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

```
df =pd.read_csv("E:\iit class\Machine learning\Cars93.csv")
```

```
df.head()
```

	Manufacturer	Model	Type	Min.Price	Price	Max.Price	MPG.city	MPG.highway	AirBags	DriveTrain	...	Fuel.tank.capacity
0	Acura	Integra	Small	12.9	15.9	18.8	25	31	NaN	Front	...	13.2
1	Acura	Legend	Midsize	29.2	33.9	38.7	18	25	Driver & Passenger	Front	...	18.0
2	Audi	90	Compact	25.9	29.1	32.3	20	26	Driver only	Front	...	16.9
3	Audi	100	Midsize	30.8	37.7	44.6	19	26	Driver & Passenger	Front	...	21.1
4	BMW	535i	Midsize	23.7	30.0	36.2	22	30	Driver only	Rear	...	21.1

5 rows × 26 columns

```
df.tail()
```

	Manufacturer	Model	Type	Min.Price	Price	Max.Price	MPG.city	MPG.highway	AirBags	DriveTrain	...	Fuel.tank.capacit
88	Volkswagen	Eurovan	Van	16.6	19.7	22.7	17	21	NaN	Front	...	21.
89	Volkswagen	Passat	Compact	17.6	20.0	22.4	21	30	NaN	Front	...	18.
90	Volkswagen	Corrado	Sporty	22.9	23.3	23.7	18	25	NaN	Front	...	18.
91	Volvo	240	Compact	21.8	22.7	23.5	21	28	Driver only	Rear	...	15.
92	Volvo	850	Midsize	24.8	26.7	28.5	20	28	Driver & Passenger	Front	...	19.

5 rows × 26 columns

```
df.shape
```

```
(93, 26)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 93 entries, 0 to 92
Data columns (total 26 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Manufacturer        93 non-null     object 
 1   Model               93 non-null     object 
 2   Type                93 non-null     object 
 3   Min.Price           93 non-null     float64
 4   Price               93 non-null     float64
 5   Max.Price           93 non-null     float64
 6   MPG.city            93 non-null     int64  
 7   MPG.highway         93 non-null     int64  
 8   AirBags             59 non-null     object 
 9   DriveTrain          93 non-null     object 
 10  Cylinders           93 non-null     object 
 11  EngineSize          93 non-null     float64
 12  Horsepower          93 non-null     int64  
 13  RPM                 93 non-null     int64  
 14  Rev.per.mile        93 non-null     int64  
 15  Man.trans.avail     93 non-null     object 
 16  Fuel.tank.capacity  93 non-null     float64
 17  Passengers          93 non-null     int64  
 18  Length              93 non-null     int64  
 19  Wheelbase           93 non-null     int64
```

```
 20  Width               93 non-null     int64
 21  Turn.circle         93 non-null     int64
 22  Rear.seat.room      91 non-null     float64
 23  Luggage.room        82 non-null     float64
 24  Weight              93 non-null     int64
 25  Origin              93 non-null     object
dtypes: float64(7), int64(11), object(8)
memory usage: 19.0+ KB
```

```
df.describe()
```

	Min.Price	Price	Max.Price	MPG.city	MPG.highway	EngineSize	Horsepower	RPM	Rev.per.mile	Fuel.tank.capacity
count	93.000000	93.000000	93.000000	93.000000	93.000000	93.000000	93.000000	93.000000	93.000000	93.000000
mean	17.125806	19.509677	21.898925	22.365591	29.086022	2.667742	143.827957	5280.645161	2332.204301	16.664516
std	8.746029	9.659430	11.030457	5.619812	5.331726	1.037363	52.374410	596.731690	496.506525	3.279370
min	6.700000	7.400000	7.900000	15.000000	20.000000	1.000000	55.000000	3800.000000	1320.000000	9.200000
25%	10.800000	12.200000	14.700000	18.000000	26.000000	1.800000	103.000000	4800.000000	1985.000000	14.500000
50%	14.700000	17.700000	19.600000	21.000000	28.000000	2.400000	140.000000	5200.000000	2340.000000	16.400000
75%	20.300000	23.300000	25.300000	25.000000	31.000000	3.300000	170.000000	5750.000000	2565.000000	18.800000
max	45.400000	61.900000	80.000000	46.000000	50.000000	5.700000	300.000000	6500.000000	3755.000000	27.000000

```
df.notnull()
```

	Manufacturer	Model	Type	Min.Price	Price	Max.Price	MPG.city	MPG.highway	AirBags	DriveTrain	...	Fuel.tank.capacity	Pass
0	True	True	True	True	True	True	True	True	False	True	...	True	
1	True	True	True	True	True	True	True	True	True	True	...	True	
2	True	True	True	True	True	True	True	True	True	True	...	True	
3	True	True	True	True	True	True	True	True	True	True	...	True	
4	True	True	True	True	True	True	True	True	True	True	...	True	
...	...	...	...	...	...	...	...	...	...	...	...	...	
88	True	True	True	True	True	True	True	True	False	True	...	True	
89	True	True	True	True	True	True	True	True	False	True	...	True	
90	True	True	True	True	True	True	True	True	False	True	...	True	
91	True	True	True	True	True	True	True	True	True	True	...	True	
92	True	True	True	True	True	True	True	True	True	True	...	True	

93 rows × 26 columns

Handling missing Value

```
df.fillna(0)
```

	Manufacturer	Model	Type	Min.Price	Price	Max.Price	MPG.city	MPG.highway	AirBags	DriveTrain	...	Fuel.tank.capacit
0	Acura	Integra	Small	12.9	15.9	18.8	25	31	0	Front	...	13.
1	Acura	Legend	Midsize	29.2	33.9	38.7	18	25	Driver & Passenger	Front	...	18.
2	Audi	90	Compact	25.9	29.1	32.3	20	26	Driver only	Front	...	16.
3	Audi	100	Midsize	30.8	37.7	44.6	19	26	Driver & Passenger	Front	...	21.
4	BMW	535i	Midsize	23.7	30.0	36.2	22	30	Driver only	Rear	...	21.
...	...	...	...	...	...	...	...	...	...	...	...	.
88	Volkswagen	Eurovan	Van	16.6	19.7	22.7	17	21	0	Front	...	21.
89	Volkswagen	Passat	Compact	17.6	20.0	22.4	21	30	0	Front	...	18.
90	Volkswagen	Corrado	Sporty	22.9	23.3	23.7	18	25	0	Front	...	18.
91	Volvo	240	Compact	21.8	22.7	23.5	21	28	Driver only	Rear	...	15.
92	Volvo	850	Midsize	24.8	26.7	28.5	20	28	Driver & Passenger	Front	...	19.

93 rows × 26 columns

```
df.describe()
```

	Min.Price	Price	Max.Price	MPG.city	MPG.highway	EngineSize	Horsepower	RPM	Rev.per.mile	Fuel.tank.capacity
count	93.000000	93.000000	93.000000	93.000000	93.000000	93.000000	93.000000	93.000000	93.000000	93.000000
mean	17.125806	19.509677	21.898925	22.365591	29.086022	2.667742	143.827957	5280.645161	2332.204301	16.664516
std	8.746029	9.659430	11.030457	5.619812	5.331726	1.037363	52.374410	596.731690	496.506525	3.279370
min	6.700000	7.400000	7.900000	15.000000	20.000000	1.000000	55.000000	3800.000000	1320.000000	9.200000
25%	10.800000	12.200000	14.700000	18.000000	26.000000	1.800000	103.000000	4800.000000	1985.000000	14.500000
50%	14.700000	17.700000	19.600000	21.000000	28.000000	2.400000	140.000000	5200.000000	2340.000000	16.400000
75%	20.300000	23.300000	25.300000	25.000000	31.000000	3.300000	170.000000	5750.000000	2565.000000	18.800000
max	45.400000	61.900000	80.000000	46.000000	50.000000	5.700000	300.000000	6500.000000	3755.000000	27.000000

```
df=df.drop_duplicates(keep='first')
df
```

	Manufacturer	Model	Type	Min.Price	Price	Max.Price	MPG.city	MPG.highway	AirBags	DriveTrain	...	Fuel.tank.capacit
0	Acura	Integra	Small	12.9	15.9	18.8	25	31	NaN	Front	...	13.
1	Acura	Legend	Midsize	29.2	33.9	38.7	18	25	Driver & Passenger	Front	...	18.
2	Audi	90	Compact	25.9	29.1	32.3	20	26	Driver only	Front	...	16.
3	Audi	100	Midsize	30.8	37.7	44.6	19	26	Driver & Passenger	Front	...	21.
4	BMW	535i	Midsize	23.7	30.0	36.2	22	30	Driver only	Rear	...	21.
...	...	...	...	...	...	...	...	...	...	...	...	.
88	Volkswagen	Eurovan	Van	16.6	19.7	22.7	17	21	NaN	Front	...	21.
89	Volkswagen	Passat	Compact	17.6	20.0	22.4	21	30	NaN	Front	...	18.
90	Volkswagen	Corrado	Sporty	22.9	23.3	23.7	18	25	NaN	Front	...	18.
91	Volvo	240	Compact	21.8	22.7	23.5	21	28	Driver only	Rear	...	15.
92	Volvo	850	Midsize	24.8	26.7	28.5	20	28	Driver & Passenger	Front	...	19.

93 rows × 26 columns

Univarite Analysis

```
df.Type.unique()
```

```
array(['Small', 'Midsize', 'Compact', 'Large', 'Sporty', 'Van'],
      dtype=object)
```

```
k=df.Type.value_counts()
k
```

```
Type
Midsize    22
Small      21
Compact    16
Sporty     14
Large      11
Van         9
Name: count, dtype: int64
```

```
A=df.groupby(by="Type")['Manufacturer'].count()
A
```

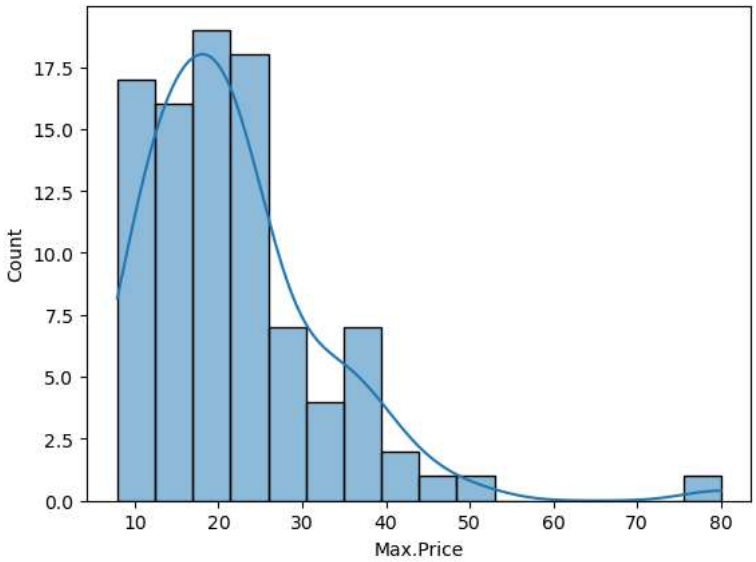
```
Type
Compact    16
Large      11
Midsize    22
Small      21
Sporty     14
Van         9
Name: Manufacturer, dtype: int64
```

```
def min_max_val (col):
    '''to individually check the min and  max values of each col
    '''
    top=df[col].idxmax()
    top_obs=pd.DataFrame(df.loc[top])

    bottom=df[col].idxmin()
    bottom_obs=pd.DataFrame(df.loc[bottom])

    min_max_obs=pd.concat([top_obs,bottom_obs],axis=1)
    return min_max_obs
```

```
min_max_val ('Max.Price')
```

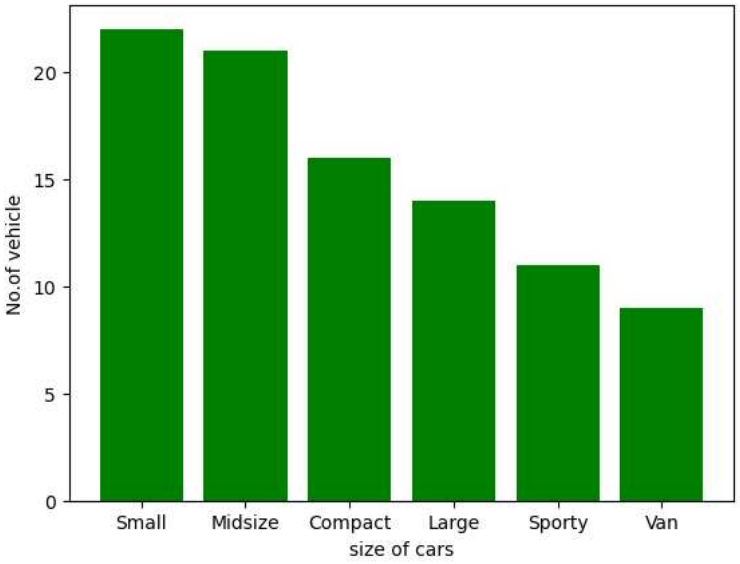
	58	30
Manufacturer	Mercedes-Benz	Ford
Model	300E	Festiva
Type	Midsize	Small
Min.Price	43.8	6.9
Price	61.9	7.4
Max.Price	80.0	7.9
MPG.city	19	31
MPG.highway	25	33
AirBags	Driver & Passenger	NaN
DriveTrain	Rear	Front
Cylinders	6	4
EngineSize	3.2	1.3
Horsepower	217	63
RPM	5500	5000
Rev.per.mile	2220	3150
Man.trans.avail	No	Yes
Fuel.tank.capacity	18.5	10.0
Passengers	5	4
Length	187	141
Wheelbase	110	90
Width	69	63
Turn.circle	37	33
Rear.seat.room	27.0	26.0
Luggage.room	15.0	12.0
Weight	3525	1845
Origin	non-USA	USA

```
sns.histplot(df['Max.Price'], kde=True)
plt.show()
```



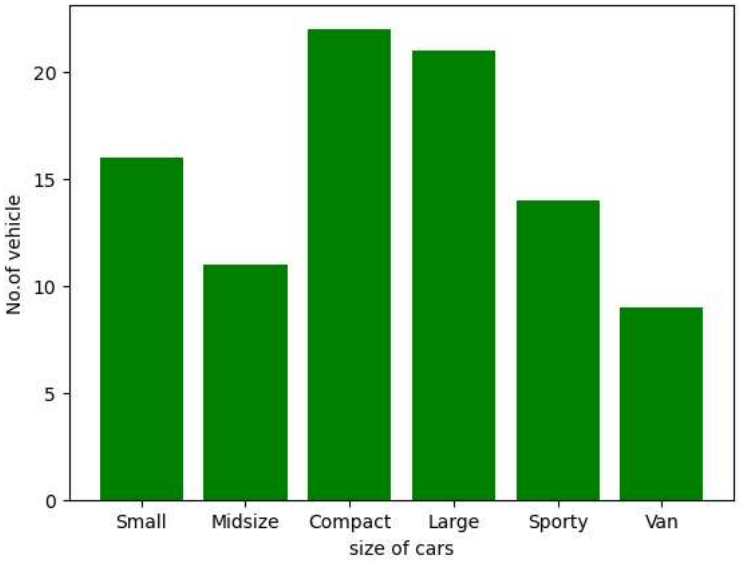

```
cat=df.Type.unique()
val=df.Type.value_counts()
plt.bar(cat,val,color="green")
plt.xlabel("size of cars")
plt.ylabel("No.of vehicle")
```

```
Text(0, 0.5, 'No.of vehicle')
```




```
cat=df.Type.unique()
val=df.groupby(by="Type")['Manufacturer'].count()
plt.bar(cat,val,color="green")
plt.xlabel("size of cars")

plt.ylabel("No.of vehicle")
```

```
Text(0, 0.5, 'No.of vehicle')
```



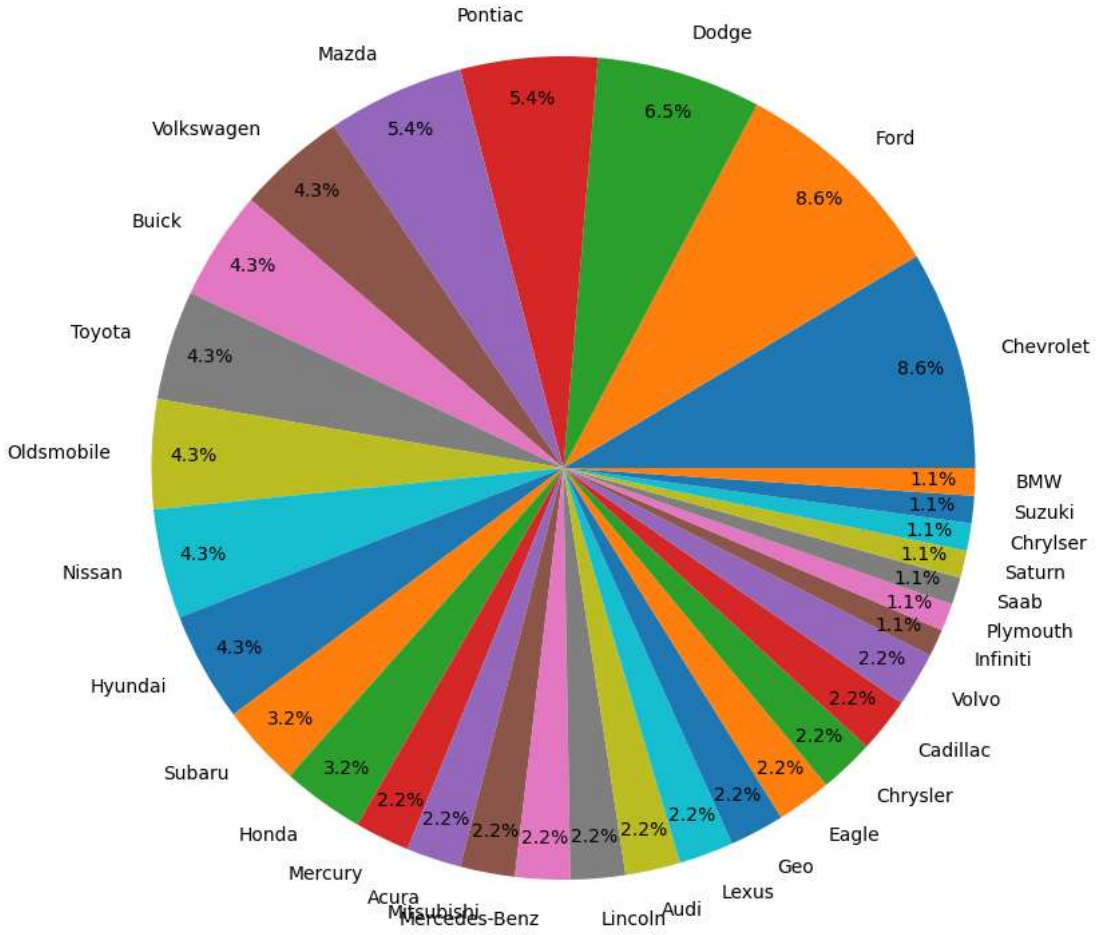

```
no_airbags_car= df.AirBags.unique()
no_airbags_car
```

```
array([nan, 'Driver & Passenger', 'Driver only'], dtype=object)
```

```
df.groupby(by="AirBags")['Manufacturer'].count()
```

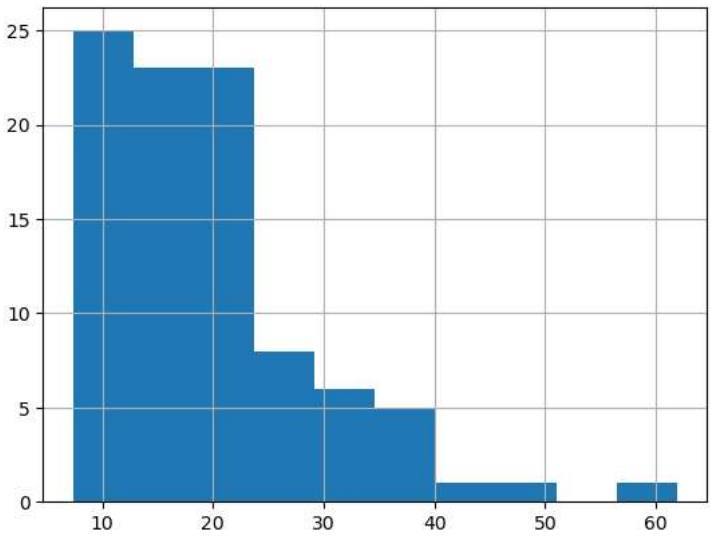
```
AirBags
Driver & Passenger    16
Driver only           43
Name: Manufacturer, dtype: int64
```

```
plt.figure(figsize=(15,10))
counts = df['Manufacturer'].value_counts()
counts.plot(kind="pie", autopct='%1.1f%%',pctdistance=.90)
#plt.legend(title="Manufacturers")
plt.axis('off')
plt.show()
```




```
df.Price.hist()
```

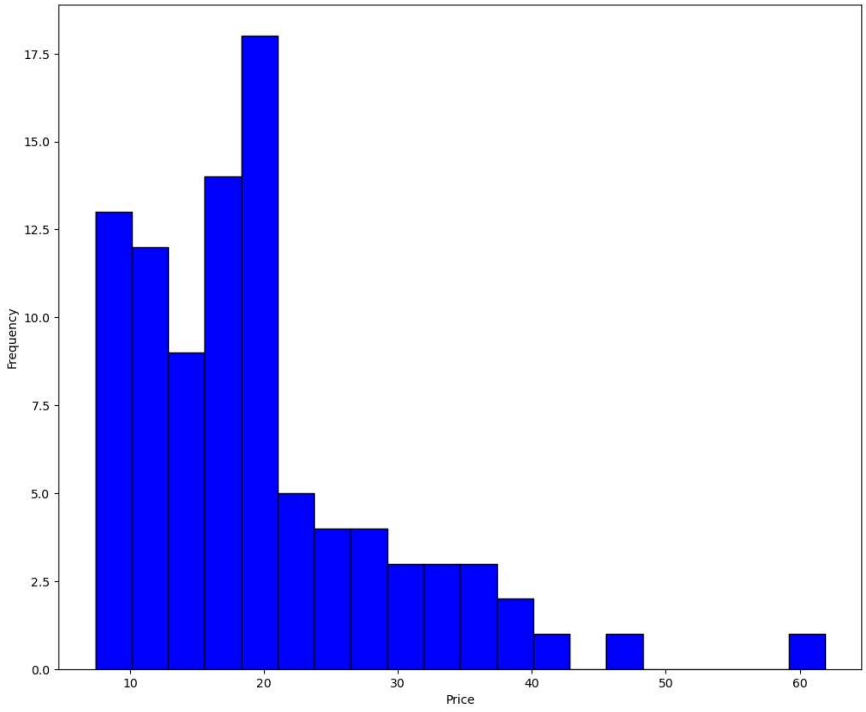
```
<Axes: >
```




```
df.columns
```

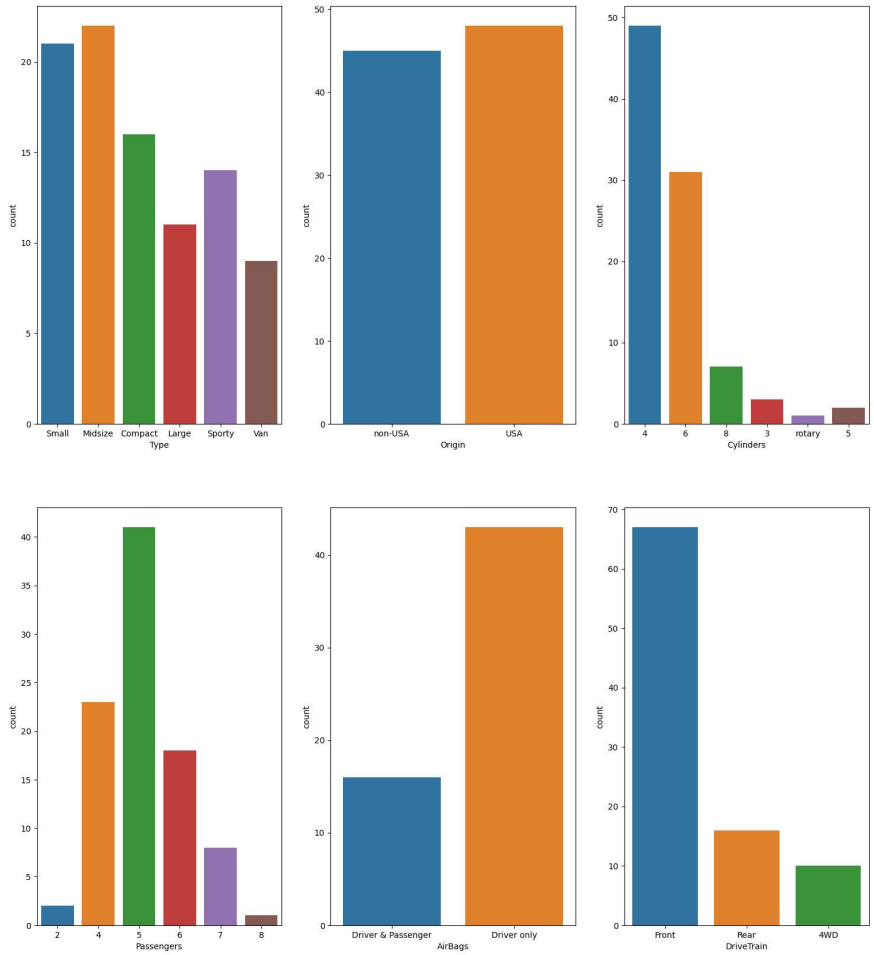
```
Index(['Manufacturer', 'Model', 'Type', 'Min.Price', 'Price', 'Max.Price',
       'MPG.city', 'MPG.highway', 'AirBags', 'DriveTrain', 'Cylinders',
       'EngineSize', 'Horsepower', 'RPM', 'Rev.per.mile', 'Man.trans.avail',
       'Fuel.tank.capacity', 'Passengers', 'Length', 'Wheelbase', 'Width',
       'Turn.circle', 'Rear.seat.room', 'Luggage.room', 'Weight', 'Origin'],
      dtype='object')
```

```
plt.figure(figsize=(12, 10))
plt.hist(df['Price'], bins=20, color='blue', edgecolor='black')
plt.xlabel('Price')
plt.ylabel('Frequency')
plt.show()
```



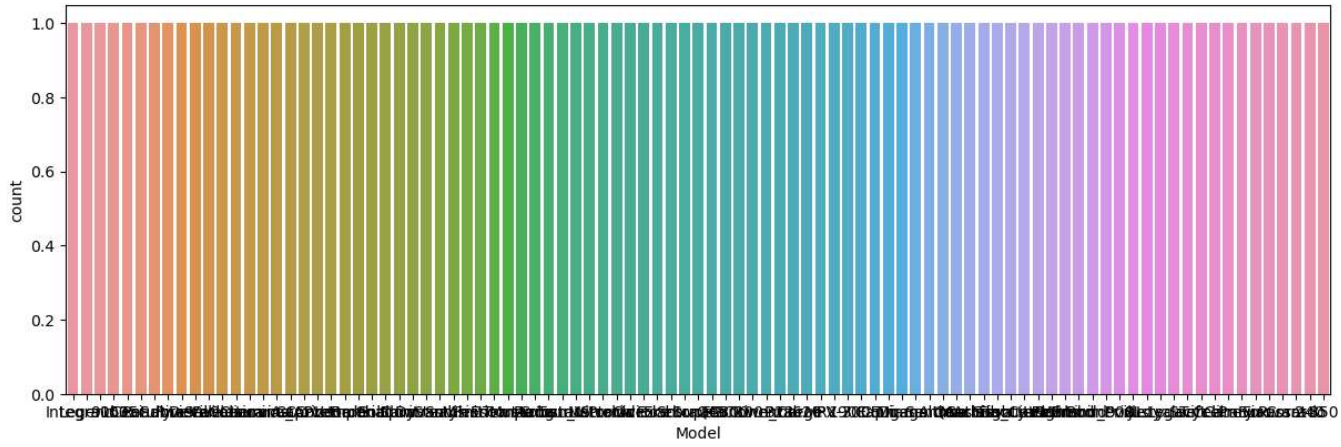

```
plt.figure(figsize=(18,20))
plt.subplot(2,3,1)
sns.countplot(x='Type', data=df)
plt.subplot(2,3,2)
sns.countplot(x='Origin',data=df)
plt.subplot(2,3,3)
sns.countplot(x='Cylinders', data=df)
plt.subplot(2,3,4)
sns.countplot(x='Passengers',data=df)
plt.subplot(2,3,5)
sns.countplot(x='AirBags',data=df)
plt.subplot(2,3,6)
sns.countplot(x='DriveTrain',data=df)
```

```
<Axes: xlabel='DriveTrain', ylabel='count'>
```



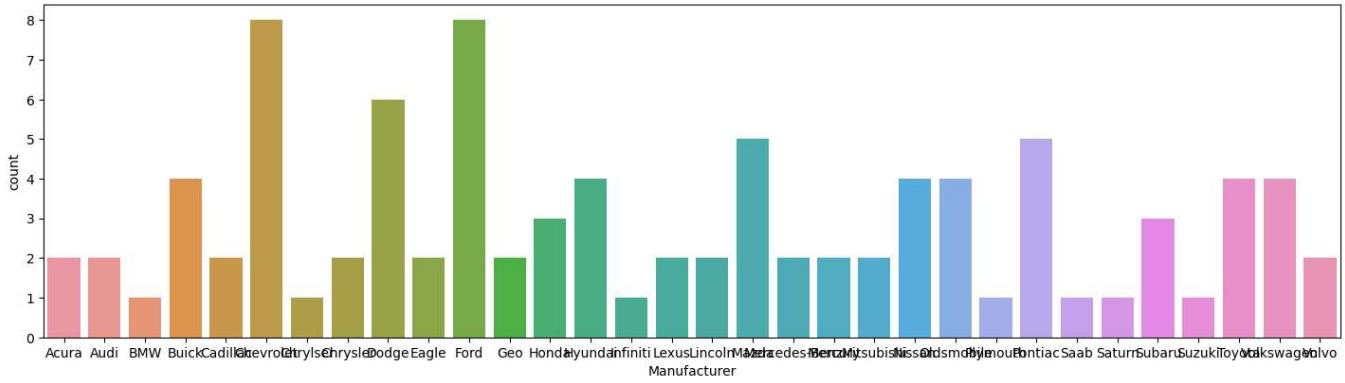
```
plt.figure(figsize=(50,10))
plt.subplot(2,3,2)
sns.countplot(x='Model',data=df)
```

```
<Axes: xlabel='Model', ylabel='count'>
```



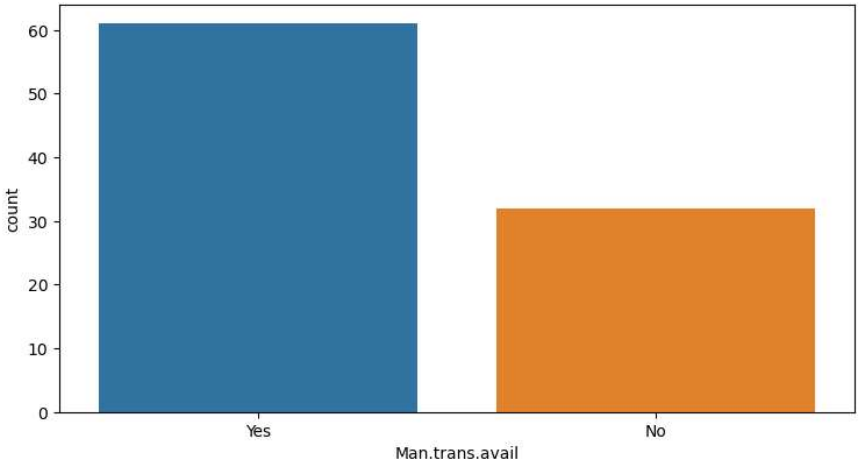

```
plt.figure(figsize=(60,10))
plt.subplot(2,3,2)
sns.countplot(x='Manufacturer',data=df)
```

```
<Axes: xlabel='Manufacturer', ylabel='count'>
```




```
plt.figure(figsize=(30,10))
plt.subplot(2,3,2)
sns.countplot(x='Man.trans.avail',data=df)
```

```
<Axes: xlabel='Man.trans.avail', ylabel='count'>
```




```
cat_col=[]
num_col=[]
for i in df.columns:
    if(df[i].dtypes=="object"):
        cat_col.append(i)
    else:
        num_col.append(i)
```

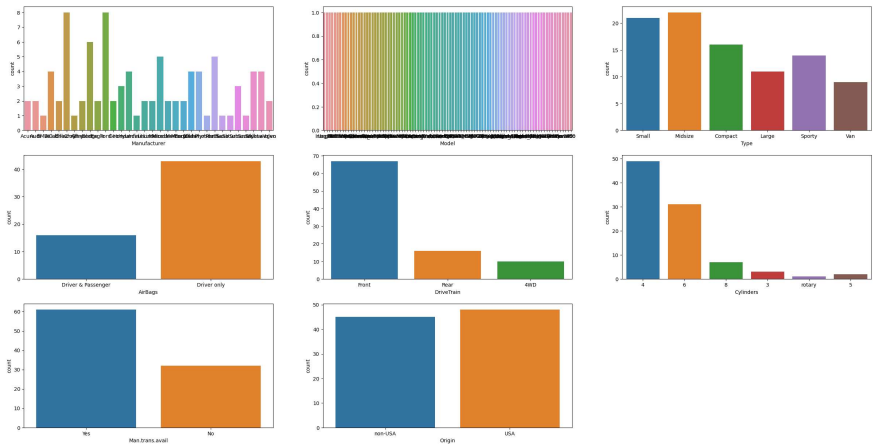
```
cat_col
```

```
['Manufacturer',
 'Model',
 'Type',
 'AirBags',
 'DriveTrain',
 'Cylinders',
 'Man.trans.avail',
 'Origin']
```

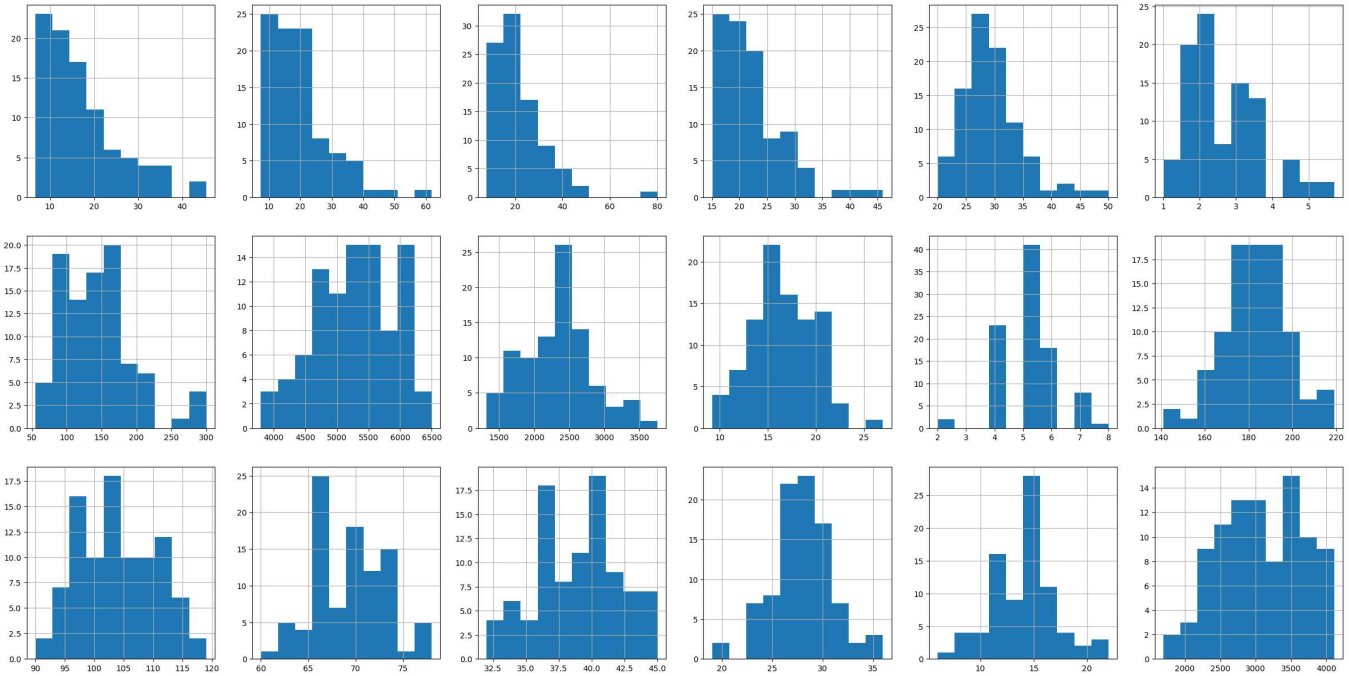
```
num_col
#len(num_col)
```

```
['Min.Price',
 'Price',
 'Max.Price',
 'MPG.city',
 'MPG.highway',
 'EngineSize',
 'Horsepower',
 'RPM',
 'Rev.per.mile',
 'Fuel.tank.capacity',
 'Passengers',
 'Length',
 'Wheelbase',
 'Width',
 'Turn.circle',
 'Rear.seat.room',
 'Luggage.room',
 'Weight']
```

```
plt.figure(figsize=(30,15))

x = 1
for i in cat_col:
    plt.subplot(3,3,x)
    sns.countplot(x=df[i],data=df)
    x = x+1
```



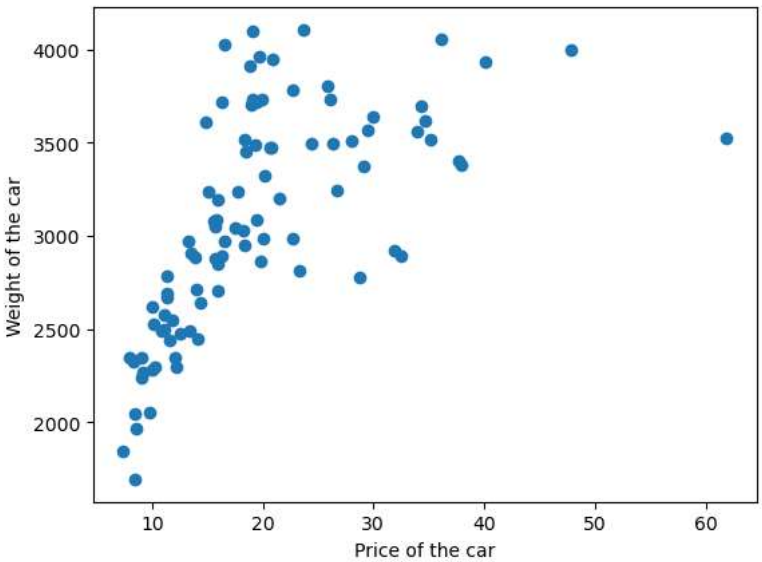

```
plt.figure(figsize=(30,15))

x = 1
for i in num_col:
    plt.subplot(3,6,x)
    df[i].hist()
    x = x+1
```



Bivariate Analysis

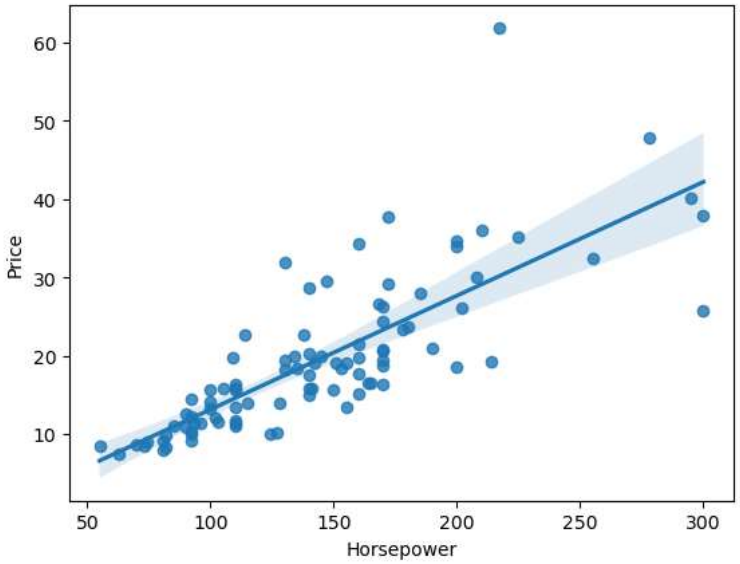
```
import matplotlib.pyplot as plt

plt.scatter(df['Price'], df['Weight'])
plt.xlabel("Price of the car")
plt.ylabel("Weight of the car")
```

```
Text(0, 0.5, 'Weight of the car')
```



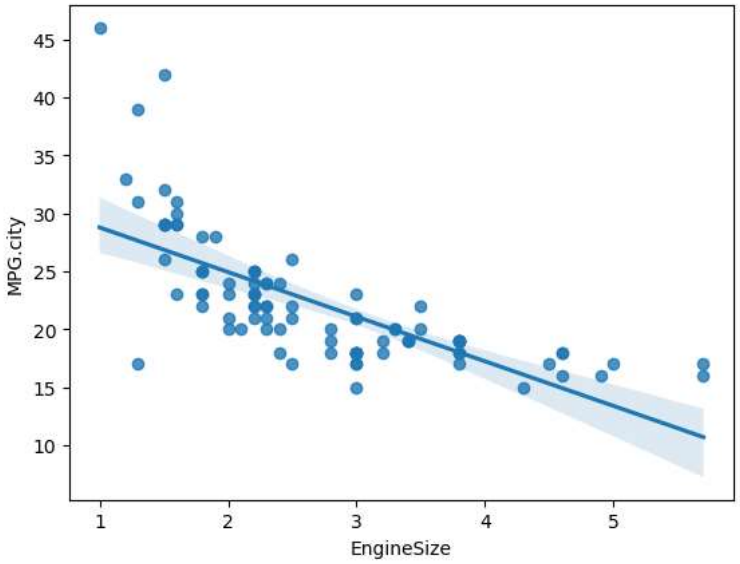

```
sns.regplot(x='Horsepower',y='Price',data=df)
```

```
<Axes: xlabel='Horsepower', ylabel='Price'>
```

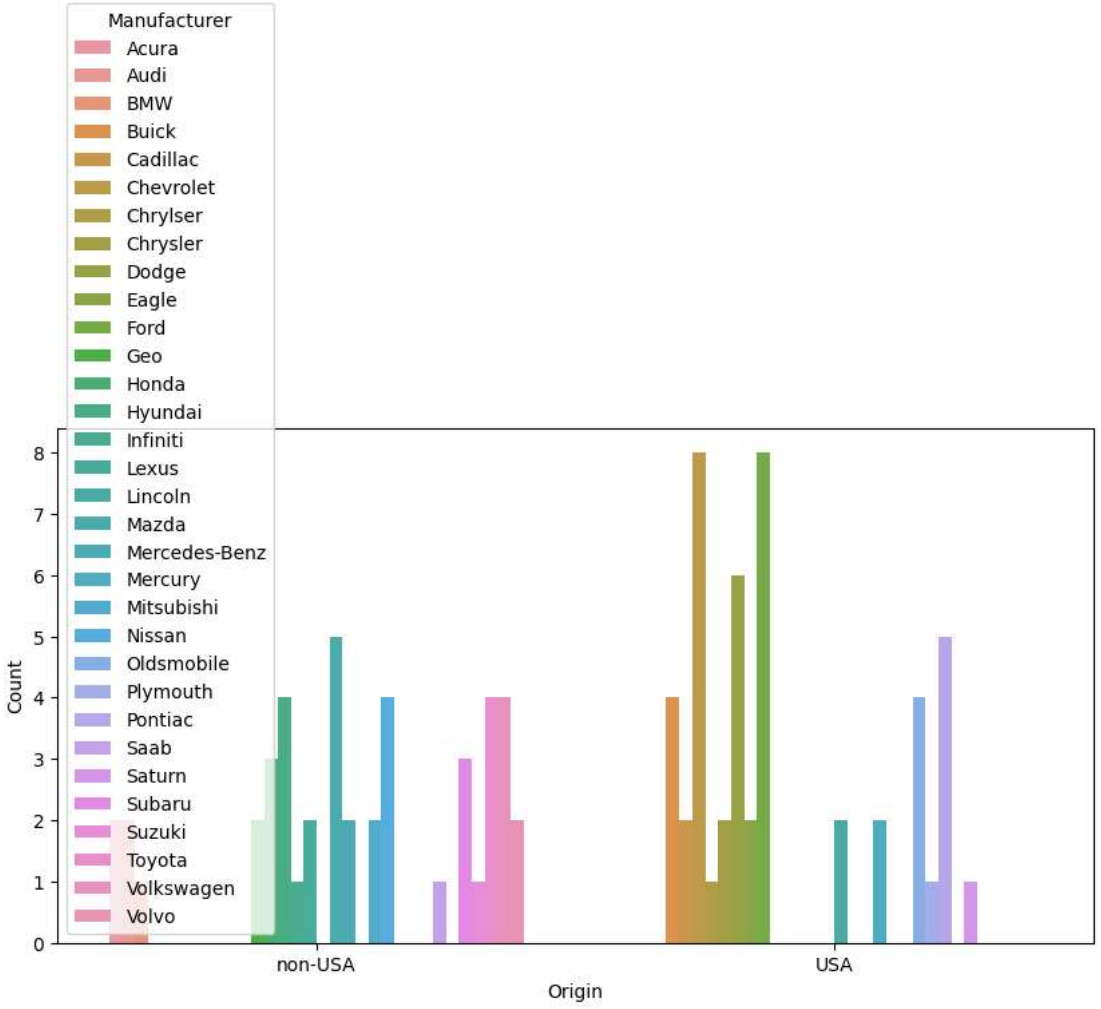



```
sns.regplot(x='EngineSize',y='MPG.city',data=df)
```

```
<Axes: xlabel='EngineSize', ylabel='MPG.city'>
```




```
for i in df.columns:
    if df[i].dtype == "object":
        plt.figure(figsize=(10,5))
        sns.countplot(x='Origin',hue= i ,data=df)
        plt.ylabel('Count')
        plt.show()
```


Manufacturer
Acura
Audi
BMW
Buick
Cadillac
Chevrolet
Chrylser
Chrysler
Dodge
Eagle
Ford
Geo
Honda
Hyundai
Infiniti
Lexus
Lincoln
Mazda
Mercedes-Benz
Mercury
Mitsubishi
Nissan
Oldsmobile
Plymouth
Pontiac
Saab
Saturn
Subaru
Suzuki
Toyota
Volkswagen
Volvo
Count
0
1
2
3
4
5
6
7
8
non-USA
USA
Origin


Model
Integra
Legend
90
100
535i
Century
LeSabre
Roadmaster
Riviera
DeVille
Seville
Cavalier
Corsica
Camaro
Lumina
Lumina_APV
Astro
Caprice
Corvette
Concorde
LeBaron
Imperial
Colt
Shadow
Spirit
Caravan
Dynasty
Stealth
Summit
Vision
Festiva
Escort
Tempo
Mustang
Probe
Aerostar
Taurus