

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df=pd.read_csv('/content/train-data.csv')
```

df

	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	Ne
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First	26.6 km/kg	998 CC	58.16 bhp	5.0	
1	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67 kmpl	1582 CC	126.2 bhp	5.0	
2	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	18.2 kmpl	1199 CC	88.7 bhp	5.0	8
3	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77 kmpl	1248 CC	88.76 bhp	7.0	
4	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.2 kmpl	1968 CC	140.8 bhp	5.0	


df.head()

	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_P
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First	26.6 km/kg	998 CC	58.16 bhp	5.0	
1	1	Hyundai Creta 1.6 CRDi SX	Pune	2015	41000	Diesel	Manual	First	19.67 kmpl	1582 CC	126.2 bhp	5.0	

df.dtypes


	0
Unnamed: 0	int64
Name	object
Location	object
Year	int64
Kilometers_Driven	int64
Fuel_Type	object
Transmission	object
Owner_Type	object
Mileage	object
Engine	object
Power	object
Seats	float64
New_Price	object
Price	float64

df.isna().sum()



	0
Unnamed: 0	0
Name	0
Location	0
Year	0
Kilometers_Driven	0
Fuel_Type	0
Transmission	0
Owner_Type	0
Mileage	2
Engine	36
Power	36
Seats	42
New_Price	5195
Price	0

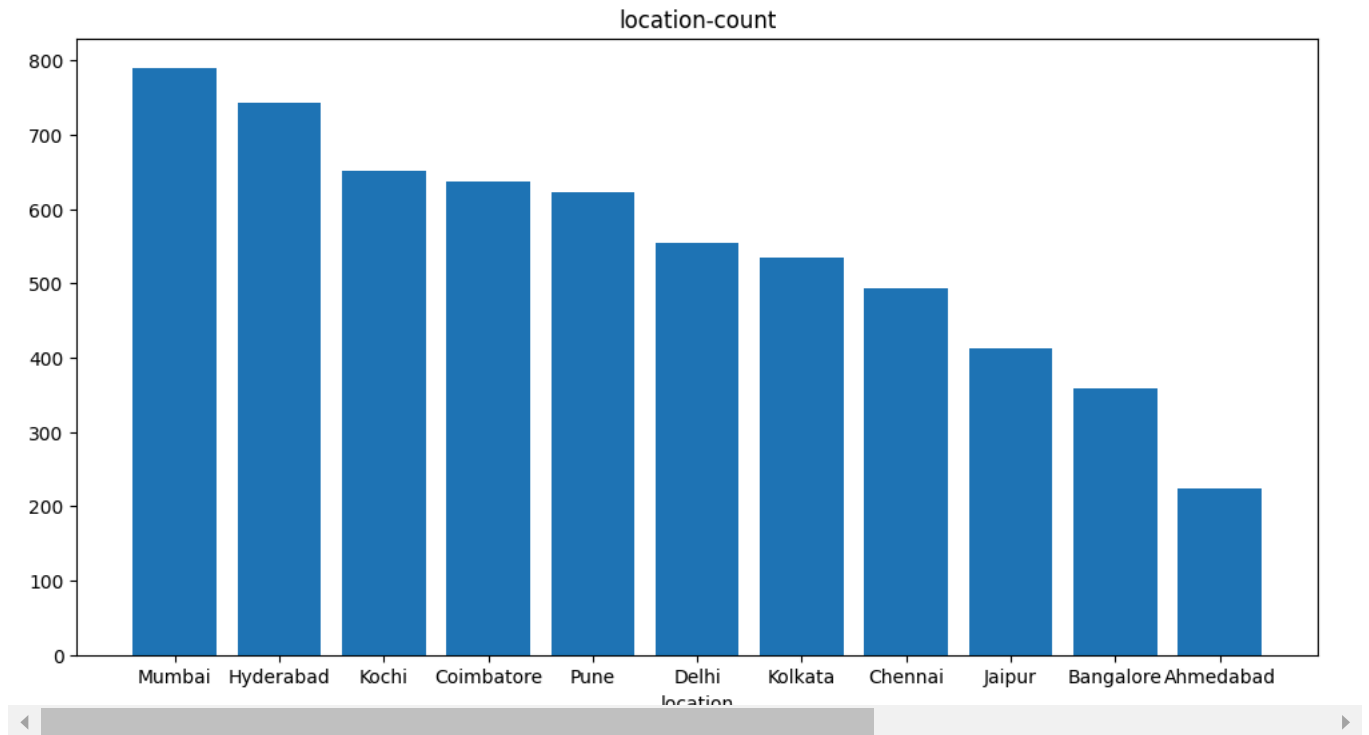
```
df1=df['Location'].value_counts()
df1
```



	count
Location	
Mumbai	790
Hyderabad	742
Kochi	651
Coimbatore	636
Pune	622
Delhi	554
Kolkata	535
Chennai	494
Jaipur	413
Bangalore	358
Ahmedabad	224

```
plt.figure(figsize=(12,6))
plt.bar(df1.index,df1)
plt.xlabel('location')
plt.title('location-count')
```

↻ Text(0.5, 1.0, 'location-count')



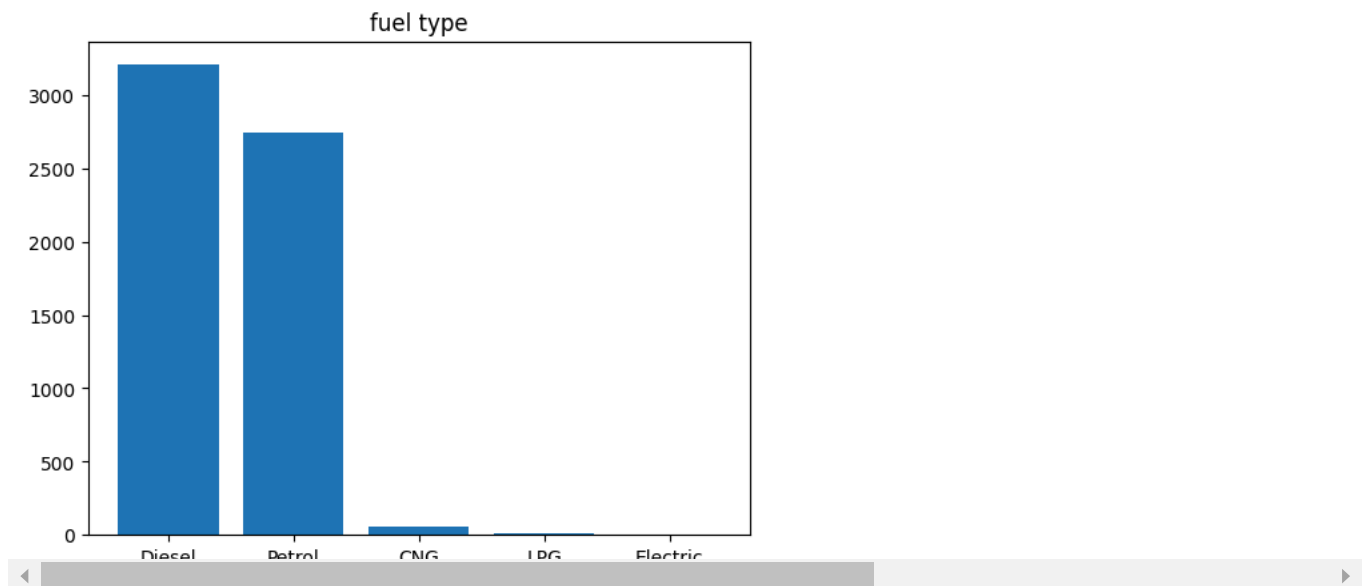
```
df2=df['Fuel_Type'].value_counts()  
df2
```

↻

	count
Fuel_Type	
Diesel	3205
Petrol	2746
CNG	56
LPG	10
Electric	2

```
plt.bar(df2.index,df2.values)  
plt.title('fuel type')
```

↻ Text(0.5, 1.0, 'fuel type')



```
df3=df['Transmission'].value_counts()  
df3
```



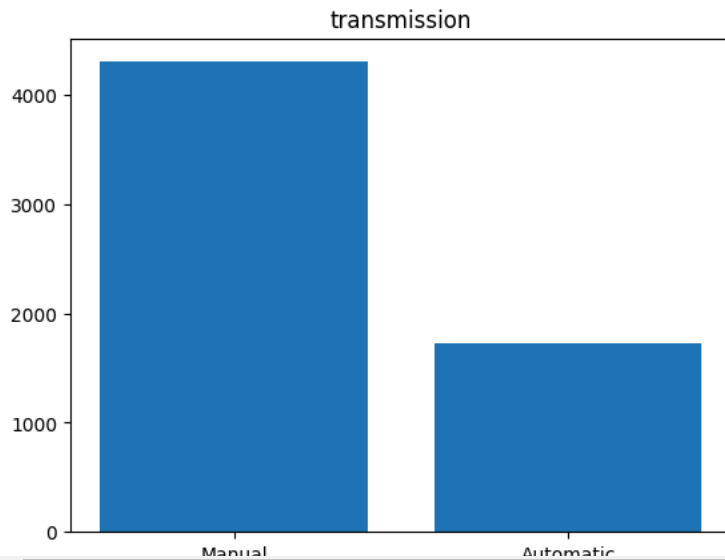
count	
Transmission	
Manual	4299
Automatic	1720



```
plt.bar(df3.index,df3)\nplt.title('transmission')
```



```
Text(0.5, 1.0, 'transmission')
```



```
df4=df['Owner_Type'].value_counts()\ndf4
```



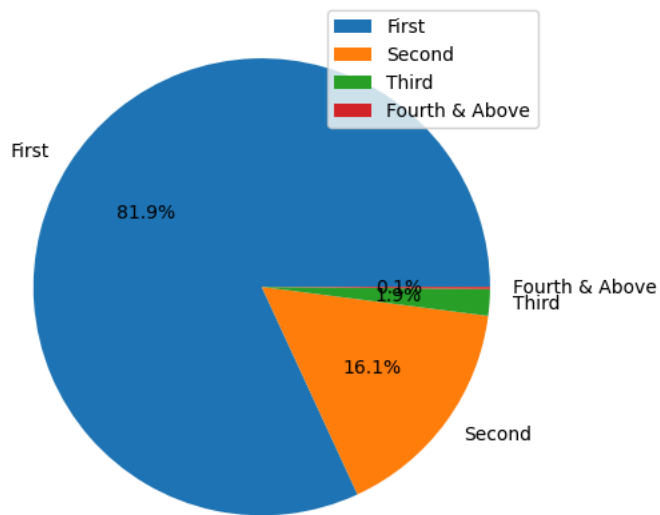
count	
Owner_Type	
First	4929
Second	968
Third	113
Fourth & Above	9



```
plt.pie(df4,autopct='%1.01f%%',labels=df4.index)\nplt.title('owner type')\nplt.legend(loc='upper right')\nplt.tight_layout()
```



owner type



```
df5=df['Mileage'].value_counts()
df5
```



count	
Mileage	
18.9 kmpl	172
17.0 kmpl	172
18.6 kmpl	119
20.36 kmpl	88
21.1 kmpl	86
...	...
27.28 kmpl	1
14.57 kmpl	1
22.8 km/kg	1
8.0 kmpl	1
17.24 kmpl	1

442 rows × 1 columns

```
df['Brand']=df['Name'].apply(lambda x:x.split())[0])
```

```
df['Brand'].value_counts()
```



	count
Brand	
Maruti	1211
Hyundai	1107
Honda	608
Toyota	411
Mercedes-Benz	318
Volkswagen	315
Ford	300
Mahindra	272
BMW	267
Audi	236
Tata	186
Skoda	173
Renault	145
Chevrolet	121
Nissan	91
Land	60
Jaguar	40
Fiat	28
Mitsubishi	27
Mini	26
Volvo	21
Porsche	18
Jeep	15
Datsun	13
Force	3
ISUZU	2
Smart	1
Ambassador	1
Isuzu	1
Bentley	1
Lamborghini	1



```
df['Brand'].shape
```



```
(6019,)
```

```
df1=pd.get_dummies(df[['Brand','Location','Fuel_Type','Transmission','Owner_Type']],dtype=int,drop_first=True)
df1
```



	Brand_Audi	Brand_BMW	Brand_Bentley	Brand_Chevrolet	Brand_Datsun	Brand_Fiat	Brand_Force	Brand_Ford	Brand_Honda	Brand_H
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	1
3	0	0	0	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0	0	0
...
6014	0	0	0	0	0	0	0	0	0	0
6015	0	0	0	0	0	0	0	0	0	0
6016	0	0	0	0	0	0	0	0	0	0
6017	0	0	0	0	0	0	0	0	0	0
6018	0	0	0	1	0	0	0	0	0	0

6019 rows × 48 columns

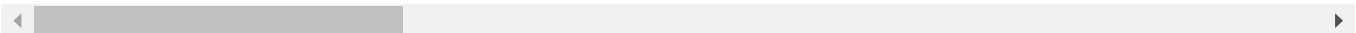


```
dfe=pd.concat([df,df1],axis=1)
dfe
```



	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	...	Location_Mu
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First	26.6 km/kg	998 CC	...	
1	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67 kmpl	1582 CC	...	
2	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	18.2 kmpl	1199 CC	...	
3	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77 kmpl	1248 CC	...	
4	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.2 kmpl	1968 CC	...	
...	
6014	6014	Maruti Swift VDI	Delhi	2014	27365	Diesel	Manual	First	28.4 kmpl	1248 CC	...	
6015	6015	Hyundai Xcent 1.1 CRDi S	Jaipur	2015	100000	Diesel	Manual	First	24.4 kmpl	1120 CC	...	
6016	6016	Mahindra Xylo D4 BSIV	Jaipur	2012	55000	Diesel	Manual	Second	14.0 kmpl	2498 CC	...	
6017	6017	Maruti Wagon R VXI	Kolkata	2013	46000	Petrol	Manual	First	18.9 kmpl	998 CC	...	
6018	6018	Chevrolet Beat Diesel	Hyderabad	2011	47000	Diesel	Manual	First	25.44 kmpl	936 CC	...	

6019 rows × 63 columns



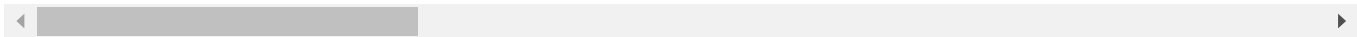
```
dfe.drop(['Unnamed: 0', 'Name', 'Location', 'Fuel_Type', 'Transmission', 'Owner_Type', 'New_Price', 'Brand'],axis=1,inplace=True)
```

dfe



	Year	Kilometers_Driven	Mileage	Engine	Power	Seats	Price	Brand_Audi	Brand_BMW	Brand_Bentley	...	Location_Mumbai	Loca
0	2010	72000	26.6 km/kg	998 CC	58.16 bhp	5.0	1.75	0	0	0	...	1	
1	2015	41000	19.67 kmpl	1582 CC	126.2 bhp	5.0	12.50	0	0	0	...	0	
2	2011	46000	18.2 kmpl	1199 CC	88.7 bhp	5.0	4.50	0	0	0	...	0	
3	2012	87000	20.77 kmpl	1248 CC	88.76 bhp	7.0	6.00	0	0	0	...	0	
4	2013	40670	15.2 kmpl	1968 CC	140.8 bhp	5.0	17.74	1	0	0	...	0	
...	
6014	2014	27365	28.4 kmpl	1248 CC	74 bhp	5.0	4.75	0	0	0	...	0	
6015	2015	100000	24.4 kmpl	1120 CC	71 bhp	5.0	4.00	0	0	0	...	0	
6016	2012	55000	14.0 kmpl	2498 CC	112 bhp	8.0	2.90	0	0	0	...	0	
6017	2013	46000	18.9 kmpl	998 CC	67.1 bhp	5.0	2.65	0	0	0	...	0	
6018	2011	47000	25.44 kmpl	936 CC	57.6 bhp	5.0	2.50	0	0	0	...	0	

6019 rows × 55 columns



dfe.dtypes



0

Year	int64
Kilometers_Driven	int64
Mileage	object
Engine	object
Power	object
Seats	float64
Price	float64
Brand_Audi	int64
Brand_BMW	int64
Brand_Bentley	int64
Brand_Chevrolet	int64
Brand_Datsun	int64
Brand_Fiat	int64
Brand_Force	int64
Brand_Ford	int64
Brand_Honda	int64
Brand_Hyundai	int64
Brand_ISUZU	int64
Brand_Isuzu	int64
Brand_Jaguar	int64
Brand_Jeep	int64
Brand_Lamborghini	int64
Brand_Land	int64
Brand_Mahindra	int64
Brand_Maruti	int64
Brand_Mercedes-Benz	int64
Brand_Mini	int64
Brand_Mitsubishi	int64
Brand_Nissan	int64
Brand_Porsche	int64
Brand_Renault	int64
Brand_Skoda	int64
Brand_Smart	int64
Brand_Tata	int64
Brand_Toyota	int64
Brand_Volkswagen	int64
Brand_Volvo	int64
Location_Bangalore	int64
Location_Chennai	int64
Location_Coimbatore	int64
Location_Delhi	int64
Location_Hyderabad	int64
Location_Jaipur	int64
Location_Kochi	int64
Location_Kolkata	int64
Location_Mumbai	int64
Location_Pune	int64
Fuel_Type_Diesel	int64
Fuel_Type_Electric	int64

Fuel_Type_LPGint64

Fuel_Type_Petrolint64

Transmission_Manualint64

Owner_Type_Fourth & Aboveint64

Owner_Type_Secondint64

Owner_Type_Thirdint64



```
dfe['Mileage']=dfe['Mileage'].str.replace('km/kg','')
dfe['Mileage']=dfe['Mileage'].str.replace('kmpl','')

dfe['Engine']=dfe['Engine'].str.replace('CC','')
dfe['Power']=dfe['Power'].str.replace('bhp','')
dfe['Mileage']=dfe['Mileage'].str.replace('null','0')
dfe['Engine']=dfe['Engine'].str.replace('null','0')
dfe['Power']=dfe['Power'].str.replace('null','0')
```

dfe

↔

	Year	Kilometers_Driven	Mileage	Engine	Power	Seats	Price	Brand_Audi	Brand_BMW	Brand_Bentley	...	Location_Mumbai	Loca
0	2010	72000	26.6	998	58.16	5.0	1.75	0	0	0	...	1	
1	2015	41000	19.67	1582	126.2	5.0	12.50	0	0	0	...	0	
2	2011	46000	18.2	1199	88.7	5.0	4.50	0	0	0	...	0	
3	2012	87000	20.77	1248	88.76	7.0	6.00	0	0	0	...	0	
4	2013	40670	15.2	1968	140.8	5.0	17.74	1	0	0	...	0	
...	
6014	2014	27365	28.4	1248	74	5.0	4.75	0	0	0	...	0	
6015	2015	100000	24.4	1120	71	5.0	4.00	0	0	0	...	0	
6016	2012	55000	14.0	2498	112	8.0	2.90	0	0	0	...	0	
6017	2013	46000	18.9	998	67.1	5.0	2.65	0	0	0	...	0	
6018	2011	47000	25.44	936	57.6	5.0	2.50	0	0	0	...	0	

6019 rows × 55 columns



dfe.dtypes



0

Year	int64
Kilometers_Driven	int64
Mileage	object
Engine	object
Power	object
Seats	float64
Price	float64
Brand_Audi	int64
Brand_BMW	int64
Brand_Bentley	int64
Brand_Chevrolet	int64
Brand_Datsun	int64
Brand_Fiat	int64
Brand_Force	int64
Brand_Ford	int64
Brand_Honda	int64
Brand_Hyundai	int64
Brand_ISUZU	int64
Brand_Isuzu	int64
Brand_Jaguar	int64
Brand_Jeep	int64
Brand_Lamborghini	int64
Brand_Land	int64
Brand_Mahindra	int64
Brand_Maruti	int64
Brand_Mercedes-Benz	int64
Brand_Mini	int64
Brand_Mitsubishi	int64
Brand_Nissan	int64
Brand_Porsche	int64
Brand_Renault	int64
Brand_Skoda	int64
Brand_Smart	int64
Brand_Tata	int64
Brand_Toyota	int64
Brand_Volkswagen	int64
Brand_Volvo	int64
Location_Bangalore	int64
Location_Chennai	int64
Location_Coimbatore	int64
Location_Delhi	int64
Location_Hyderabad	int64
Location_Jaipur	int64
Location_Kochi	int64
Location_Kolkata	int64
Location_Mumbai	int64
Location_Pune	int64
Fuel_Type_Diesel	int64
Fuel_Type_Electric	int64

Fuel_Type_LPG	int64
Fuel_Type_Petrol	int64
Transmission_Manual	int64
Owner_Type_Fourth & Above	int64
Owner_Type_Second	int64
Owner_Type_Third	int64

dtype: object

```
dfe['Mileage']=dfe['Mileage'].astype('float')  
dfe['Engine']=dfe['Engine'].astype('float')  
dfe['Power']=dfe['Power'].astype('float')
```

```
dfe.isna().sum()
```



	0
Year	0
Kilometers_Driven	0
Mileage	2
Engine	36
Power	36
Seats	42
Price	0
Brand_Audi	0
Brand_BMW	0
Brand_Bentley	0
Brand_Chevrolet	0
Brand_Datsun	0
Brand_Fiat	0
Brand_Force	0
Brand_Ford	0
Brand_Honda	0
Brand_Hyundai	0
Brand_ISUZU	0
Brand_Isuzu	0
Brand_Jaguar	0
Brand_Jeep	0
Brand_Lamborghini	0
Brand_Land	0
Brand_Mahindra	0
Brand_Maruti	0
Brand_Mercedes-Benz	0
Brand_Mini	0
Brand_Mitsubishi	0
Brand_Nissan	0
Brand_Porsche	0
Brand_Renault	0
Brand_Skoda	0
Brand_Smart	0
Brand_Tata	0
Brand_Toyota	0
Brand_Volkswagen	0
Brand_Volvo	0
Location_Bangalore	0
Location_Chennai	0
Location_Coimbatore	0
Location_Delhi	0
Location_Hyderabad	0
Location_Jaipur	0
Location_Kochi	0
Location_Kolkata	0
Location_Mumbai	0
Location_Pune	0
Fuel_Type_Diesel	0
Fuel_Type_Electric	0

Fuel_Type_LPG	0
Fuel_Type_Petrol	0
Transmission_Manual	0
Owner_Type_Fourth & Above	0
Owner_Type_Second	0
Owner_Type_Third	0

dtype: int64

```
#null ==>'0' ==>null ==>0.0  
dfe.loc[dfe.Engine==0, 'Engine']=np.NaN  
dfe.loc[dfe.Mileage==0, 'Mileage']=np.NaN  
dfe.loc[dfe.Power==0, 'Power']=np.NaN
```

```
dfe.isna().sum()
```



	0
Year	0
Kilometers_Driven	0
Mileage	70
Engine	36
Power	143
Seats	42
Price	0
Brand_Audi	0
Brand_BMW	0
Brand_Bentley	0
Brand_Chevrolet	0
Brand_Datsun	0
Brand_Fiat	0
Brand_Force	0
Brand_Ford	0
Brand_Honda	0
Brand_Hyundai	0
Brand_ISUZU	0
Brand_Isuzu	0
Brand_Jaguar	0
Brand_Jeep	0
Brand_Lamborghini	0
Brand_Land	0
Brand_Mahindra	0
Brand_Maruti	0
Brand_Mercedes-Benz	0
Brand_Mini	0
Brand_Mitsubishi	0
Brand_Nissan	0
Brand_Porsche	0
Brand_Renault	0
Brand_Skoda	0
Brand_Smart	0
Brand_Tata	0
Brand_Toyota	0
Brand_Volkswagen	0
Brand_Volvo	0
Location_Bangalore	0
Location_Chennai	0
Location_Coimbatore	0
Location_Delhi	0
Location_Hyderabad	0
Location_Jaipur	0
Location_Kochi	0
Location_Kolkata	0
Location_Mumbai	0
Location_Pune	0
Fuel_Type_Diesel	0
Fuel_Type_Electric	0

```

Fuel_Type_LPG      0
Fuel_Type_Petrol   0
Transmission_Manual 0
Owner_Type_Fourth & Above 0
Owner_Type_Second  0
Owner_Type_Third   0

```


```
dfe['Mileage'].unique()
```

```

array([26.6 , 19.67, 18.2 , 20.77, 15.2 , 21.1 , 23.08, 11.36, 20.54,
       22.3 , 21.56, 16.8 , 25.2 , 12.7 , nan, 13.5 , 25.8 , 28.4 ,
       20.45, 14.84, 22.69, 23.65, 13.53, 18.5 , 14.4 , 20.92, 17.5 ,
       12.8 , 19.01, 14.53, 11.18, 12.4 , 16.09, 14. , 24.3 , 18.15,
       11.74, 22.07, 19.7 , 25.4 , 25.32, 14.62, 14.28, 14.9 , 11.25,
       24.4 , 16.55, 17.11, 22.9 , 17.8 , 18.9 , 15.04, 25.17, 20.36,
       13.29, 13.68, 20. , 15.8 , 25. , 16.4 , 24.52, 22.1 , 8.5 ,
       15.1 , 16.95, 19.64, 16.5 , 18.53, 17.57, 18. , 23.2 , 16.73,
       17. , 13. , 17.68, 22.7 , 16.2 , 15.26, 23. , 19.83, 14.94,
       17.71, 14.74, 16. , 22.32, 12.99, 23.3 , 19.15, 10.8 , 15. ,
       22. , 21.9 , 12.05, 11.7 , 21.21, 20.73, 24.07, 19. , 20.58,
       19.27, 11.5 , 18.6 , 21.14, 11.05, 21.76, 7.81, 21.66, 17.2 ,
       20.63, 19.4 , 14.8 , 26. , 20.4 , 21.5 , 15.3 , 17.9 , 16.6 ,
       22.54, 25.44, 13.7 , 22.48, 12.9 , 19.98, 21.4 , 19.81, 15.4 ,
       25.47, 19.87, 17.45, 14.7 , 15.64, 15.73, 23.59, 16.1 , 27.4 ,
       20.46, 15.29, 20.51, 11.8 , 14.3 , 14.67, 17.19, 21.03, 22.5 ,
       16.82, 11.72, 17.4 , 17.05, 24. , 28.09, 20.5 , 13.1 , 19.91,
       18.7 , 16.38, 11.57, 17.3 , 22.95, 18.88, 23.4 , 22.74, 12.07,
       17.1 , 18.48, 16.47, 23.1 , 14.07, 16.02, 19.3 , 17.7 , 9.52,
       14.75, 26.3 , 11.3 , 21.12, 21.02, 14.45, 19.33, 13.8 , 24.7 ,
       11. , 11.07, 21.43, 14.21, 18.86, 16.07, 13.49, 20.38, 12. ,
       17.01, 13.2 , 20.37, 15.96, 14.16, 13.17, 27.62, 25.1 , 15.17,
       11.33, 17.92, 12.55, 12.6 , 17.72, 18.16, 15.68, 15.5 , 12.1 ,
       14.83, 17.6 , 14.6 , 14.66, 10.93, 20.68, 9.9 , 21.13, 20.14,
       19.2 , 27.3 , 16.36, 26.59, 12.5 , 13.6 , 15.06, 10.13, 17.21,
       15.97, 10.5 , 14.69, 23.9 , 19.1 , 21.27, 15.9 , 20.7 , 14.1 ,
       20.89, 18.12, 12.3 , 19.71, 9.43, 13.4 , 13.14, 18.1 , 22.77,
       14.49, 12.39, 10.91, 20.85, 15.63, 27.39, 18.3 , 16.78, 25.5 ,
       10. , 13.73, 24.2 , 14.02, 26.83, 16.77, 24.5 , 20.34, 21.7 ,
       9.7 , 14.33, 21.64, 19.16, 16.93, 9. , 26.2 , 16.3 , 12.62,
       20.64, 14.24, 18.06, 10.2 , 10.1 , 18.25, 13.93, 25.83, 8.6 ,
       13.24, 17.09, 23.84, 8.45, 19.6 , 19.5 , 20.3 , 16.05, 11.2 ,
       27.03, 18.78, 12.35, 14.59, 17.32, 14.95, 13.22, 23.03, 33.44,
       15.6 , 19.12, 10.98, 33.54, 16.46, 18.4 , 11.1 , 13.01, 18.8 ,
       16.52, 18.44, 19.49, 23.5 , 23.8 , 12.65, 20.65, 21.72, 12.19,
       26.1 , 18.33, 12.81, 17.06, 17.67, 19.34, 8.3 , 16.96, 11.79,
       20.86, 16.98, 11.68, 15.74, 15.7 , 18.49, 10.9 , 19.59, 11.4 ,
       13.06, 21. , 15.15, 16.9 , 18.23, 17.16, 17.43, 19.08, 18.56,
       11.9 , 24.6 , 21.79, 12.95, 25.6 , 13.45, 26.21, 13.58, 16.25,
       10.4 , 17.44, 22.71, 17.54, 15.87, 9.5 , 11.56, 14.39, 19.09,
       17.85, 31.79, 18.18, 21.19, 21.8 , 15.42, 14.47, 19.69, 12.83,
       8. , 22.8 , 12.63, 14.57, 27.28, 15.41, 32.26, 18.19, 13.33,
       16.7 , 17.84, 23.19, 11.49, 18.51, 13.44, 8.7 , 8.77, 17.97,
       23.57, 12.37, 9.1 , 12.51, 19.44, 21.38, 16.51, 24.8 , 14.42,
       26.8 , 24.04, 9.8 , 19.68, 21.2 , 19.72, 14.2 , 12.98, 23.01,
       16.12, 9.3 , 15.85, 17.88, 10.6 , 11.78, 7.94, 25.01, 8.1 ,
       13.9 , 11.62, 20.62, 15.11, 10.37, 18.59, 9.74, 14.81, 8.2 ,
       12.97, 7.5 , 30.46, 6.4 , 12.85, 18.69, 17.24])

```

```
dfe['Mileage'].fillna(dfe['Mileage'].mean(),inplace=True)
```

 <ipython-input-31-a8d9df8547f5>:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained ass The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col

```
dfe['Mileage'].fillna(dfe['Mileage'].mean(),inplace=True)
```

```
dfe['Engine'].unique()
```

```


array([ 998., 1582., 1199., 1248., 1968., 814., 1461., 2755., 1598.,
       1462., 1497., 2179., 2477., 1498., 2143., 1995., 1984., 1197.,
       2494., 1798., 2696., 2698., 1061., 1198., 2987., 796., 624.,
       1999., 1991., 2694., 1120., 2498., 799., 2393., 1399., 1796.,
       2148., 1396., 1950., 4806., 1998., 1086., 1193., 2982., 1493.,
       2967., 2993., 1196., 1799., 2497., 2354., 1373., 2996., 1591.,
       2894., 5461., 1595., 936., 1997., nan, 1896., 1390., 1364.,
       2199., 993., 999., 1405., 2956., 1794., 995., 2496., 1599.,
       2400., 1495., 2523., 793., 4134., 1596., 1395., 2953., 1586.,

```



```
2362., 1496., 1368., 1298., 1956., 1299., 3498., 2835., 1150.,
3198., 1343., 1499., 1186., 1590., 2609., 2499., 2446., 1978.,
2360., 3436., 2198., 4367., 2706., 1422., 2979., 1969., 1489.,
2489., 1242., 1388., 1172., 2495., 1194., 3200., 1781., 1341.,
2773., 3597., 1985., 2147., 1047., 2999., 2995., 2997., 1948.,
2359., 4395., 2349., 2720., 1468., 3197., 2487., 1597., 2771.,
72., 4951., 970., 2925., 2200., 5000., 2149., 5998., 2092.,
5204., 2112., 1797.])
```


```
dfe['Engine'].fillna(dfe['Engine'].mean(),inplace=True)
```

 <ipython-input-33-88559e7dbbe5>:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment. The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting


For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col]

```
dfe['Engine'].fillna(dfe['Engine'].mean(),inplace=True)
```

```
dfe['Power'].unique()
```

 array([58.16 , 126.2 , 88.7 , 88.76 , 140.8 , 55.2 , 63.1 ,
171.5 , 103.6 , 74. , 103.25 , 116.3 , 187.7 , 115. ,
175.56 , 98.6 , 83.8 , 167.62 , 190. , 88.5 , 177.01 ,
80. , 67.1 , 102. , 108.45 , 138.1 , 184. , 179.5 ,
103.5 , 64. , 82. , 254.8 , 73.9 , 46.3 , 37.5 ,
77. , 82.9 , 149.92 , 138.03 , 112.2 , 163.7 , 71. ,
105. , 174.33 , 75. , 103.2 , 53.3 , 78.9 , 147.6 ,
147.8 , 68. , 186. , 170. , 69. , 140. , 78. ,
194. , 500. , 108.5 , 86.8 , 187.74 , nan, 132. ,
86.7 , 73.94 , 117.3 , 218. , 168.5 , 89.84 , 110. ,
90. , 82.85 , 67. , 241.4 , 35. , 270.9 , 126.32 ,
73. , 130. , 100.6 , 150. , 75.94 , 215. , 107.3 ,
37.48 , 120. , 178. , 152. , 91.1 , 85.8 , 362.07 ,
121.3 , 143. , 81.8 , 171. , 76.8 , 103.52 , 444. ,
362.9 , 67.06 , 120.7 , 258. , 81.86 , 112. , 88.73 ,
57.6 , 157.75 , 102.5 , 201.1 , 83.1 , 68.05 , 106. ,
100. , 81.83 , 85. , 64.1 , 177.5 , 246.7 , 177.46 ,
65. , 67.04 , 189.08 , 99. , 53.5 , 194.3 , 70. ,
183. , 254.79 , 66.1 , 76. , 60. , 123.24 , 47.3 ,
118. , 88.8 , 177. , 136. , 201.15 , 93.7 , 177.6 ,
313. , 245. , 125. , 141. , 227. , 62. , 141.1 ,
83.14 , 192. , 67.05 , 47. , 235. , 37. , 87.2 ,
203. , 204. , 246.74 , 122. , 282. , 181. , 224. ,
94. , 367. , 98.79 , 62.1 , 174.3 , 114. , 335.2 ,
169. , 191.34 , 108.49 , 138.02 , 156. , 187.4 , 66. ,
103.3 , 164.7 , 79.4 , 198.5 , 154. , 73.8 , 181.43 ,
207.8 , 108.4 , 88. , 63. , 82.5 , 364.9 , 107.2 ,
113.98 , 126.3 , 185. , 237.4 , 99.6 , 66.7 , 160. ,
306. , 98.59 , 92.7 , 147.51 , 197.2 , 167.6 , 165. ,
110.4 , 73.97 , 147.9 , 116.6 , 148. , 34.2 , 155. ,
197. , 108.62 , 118.3 , 38.4 , 241.38 , 153.86 , 163.5 ,
226.6 , 84.8 , 53.64 , 158.2 , 69.01 , 181.03 , 58.2 ,
104.68 , 126.24 , 73.75 , 158. , 130.2 , 57.5 , 97.7 ,
121.4 , 98.96 , 174.5 , 308. , 121.36 , 138. , 265. ,
84. , 321. , 91.72 , 65.3 , 88.2 , 93. , 35.5 ,
86.79 , 157.7 , 40.3 , 91.7 , 180. , 114.4 , 158.8 ,
157.8 , 123.7 , 56.3 , 189. , 104. , 210. , 270.88 ,
142. , 255. , 236. , 167.7 , 148.31 , 80.46 , 138.08 ,
250. , 74.9 , 91.2 , 102.57 , 97.6 , 102.53 , 240. ,
254. , 112.4 , 73.74 , 108.495, 116.9 , 101. , 320. ,
70.02 , 261.49 , 105.5 , 550. , 168.7 , 55.23 , 94.68 ,
152.88 , 163.2 , 203.2 , 241. , 95. , 200. , 271.23 ,
63.12 , 85.7 , 308.43 , 118.6 , 199.3 , 83.83 , 55. ,
83. , 300. , 201. , 262.6 , 163. , 58.33 , 86.76 ,
76.9 , 174.57 , 301.73 , 68.1 , 162. , 394.3 , 80.9 ,
147.5 , 272. , 340. , 120.33 , 82.4 , 231.1 , 335.3 ,
333. , 198.25 , 224.34 , 402. , 261. , 61. , 144. ,
71.01 , 271.72 , 134. , 135.1 , 92. , 64.08 , 261.5 ,
123.37 , 175.67 , 53. , 110.5 , 178.4 , 193.1 , 41. ,
395. , 48.21 , 450. , 421. , 89.75 , 387.3 , 130.3 ,
281.61 , 52.8 , 139.01 , 208. , 503. , 168. , 98.82 ,
139.07 , 83.11 , 74.93 , 382. , 74.96 , 552. , 127. ,
560. , 116.4 , 161.6 , 488.1 , 103. , 181.04])

```
dfe['Power'].fillna(dfe['Power'].mean(),inplace=True)
```

 <ipython-input-35-ab9a6ebf03a6>:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment. The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col]

```
dfe['Power'].fillna(dfe['Power'].mean(),inplace=True)
```

```
dfe['Seats'].unique()
```

```
↵ array([ 5.,  7.,  8.,  4.,  6.,  2., nan, 10.,  9.,  0.])
```

```
dfe['Seats'].fillna(dfe['Seats'].mode()[0],inplace=True)
```

↵ <ipython-input-37-391d9529cdfa>:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment. The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting is a copy. For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or 'df[col] = df[col].method(value, inplace=True)'

```
dfe['Seats'].fillna(dfe['Seats'].mode()[0],inplace=True)
```

```
dfe.isna().sum()
```



	0
Year	0
Kilometers_Driven	0
Mileage	0
Engine	0
Power	0
Seats	0
Price	0
Brand_Audi	0
Brand_BMW	0
Brand_Bentley	0
Brand_Chevrolet	0
Brand_Datsun	0
Brand_Fiat	0
Brand_Force	0
Brand_Ford	0
Brand_Honda	0
Brand_Hyundai	0
Brand_ISUZU	0
Brand_Isuzu	0
Brand_Jaguar	0
Brand_Jeep	0
Brand_Lamborghini	0
Brand_Land	0
Brand_Mahindra	0
Brand_Maruti	0
Brand_Mercedes-Benz	0
Brand_Mini	0
Brand_Mitsubishi	0
Brand_Nissan	0
Brand_Porsche	0
Brand_Renault	0
Brand_Skoda	0
Brand_Smart	0
Brand_Tata	0
Brand_Toyota	0
Brand_Volkswagen	0
Brand_Volvo	0
Location_Bangalore	0
Location_Chennai	0
Location_Coimbatore	0
Location_Delhi	0
Location_Hyderabad	0
Location_Jaipur	0
Location_Kochi	0
Location_Kolkata	0
Location_Mumbai	0
Location_Pune	0
Fuel_Type_Diesel	0
Fuel_Type_Electric	0

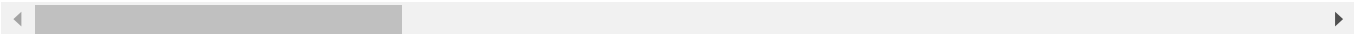
```
Fuel_Type_LPG      0
Fuel_Type_Petrol   0
Transmission_Manual 0
Owner_Type_Fourth & Above 0
Owner_Type_Second  0
Owner_Type_Third   0
```



```
x=dfe.drop(['Price'],axis=1)
x
```

	Year	Kilometers_Driven	Mileage	Engine	Power	Seats	Brand_Audi	Brand_BMW	Brand_Bentley	Brand_Chevrolet	...	Location_M
0	2010	72000	26.60	998.0	58.16	5.0	0	0	0	0	...	
1	2015	41000	19.67	1582.0	126.20	5.0	0	0	0	0	...	
2	2011	46000	18.20	1199.0	88.70	5.0	0	0	0	0	...	
3	2012	87000	20.77	1248.0	88.76	7.0	0	0	0	0	...	
4	2013	40670	15.20	1968.0	140.80	5.0	1	0	0	0	...	
...	
6014	2014	27365	28.40	1248.0	74.00	5.0	0	0	0	0	...	
6015	2015	100000	24.40	1120.0	71.00	5.0	0	0	0	0	...	
6016	2012	55000	14.00	2498.0	112.00	8.0	0	0	0	0	...	
6017	2013	46000	18.90	998.0	67.10	5.0	0	0	0	0	...	
6018	2011	47000	25.44	936.0	57.60	5.0	0	0	0	1	...	

6019 rows × 54 columns



```
y=dfe['Price']
y
```

	Price
0	1.75
1	12.50
2	4.50
3	6.00
4	17.74
...	...
6014	4.75
6015	4.00
6016	2.90
6017	2.65
6018	2.50

6019 rows × 1 columns



- Start coding or [generate](#) with AI.
- Start coding or [generate](#) with AI.
- Start coding or [generate](#) with AI.
- Start coding or [generate](#) with AI.

```
test=pd.read_csv('/content/test-data.csv')
test
```

	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_Price
0	0	Maruti Alto K10 LXI CNG	Delhi	2014	40929	CNG	Manual	First	32.26 km/kg	998 CC	58.2 bhp	4.0	1052
1	1	Maruti Alto 800 2016-2019 LXI	Coimbatore	2013	54493	Petrol	Manual	Second	24.7 kmpl	796 CC	47.3 bhp	5.0	1052
2	2	Toyota Innova Crysta Touring Sport 2.4 MT	Mumbai	2017	34000	Diesel	Manual	First	13.68 kmpl	2393 CC	147.8 bhp	7.0	1052
3	3	Toyota Etios Liva GD	Hyderabad	2012	139000	Diesel	Manual	First	23.59 kmpl	1364 CC	null bhp	5.0	1052
4	4	Hyundai i20	Mumbai	2014	30000	Petrol	Manual	First	18.5	1197	82.85	5.0	1052

```
test.head()
```

	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_Price
0	0	Maruti Alto K10 LXI CNG	Delhi	2014	40929	CNG	Manual	First	32.26 km/kg	998 CC	58.2 bhp	4.0	N
1	1	Maruti Alto 800	Coimbatore	2013	54493	Petrol	Manual	Second	24.7 kmpl	796 CC	47.3 bhp	5.0	N

```
test.isna().sum()
```

	0
Unnamed: 0	0
Name	0
Location	0
Year	0
Kilometers_Driven	0
Fuel_Type	0
Transmission	0
Owner_Type	0
Mileage	0
Engine	10
Power	10
Seats	11
New_Price	1052

```
test.dtypes
```



0	
Unnamed: 0	int64
Name	object
Location	object
Year	int64
Kilometers_Driven	int64
Fuel_Type	object
Transmission	object
Owner_Type	object
Mileage	object
Engine	object
Power	object
Seats	float64
New_Price	object



```
df1=test['Location'].value_counts()  
df1
```

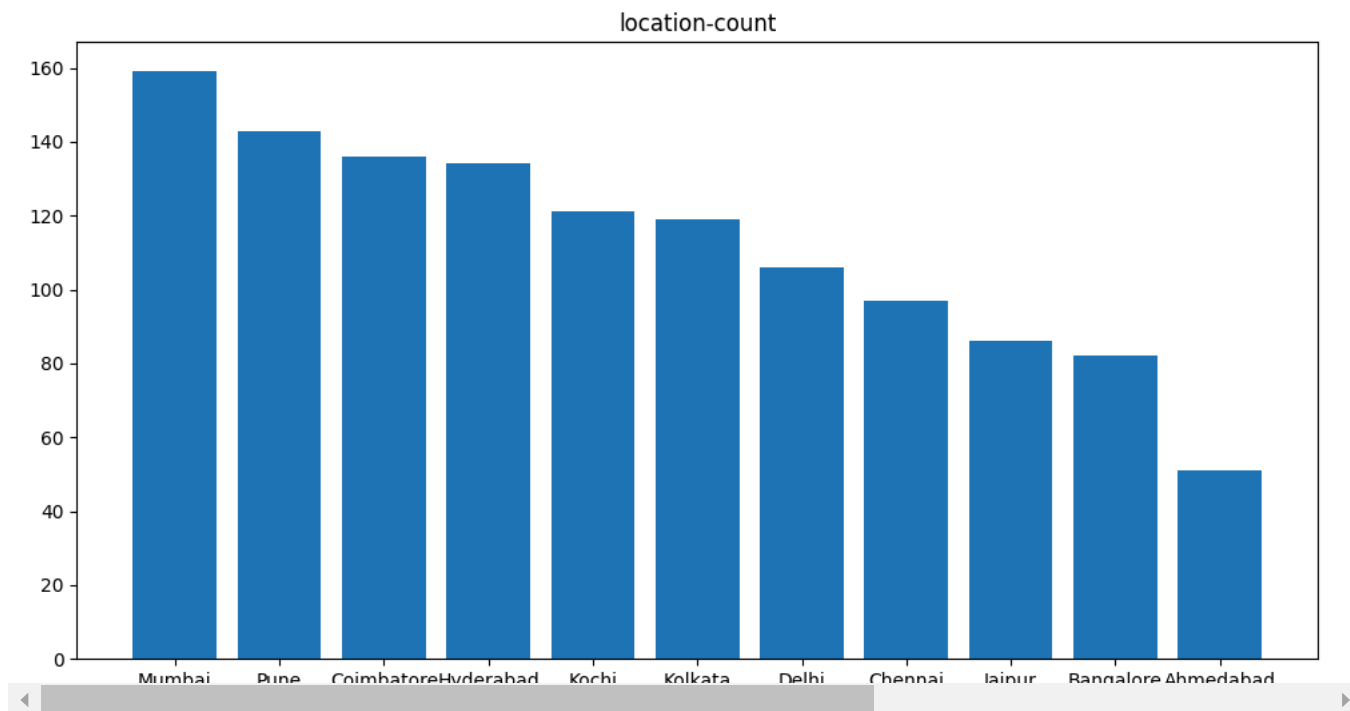


	count
Location	
Mumbai	159
Pune	143
Coimbatore	136
Hyderabad	134
Kochi	121
Kolkata	119
Delhi	106
Chennai	97
Jaipur	86
Bangalore	82
Ahmedabad	51



```
plt.figure(figsize=(12,6))  
plt.bar(df1.index,df1)  
plt.title('location-count')
```

↻ Text(0.5, 1.0, 'location-count')



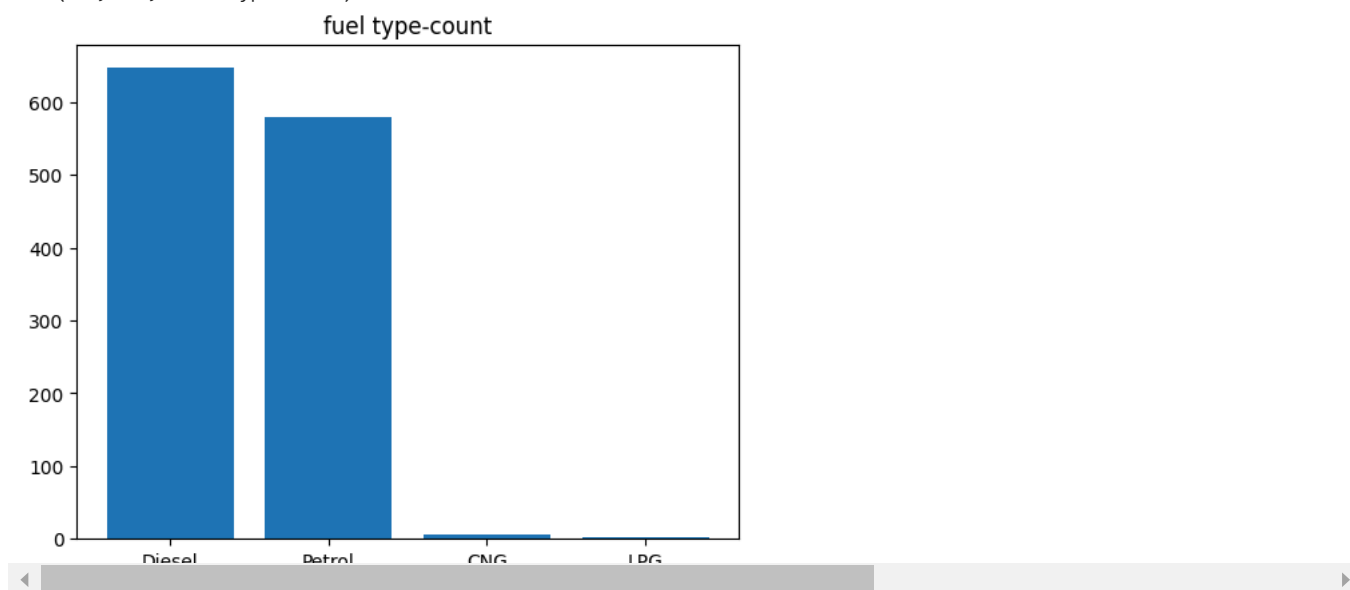
```
df2=test['Fuel_Type'].value_counts()  
df2
```

↻

	count
Fuel_Type	
Diesel	647
Petrol	579
CNG	6
LPG	2

```
#plt.pie(df2,autopct='%1.01f%%',labels=df2.index)  
plt.bar(df2.index,df2)  
plt.title('fuel type-count')
```

↻ Text(0.5, 1.0, 'fuel type-count')



```
df3=test['Transmission'].value_counts()  
df3
```

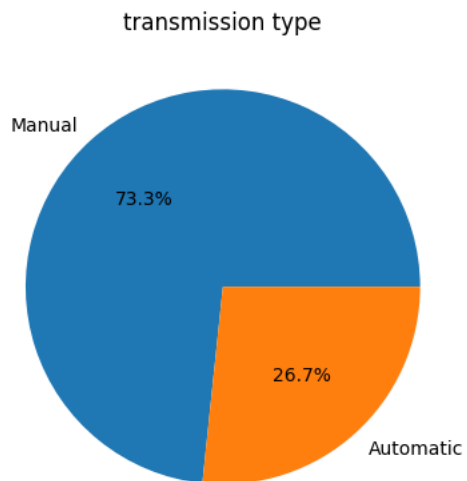


	count
Transmission	
Manual	905
Automatic	329

```
plt.pie(df3, autopct='%1.01f%%', labels=df3.index)  
plt.title('transmission type')
```



Text(0.5, 1.0, 'transmission type')



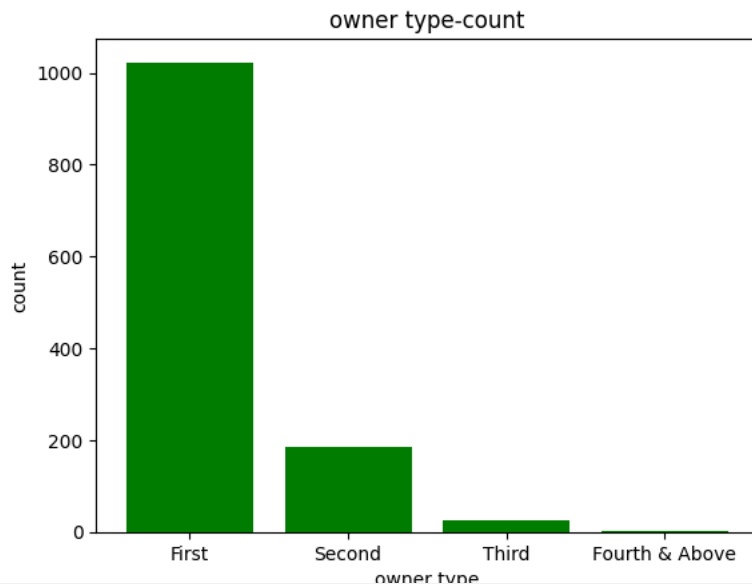
```
df4=test['Owner_Type'].value_counts()  
df4
```



	count
Owner_Type	
First	1023
Second	184
Third	24
Fourth & Above	3

```
plt.bar(df4.index, df4, color='g')  
plt.title('owner type-count')  
plt.xlabel('owner type')  
plt.ylabel('count')
```



Text(0, 0.5, 'count')



```
test['Name'].unique()
```

```
array(['Maruti Alto K10 LXI CNG', 'Maruti Alto 800 2016-2019 LXI',
      'Toyota Innova Crysta Touring Sport 2.4 MT',
      'Toyota Etios Liva GD', 'Hyundai i20 Magna',
      'Mahindra XUV500 W8 2WD', 'Toyota Fortuner 4x2 AT TRD Sportivo',
      'Hyundai EON Era Plus', 'Honda City 1.5 S MT',
      'Mahindra XUV500 W6 2WD', 'Audi Q5 2008-2012 2.0 TDI',
      'Hyundai Grand i10 Magna', 'Toyota Corolla H5',
      'Maruti Swift Vdi BSIII', 'Nissan Terrano XL', 'BMW X1 sDrive20d',
      'BMW 3 Series GT 320d Luxury Line', 'Ford Ikon 1.4 TDCi DuraTorq',
      'Maruti Swift AMT ZXI', 'Maruti Swift Dzire VXI',
      'Maruti Ritz LDi', 'Nissan Micra XV',
      'Skoda Laura 1.8 TSI Ambition',
      'Honda Civic 2010-2013 1.8 S MT Inspire', 'Ford Ikon 1.3 Flair',
      'Maruti Alto LX', 'Honda Jazz 1.5 V i DTEC',
      'Skoda Rapid 1.5 TDI AT Ambition', 'Hyundai i20 1.2 Sportz',
      'Toyota Etios Cross 1.4L VD', 'Volkswagen Polo GT TDI',
      'Hyundai i10 Era 1.1 iTech SE', 'Mitsubishi Pajero 2.8 SFX',
      'Maruti Swift Dzire Tour LDI', 'Maruti Swift Dzire VXI',
      'Skoda Rapid 1.6 MPI AT Style',
      'Hyundai Creta 1.6 SX Plus Dual Tone Petrol',
      'Toyota Innova 2.5 G (Diesel) 7 Seater BS IV',
      'BMW 5 Series 520d Luxury Line', 'Maruti Baleno Alpha 1.3',
      'Audi A4 2.0 TDI 177 Bhp Technology Edition',
      'Audi A4 2.0 TDI Multitronic', 'Skoda Rapid 1.6 MPI Ambition',
      'Maruti Ertiga SHVS VDI', 'Maruti Ciaz 1.4 Zeta',
      'Maruti Ciaz ZXI', 'Mercedes-Benz A Class A200 CDI Sport',
      'Toyota Innova 2.5 ZX Diesel 7 Seater',
      'BMW 3 Series 320d Prestige', 'Ford Ecosport 1.5 DV5 MT Titanium',
      'Ford Endeavour 3.0L 4X4 AT', 'Audi Q5 2.0 TDI',
      'Hyundai Verna 1.6 SX VTVT', 'Mahindra XUV500 W8 4WD',
      'Maruti Vitara Brezza ZDi', 'Toyota Etios Liva 1.4 VXD',
      'Tata Indigo CS LX (TDI) BS III', 'Honda City ZX GXi',
      'Toyota Corolla Altis G', 'Hyundai Grand i10 Asta',
      'Hyundai i20 Sportz 1.2', 'Volkswagen Polo 1.5 TDI Comfortline',
      'Maruti Celerio X VXI Option', 'Chevrolet Sail Hatchback 1.2',
      'Mahindra Scorpio VLX Special Edition BS-IV',
      'Volkswagen Ameo 1.5 TDI Comfortline', 'Maruti Wagon R LXI',
      'Honda Brio S Option MT', 'Maruti Ertiga VXI',
      'Hyundai Santro Xing GL', 'Ford Figo Aspire 1.2 Ti-VCT Titanium',
      'Tata Tiago AMT 1.2 Revotron XTA', 'Hyundai Santro Xing XL',
      'Chevrolet Beat LT', 'Nissan Micra Diesel XV',
      'Toyota Innova 2.5 GX (Diesel) 8 Seater',
      'Hyundai Xcent 1.2 VTVT E', 'Maruti Ciaz VXI',
      'BMW 5 Series 2013-2017 520d Luxury Line',
      'Toyota Camry Hybrid 2.5', 'Toyota Corolla Altis D-4D G',
      'Hyundai Verna 1.6 SX CRDi (O)',
      'Volkswagen Vento 1.5 TDI Highline AT', 'Nissan Teana XL',
      'Hyundai i20 1.2 Sportz Option',
      'Tata Manza Club Class Safire90 LX', 'Ford Figo Diesel ZXI',
      'Hyundai i20 Asta 1.2', 'Ford Fiesta Classic 1.6 SXi Duratec',
      'Mercedes-Benz B Class B180 Sports', 'Maruti Swift Dzire VDI',
      'Hyundai Santro Xing GLS', 'Maruti Baleno Alpha',
      'Honda Accord 2.4 Elegance M/T', 'Nissan Micra XL CVT',
      'Maruti Zen Estilo LXI BS IV', 'Hyundai Elantra CRDi SX',
      'Tata Indica V2 DiCOR DLG BS-III', 'Tata Nano Lx',
      'Honda Amaze S i-Dtech', 'Volkswagen Polo Diesel Highline 1.2L',
      'Tata Indica Vista Quadrajet LX', 'Audi Q3 30 TDI S Edition',
      'Honda City i DTEC SV', 'Honda Amaze SX i-DTEC',
```

```
test['Brand']=test['Name'].apply(lambda x:x.split()[0])
test
```



	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	Brand
0	0	Maruti Alto K10 LXI CNG	Delhi	2014	40929	CNG	Manual	First	32.26 km/kg	998 CC	58.2 bhp	4.0	Maruti
1	1	Maruti Alto 800 2016-2019 LXI	Coimbatore	2013	54493	Petrol	Manual	Second	24.7 kmpl	796 CC	47.3 bhp	5.0	Maruti
2	2	Toyota Innova Crysta Touring Sport 2.4 MT	Mumbai	2017	34000	Diesel	Manual	First	13.68 kmpl	2393 CC	147.8 bhp	7.0	Toyota
3	3	Toyota Etios Liva GD	Hyderabad	2012	139000	Diesel	Manual	First	23.59 kmpl	1364 CC	null bhp	5.0	Toyota
4	4	Hyundai i20	Mumbai	2014	20000	Petrol	Manual	First	18.5	1197	82.85	5.0	Hyundai

```
test.dtypes
```



		0
Unnamed: 0		int64
Name		object
Location		object
Year		int64
Kilometers_Driven		int64
Fuel_Type		object
Transmission		object
Owner_Type		object
Mileage		object
Engine		object
Power		object
Seats		float64
New_Price		object
Brand		object

```
test['Brand'].value_counts()
```



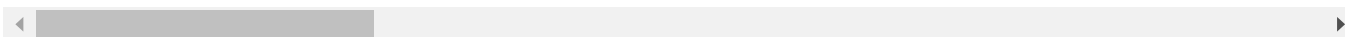
	count
Brand	
Maruti	233
Hyundai	233
Honda	135
Toyota	96
Mercedes-Benz	62
Mahindra	59
Volkswagen	59
Ford	51
Audi	49
BMW	45
Tata	42
Chevrolet	30
Skoda	29
Nissan	26
Renault	25
Fiat	10
Mitsubishi	9
Jaguar	8
Volvo	7
Land	7
Mini	5
Datsun	4
Jeep	4
Bentley	1
Hindustan	1
Isuzu	1
Porsche	1
ISUZU	1
OpelCorsa	1

```
df1=pd.get_dummies(test[['Brand','Location','Fuel_Type','Transmission','Owner_Type']],dtype=int,drop_first=True)
df1
```



	Brand_BMW	Brand_Bentley	Brand_Chevrolet	Brand_Datsun	Brand_Fiat	Brand_Ford	Brand_Hindustan	Brand_Honda	Brand_Hyundai
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	1
...
1229	0	0	0	0	0	0	0	0	0
1230	0	0	0	0	0	0	0	0	0
1231	0	0	0	0	0	0	0	0	0
1232	0	0	0	0	0	0	0	0	0
1233	0	0	0	0	0	0	0	0	0

1234 rows × 45 columns

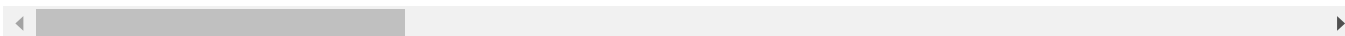


```
dft=pd.concat((test,df1),axis=1)
dft
```



	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	...	Location_I
0	0	Maruti Alto K10 LXI CNG	Delhi	2014	40929	CNG	Manual	First	32.26 km/kg	998 CC	...	
1	1	Maruti Alto 800 2016-2019 LXI	Coimbatore	2013	54493	Petrol	Manual	Second	24.7 kmpl	796 CC	...	
2	2	Toyota Innova Crysta Touring Sport 2.4 MT	Mumbai	2017	34000	Diesel	Manual	First	13.68 kmpl	2393 CC	...	
3	3	Toyota Etios Liva GD	Hyderabad	2012	139000	Diesel	Manual	First	23.59 kmpl	1364 CC	...	
4	4	Hyundai i20 Magna	Mumbai	2014	29000	Petrol	Manual	First	18.5 kmpl	1197 CC	...	
...	
1229	1229	Volkswagen Vento Diesel Trendline	Hyderabad	2011	89411	Diesel	Manual	First	20.54 kmpl	1598 CC	...	
1230	1230	Volkswagen Polo GT TSI	Mumbai	2015	59000	Petrol	Automatic	First	17.21 kmpl	1197 CC	...	
1231	1231	Nissan Micra Diesel XV	Kolkata	2012	28000	Diesel	Manual	First	23.08 kmpl	1461 CC	...	
1232	1232	Volkswagen Polo GT TSI	Pune	2013	52262	Petrol	Automatic	Third	17.2 kmpl	1197 CC	...	
1233	1233	Mercedes-Benz E-Class 2009-2013 E 220 CDI Avan...	Kochi	2014	72443	Diesel	Automatic	First	10.0 kmpl	2148 CC	...	

1234 rows × 59 columns



```
dft.isna().sum()
```



	0
Unnamed: 0	0
Name	0
Location	0
Year	0
Kilometers_Driven	0
Fuel_Type	0
Transmission	0
Owner_Type	0
Mileage	0
Engine	10
Power	10
Seats	11
New_Price	1052
Brand	0
Brand_BMW	0
Brand_Bentley	0
Brand_Chevrolet	0
Brand_Datsun	0
Brand_Fiat	0
Brand_Ford	0
Brand_Hindustan	0
Brand_Honda	0
Brand_Hyundai	0
Brand_ISUZU	0
Brand_Isuzu	0
Brand_Jaguar	0
Brand_Jeep	0
Brand_Land	0
Brand_Mahindra	0
Brand_Maruti	0
Brand_Mercedes-Benz	0
Brand_Mini	0
Brand_Mitsubishi	0
Brand_Nissan	0
Brand_OpelCorsa	0
Brand_Porsche	0
Brand_Renault	0
Brand_Skoda	0
Brand_Tata	0
Brand_Toyota	0
Brand_Volkswagen	0
Brand_Volvo	0
Location_Bangalore	0
Location_Chennai	0
Location_Coimbatore	0
Location_Delhi	0
Location_Hyderabad	0
Location_Jaipur	0
Location_Kochi	0