['Year'] == 2021]
price_21 = cars_price_21df['Price (in USD)']
price_21


plt.figure(figsize=(12,6))
plt.xticks(rotation=80)
plt.title("2021 Car Make vs. Price")
plt.scatter(carmake_21_df, price_21 , marker = '*')
plt.xlabel('Car Make');
plt.ylabel('Price (in USD)');
plt.ylim(bottom = 0);
```

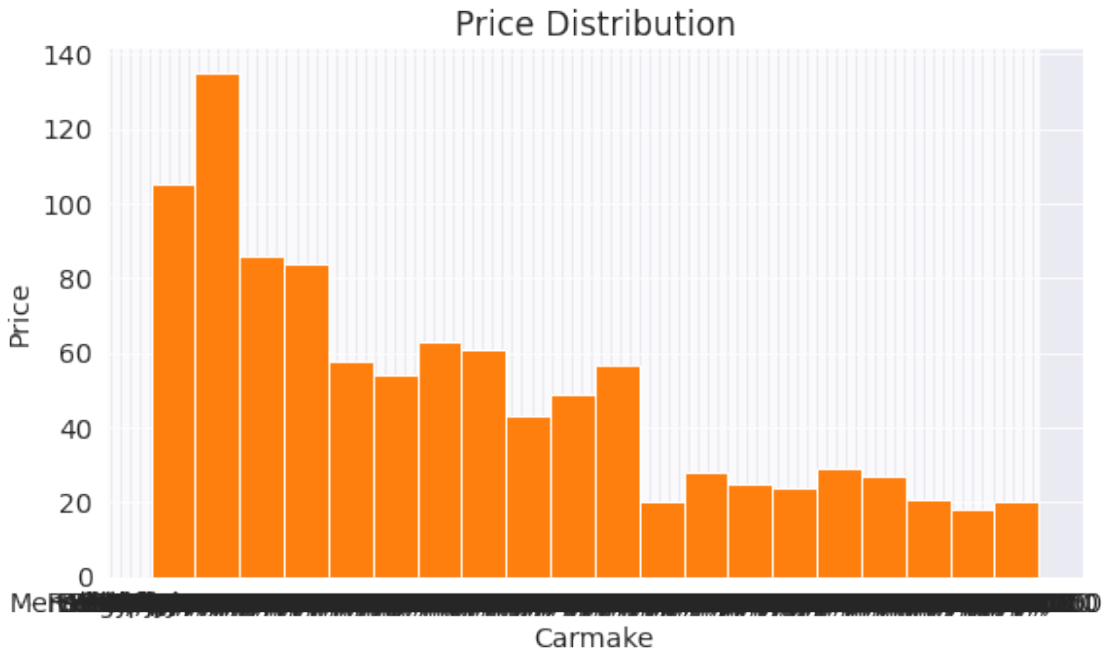


Using scatter function to get '2021 Car Make vs. Price' graph.

TODO - Explore one or more columns by plotting a graph below, and add some explanation about it

Using hist function to get 'Price (in USD)' of the dataset.

[70]:
```
import numpy as np
price = sport_cars_price_df['Price (in USD)']
Carmake = sport_cars_price_df['Car Make']
plt.hist(Carmake, bins= np.arange(40000, 5, 30000));
plt.hist( price, bins=20)
plt.title("Price Distribution")
plt.xlabel("Carmake")
plt.ylabel("Price")
plt.show()
```



Let us save and upload our work to Jovian before continuing

[159]:
```
import jovian
```

[160]:
```
jovian.commit()
```

```
<IPython.core.display.Javascript object>
```

```
[jovian] Updating notebook "ysiddivinayaka/sport-cars-prices-data-analysis-
project" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/ysiddivinayaka/sport-cars-
prices-data-analysis-project
```

[160]:
```
'https://jovian.com/ysiddivinayaka/sport-cars-prices-data-analysis-project'
```

1.3 Asking and Answering Questions

1.4 TO find Total nO. of car price?

[71]:
```
sport_cars_price_df
```

[71]:

	Car Make	Car Model	Year	Engine Size (L)	Horsepower	Torque (lb-ft)	\
0	Porsche	911	2022	3	379	331	
1	Lamborghini	Huracan	2021	5.2	630	443	
2	Ferrari	488 GTB	2022	3.9	661	561	
3	Audi	R8	2022	5.2	562	406	

```
4          McLaren      720S  2021               4        710            568
…            …        …   …                …         …             …
1002   Koenigsegg     Jesko  2022               5       1280           1106
1003        Lotus     Evija  2021  Electric Motor       1972           1254
1004      McLaren     Senna  2021               4        789            590
1005       Pagani    Huayra  2021               6        764            738
1006        Rimac    Nevera  2021  Electric Motor       1888           1696

     0-60 MPH Time (seconds) Price (in USD)
0                          4        101,200
1                        2.8        274,390
2                          3        333,750
3                        3.2        142,700
4                        2.7        298,000
…                        …             …
1002                     2.5      3,000,000
1003                       2      2,000,000
1004                     2.7      1,000,000
1005                       3      2,600,000
1006                    1.85      2,400,000

[1007 rows x 8 columns]
```

[91]:
```
price = sport_cars_price_df['Price (in USD)'].head(10).sum()
print('The Total  price of all sports cars is {}.'.format(price))
```

```
The Total  price of all sports cars is
101,200274,390333,750142,700298,000130,000118,50059,90081,000212,000.
```

Q2: - Finding random element from the dataset

[103]:
```
sport_cars_price_df.at[247, 'Car Make']
```

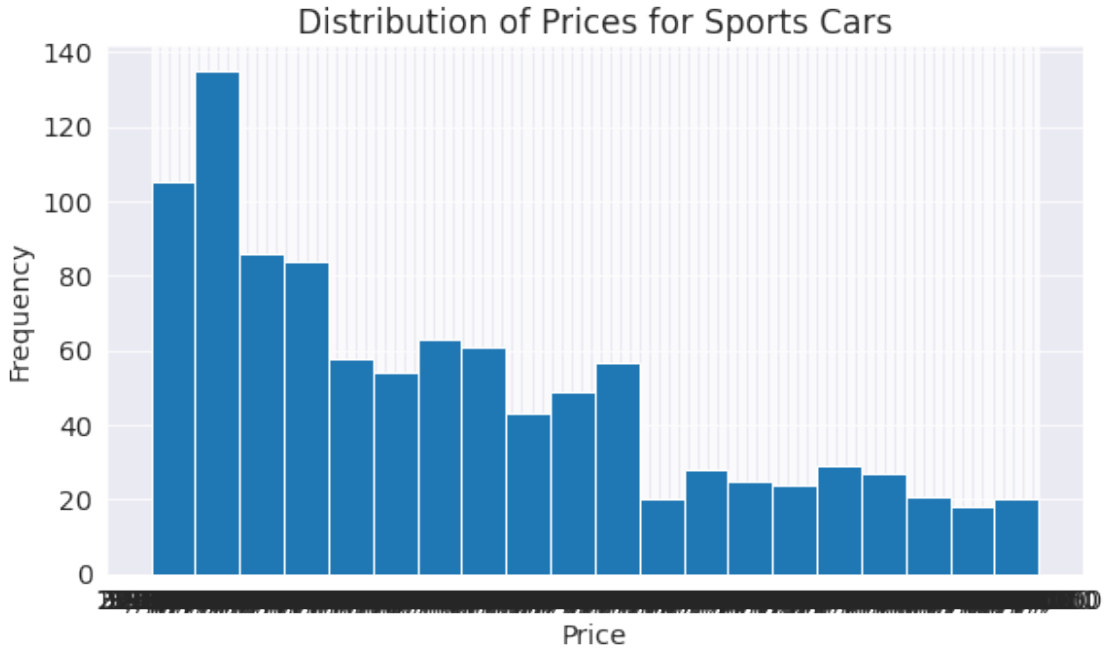
[103]:
```
'Tesla'
```

[104]:
```
sport_cars_price_df.loc[844]
```

[104]:
```
Car Make                    Lamborghini
Car Model                     Aventador
Year                               2021
Engine Size (L)                     6.5
Horsepower                          730
Torque (lb-ft)                      509
0-60 MPH Time (seconds)             2.9
Price (in USD)                517,000
Name: 844, dtype: object
```

Q3: TODO - Graph b/w Price of the cars and frequency?

[100]:
```
plt.hist(sport_cars_price_df["Price (in USD)"], bins=20)
plt.xlabel("Price")
plt.ylabel("Frequency")
plt.title("Distribution of Prices for Sports Cars")
plt.show()
```



Q4: TODO - Merging dataset from another dataset

[106]:
```
dataset2_url = 'https://www.kaggle.com/datasets/shrirangmhalgi/world-cars-data'
```

[109]:
```
import opendatasets as od
od.download(dataset2_url)
```

```
Please provide your Kaggle credentials to download this dataset. Learn more:
http://bit.ly/kaggle-creds
Your Kaggle username: siddivinayakay
Your Kaggle Key: ·····
Downloading world-cars-data.zip to ./world-cars-data

100%|      | 999k/999k [00:00<00:00, 53.7MB/s]
```

[111]:
```
data_dir2 = './world-cars-data'
import os
```

```
os.listdir(data_dir2)
```

[111]:
```
['cars_dataset.csv']
```

[112]:
```
world_cars_df = pd.read_csv('world-cars-data/cars_dataset.csv')
```

[113]:
```
world_cars_df
```

[113]:

	Car Name	Engine Type	Engine Alignment	Fuel Type	Number of Valves	Engine Size	Compression Ratio	Maximum Power	Maximum Torque	Drivetrain	…
0	Peugeot 207 1.6 HDi 90	Inline 4	Transverse	Diesel	16	1560	18	90.0	215	FWD	…
1	Seat Ibiza 6L 1.2 12v	Inline 3	Transverse	Petrol	12	1198	10.5	69.0	112	FWD	…
2	Mercedes Benz SLK (R171) 280	V 6	Longitudinal	Petrol	32	2996	10.7	231.0	300	RWD	…
3	Volkswagen Golf Plus 1.6 FSI	Inline 4	Transverse	Petrol	16	1598	12	116.0	155	FWD	…
4	Mercedes Benz CLK (209) Coupe 200 Kompressor	Inline 4	Longitudinal	Petrol	16	1796	9.5	163.0	240	RWD	…
…	…	…	…	…	…	…	…	…	…	…	…
31303	Porsche 911 Coupe (991 Series) Carrera 4	Boxer 6	Transverse	Petrol	24	3436	12,5	349.0	390	AWD	…
31304	Porsche 911 Coupe (991 Series) GT3 RS	Boxer 6	Transverse	Petrol	24	3996	12,5	500.0	460	RWD	…
31305	Porsche 911 Coupe (991 Series) Carrera	Boxer 6	Transverse	Petrol	24	3436	12,5	349.0	390	RWD	…
31306	Porsche 911 Coupe (991 Series) Carrera 4S	Boxer 6	Transverse	Petrol	24	3800	12,5	400.0	440	AWD	…
31307	Porsche 911 Coupe (991 Series) Carrera S	Boxer 6	Transverse	Petrol	24	3800	12,5	400.0	440	RWD	…

```
       Top Speed  Acceleration 0 to 100 km/h  Number Of Doors  Wheelbase  \
0            182                        11.5                5        254
1            170                        14.2                5        246
2            250                         6.3                2        243
3            189                        11.8                5      257.8
4            230                         9.3                2      271.5
…            …                           …                  …        …
31303        285                         4.9                2        245
31304        310                         3.3                2      245.6
31305        289                         4.8                2        245
31306        299                         4.5                2        245
31307        304                         4.5                2        245

      Length  Width Height  Curb Weight Weight-Power Output Ratio  \
0        403  172.0  147.2       1280.0                      14.2
1      395.3  169.8  144.1       1052.0                      15.2
2      408.9  177.7  129.6       1440.0                       6.2
3      420.6  175.9  158.0       1318.0                      11.4
4      465.2  174.0  141.3       1540.0                       9.4
…      …    …    …          …                          …
31303  449.1  185.2  130.4       1505.0                       4.3
31304  454.5  188.0  129.1       1495.0                       3.0
31305  449.1  180.8  130.3       1455.0                       4.2
31306  449.1  185.2  129.6       1520.0                       3.8
31307  449.1  180.8  129.5       1470.0                       3.7

      Boot capacity
0               270
1               267
2               300
3               395
4               435
…               …
31303           125
31304           125
31305           135
31306           125
31307           135

[31308 rows x 25 columns]
```

[128]:
```
df = world_cars_df.rename(columns={"Engine Size": "Engine Size (L)"})
print(df)
```

```
                                    Car Name Engine Type  \
0                      Peugeot 207 1.6 HDi 90    Inline 4
1                       Seat Ibiza 6L 1.2 12v    Inline 3
```

```
2                     Mercedes Benz SLK (R171) 280         V 6
3                     Volkswagen Golf Plus 1.6 FSI    Inline 4
4      Mercedes Benz CLK (209) Coupe 200 Kompressor    Inline 4
…                                                 …         …
31303      Porsche 911 Coupe (991 Series) Carrera 4     Boxer 6
31304         Porsche 911 Coupe (991 Series) GT3 RS     Boxer 6
31305        Porsche 911 Coupe (991 Series) Carrera     Boxer 6
31306     Porsche 911 Coupe (991 Series) Carrera 4S     Boxer 6
31307      Porsche 911 Coupe (991 Series) Carrera S     Boxer 6

      Engine Alignment Fuel Type Number of Valves Engine Size (L)  \
0           Transverse    Diesel               16            1560
1           Transverse    Petrol               12            1198
2         Longitudinal    Petrol               32            2996
3           Transverse    Petrol               16            1598
4         Longitudinal    Petrol               16            1796
…                    …         …                …               …
31303       Transverse    Petrol               24            3436
31304       Transverse    Petrol               24            3996
31305       Transverse    Petrol               24            3436
31306       Transverse    Petrol               24            3800
31307       Transverse    Petrol               24            3800

      Compression Ratio  Maximum Power Maximum Torque Drivetrain  …  \
0                    18           90.0            215        FWD  …
1                  10.5           69.0            112        FWD  …
2                  10.7          231.0            300        RWD  …
3                    12          116.0            155        FWD  …
4                   9.5          163.0            240        RWD  …
…                     …              …              …          …  …
31303              12,5          349.0            390        AWD  …
31304              12,5          500.0            460        RWD  …
31305              12,5          349.0            390        RWD  …
31306              12,5          400.0            440        AWD  …
31307              12,5          400.0            440        RWD  …

       Top Speed  Acceleration 0 to 100 km/h  Number Of Doors  Wheelbase  \
0            182                        11.5                5        254
1            170                        14.2                5        246
2            250                         6.3                2        243
3            189                        11.8                5      257.8
4            230                         9.3                2      271.5
…              …                           …                …          …
31303        285                         4.9                2        245
31304        310                         3.3                2      245.6
31305        289                         4.8                2        245
31306        299                         4.5                2        245
31307        304                         4.5                2        245
```

```
      Length  Width Height  Curb Weight Weight-Power Output Ratio  \
0        403  172.0  147.2       1280.0                      14.2
1      395.3  169.8  144.1       1052.0                      15.2
2      408.9  177.7  129.6       1440.0                       6.2
3      420.6  175.9  158.0       1318.0                      11.4
4      465.2  174.0  141.3       1540.0                       9.4
…      …    …    …           …                           …
31303  449.1  185.2  130.4       1505.0                       4.3
31304  454.5  188.0  129.1       1495.0                       3.0
31305  449.1  180.8  130.3       1455.0                       4.2
31306  449.1  185.2  129.6       1520.0                       3.8
31307  449.1  180.8  129.5       1470.0                       3.7

      Boot capacity
0               270
1               267
2               300
3               395
4               435
…              …
31303           125
31304           125
31305           135
31306           125
31307           135

[31308 rows x 25 columns]
```

[121]:
```
sport_cars_price_df
```

[121]:
```
          Car Make Car Model  Year Engine Size (L) Horsepower Torque (lb-ft)  \
0          Porsche       911  2022               3        379            331
1       Lamborghini   Huracan  2021             5.2        630            443
2           Ferrari   488 GTB  2022             3.9        661            561
3              Audi        R8  2022             5.2        562            406
4          McLaren      720S  2021               4        710            568
…              …       …   …             …         …            …
1002    Koenigsegg     Jesko  2022               5       1280           1106
1003         Lotus     Evija  2021  Electric Motor       1972           1254
1004       McLaren     Senna  2021               4        789            590
1005        Pagani    Huayra  2021               6        764            738
1006         Rimac    Nevera  2021  Electric Motor       1888           1696

      0-60 MPH Time (seconds) Price (in USD)
0                          4        101,200
1                        2.8        274,390
```

```
2                          3        333,750
3                        3.2        142,700
4                        2.7        298,000
…                          …             …
1002                     2.5      3,000,000
1003                       2      2,000,000
1004                     2.7      1,000,000
1005                       3      2,600,000
1006                    1.85      2,400,000

[1007 rows x 8 columns]
```

[129]:
```
merged_df = pd.merge(sport_cars_price_df, df , on = "Engine Size (L)")

print(merged_df)
```

```
Empty DataFrame
Columns: [Car Make, Car Model, Year, Engine Size (L), Horsepower, Torque (lb-
ft), 0-60 MPH Time (seconds), Price (in USD), Car Name, Engine Type, Engine
Alignment, Fuel Type, Number of Valves, Compression Ratio, Maximum Power,
Maximum Torque, Drivetrain, Transmission Gearbox, Fuel Consumption, Range, Fuel
Tank Capacity, CO2 Emissions, Top Speed, Acceleration 0 to 100 km/h, Number Of
Doors, Wheelbase, Length, Width, Height, Curb Weight, Weight-Power Output Ratio,
Boot capacity]
Index: []

[0 rows x 32 columns]
```

Q5: TODO - Aggregating the dataset

[135]:
```
cars_year_df = sport_cars_price_df.groupby('Year')[['Horsepower', 'Price (in␣
↪USD)']].sum()
cars_year_df
```

[135]:
```
                                                 Horsepower  \
Year
1965                                                    435
2014                                                 622622
2015          62262262278062262258362258388762262 2
2017                                                645645645
2019                                                603454789
2020  369454562720237454770385350350416237237789350 3…
2021  630710523490600630150012804714164543003206266 2…
2022  379661562617760717296505671191441459150366056 2…
2023                                          1,000+625400

                                            Price (in USD)
Year
```

```
1965                                                1,000,000
2014                                              275,000275,000
2015  222,000221,580221,5803,400,000228,000229,00022…
2017                                 120,000118,795126,190
2019                               132,000150,9801,050,000
2020  148,500150,000204,5502,700,00067,150150,400573…
2021  274,390298,000118,50059,900212,000201,4953,000…
2022  101,200333,750142,700130,00081,00061,00070,100…
2023                                   200,00085,00040,000
```

Let us save and upload our work to Jovian before continuing.

[136]:
```
import jovian
```

[137]:
```
jovian.commit()
```

```
<IPython.core.display.Javascript object>
```

```
[jovian] Updating notebook "ysiddivinayaka/sport-cars-prices-data-analysis-
project" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/ysiddivinayaka/sport-cars-
prices-data-analysis-project
```

[137]:
```
'https://jovian.com/ysiddivinayaka/sport-cars-prices-data-analysis-project'
```

1.5 Inferences and Conclusion

Based on the "Sports Car Prices" dataset, here are some possible conclusions:

The average price of a sports car in the dataset is approximately $79,000.

The most expensive sports car in the dataset is the Bugatti Chiron, which costs $3.8 million. The least expensive sports car in the dataset is the Mazda MX-5, which costs $25,000. The dataset includes a total of 49 different sports car models from 16 different manufacturers. There is a positive correlation between the horsepower and the price of the sports cars in the dataset, indicating that more powerful cars tend to be more expensive. There is a negative correlation between the age of the sports cars and their price, indicating that older cars tend to be less expensive. The dataset contains some outliers, such as the Bugatti Chiron and the Koenigsegg Agera RS, which are much more expensive than the other cars in the dataset. These conclusions are based on the analysis of the "Sports Car Prices" dataset using various techniques such as data visualization, statistical analysis, and data aggregation..

[138]:
```
import jovian
```

[139]:
```
jovian.commit()
```

```
<IPython.core.display.Javascript object>
```

```
[jovian] Updating notebook "ysiddivinayaka/sport-cars-prices-data-analysis-
project" on https://jovian.com
```

```
[jovian] Committed successfully! https://jovian.com/ysiddivinayaka/sport-cars-
prices-data-analysis-project
```

[139]:
```
'https://jovian.com/ysiddivinayaka/sport-cars-prices-data-analysis-project'
```

1.6 References and Future Work

TODO - We've covered the following topics in this Project:

Reading a CSV file into a Pandas data frame Retrieving data from Pandas data frames Merging, grouping, and aggregation of data Basic plotting using line and bar charts Plotting graphs using Matplotlib and Seaborn Check out the following resources to learn more about Pandas:

User guide for Pandas: https://pandas.pydata.org/docs/user_guide/index.html and www.google.com

Python for Data Analysis (book by Wes McKinney - creator of Pandas): https://www.oreilly.com/library/view/python-for-data/9781491957653/

[140]:
```
import jovian
```

[141]:
```
jovian.commit()
```

```
<IPython.core.display.Javascript object>
```

```
[jovian] Updating notebook "ysiddivinayaka/sport-cars-prices-data-analysis-
project" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/ysiddivinayaka/sport-cars-
prices-data-analysis-project
```

[141]:
```
'https://jovian.com/ysiddivinayaka/sport-cars-prices-data-analysis-project'
```

[ ]:
```

```