

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

```
In [2]: df = pd.read_csv("my_data.csv")
```

```
In [3]: df
```

Out[3]:

	name	year	km_driven	fuel	seller_type	transmission	owner	torque	se
0	Maruti Swift Dzire VDI	2014	145500	Diesel	Individual	Manual	First Owner	190Nm@ 2000rpm	
1	Skoda Rapid 1.5 TDI Ambition	2014	120000	Diesel	Individual	Manual	Second Owner	250Nm@ 1500- 2500rpm	
2	Honda City 2017- 2020 EXi	2006	140000	Petrol	Individual	Manual	Third Owner	12.7@ 2,700(kgm@ rpm)	
3	Honda City 2017- 2020 EXi	2006	140000	Petrol	Individual	Manual	Third Owner	12.7@ 2,700(kgm@ rpm)	
4	Hyundai i20 Sportz Diesel	2010	127000	Diesel	Individual	Manual	First Owner	22.4 kgm at 1750- 2750rpm	
...
7407	Maruti Wagon R VXI BS IV with ABS	2013	50000	Petrol	Individual	Manual	Second Owner	90Nm@ 3500rpm	
7408	Hyundai i20 Magna 1.4 CRDi	2014	80000	Diesel	Individual	Manual	Second Owner	219.7Nm@ 1500- 2750rpm	
7409	Hyundai i20 Magna	2013	110000	Petrol	Individual	Manual	First Owner	113.7Nm@ 4000rpm	
7410	Hyundai Verna CRDi SX	2007	119000	Diesel	Individual	Manual	Fourth & Above Owner	24@ 1,900- 2,750(kgm@ rpm)	
7411	Maruti Swift Dzire ZDi	2009	120000	Diesel	Individual	Manual	First Owner	190Nm@ 2000rpm	

7412 rows × 13 columns



In [4]: df.info()

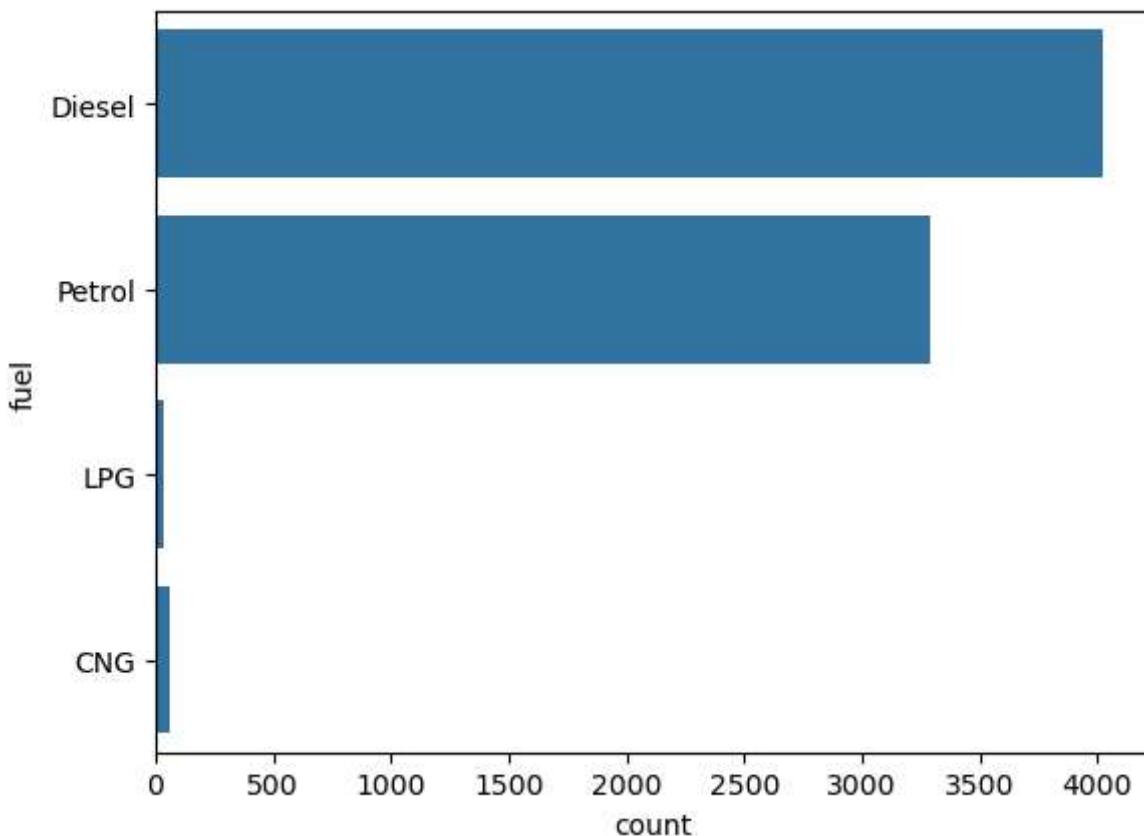
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7412 entries, 0 to 7411
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype  
--- 
 0   name              7412 non-null    object  
 1   year              7412 non-null    int64  
 2   km_driven         7412 non-null    int64  
 3   fuel              7412 non-null    object  
 4   seller_type       7412 non-null    object  
 5   transmission      7412 non-null    object  
 6   owner             7412 non-null    object  
 7   torque            7412 non-null    object  
 8   seats              7412 non-null    float64 
 9   mileage(kmpl)    7412 non-null    float64 
 10  engine(CC)        7412 non-null    float64 
 11  max_power(bhp)   7412 non-null    float64 
 12  selling_price    7412 non-null    int64  
dtypes: float64(4), int64(3), object(6)
memory usage: 752.9+ KB
```

```
In [5]: df.nunique()
```

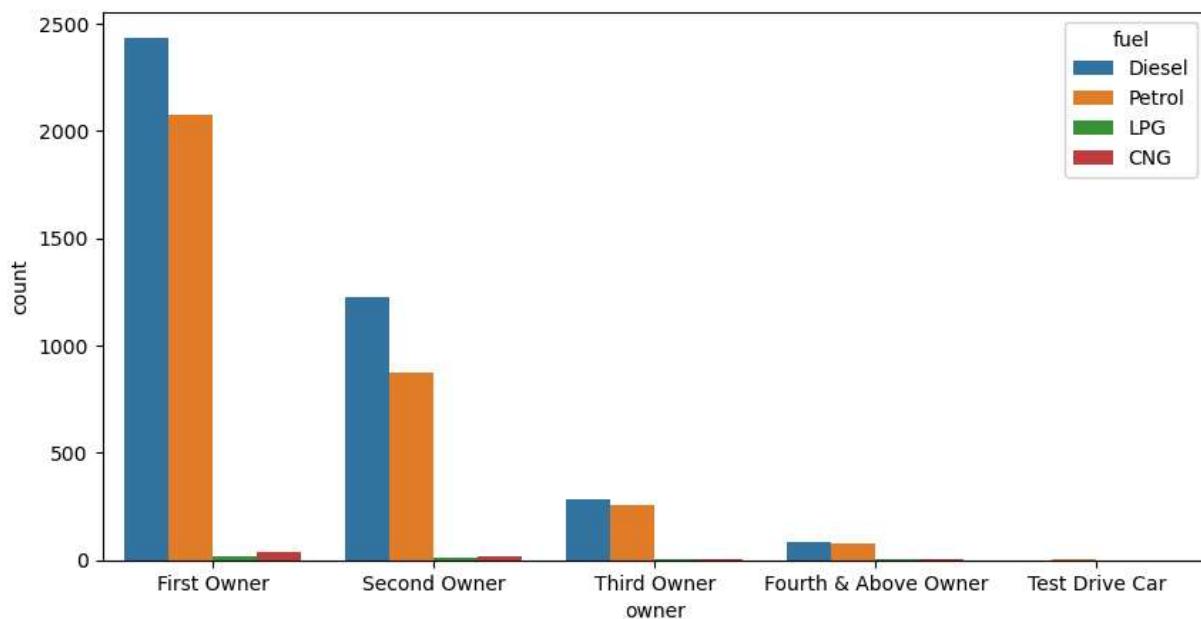
```
Out[5]: name          2058
year           29
km_driven     921
fuel            4
seller_type    3
transmission   2
owner           5
torque          442
seats           9
mileage(kmpl) 381
engine(CC)      121
max_power(bhp) 319
selling_price  677
dtype: int64
```

```
In [6]: sns.countplot(df["fuel"])
```

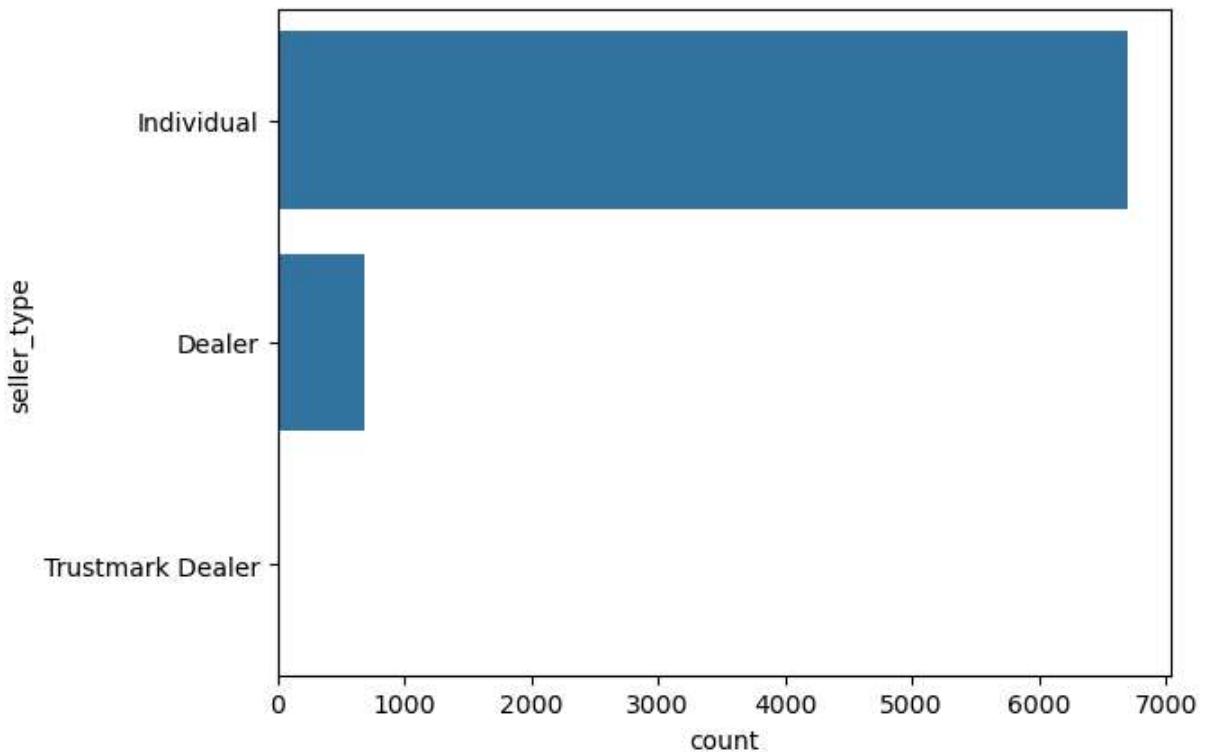
```
Out[6]: <Axes: xlabel='count', ylabel='fuel'>
```



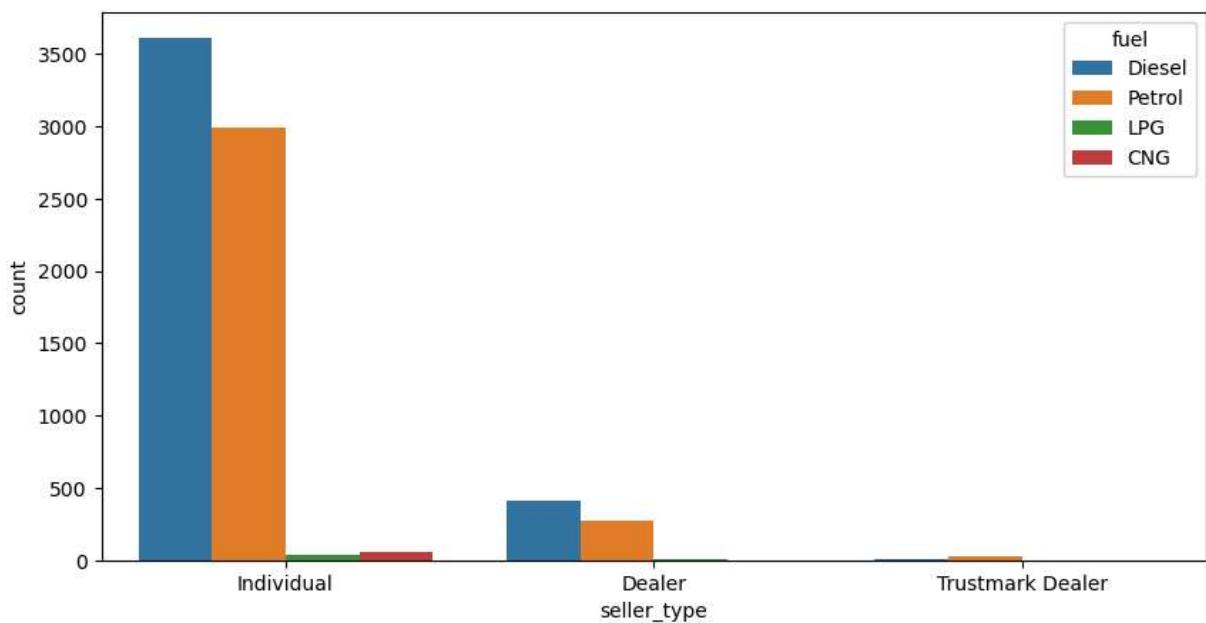
```
In [7]: plt.figure(figsize=(10,5))
sns.countplot(df,x="owner",hue="fuel")
plt.show()
```



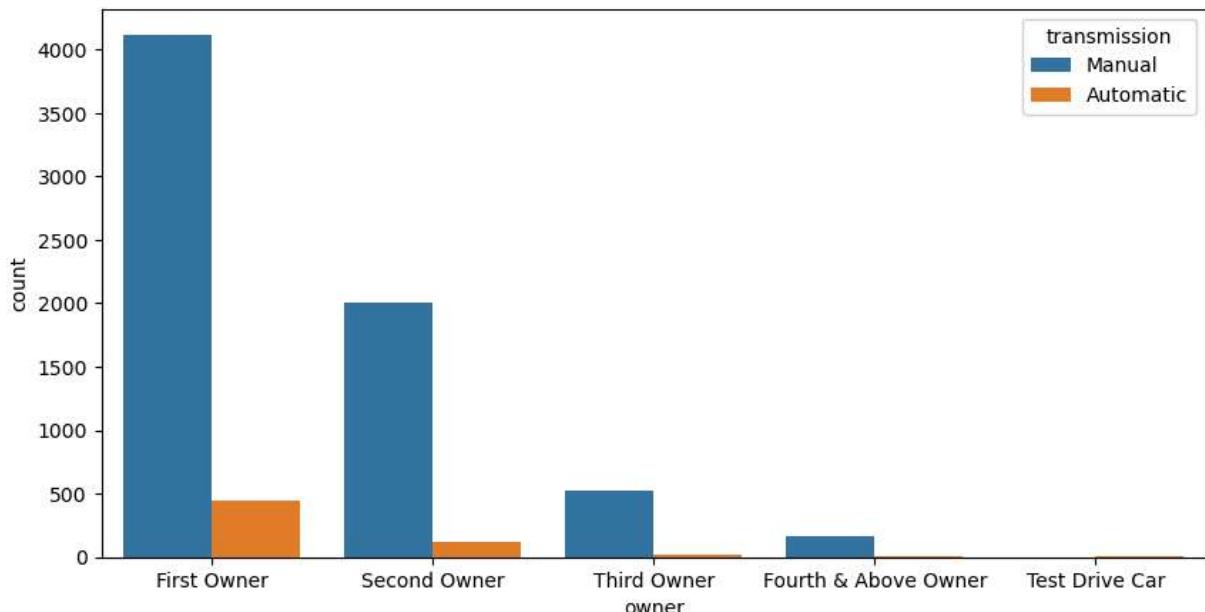
```
In [8]: sns.countplot(df["seller_type"])
plt.show()
```



```
In [9]: plt.figure(figsize=(10,5))
sns.countplot(df,x="seller_type",hue="fuel")
plt.show()
```



```
In [10]: plt.figure(figsize=(10,5))
sns.countplot(df,x="owner",hue="transmission")
plt.show()
```



```
In [11]: df["year"].nunique()
```

```
Out[11]: 29
```

```
In [12]: for index, row in df.iterrows():
    if row["selling_price"] == 1000000:
        print(index, row["name"])
```

272 Maruti Ertiga SHVS ZDI Plus
564 Toyota Innova 2.5 G (Diesel) 8 Seater
969 Hyundai Creta 1.4 EX Diesel
1193 Maruti Ciaz Alpha Diesel
1530 Toyota Yaris V BSIV
1531 Toyota Yaris V BSIV
1651 Volkswagen Vento 1.5 TDI Highline AT
1829 Maruti Ciaz S 1.3
2273 Toyota Innova 2.5 G (Diesel) 8 Seater
2429 Mahindra Ssangyong Rexton RX7
2624 Mahindra Scorpio 1.99 S4
2987 Hyundai Verna 1.6 VTIVT AT SX
3204 Mahindra XUV500 W8 2WD
3214 Toyota Innova 2.5 GX (Diesel) 8 Seater
3270 Hyundai Verna CRDi 1.6 AT SX Plus
3273 Renault Koleos 2.0 Diesel
3400 Mahindra Scorpio S3 9 Seater BSIV
3564 Maruti XL6 Alpha
3853 Mahindra XUV300 W8 Option BSIV
3945 Mahindra XUV500 W6 2WD
3992 Hyundai Verna VTIVT 1.6 SX Option
4271 Maruti S-Cross 2017-2020 Zeta DDIS 200 SH
4375 Mahindra Scorpio 1.99 S10 4WD
4414 Mahindra XUV500 W8 AWD
4528 BMW 5 Series 525d
4540 Mahindra XUV500 W10 2WD
5005 Mercedes-Benz New C-Class C 250 CDI Elegance
5269 Nissan Terrano XL 85 PS
5390 Hyundai Creta 1.6 CRDi AT SX Plus
5543 Mercedes-Benz E-Class E350 Petrol
5716 Hyundai Creta 1.6 CRDi SX Option
5848 BMW X1 sDrive20d
5925 Honda WR-V i-DTEC VX
6011 Hyundai Creta 1.6 CRDi SX Option
6435 Mahindra Scorpio S10 7 Seater
6644 Honda City i DTec VX
6732 Mahindra Marazzo M2 8Str BSIV
6770 Honda Mobilio RS Option i DTEC
6936 Hyundai Creta 1.6 CRDi SX Option
7036 Mahindra Scorpio S5 BSIV

In [13]: SP=df["selling_price"].unique()

In [14]: SP.sort()

In [15]: SP

```
Out[15]: array([
 29999, 30000, 31000, 31504, 33351, 33983,
 35000, 39000, 40000, 42000, 45000, 45957,
 46000, 50000, 52000, 54000, 55000, 55599,
 56000, 57000, 58000, 59000, 59259, 60000,
 64000, 65000, 66000, 67000, 67500, 68000,
 70000, 72000, 75000, 75527, 78000, 80000,
 80500, 81000, 83000, 85000, 86000, 88000,
 89000, 90000, 90177, 92000, 93000, 93150,
 94500, 95000, 96000, 98000, 99000, 100000,
 101000, 102000, 105000, 108000, 110000, 111000,
 112999, 114999, 115999, 118000, 119000, 120000,
 121000, 122000, 124000, 125000, 126000, 127000,
 128000, 129000, 130000, 131000, 132000, 135000,
 136000, 138000, 140000, 141000, 142000, 144000,
 145000, 148000, 149000, 150000, 151000, 152000,
 153000, 155000, 156000, 157000, 158000, 160000,
 161000, 163000, 165000, 166000, 168000, 169000,
 170000, 172000, 174000, 175000, 178000, 179000,
 180000, 181000, 184000, 185000, 187000, 189000,
 190000, 191000, 194000, 195000, 196000, 198000,
 199000, 200000, 202999, 204000, 204999, 206000,
 209000, 210000, 211000, 212000, 215000, 217000,
 219000, 220000, 221000, 222000, 225000, 227000,
 227999, 229999, 231000, 235000, 236000, 238000,
 240000, 241000, 242000, 244000, 245000, 246000,
 248000, 250000, 250999, 252000, 252999, 254000,
 254999, 256000, 256999, 258000, 259000, 260000,
 261000, 262000, 265000, 267000, 268000, 270000,
 272000, 275000, 276000, 277000, 278000, 279000,
 280000, 282000, 285000, 287000, 290000, 291000,
 292000, 295000, 297000, 298000, 299000, 300000,
 302000, 305000, 307000, 308000, 310000, 311000,
 312000, 313000, 315000, 316000, 317000, 318000,
 319000, 320000, 321000, 325000, 327000, 328000,
 329000, 330000, 335000, 337000, 339000, 340000,
 341000, 345000, 346000, 347000, 348000, 350000,
 351000, 352000, 355000, 356000, 357000, 358000,
 359000, 360000, 361000, 362000, 365000, 366000,
 369000, 370000, 371000, 372000, 374000, 375000,
 376000, 377000, 378000, 380000, 382000, 385000,
 386000, 387000, 388000, 389000, 390000, 391000,
 392000, 395000, 399000, 400000, 401000, 403000,
 404000, 405000, 408000, 409000, 409999, 411000,
 412000, 415000, 416000, 420000, 421000, 423000,
 425000, 426000, 426999, 428000, 429000, 430000,
 430999, 432000, 433000, 434000, 434999, 438999,
 440000, 441000, 444000, 445000, 448000, 449000,
 450000, 451000, 451999, 453000, 454000, 455000,
 455999, 458000, 459999, 465000, 466000, 467000,
 469000, 470000, 472000, 475000, 476999, 479000,
 480000, 482000, 483000, 484999, 486000, 487000,
 488000, 488999, 490000, 493000, 495000, 497000,
 499000, 500000, 501000, 503000, 509999, 511000,
 512000, 515000, 516000, 520000, 521000, 522000,
 524000, 525000, 526000, 528000, 530000, 531000,
 532000, 535000, 537000, 539000, 540000, 541000,
```

545000,	549000,	550000,	551000,	555000,	556000,
558000,	560000,	561000,	562000,	563000,	565000,
567000,	570000,	571000,	573000,	574000,	575000,
576000,	577000,	578000,	579000,	580000,	585000,
589000,	590000,	591000,	595000,	597000,	599000,
600000,	604000,	605000,	610000,	611000,	615000,
616000,	620000,	624000,	625000,	626000,	630000,
631000,	632000,	634000,	635000,	640000,	643000,
645000,	649000,	650000,	651000,	654000,	655000,
660000,	665000,	666000,	667000,	669000,	670000,
671000,	675000,	678000,	679000,	680000,	681000,
684000,	685000,	690000,	693000,	694000,	695000,
698000,	699000,	700000,	710000,	711000,	715000,
720000,	721000,	722000,	725000,	730000,	731000,
735000,	736000,	737000,	740000,	741000,	745000,
746000,	749000,	750000,	751000,	752000,	754000,
755000,	756000,	757000,	760000,	764000,	765000,
770000,	773000,	775000,	778000,	779000,	780000,
785000,	786000,	790000,	791000,	793000,	795000,
797000,	799000,	800000,	801000,	802999,	803999,
805000,	808000,	810000,	811000,	813000,	819999,
822000,	825000,	830000,	833000,	834000,	835000,
839000,	840000,	844999,	849000,	850000,	851000,
860000,	861000,	861999,	866000,	869999,	875000,
880000,	885000,	889000,	890000,	891000,	892000,
894999,	899000,	900000,	905000,	906000,	910000,
911000,	911999,	919999,	925000,	930000,	934000,
940000,	944999,	946000,	949000,	950000,	955000,
957000,	960000,	965000,	969999,	974000,	975000,
977000,	978999,	980000,	990000,	999000,	1000000,
1019999,	1025000,	1030000,	1031000,	1035000,	1040000,
1044999,	1050000,	1051000,	1075000,	1080000,	1085000,
1090000,	1100000,	1110000,	1125000,	1132000,	1140000,
1143000,	1145000,	1147000,	1149000,	1150000,	1151000,
1160000,	1165000,	1175000,	1180000,	1190000,	1200000,
1210000,	1220000,	1225000,	1227000,	1237000,	1250000,
1251000,	1265000,	1270000,	1275000,	1282000,	1290000,
1295000,	1300000,	1325000,	1350000,	1365000,	1380000,
1385000,	1390000,	1400000,	1405000,	1425000,	1445000,
1450000,	1460000,	1465000,	1475000,	1490000,	1500000,
1511000,	1515000,	1516000,	1520000,	1525000,	1530000,
1550000,	1560000,	1575000,	1576000,	1590000,	1594000,
1600000,	1625000,	1630000,	1650000,	1670000,	1675000,
1680000,	1689999,	1700000,	1745000,	1748999,	1750000,
1757000,	1789999,	1800000,	1825000,	1850000,	1859000,
1864999,	1888000,	1898999,	1900000,	1920000,	1925000,
1938000,	1950000,	2000000,	2051000,	2064000,	2100000,
2125000,	2150000,	2175000,	2199000,	2200000,	2280000,
2300000,	2375000,	2400000,	2450000,	2475000,	2500000,
2550000,	2575000,	2600000,	2625000,	2650000,	2700000,
2711000,	2750000,	2800000,	2825000,	2850000,	2900000,
2940000,	2950000,	3000000,	3100000,	3200000,	3250000,
3251000,	3300000,	3350000,	3400000,	3500000,	3600000,
3750000,	3790000,	3800000,	3900000,	3975000,	4000000,
4090000,	4100000,	4500000,	4600000,	5150000,	5200000,

```
5400000, 5500000, 5800000, 5830000, 5850000, 5923000,  
6000000, 6223000, 6523000, 7200000, 10000000], dtype=int64)
```

```
In [16]: for index, row in df.iterrows():  
    if row["selling_price"] in [ 29999,     30000,     31000,     31504,     33351,  
        46000,     50000,     52000,     54000,     55000,     55599, 56000,     57000,     580  
    print(index, row["name"])
```

8 Maruti 800 DX BSII
83 Tata Indica DLS
192 Maruti 800 Std
226 Maruti Omni E 8 Str STD
716 Tata Nano Cx
766 Hyundai Santro Xing XP
876 Maruti Esteem Vxi - BSII
921 Maruti 800 Std
1026 Maruti 800 AC
1194 Tata Indica V2 DLS TC
1212 Maruti 800 Std BSII
1227 Tata Nano Cx BSIV
1432 Maruti 800 DX
1438 Maruti 800 AC
1467 Maruti 800 Std BSII
1738 Maruti 800 AC
1814 Maruti 800 Std MPFi
1842 Maruti 800 Std MPFi
1880 Maruti 800 Std
1964 Maruti 800 AC BSII
2012 Maruti Wagon R LXI
2099 Maruti Zen LXI
2217 Maruti 800 AC
2300 Maruti Zen LX
2376 Maruti Baleno LXI
2397 Chevrolet Spark 1.0
2589 Maruti Wagon R LXI
2773 Maruti 800 Std
3000 Maruti 800 DX
3397 Hyundai Accent GLS
3476 Maruti 800 EX BSII
3745 Maruti 800 Std
3776 Hyundai Santro LE
3777 Maruti Zen LX
3875 Daewoo Matiz SD
3939 Maruti 800 DX
3991 Maruti 800 Std
4070 Hyundai Santro GLS I - Euro II
4094 Hyundai Santro LE
4309 Maruti 800 Std MPFi
4476 Maruti Wagon R VXI
4478 Maruti 800 Std
4523 Honda City 1.3 EXI
4578 Maruti 800 Std BSIII
4640 Honda City 1.3 EXI
4651 Maruti Zen LXI
4760 Maruti 800 AC Uniq
4927 Maruti Omni LPG STD BSIV
5016 Maruti 800 Std
5051 Maruti 800 Std
5064 Maruti 800 AC
5069 Maruti 800 Std MPFi
5252 Tata Indica DLX
5258 Hyundai Santro DX
5259 Maruti Zen LXI
5290 Tata Indica V2 DLE BSII

5293 Maruti 800 AC
5332 Maruti 800 Std
5396 Hyundai Accent GLE
5438 Hyundai Santro LP - Euro II
5467 Tata Indica V2 DiCOR DLG BS-III
5547 Maruti 800 Std
5559 Maruti Esteem Lxi - BSIII
5596 Maruti Omni LPG CARGO BSIII W IMMOBILISER
5650 Maruti 800 AC
5708 Maruti 800 Std
5711 Maruti 800 AC
5748 Maruti Alto LX
5832 Maruti 800 AC
5859 Hyundai Santro LS zipPlus
5896 Maruti Zen Std
5904 Maruti 800 Std
5942 Maruti 800 Std
5947 Maruti Zen VXI
6036 Maruti 800 DX
6090 Tata Indica V2 DLE BSII
6105 Maruti 800 AC
6185 Maruti 800 AC
6189 Peugeot 309 GLD
6198 Tata Nano Lx BSIV
6237 Hyundai Santro GLS I - Euro I
6244 Tata Indica V2 DLS TC
6394 Tata Indica V2 2001-2011 DLS BSIII
6395 Maruti 800 AC
6506 Maruti 800 AC
6670 Maruti 800 AC
6683 Tata Indica V2 DLX
6795 Tata Indica V2 DLX TC
6796 Ford Fiesta 1.4 ZXi TDCi Limited Edition
7040 Maruti 800 AC
7118 Maruti Zen Classic
7167 Tata Nano LX SE
7237 Maruti 800 AC
7259 Maruti Alto LX
7308 Maruti 800 AC
7374 Tata Nano Cx
7402 Maruti 800 AC

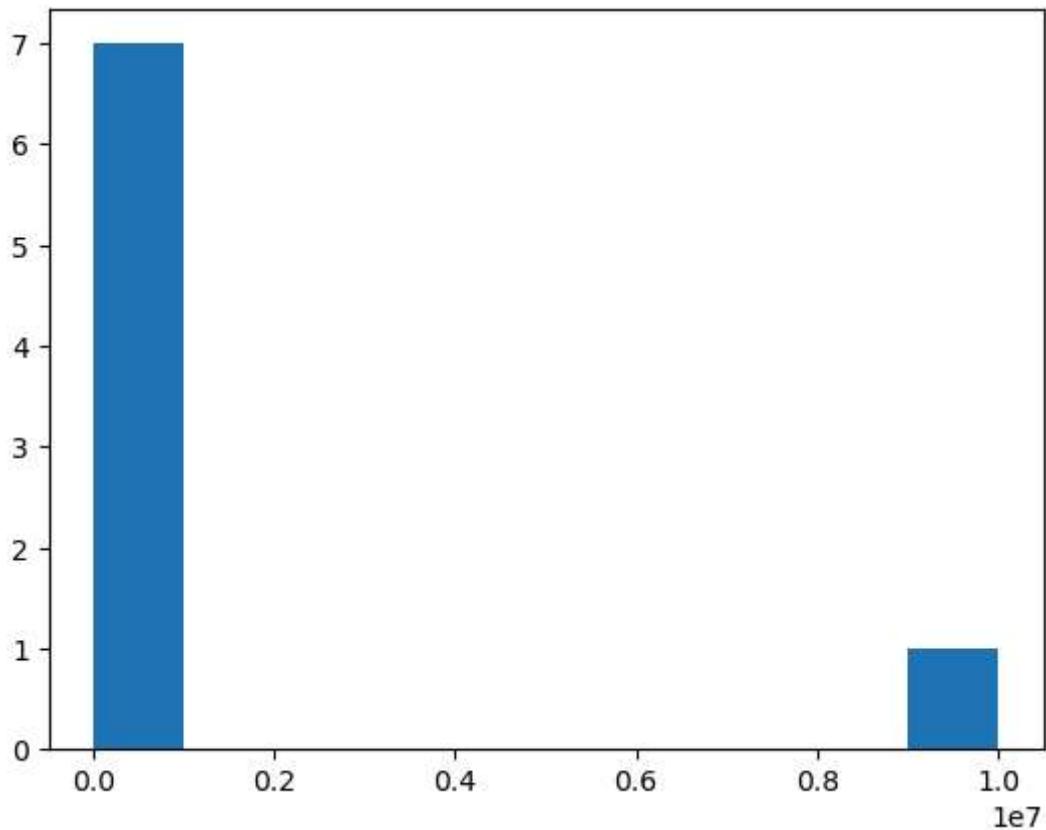
```
In [17]: for index, row in df.iterrows():
    if row["selling_price"] in [ 29999,      30000,      31000,      31504,      33351,
        46000,      50000,      52000,      54000,      55000,      55599, 56000,      57000,      580
        print(row["name"])
```

Maruti 800 DX BSII
Tata Indica DLS
Maruti 800 Std
Maruti Omni E 8 Str STD
Tata Nano Cx
Hyundai Santro Xing XP
Maruti Esteem Vxi - BSII
Maruti 800 Std
Maruti 800 AC
Tata Indica V2 DLS TC
Maruti 800 Std BSII
Tata Nano Cx BSIV
Maruti 800 DX
Maruti 800 AC
Maruti 800 Std BSII
Maruti 800 AC
Maruti 800 Std MPFi
Maruti 800 Std MPFi
Maruti 800 Std
Maruti 800 AC BSII
Maruti Wagon R LXI
Maruti Zen LXI
Maruti 800 AC
Maruti Zen LX
Maruti Baleno LXI
Chevrolet Spark 1.0
Maruti Wagon R LXI
Maruti 800 Std
Maruti 800 DX
Hyundai Accent GLS
Maruti 800 EX BSII
Maruti 800 Std
Hyundai Santro LE
Maruti Zen LX
Daewoo Matiz SD
Maruti 800 DX
Maruti 800 Std
Hyundai Santro GLS I - Euro II
Hyundai Santro LE
Maruti 800 Std MPFi
Maruti Wagon R VXI
Maruti 800 Std
Honda City 1.3 EXI
Maruti 800 Std BSIII
Honda City 1.3 EXI
Maruti Zen LXI
Maruti 800 AC Uniq
Maruti Omni LPG STD BSIV
Maruti 800 Std
Maruti 800 Std
Maruti 800 AC
Maruti 800 Std MPFi
Tata Indica DLX
Hyundai Santro DX
Maruti Zen LXI
Tata Indica V2 DLE BSII

Maruti 800 AC
 Maruti 800 Std
 Hyundai Accent GLE
 Hyundai Santro LP - Euro II
 Tata Indica V2 DiCOR DLG BS-III
 Maruti 800 Std
 Maruti Esteem Lxi - BSIII
 Maruti Omni LPG CARGO BSIII W IMMOBILISER
 Maruti 800 AC
 Maruti 800 Std
 Maruti 800 AC
 Maruti Alto LX
 Maruti 800 AC
 Hyundai Santro LS zipPlus
 Maruti Zen Std
 Maruti 800 Std
 Maruti 800 Std
 Maruti Zen VXI
 Maruti 800 DX
 Tata Indica V2 DLE BSII
 Maruti 800 AC
 Maruti 800 AC
 Peugeot 309 GLD
 Tata Nano Lx BSIV
 Hyundai Santro GLS I - Euro I
 Tata Indica V2 DLS TC
 Tata Indica V2 2001-2011 DLS BSIII
 Maruti 800 AC
 Maruti 800 AC
 Maruti 800 AC
 Tata Indica V2 DLX
 Tata Indica V2 DLX TC
 Ford Fiesta 1.4 ZXi TDCi Limited Edition
 Maruti 800 AC
 Maruti Zen Classic
 Tata Nano LX SE
 Maruti 800 AC
 Maruti Alto LX
 Maruti 800 AC
 Tata Nano Cx
 Maruti 800 AC

```
In [38]: #What is the distribution of selling_price?
plt.hist(df["selling_price"].describe())
```

```
Out[38]: (array([7., 0., 0., 0., 0., 0., 0., 0., 1.]),
 array([7.4120000e+03, 1.0066708e+06, 2.0059296e+06, 3.0051884e+06,
        4.0044472e+06, 5.0037060e+06, 6.0029648e+06, 7.0022236e+06,
        8.0014824e+06, 9.0007412e+06, 1.0000000e+07]),
 <BarContainer object of 10 artists>)
```



```
In [42]: #What are the average km_driven for each fuel type?
df.groupby("fuel")["km_driven"].mean().round()
```

```
Out[42]: fuel
CNG      69008.0
Diesel   88296.0
LPG      89174.0
Petrol   55694.0
Name: km_driven, dtype: float64
```

```
In [44]: #How many cars of each seller_type are there?
df["seller_type"].value_counts()
```

```
Out[44]: seller_type
Individual      6698
Dealer          687
Trustmark Dealer    27
Name: count, dtype: int64
```

```
In [54]: #What is the range of year values in the dataset?
df["year"].min() , df["year"].max() , df["year"].nunique()
```

```
Out[54]: (1983, 2020, 29)
```

```
In [64]: #What are the most common car names?
df["name"].value_counts().head(10)
```

```
Out[64]: name
Maruti Swift Dzire VDI      168
Maruti Alto 800 LXI         92
Maruti Alto LXi             77
Maruti Swift VDI BSIV       68
Maruti Swift VDI            66
Maruti Alto K10 VXI         53
Hyundai EON Era Plus        50
Maruti Alto LX               49
Maruti Wagon R VXI BS IV    47
Hyundai Verna 1.6 SX         45
Name: count, dtype: int64
```

```
In [80]: #What is the average mileage and engine size per fuel type?
df.groupby("fuel")[["mileage(kmpl)", "engine(CC)"]].mean()
```

fuel	mileage(kmpl)	engine(CC)
CNG	23.781724	1096.017241
Diesel	19.874258	1651.160497
LPG	18.665789	1047.605263
Petrol	19.160109	1140.693710

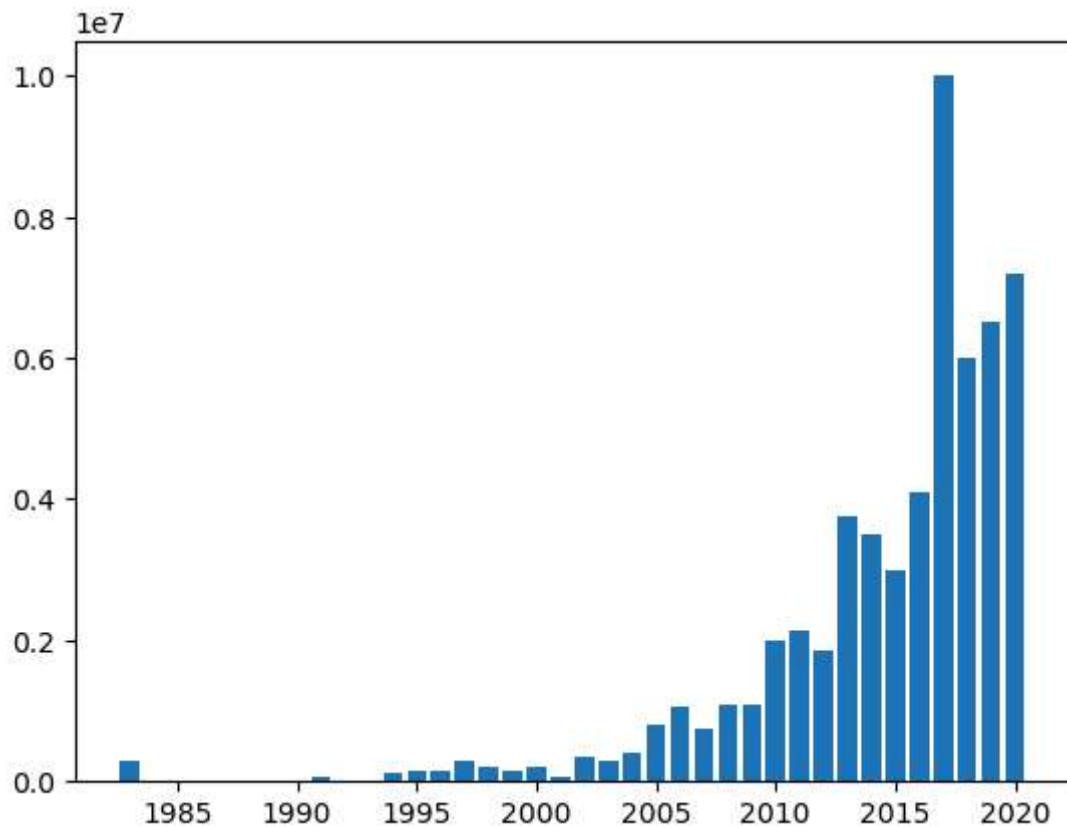
```
In [88]: #Is there a correlation between km_driven and selling_price?
df["km_driven"].corr(df["selling_price"])
```

```
Out[88]: -0.16611899163278507
```

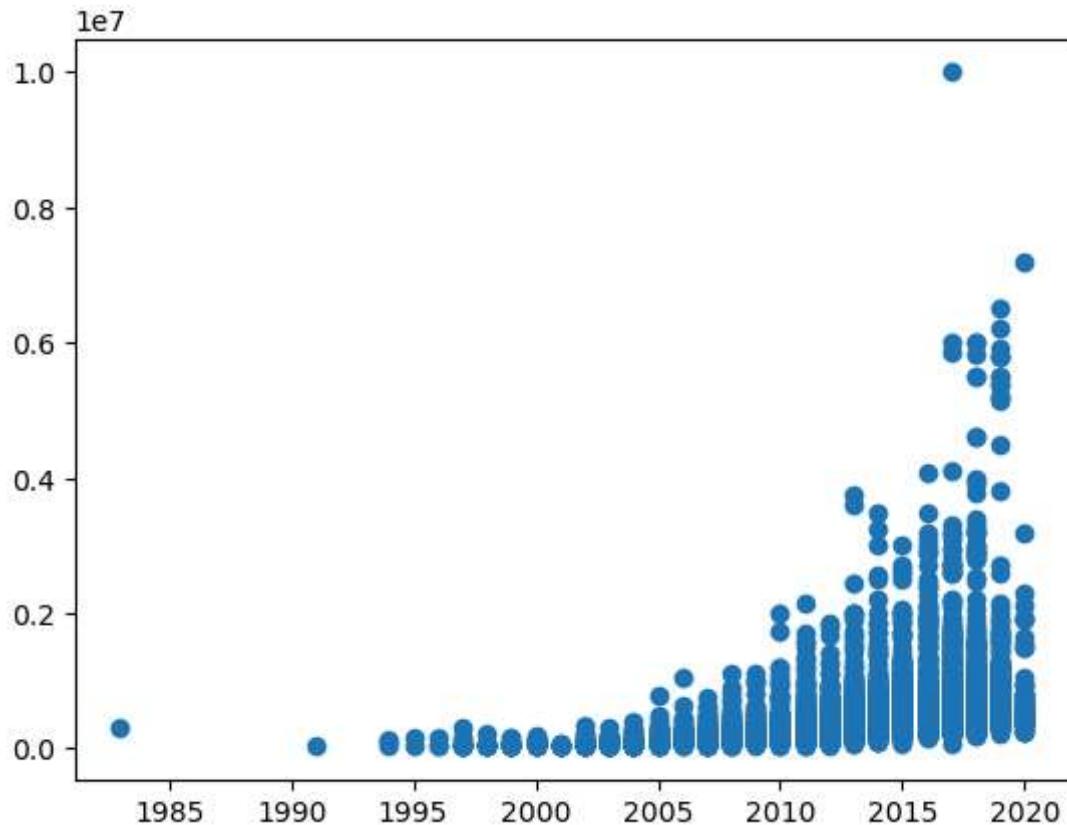
```
In [114...]: df["km_driven"].describe()
```

```
Out[114...]: count    7.412000e+03
mean     7.367432e+04
std      5.732998e+04
min      1.000000e+00
25%      4.000000e+04
50%      7.000000e+04
75%      1.000000e+05
max      2.360457e+06
Name: km_driven, dtype: float64
```

```
In [132...]: #How does selling_price vary with year?
plt.bar(df["year"], df["selling_price"])
plt.show()
```



```
In [134]:  
plt.scatter(df["year"], df["selling_price"])  
plt.show()
```

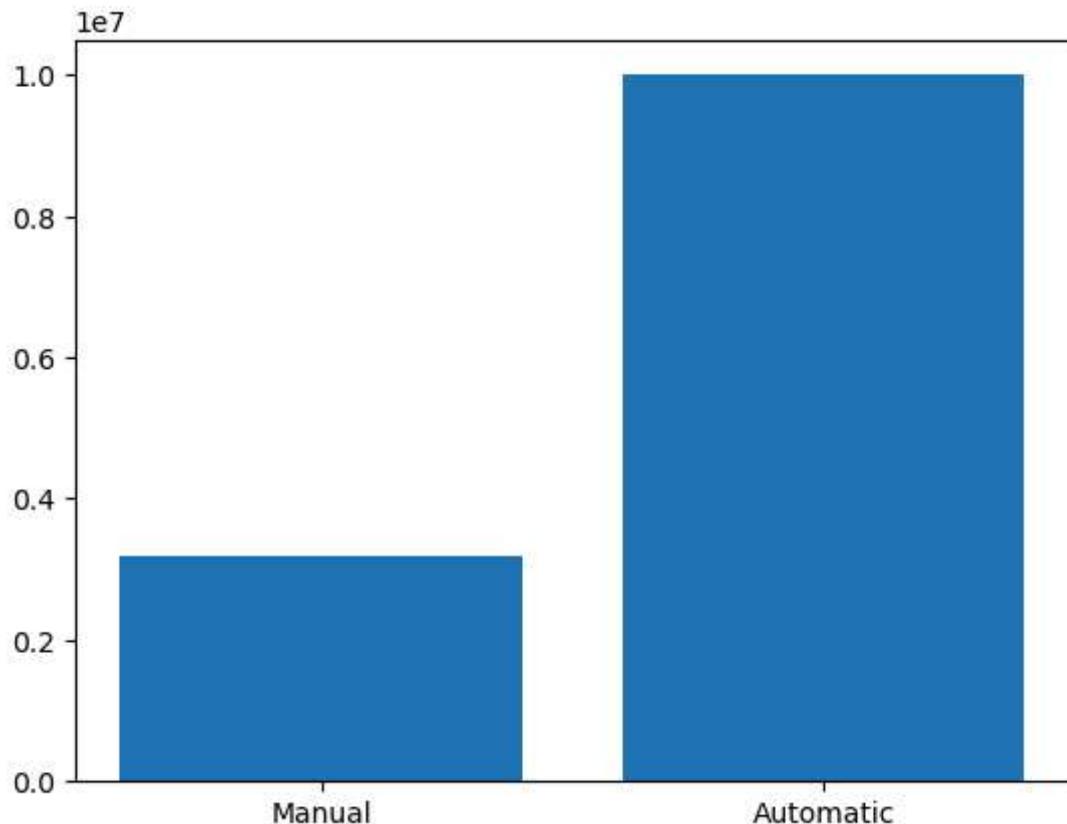


```
In [146... #Does transmission type affect selling_price?  
df.groupby("transmission")["selling_price"].describe()
```

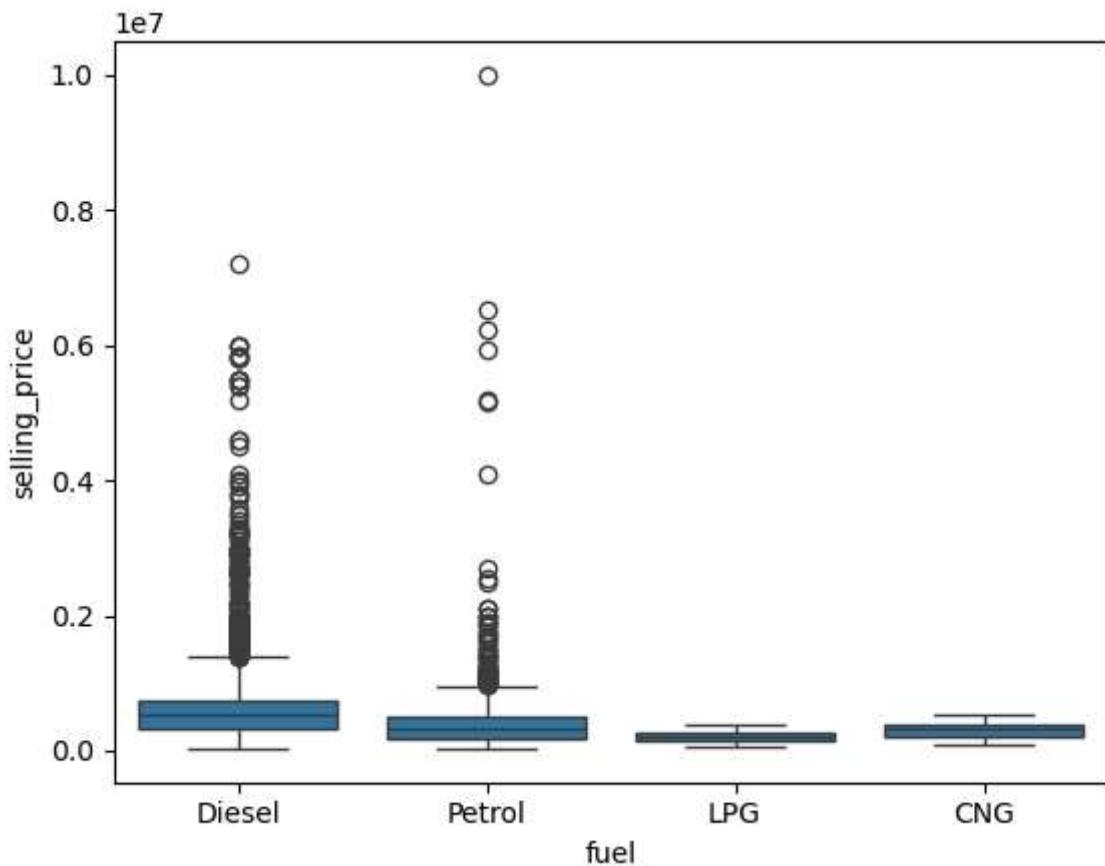
Out[146...]

transmission	count	mean	std	min	25%	50%	75%
Automatic	598.0	1.317120e+06	1.270499e+06	75000.0	550000.0	850000.0	1650000.0
Manual	6814.0	4.443532e+05	2.917113e+05	29999.0	235000.0	390000.0	600000.0

```
In [149... plt.bar(df["transmission"],df["selling_price"])  
plt.show()
```



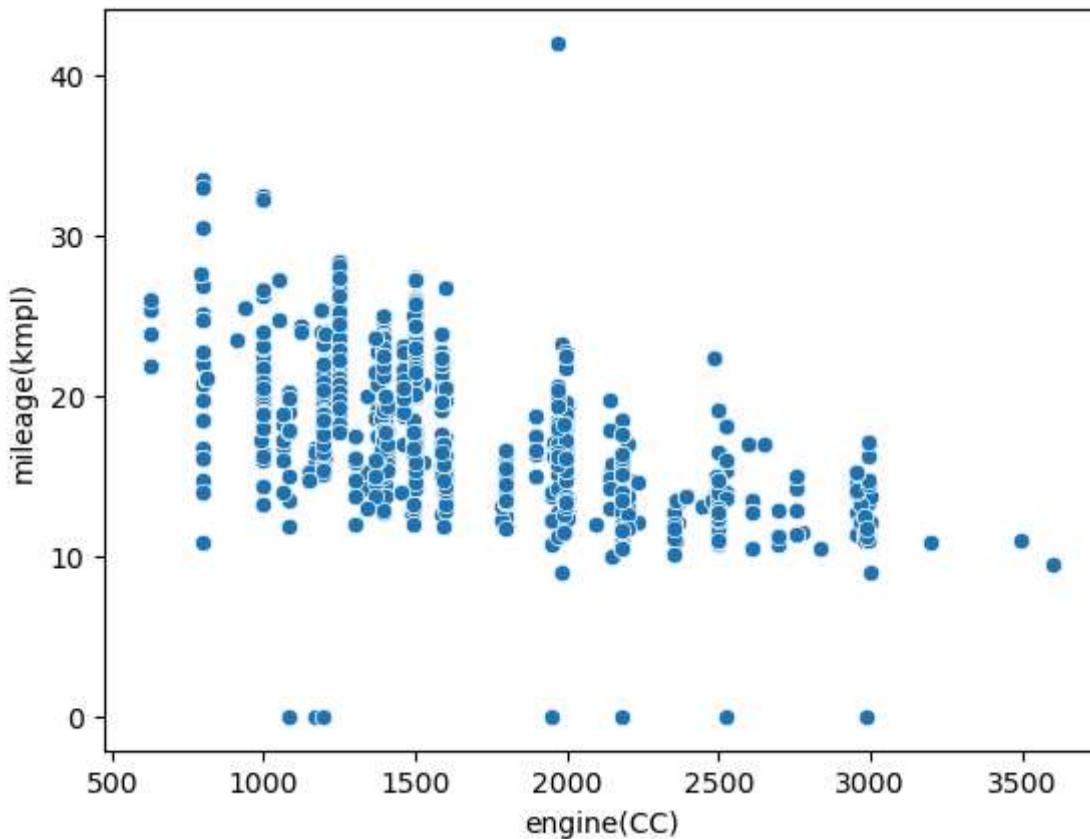
```
In [155... #How does fuel type affect selling_price?  
sns.boxplot(x = df["fuel"], y = df["selling_price"])  
plt.show()
```



```
In [161]: df["engine(CC)"].astype(int)
```

```
Out[161]: 0      1248  
1      1498  
2      1497  
3      1497  
4      1396  
...  
7407    998  
7408    1396  
7409    1197  
7410    1493  
7411    1248  
Name: engine(CC), Length: 7412, dtype: int32
```

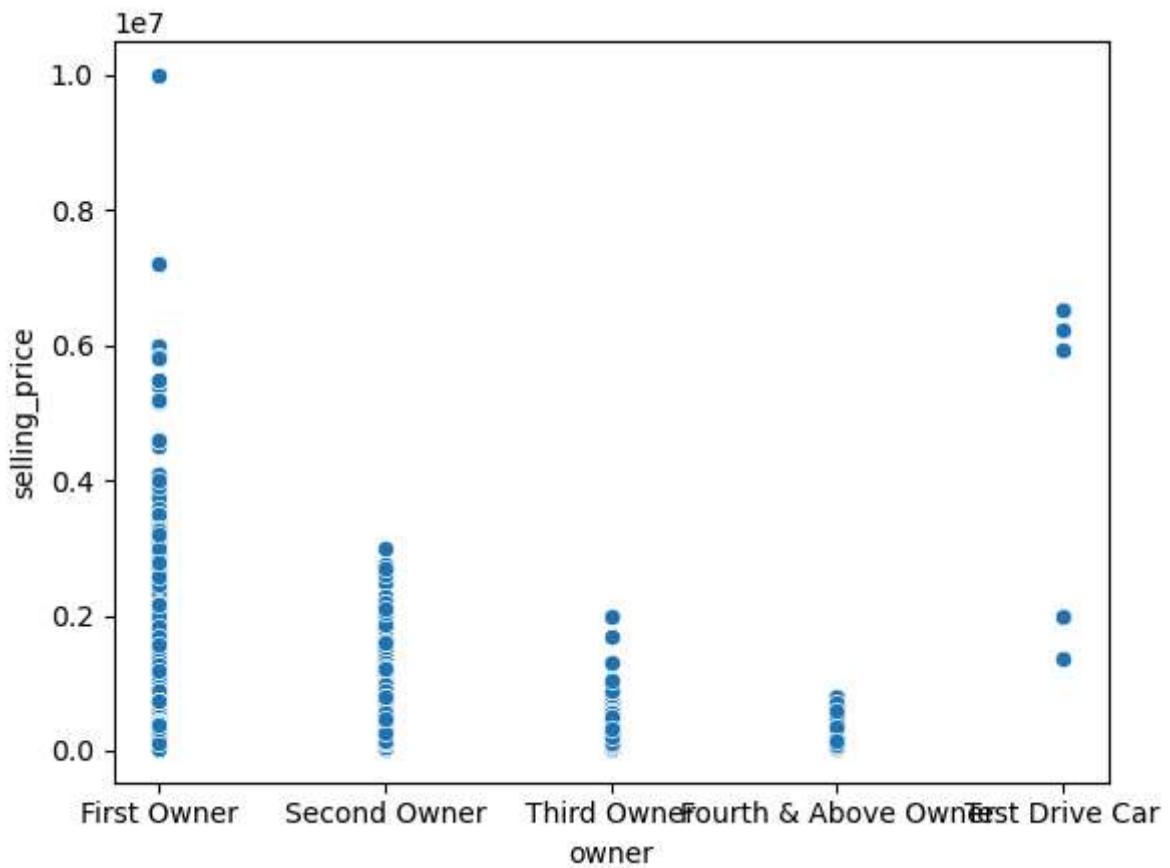
```
In [171]: #Is there a relationship between engine(CC) and max_power(bhp)?  
sns.scatterplot(df,x="engine(CC)",y="mileage(kmpl)")  
plt.show()
```



```
In [177]: #Does the number of owners affect the selling_price?
df.groupby("owner")["selling_price"].describe()
```

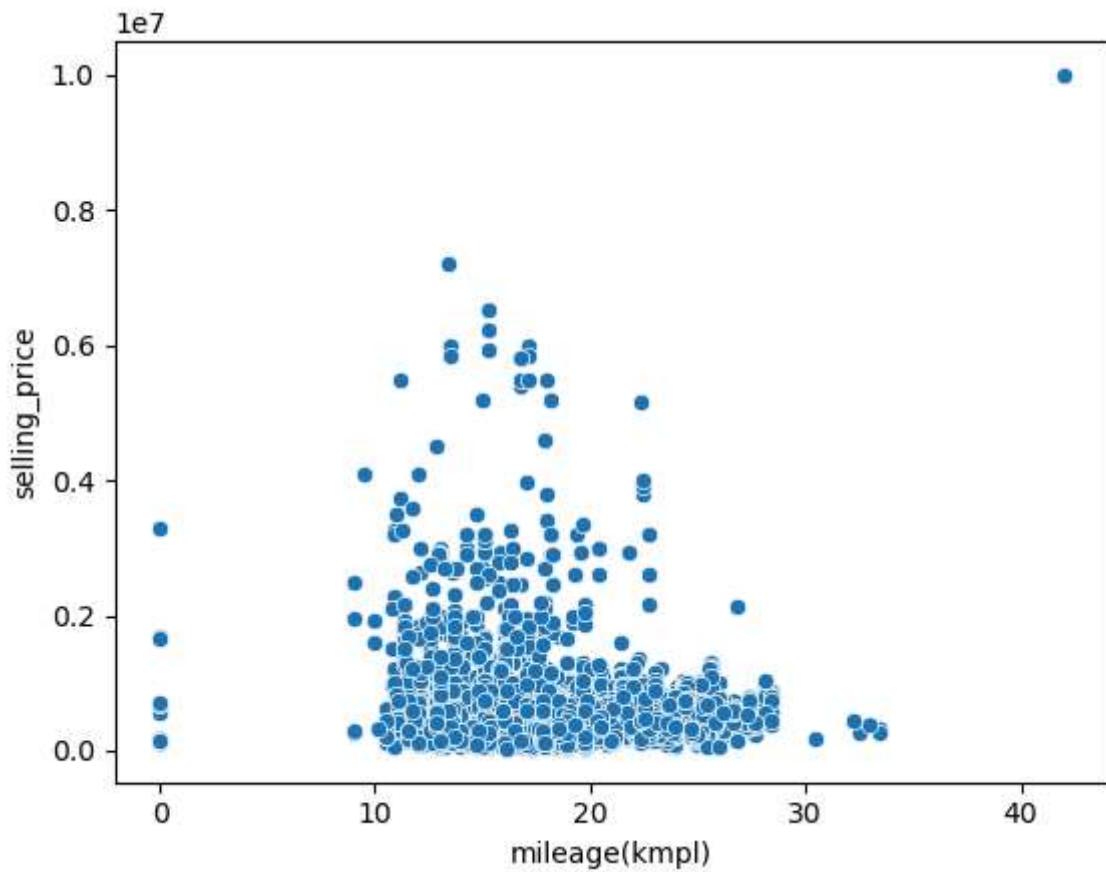
	count	mean	std	min	25%	50%	75%
owner							
First Owner	4562.0	6.121627e+05	5.723397e+05	30000.0	320000.0	500000.0	700000.0
Fourth & Above Owner	171.0	2.224883e+05	1.550226e+05	31000.0	100000.0	170000.0	320000.0
Second Owner	2126.0	3.808943e+05	3.148198e+05	33983.0	185000.0	300000.0	480000.0
Test Drive Car	5.0	4.403800e+06	2.510598e+06	1350000.0	2000000.0	5923000.0	6223000.0
Third Owner	548.0	2.790666e+05	2.186593e+05	29999.0	140000.0	220000.0	350000.0

```
In [181]: sns.scatterplot(df,x = "owner" , y = "selling_price")
plt.show()
```



In [183...]

```
#How does mileage affect selling price? (scatter plot and correlation)
sns.scatterplot(df,x = "mileage(kmpl)" , y = "selling_price")
plt.show()
```



```
In [185]: df.loc[df["selling_price"] == 1000000]
```

Out[185...]

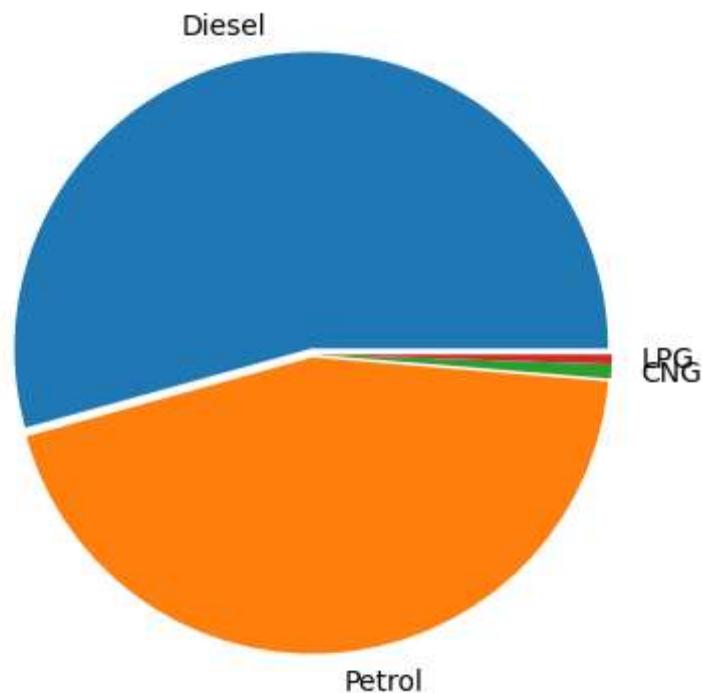
		name	year	km_driven	fuel	seller_type	transmission	owner	torque
272		Maruti Ertiga SHVS ZDI Plus	2016	70000	Diesel	Individual	Manual	First Owner	200Nm@ 1750rpm
564		Toyota Innova 2.5 G (Diesel) 8 Seater	2013	190000	Diesel	Individual	Manual	First Owner	200Nm@ 1200-3600rpm
969		Hyundai Creta 1.4 EX Diesel	2019	15000	Diesel	Individual	Manual	First Owner	219.7Nm@ 1500-2750rpm
1193		Maruti Ciaz Alpha Diesel	2018	50000	Diesel	Individual	Manual	First Owner	200Nm@ 1750rpm
1530		Toyota Yaris V BSIV	2018	4295	Petrol	Trustmark Dealer	Manual	First Owner	140Nm@ 4200rpm
1531		Toyota Yaris V BSIV	2018	5229	Petrol	Trustmark Dealer	Manual	First Owner	140Nm@ 4200rpm
1651		Volkswagen Vento 1.5 TDI Highline AT	2016	60000	Diesel	Individual	Automatic	First Owner	250Nm@ 1500-3000rpm
1829		Maruti Ciaz S 1.3	2018	40000	Diesel	Individual	Manual	First Owner	200Nm@ 1750rpm
2273		Toyota Innova 2.5 G (Diesel) 8 Seater	2016	65000	Diesel	Individual	Manual	First Owner	200Nm@ 1200-3600rpm
2429		Mahindra Ssangyong Rexton RX7	2013	40000	Diesel	Individual	Automatic	First Owner	402Nm@ 1600-3000rpm
2624		Mahindra Scorpio 1.99 S4	2015	35000	Diesel	Individual	Manual	First Owner	280Nm@ 1800-2800rpm
2987		Hyundai Verna 1.6 VTVT AT SX	2017	15000	Petrol	Individual	Automatic	First Owner	154.9Nm@ 4200rpm
3204		Mahindra XUV500 W8 2WD	2014	120000	Diesel	Individual	Manual	Third Owner	330Nm@ 1600-2800rpm
3214		Toyota Innova 2.5 GX (Diesel) 8 Seater	2015	50000	Diesel	Individual	Manual	First Owner	200Nm@ 1200-3600rpm

		name	year	km_driven	fuel	seller_type	transmission	owner	torque
3270		Hyundai Verna CRDi 1.6 AT SX Plus	2018	30000	Diesel	Individual	Automatic	First Owner	259.87Nm@ 1500-3000rpm
3273		Renault Koleos 2.0 Diesel	2013	75010	Diesel	Dealer	Automatic	First Owner	320Nm@ 2000rpm
3400		Mahindra Scorpio S3 9 Seater BSIV	2018	35000	Diesel	Individual	Manual	First Owner	200Nm@ 1400-2200rpm
3564		Maruti XL6 Alpha	2019	15000	Petrol	Individual	Manual	First Owner	138nm@ 4400rpm
3853		Mahindra XUV300 W8 Option BSIV	2019	22000	Petrol	Individual	Manual	First Owner	200Nm@ 2000-3500rpm
3945		Mahindra XUV500 W6 2WD	2016	100000	Diesel	Individual	Manual	First Owner	330Nm@ 1600-2800rpm
3992		Hyundai Verna VTiT 1.6 SX Option	2018	10000	Petrol	Individual	Manual	First Owner	151Nm@ 4850rpm
4271		Maruti S-Cross 2017-2020 Zeta DDiS 200 SH	2018	70000	Diesel	Individual	Manual	First Owner	200Nm@ 1750rpm
4375		Mahindra Scorpio 1.99 S10 4WD	2016	112000	Diesel	Dealer	Manual	Second Owner	280Nm@ 1800-2800rpm
4414		Mahindra XUV500 W8 AWD	2014	50000	Diesel	Individual	Manual	First Owner	330Nm@ 1600-2800rpm
4528		BMW 5 Series 525d	2010	60000	Diesel	Individual	Automatic	First Owner	450Nm@ 1750-2500rpm
4540		Mahindra XUV500 W10 2WD	2015	24700	Diesel	Individual	Manual	First Owner	330Nm@ 1600-2800rpm

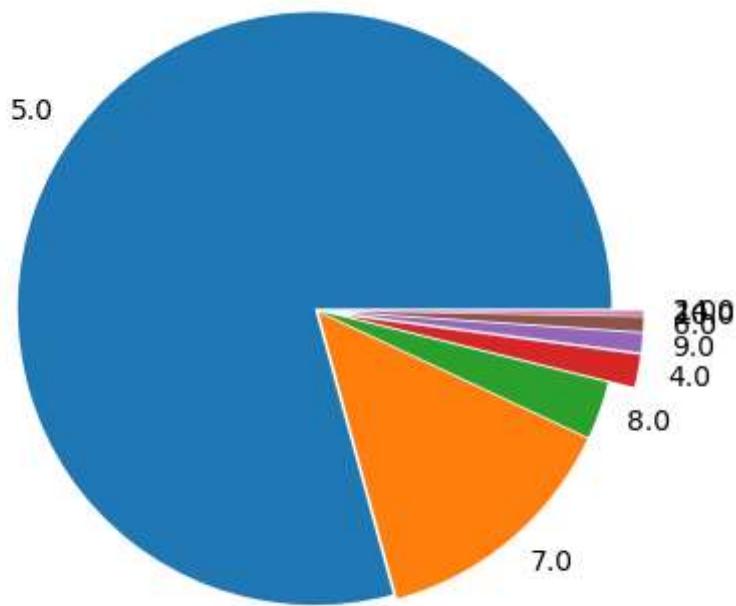
		name	year	km_driven	fuel	seller_type	transmission	owner	torque
5005		Mercedes-Benz New C-Class C 250 CDI Elegance	2011	90000	Diesel	Dealer	Automatic	First Owner	500Nm@ 1600-1800rpm
5269		Nissan Terrano XL 85 PS	2015	110000	Diesel	Individual	Manual	First Owner	200Nm@ 1900rpm
5390		Hyundai Creta 1.6 CRDi AT SX Plus	2017	60000	Diesel	Individual	Automatic	Second Owner	259.9Nm@ 1900-2750rpm
5543		Mercedes-Benz E-Class E350 Petrol	2009	80000	Petrol	Individual	Automatic	Third Owner	355Nm@ 4500rpm
5716		Hyundai Creta 1.6 CRDi SX Option	2015	70000	Diesel	Individual	Manual	First Owner	259.9Nm@ 1900-2750rpm
5848		BMW X1 sDrive20d	2011	50000	Diesel	Individual	Automatic	First Owner	380Nm@ 1750-2750rpm
5925		Honda WR-V i-DTEC VX	2019	72000	Diesel	Individual	Manual	First Owner	200Nm@ 1750rpm
6011		Hyundai Creta 1.6 CRDi SX Option	2015	50000	Diesel	Individual	Manual	First Owner	259.9Nm@ 1900-2750rpm
6435		Mahindra Scorpio S10 7 Seater	2015	133000	Diesel	Individual	Manual	Second Owner	280Nm@ 1800-2800rpm
6644		Honda City i DTec VX	2016	50000	Diesel	Individual	Manual	Second Owner	200Nm@ 1750rpm
6732		Mahindra Marazzo M2 8Str BSIV	2019	17000	Diesel	Dealer	Manual	First Owner	300Nm@ 1750-2500rpm
6770		Honda Mobilio RS Option i DTEC	2015	46000	Diesel	Individual	Manual	First Owner	200Nm@ 1750rpm

	name	year	km_driven	fuel	seller_type	transmission	owner	torque
6936	Hyundai Creta 1.6 CRDi SX Option	2015	73500	Diesel	Individual	Manual	First Owner	259.9Nm@ 1900- 2750rpm
7036	Mahindra Scorpio S5 BSIV	2019	60000	Diesel	Individual	Manual	First Owner	280Nm@ 1800- 2800rpm

```
In [215]: plt.pie(x=df["fuel"].value_counts(), labels = ["Diesel","Petrol","CNG","LPG"], explode=0, autopct='%.2f', shadow=True)
```



```
In [235]: plt.pie(x=df["seats"].value_counts(), labels=[5.0,7.0,8.0,4.0,9.0,6.0 ,10.0,2.0,14.0], autopct='%.2f')
```



In [159]: df.info

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7412 entries, 0 to 7411
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   name             7412 non-null   object  
 1   year              7412 non-null   int64  
 2   km_driven         7412 non-null   int64  
 3   fuel              7412 non-null   object  
 4   seller_type        7412 non-null   object  
 5   transmission       7412 non-null   object  
 6   owner              7412 non-null   object  
 7   torque             7412 non-null   object  
 8   seats              7412 non-null   float64 
 9   mileage(kmpl)     7412 non-null   float64 
 10  engine(CC)        7412 non-null   float64 
 11  max_power(bhp)    7412 non-null   float64 
 12  selling_price     7412 non-null   int64  
dtypes: float64(4), int64(3), object(6)
memory usage: 752.9+ KB
```

In [84]: df.head(1)

Out[84]:

	name	year	km_driven	fuel	seller_type	transmission	owner	torque	seats	mile
0	Maruti Swift Dzire VDI	2014	145500	Diesel	Individual	Manual	First Owner	190Nm@ 2000rpm		5.0

In []: