

```
In [1]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
In [2]: df=pd.read_csv(r"C:\Users\HP\Downloads\used_cars_data.csv")
df
```

Out[2]:

	S.No.	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	
1	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	
2	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	
3	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	
4	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Se
...	...	...	...	...	...	...	...	
7248	7248	Volkswagen Vento Diesel Trendline	Hyderabad	2011	89411	Diesel	Manual	
7249	7249	Volkswagen Polo GT TSI	Mumbai	2015	59000	Petrol	Automatic	
7250	7250	Nissan Micra Diesel XV	Kolkata	2012	28000	Diesel	Manual	
7251	7251	Volkswagen Polo GT TSI	Pune	2013	52262	Petrol	Automatic	
7252	7252	Mercedes-Benz E-Class 2009-2013 E 220 CDI Avan...	Kochi	2014	72443	Diesel	Automatic	

7253 rows × 14 columns



```
In [3]: df.head()
```

```
Out[3]:
```

	S.No.	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First
1	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First
2	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First
3	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First
4	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second



```
In [4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7253 entries, 0 to 7252
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   S.No.                 7253 non-null  int64
1   Name                  7253 non-null  object
2   Location              7253 non-null  object
3   Year                  7253 non-null  int64
4   Kilometers_Driven     7253 non-null  int64
5   Fuel_Type             7253 non-null  object
6   Transmission          7253 non-null  object
7   Owner_Type            7253 non-null  object
8   Mileage               7251 non-null  object
9   Engine                7207 non-null  object
10  Power                 7207 non-null  object
11  Seats                 7200 non-null  float64
12  New_Price             1006 non-null  object
13  Price                 6019 non-null  float64
dtypes: float64(2), int64(3), object(9)
memory usage: 793.4+ KB
```

```
In [5]: df.dropna(inplace=True)
```

```
In [6]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 823 entries, 2 to 6014
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   S.No.                 823 non-null   int64  
1   Name                  823 non-null   object  
2   Location              823 non-null   object  
3   Year                  823 non-null   int64  
4   Kilometers_Driven     823 non-null   int64  
5   Fuel_Type             823 non-null   object  
6   Transmission          823 non-null   object  
7   Owner_Type            823 non-null   object  
8   Mileage               823 non-null   object  
9   Engine                823 non-null   object  
10  Power                 823 non-null   object  
11  Seats                 823 non-null   float64 
12  New_Price             823 non-null   object  
13  Price                 823 non-null   float64 
dtypes: float64(2), int64(3), object(9)
memory usage: 96.4+ KB
```

```
In [7]: df.describe()
```

```
Out[7]:
```

	S.No.	Year	Kilometers_Driven	Seats	Price
<b>count</b>	823.000000	823.000000	823.000000	823.000000	823.000000
<b>mean</b>	3017.058323	2015.869988	40498.402187	5.304982	13.647606
<b>std</b>	1775.572600	2.378709	28791.171759	0.795707	14.860539
<b>min</b>	2.000000	2001.000000	600.000000	2.000000	0.700000
<b>25%</b>	1522.000000	2015.000000	20487.500000	5.000000	5.000000
<b>50%</b>	2916.000000	2016.000000	34895.000000	5.000000	8.200000
<b>75%</b>	4650.500000	2017.000000	53829.000000	5.000000	17.500000
<b>max</b>	6014.000000	2019.000000	215000.000000	8.000000	160.000000

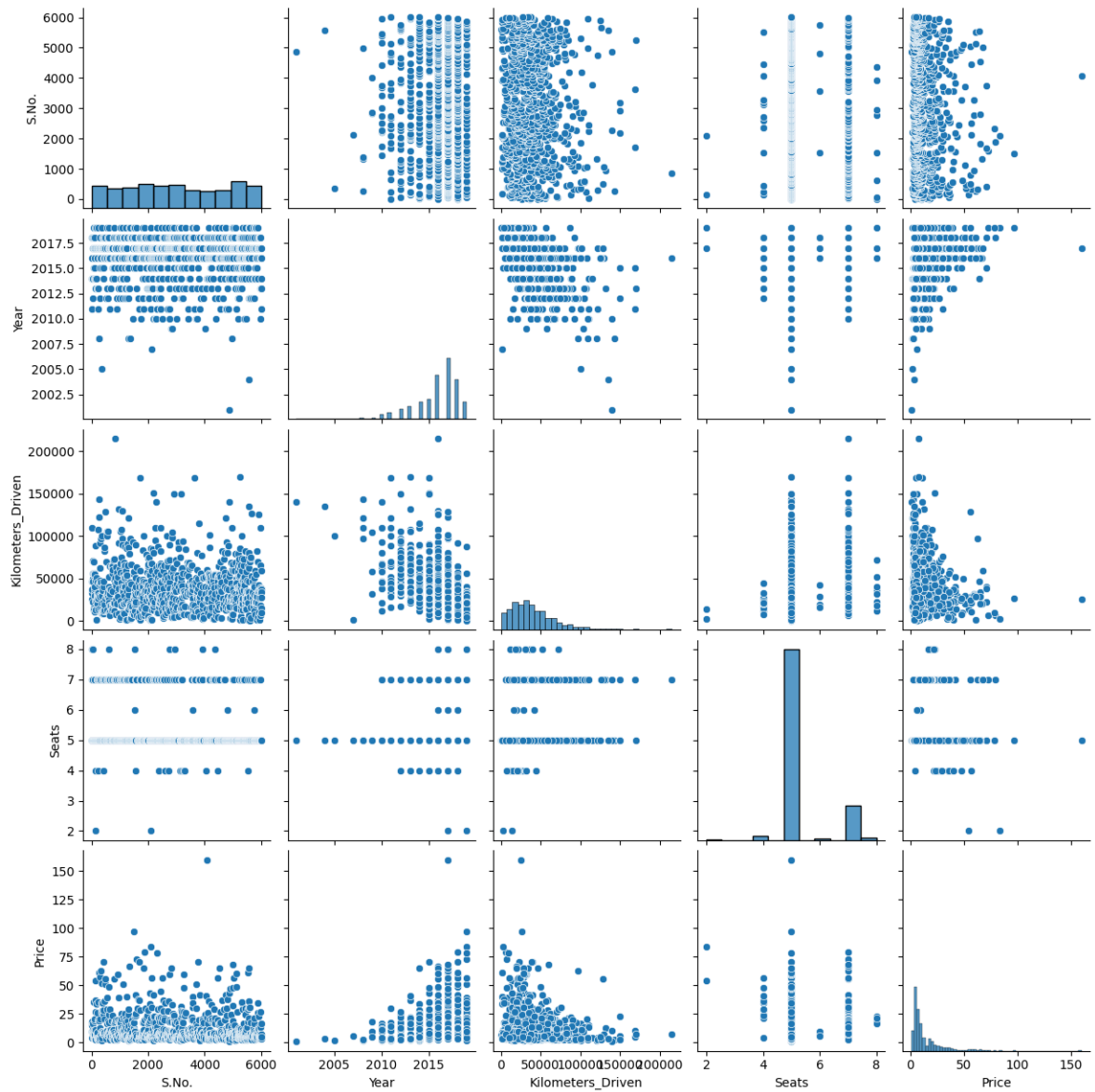
```
In [8]: df.columns
```

```
Out[8]: Index(['S.No.', 'Name', 'Location', 'Year', 'Kilometers_Driven', 'Fuel_Type',
               'Transmission', 'Owner_Type', 'Mileage', 'Engine', 'Power', 'Seats',
               'New_Price', 'Price'],
              dtype='object')
```

## Exploratory Data Analysis:

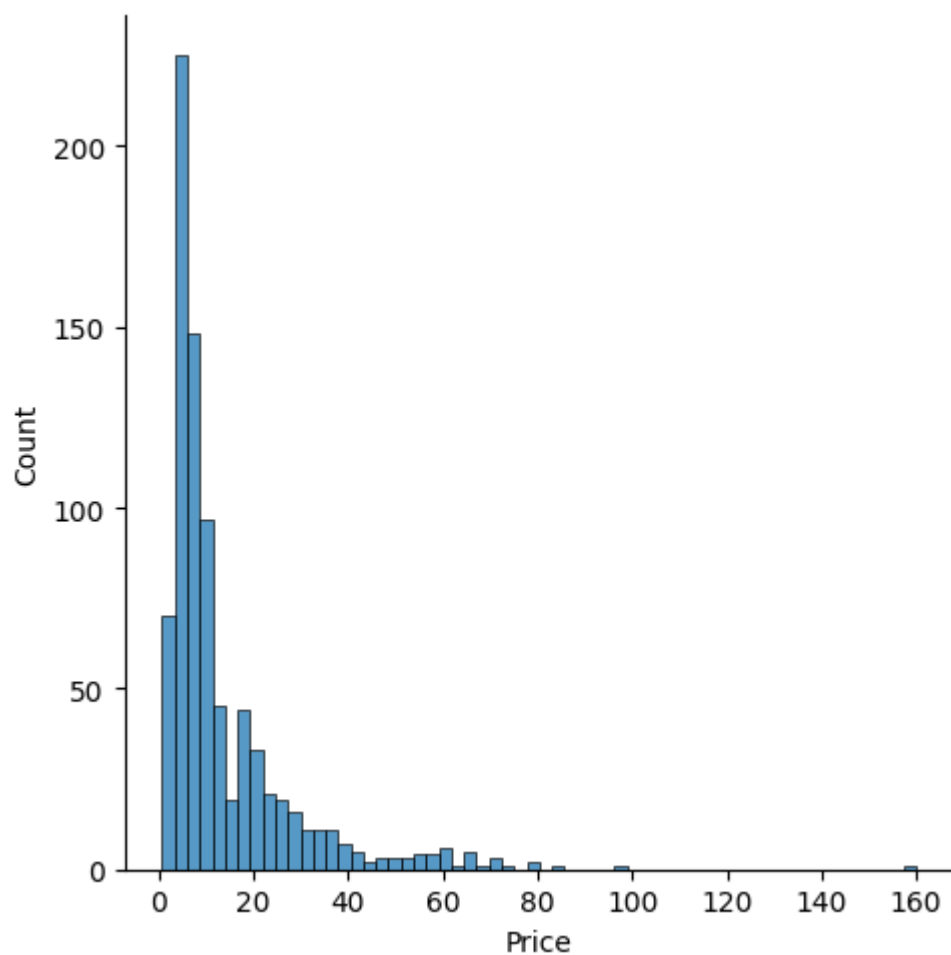
```
In [9]: sns.pairplot(df)
```

```
Out[9]: <seaborn.axisgrid.PairGrid at 0x1c1b9256110>
```



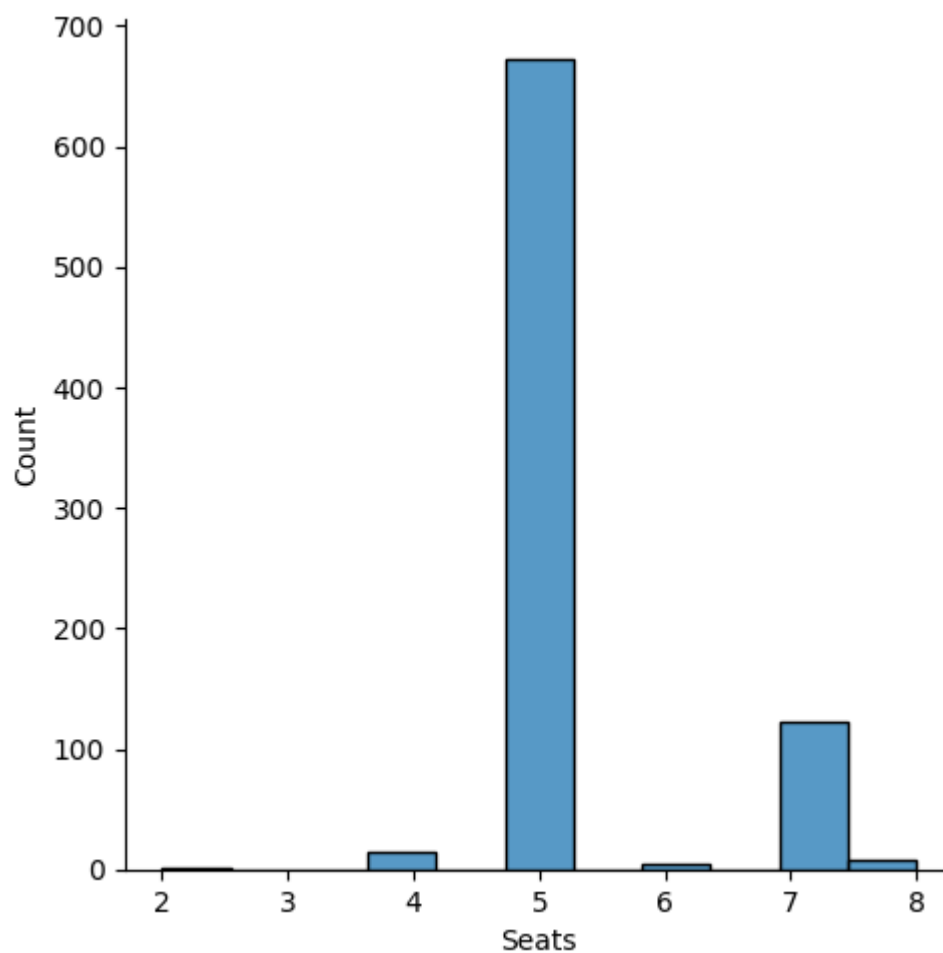
```
In [10]: sns.displot(df['Price'])
```

```
Out[10]: <seaborn.axisgrid.FacetGrid at 0x1c1bd4a3ee0>
```



```
In [11]: sns.displot(df['Seats'])
```

```
Out[11]: <seaborn.axisgrid.FacetGrid at 0x1c1bd54f190>
```



```
In [12]: carsdf=df[['S.No.', 'Name', 'Location', 'Year', 'Kilometers_Driven', 'Fuel_Type',  
                    'Transmission', 'Owner_Type', 'Mileage', 'Engine', 'Power', 'Seats',  
                    'New_Price', 'Price']]
```

```
In [13]: sns.heatmap(carsdf.corr())
```

```
-----  
ValueError                                Traceback (most recent call last)  
Cell In[13], line 1  
----> 1 sns.heatmap(carsdf.corr())  
  
File ~\AppData\Roaming\Python\Python310\site-packages\pandas\core\frame.py:1  
0059, in DataFrame.corr(self, method, min_periods, numeric_only)  
    10057 cols = data.columns  
    10058 idx = cols.copy()  
> 10059 mat = data.to_numpy(dtype=float, na_value=np.nan, copy=False)  
    10061 if method == "pearson":  
    10062     correl = libalgos.nancorr(mat, minp=min_periods)  
  
File ~\AppData\Roaming\Python\Python310\site-packages\pandas\core\frame.py:1  
838, in DataFrame.to_numpy(self, dtype, copy, na_value)  
    1836 if dtype is not None:  
    1837     dtype = np.dtype(dtype)  
-> 1838 result = self._mgr.as_array(dtype=dtype, copy=copy, na_value=na_valu  
e)  
    1839 if result.dtype is not dtype:  
    1840     result = np.array(result, dtype=dtype, copy=False)  
  
File ~\AppData\Roaming\Python\Python310\site-packages\pandas\core\internals  
\managers.py:1732, in BlockManager.as_array(self, dtype, copy, na_value)  
    1730     arr.flags.writeable = False  
    1731 else:  
-> 1732     arr = self._interleave(dtype=dtype, na_value=na_value)  
    1733     # The underlying data was copied within _interleave, so no need  
    1734     # to further copy if copy=True or setting na_value  
    1736 if na_value is not lib.no_default:  
  
File ~\AppData\Roaming\Python\Python310\site-packages\pandas\core\internals  
\managers.py:1794, in BlockManager._interleave(self, dtype, na_value)  
    1792     else:  
    1793         arr = blk.get_values(dtype)  
-> 1794     result[r1.indexer] = arr  
    1795     itemmask[r1.indexer] = 1  
    1797 if not itemmask.all():  
  
ValueError: could not convert string to float: 'Honda Jazz V'
```

```
In [ ]: x=carsdf[['S.No.', 'Name', 'Location', 'Year', 'Kilometers_Driven', 'Fuel_Type',  
                'Transmission', 'Owner_Type', 'Mileage', 'Engine', 'Power', 'Seats',  
                'New_Price', 'Price']]  
y=df['Price']
```

```
In [ ]: from sklearn.model_selection import train_test_split  
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=
```

```
In [ ]: from sklearn.linear_model import LinearRegression  
lm=LinearRegression  
lm.fit(x_train,y_train)
```

```
In [ ]: print(lm.intercept_)
```

```
In [ ]:
```