EDA on Car Dekhlo Dataset

Exploratory Data Analysis

In [ ]:

In [1]:
```
# Business probles:
```

In [2]:
```
!pip install kagglehub
```

```
Requirement already satisfied: kagglehub in c:\users\administrator\anaconda3\pyth
on 1\lib\site-packages (0.3.13)
Requirement already satisfied: packaging in c:\users\administrator\anaconda3\pyth
on 1\lib\site-packages (from kagglehub) (24.2)
Requirement already satisfied: pyyaml in c:\users\administrator\anaconda3\python
1\lib\site-packages (from kagglehub) (6.0.2)
Requirement already satisfied: requests in c:\users\administrator\anaconda3\pytho
n 1\lib\site-packages (from kagglehub) (2.32.3)
Requirement already satisfied: tqdm in c:\users\administrator\anaconda3\python 1
\lib\site-packages (from kagglehub) (4.67.1)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\administrator
\anaconda3\python 1\lib\site-packages (from requests->kagglehub) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in c:\users\administrator\anaconda3\p
ython 1\lib\site-packages (from requests->kagglehub) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\administrator\anaco
nda3\python 1\lib\site-packages (from requests->kagglehub) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\administrator\anaco
nda3\python 1\lib\site-packages (from requests->kagglehub) (2025.4.26)
Requirement already satisfied: colorama in c:\users\administrator\anaconda3\pytho
n 1\lib\site-packages (from tqdm->kagglehub) (0.4.6)
```

In [3]:
```
import kagglehub

# Download latest version
path = kagglehub.dataset_download("manishkr1754/cardekho-used-car-data")
```

```
print("Path to dataset files:", path)
```

```
Path to dataset files: C:\Users\Administrator\.cache\kagglehub\datasets\manishkr1754\cardekho-used-car-data\versions\2
```

In [4]:
```
path
```

Out[4]:
```
'C:\\Users\\Administrator\\.cache\\kagglehub\\datasets\\manishkr1754\\cardekho-used-car-data\\versions\\2'
```

In [5]:
```
import os
```

In [6]:
```
os.listdir(path)[0]
```

Out[6]:
```
'cardekho_dataset.csv'
```

In [7]:
```
file_name=os.listdir(path)[0]
```

In [8]:
```
print(file_name)
```

```
cardekho_dataset.csv
```

In [9]:
```
file_path =path + '\\' + file_name
```

In [10]:
```
print(file_path)
```

```
C:\Users\Administrator\.cache\kagglehub\datasets\manishkr1754\cardekho-used-car-data\versions\2\cardekho_dataset.csv
```

Step 2: import important Modules

In [11]:
```
import pandas as pd
```

Step 3 : Load DataFrame

In [12]:
```
df =pd.read_csv(file_path)
```

In [13]:
```
df
```

Out[13]:

	Unnamed: 0	car_name	brand	model	vehicle_age	km_driven	seller_type	f
0	0	Maruti Alto	Maruti	Alto	9	120000	Individual	
1	1	Hyundai Grand	Hyundai	Grand	5	20000	Individual	
2	2	Hyundai i20	Hyundai	i20	11	60000	Individual	
3	3	Maruti Alto	Maruti	Alto	9	37000	Individual	
4	4	Ford Ecosport	Ford	Ecosport	6	30000	Dealer	
...	...	...	...	...	...	...	...	
15406	19537	Hyundai i10	Hyundai	i10	9	10723	Dealer	
15407	19540	Maruti Ertiga	Maruti	Ertiga	2	18000	Dealer	
15408	19541	Skoda Rapid	Skoda	Rapid	6	67000	Dealer	
15409	19542	Mahindra XUV500	Mahindra	XUV500	5	3800000	Dealer	
15410	19543	Honda City	Honda	City	2	13000	Dealer	

15411 rows × 14 columns

| Column name | Description | | ---------- | ---------- | | **car_name** | The name of the car (includes brand + model + variant). Example: *Maruti Swift VDI*. | | **brand** | Manufacturer/brand of the car. Example: *Maruti, Hyundai, Honda, Toyota*. | | **model** | Specific model name of the car. Example: *Swift, i20, City, Innova*. | | **vehicle_age** | Age of the car in years (current year − year of manufacture). Example: *5* means the car is 5 years old. | | **km_driven** | Total kilometers driven by the car (odometer reading). Usually numeric. | | **seller_type** | Type of seller — *Individual*, *Dealer*, or *Trustmark Dealer*. | | **fuel_type** | Type of fuel used — *Petrol, Diesel, CNG, LPG, Electric, Hybrid*. | | **transmission_type** | Type of transmission — *Manual* or *Automatic*. | | **mileage** | Mileage of the car (fuel efficiency) in km/l (for petrol/diesel) or km/kg (for CNG). | | **engine** | Engine displacement (size) in CC (cubic centimeters). Example: *1197 CC*. | | **max_power** | Maximum engine power output, usually in bhp (brake horsepower) or PS/kW. | | **seats** | Number of seats in the car. Example: *5, 7*. | | **selling_price** | Price at which the car is being sold (target variable for prediction). |

EDA

In [14]:
```
df.shape
```

Out[14]:
```
(15411, 14)
```

In [15]:
```
df.size
```

Out[15]:
```
215754
```

In [16]:
```
df.dtypes
```

Out[16]:
```
Unnamed: 0            int64
car_name             object
brand                object
model                object
vehicle_age           int64
km_driven             int64
seller_type          object
fuel_type            object
transmission_type    object
mileage             float64
engine                int64
max_power           float64
seats                 int64
selling_price         int64
dtype: object
```

In [17]:
```
df.columns
```

Out[17]:
```
Index(['Unnamed: 0', 'car_name', 'brand', 'model', 'vehicle_age', 'km_driven',
       'seller_type', 'fuel_type', 'transmission_type', 'mileage', 'engine',
       'max_power', 'seats', 'selling_price'],
      dtype='object')
```

In [18]:
```
df.index
```

Out[18]:
```
RangeIndex(start=0, stop=15411, step=1)
```

In [19]:
```
df.drop('Unnamed: 0',axis =1, inplace = True)
print('done')
```

```
done
```

In [20]:
```
df
```

Out[20]:

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	tr
0	Maruti Alto	Maruti	Alto	9	120000	Individual	Petrol	
1	Hyundai Grand	Hyundai	Grand	5	20000	Individual	Petrol	
2	Hyundai i20	Hyundai	i20	11	60000	Individual	Petrol	
3	Maruti Alto	Maruti	Alto	9	37000	Individual	Petrol	
4	Ford Ecosport	Ford	Ecosport	6	30000	Dealer	Diesel	
...	...	...	...	...	...	...	...	
15406	Hyundai i10	Hyundai	i10	9	10723	Dealer	Petrol	
15407	Maruti Ertiga	Maruti	Ertiga	2	18000	Dealer	Petrol	
15408	Skoda Rapid	Skoda	Rapid	6	67000	Dealer	Diesel	
15409	Mahindra XUV500	Mahindra	XUV500	5	3800000	Dealer	Diesel	
15410	Honda City	Honda	City	2	13000	Dealer	Petrol	

15411 rows × 13 columns

In [21]:
```
df.head(10)
```

Out[21]:

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	transmis
0	Maruti Alto	Maruti	Alto	9	120000	Individual	Petrol	
1	Hyundai Grand	Hyundai	Grand	5	20000	Individual	Petrol	
2	Hyundai i20	Hyundai	i20	11	60000	Individual	Petrol	
3	Maruti Alto	Maruti	Alto	9	37000	Individual	Petrol	
4	Ford Ecosport	Ford	Ecosport	6	30000	Dealer	Diesel	
5	Maruti Wagon R	Maruti	Wagon R	8	35000	Individual	Petrol	
6	Hyundai i10	Hyundai	i10	8	40000	Dealer	Petrol	
7	Maruti Wagon R	Maruti	Wagon R	3	17512	Dealer	Petrol	
8	Hyundai Venue	Hyundai	Venue	2	20000	Individual	Petrol	
9	Maruti Swift	Maruti	Swift	4	28321	Dealer	Petrol	

In [22]:
```
df.tail()
```

Out[22]:

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	tra
15406	Hyundai i10	Hyundai	i10	9	10723	Dealer	Petrol	
15407	Maruti Ertiga	Maruti	Ertiga	2	18000	Dealer	Petrol	
15408	Skoda Rapid	Skoda	Rapid	6	67000	Dealer	Diesel	
15409	Mahindra XUV500	Mahindra	XUV500	5	3800000	Dealer	Diesel	
15410	Honda City	Honda	City	2	13000	Dealer	Petrol	

In [23]:
```
df.sample(5)
```

Out[23]:

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type
10885	Renault KWID	Renault	KWID	5	12000	Dealer	Petrol
3366	BMW X5	BMW	X5	13	61000	Dealer	Diesel
14026	Mahindra Scorpio	Mahindra	Scorpio	8	71000	Dealer	Diesel
8634	Volkswagen Polo	Volkswagen	Polo	7	72000	Individual	Petrol
11282	Maruti Wagon R	Maruti	Wagon R	8	55000	Dealer	Petrol

In [24]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15411 entries, 0 to 15410
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   car_name           15411 non-null  object 
 1   brand              15411 non-null  object 
 2   model              15411 non-null  object 
 3   vehicle_age        15411 non-null  int64  
 4   km_driven          15411 non-null  int64  
 5   seller_type        15411 non-null  object 
 6   fuel_type          15411 non-null  object 
 7   transmission_type  15411 non-null  object 
 8   mileage            15411 non-null  float64
 9   engine             15411 non-null  int64  
 10  max_power          15411 non-null  float64
 11  seats              15411 non-null  int64  
 12  selling_price      15411 non-null  int64  
dtypes: float64(2), int64(5), object(6)
memory usage: 1.5+ MB
```

In [25]:
```
df.dtypes
```

Out[25]:
```
car_name              object
brand                 object
model                 object
vehicle_age            int64
km_driven              int64
seller_type           object
fuel_type             object
transmission_type     object
mileage              float64
engine                 int64
max_power            float64
seats                  int64
selling_price          int64
dtype: object
```

In [26]:
```
df.dtypes == 'object'
```

Out[26]:
```
car_name              True
brand                 True
model                 True
vehicle_age          False
km_driven            False
seller_type           True
fuel_type             True
transmission_type     True
mileage              False
engine               False
max_power            False
seats                False
selling_price        False
dtype: bool
```

In [27]:
```
object_cols =df.select_dtypes('object').columns
```

In [28]:
```
number_cols =df.select_dtypes('number').columns
```

In [29]:
```
df.describe()
```

Out[29]:

	vehicle_age	km_driven	mileage	engine	max_power	s
count	15411.000000	1.541100e+04	15411.000000	15411.000000	15411.000000	15411.000
mean	6.036338	5.561648e+04	19.701151	1486.057751	100.588254	5.325
std	3.013291	5.161855e+04	4.171265	521.106696	42.972979	0.807
min	0.000000	1.000000e+02	4.000000	793.000000	38.400000	0.000
25%	4.000000	3.000000e+04	17.000000	1197.000000	74.000000	5.000
50%	6.000000	5.000000e+04	19.670000	1248.000000	88.500000	5.000
75%	8.000000	7.000000e+04	22.700000	1582.000000	117.300000	5.000
max	29.000000	3.800000e+06	33.540000	6592.000000	626.000000	9.000

In [30]:
```
df.describe().columns
```

Out[30]:
```
Index(['vehicle_age', 'km_driven', 'mileage', 'engine', 'max_power', 'seats',
       'selling_price'],
      dtype='object')
```

In [ ]:
```

```

In [31]:
```
number_cols
```

Out[31]:
```
Index(['vehicle_age', 'km_driven', 'mileage', 'engine', 'max_power', 'seats',
       'selling_price'],
      dtype='object')
```

In [32]:
```
df.describe().round(2)
```

Out[32]:

	vehicle_age	km_driven	mileage	engine	max_power	seats	selling_price
count	15411.00	15411.00	15411.00	15411.00	15411.00	15411.00	15411.00
mean	6.04	55616.48	19.70	1486.06	100.59	5.33	774971.12
std	3.01	51618.55	4.17	521.11	42.97	0.81	894128.36
min	0.00	100.00	4.00	793.00	38.40	0.00	40000.00
25%	4.00	30000.00	17.00	1197.00	74.00	5.00	385000.00
50%	6.00	50000.00	19.67	1248.00	88.50	5.00	556000.00
75%	8.00	70000.00	22.70	1582.00	117.30	5.00	825000.00
max	29.00	3800000.00	33.54	6592.00	626.00	9.00	39500000.00

In [33]:
```
df.describe().round()
```

Out[33]:

	vehicle_age	km_driven	mileage	engine	max_power	seats	selling_price
count	15411.0	15411.0	15411.0	15411.0	15411.0	15411.0	15411.0
mean	6.0	55616.0	20.0	1486.0	101.0	5.0	774971.0
std	3.0	51619.0	4.0	521.0	43.0	1.0	894128.0
min	0.0	100.0	4.0	793.0	38.0	0.0	40000.0
25%	4.0	30000.0	17.0	1197.0	74.0	5.0	385000.0
50%	6.0	50000.0	20.0	1248.0	88.0	5.0	556000.0
75%	8.0	70000.0	23.0	1582.0	117.0	5.0	825000.0
max	29.0	3800000.0	34.0	6592.0	626.0	9.0	39500000.0

In [34]:
```
df.describe(include="object")
```

Out[34]:

	car_name	brand	model	seller_type	fuel_type	transmission_type
count	15411	15411	15411	15411	15411	15411
unique	121	32	120	3	5	2
top	Hyundai i20	Maruti	i20	Dealer	Petrol	Manual
freq	906	4992	906	9539	7643	12225

In [35]:
```
df.describe(include="object").columns
```

Out[35]:
```
Index(['car_name', 'brand', 'model', 'seller_type', 'fuel_type',
       'transmission_type'],
      dtype='object')
```

In [36]:
```
df.isna().sum()
```

Out[36]:
```
car_name             0
brand                0
model                0
vehicle_age          0
km_driven            0
seller_type          0
fuel_type            0
transmission_type    0
mileage              0
engine               0
max_power            0
seats                0
selling_price        0
dtype: int64
```

In [37]:
```
df.isnull().sum()
```

Out[37]:
```
car_name             0
brand                0
model                0
vehicle_age          0
km_driven            0
seller_type          0
fuel_type            0
transmission_type    0
mileage              0
engine               0
max_power            0
seats                0
selling_price        0
dtype: int64
```

In [38]:
```
df.isnull().mean()
```

Out[38]:
```
car_name             0.0
brand                0.0
model                0.0
vehicle_age          0.0
km_driven            0.0
seller_type          0.0
fuel_type            0.0
transmission_type    0.0
mileage              0.0
engine               0.0
max_power            0.0
seats                0.0
selling_price        0.0
dtype: float64
```

In [39]:
```
# Slince no null values found , so we can  procedd for further analysis,
# hence no requirment of fillna
```

Univariate Analysis

- Single

Bivariate Analysis

- Double Colum Analysis

Multivariate Analysis

-Multiple cols Analysis

Univariate Analysys

In [40]:
```
df['brand'].unique()
```

Out[40]:
```
array(['Maruti', 'Hyundai', 'Ford', 'Renault', 'Mini', 'Mercedes-Benz',
       'Toyota', 'Volkswagen', 'Honda', 'Mahindra', 'Datsun', 'Tata',
       'Kia', 'BMW', 'Audi', 'Land Rover', 'Jaguar', 'MG', 'Isuzu',
       'Porsche', 'Skoda', 'Volvo', 'Lexus', 'Jeep', 'Maserati',
       'Bentley', 'Nissan', 'ISUZU', 'Ferrari', 'Mercedes-AMG',
       'Rolls-Royce', 'Force'], dtype=object)
```

In [41]:
```
df['brand'].nunique()
```

Out[41]:
```
32
```

In [42]:
```
df['brand'].value_counts()
```

Out[42]:
```
brand
Maruti           4992
Hyundai          2982
Honda            1485
Mahindra         1011
Toyota            793
Ford              790
Volkswagen        620
Renault           536
BMW               439
Tata              430
Mercedes-Benz     337
Skoda             334
Audi              192
Datsun            170
Jaguar             59
Land Rover         51
Jeep               41
Kia                32
Porsche            21
Volvo              20
MG                 19
Mini               17
Nissan             11
Lexus              10
Isuzu               8
Bentley             3
Maserati            2
ISUZU               2
Ferrari             1
Mercedes-AMG        1
Rolls-Royce         1
Force               1
Name: count, dtype: int64
```

In [43]:
```
df['brand'].value_counts().head(10)
```

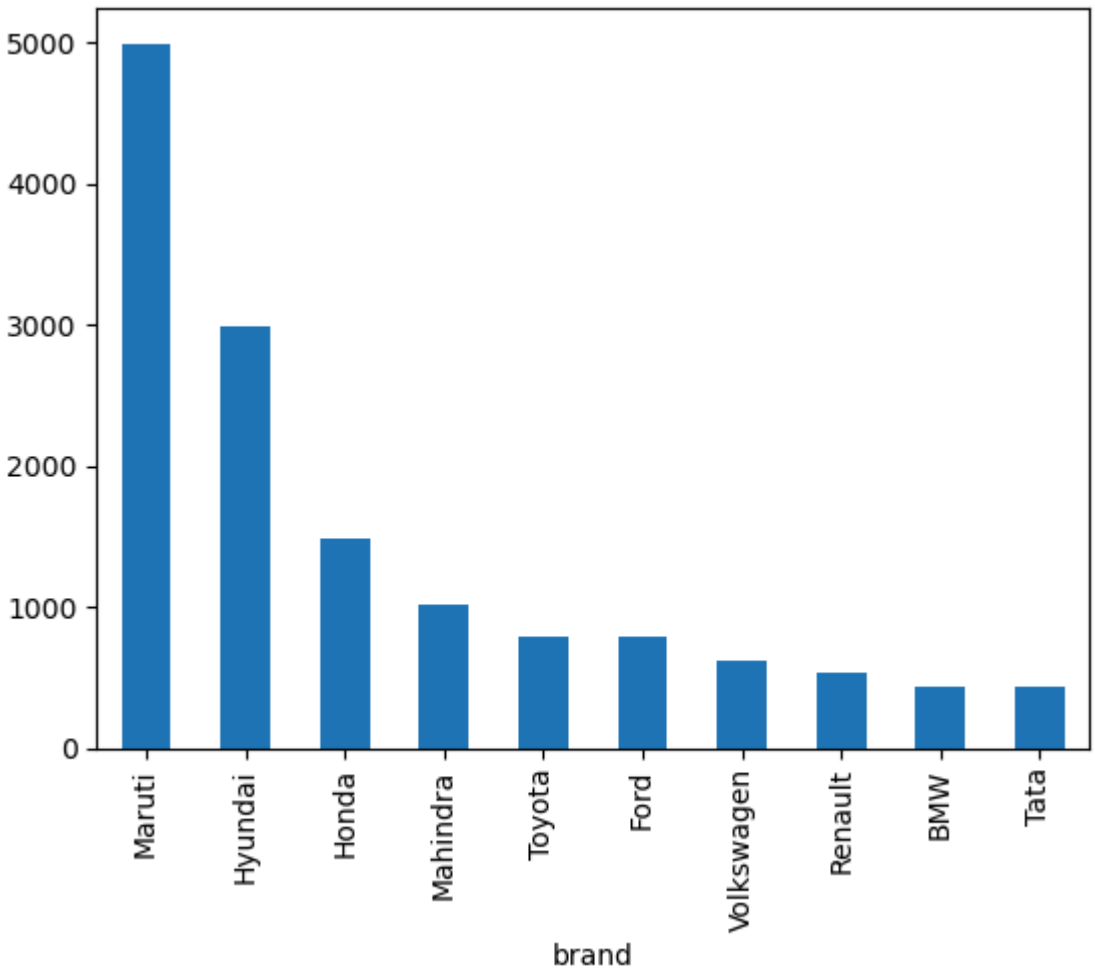
Out[43]:
```
brand
Maruti        4992
Hyundai       2982
Honda         1485
Mahindra      1011
Toyota         793
Ford           790
Volkswagen     620
Renault        536
BMW            439
Tata           430
Name: count, dtype: int64
```

In [44]:
```
df['brand'].value_counts().head(10).sum()
```

Out[44]:
```
np.int64(14078)
```

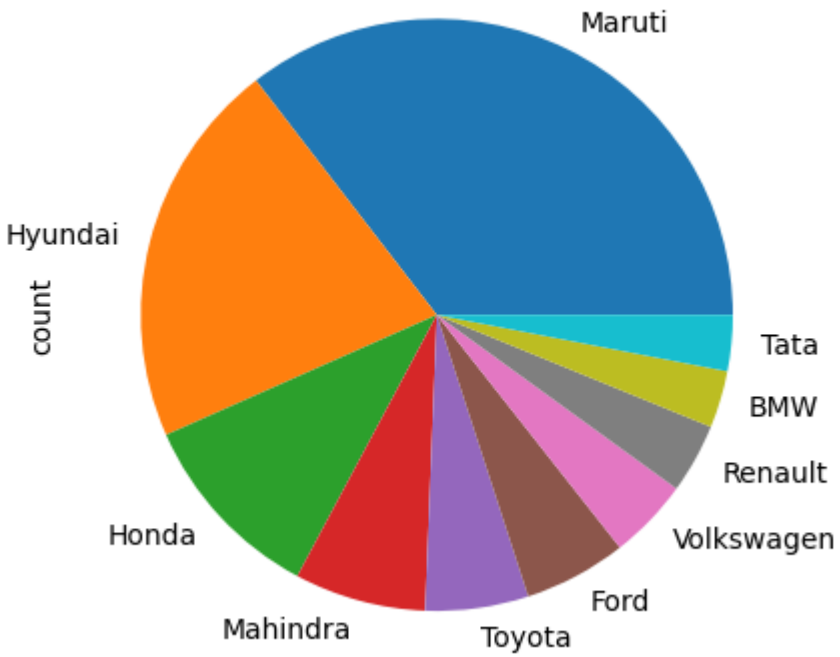
In [45]:
```
df['brand'].value_counts().head(10).plot(kind='bar')
```

Out[45]:
```
<Axes: xlabel='brand'>
```



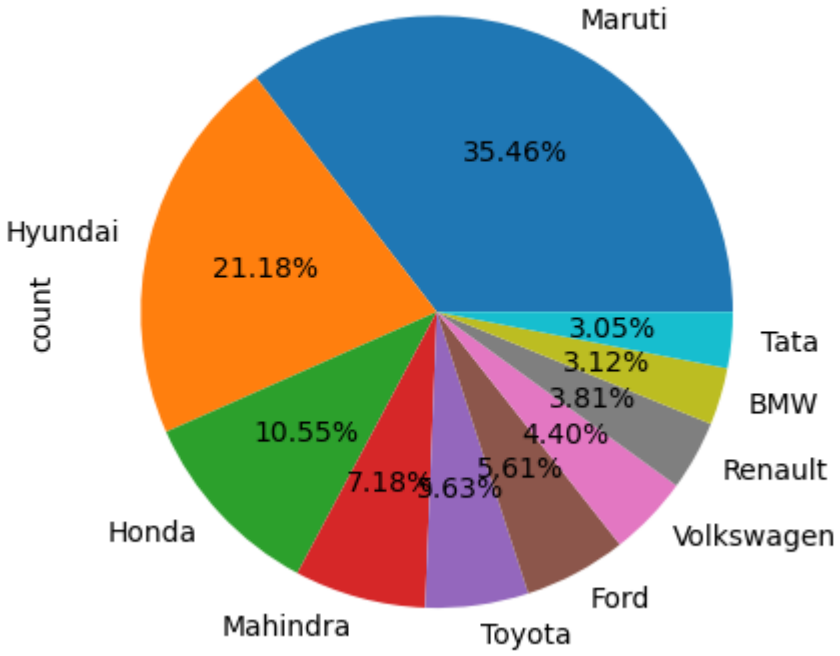
In [46]:
```
df['brand'].value_counts().head(10).plot(kind='pie')
```

Out[46]:
```
<Axes: ylabel='count'>
```



In [47]:
```
df['brand'].value_counts().head(10).plot(kind='pie',autopct ='%.2f%%')
```

Out[47]: <Axes: ylabel='count'>



All Object Cols Unique values

In [48]:
```
for i in object_cols:
    print(f'{i.upper()}: >> {df[i].unique()}')
    print('-------------------------------\n\n')
```

```
CAR_NAME: >> ['Maruti Alto' 'Hyundai Grand' 'Hyundai i20' 'Ford Ecosport'
 'Maruti Wagon R' 'Hyundai i10' 'Hyundai Venue' 'Maruti Swift'
 'Hyundai Verna' 'Renault Duster' 'Mini Cooper' 'Maruti Ciaz'
 'Mercedes-Benz C-Class' 'Toyota Innova' 'Maruti Baleno'
 'Maruti Swift Dzire' 'Volkswagen Vento' 'Hyundai Creta' 'Honda City'
 'Mahindra Bolero' 'Toyota Fortuner' 'Renault KWID' 'Honda Amaze'
 'Hyundai Santro' 'Mahindra XUV500' 'Mahindra KUV100' 'Maruti Ignis'
 'Datsun RediGO' 'Mahindra Scorpio' 'Mahindra Marazzo' 'Ford Aspire'
 'Ford Figo' 'Maruti Vitara' 'Tata Tiago' 'Volkswagen Polo' 'Kia Seltos'
 'Maruti Celerio' 'Datsun GO' 'BMW 5' 'Honda CR-V' 'Ford Endeavour'
 'Mahindra KUV' 'Honda Jazz' 'BMW 3' 'Audi A4' 'Tata Tigor'
 'Maruti Ertiga' 'Tata Safari' 'Mahindra Thar' 'Tata Hexa'
 'Land Rover Rover' 'Maruti Eeco' 'Audi A6' 'Mercedes-Benz E-Class'
 'Audi Q7' 'BMW Z4' 'BMW 6' 'Jaguar XF' 'BMW X5' 'MG Hector' 'Honda Civic'
 'Isuzu D-Max' 'Porsche Cayenne' 'BMW X1' 'Skoda Rapid' 'Ford Freestyle'
 'Skoda Superb' 'Tata Nexon' 'Mahindra XUV300' 'Maruti Dzire VXI'
 'Volvo S90' 'Honda WR-V' 'Maruti XL6' 'Renault Triber' 'Lexus ES'
 'Jeep Wrangler' 'Toyota Camry' 'Hyundai Elantra' 'Toyota Yaris'
 'Mercedes-Benz GL-Class' 'BMW 7' 'Maruti S-Presso' 'Maruti Dzire LXI'
 'Hyundai Aura' 'Volvo XC' 'Maserati Ghibli' 'Bentley Continental'
 'Honda CR' 'Nissan Kicks' 'Mercedes-Benz S-Class' 'Hyundai Tucson'
 'Tata Harrier' 'BMW X3' 'Skoda Octavia' 'Jeep Compass'
 'Mercedes-Benz CLS' 'Datsun redi-GO' 'Toyota Glanza' 'Porsche Macan'
 'BMW X4' 'Maruti Dzire ZXI' 'Volvo XC90' 'Jaguar F-PACE' 'Audi A8'
 'ISUZU MUX' 'Ferrari GTC4Lusso' 'Mercedes-Benz GLS' 'Nissan X-Trail'
 'Jaguar XE' 'Volvo XC60' 'Porsche Panamera' 'Mahindra Alturas'
 'Tata Altroz' 'Lexus NX' 'Kia Carnival' 'Mercedes-AMG C' 'Lexus RX'
 'Rolls-Royce Ghost' 'Maserati Quattroporte' 'Isuzu MUX' 'Force Gurkha']
--------------------------------


BRAND: >> ['Maruti' 'Hyundai' 'Ford' 'Renault' 'Mini' 'Mercedes-Benz' 'Toyota'
 'Volkswagen' 'Honda' 'Mahindra' 'Datsun' 'Tata' 'Kia' 'BMW' 'Audi'
 'Land Rover' 'Jaguar' 'MG' 'Isuzu' 'Porsche' 'Skoda' 'Volvo' 'Lexus'
 'Jeep' 'Maserati' 'Bentley' 'Nissan' 'ISUZU' 'Ferrari' 'Mercedes-AMG'
 'Rolls-Royce' 'Force']
--------------------------------


MODEL: >> ['Alto' 'Grand' 'i20' 'Ecosport' 'Wagon R' 'i10' 'Venue' 'Swift' 'Vern
a'
 'Duster' 'Cooper' 'Ciaz' 'C-Class' 'Innova' 'Baleno' 'Swift Dzire'
 'Vento' 'Creta' 'City' 'Bolero' 'Fortuner' 'KWID' 'Amaze' 'Santro'
 'XUV500' 'KUV100' 'Ignis' 'RediGO' 'Scorpio' 'Marazzo' 'Aspire' 'Figo'
 'Vitara' 'Tiago' 'Polo' 'Seltos' 'Celerio' 'GO' '5' 'CR-V' 'Endeavour'
 'KUV' 'Jazz' '3' 'A4' 'Tigor' 'Ertiga' 'Safari' 'Thar' 'Hexa' 'Rover'
 'Eeco' 'A6' 'E-Class' 'Q7' 'Z4' '6' 'XF' 'X5' 'Hector' 'Civic' 'D-Max'
 'Cayenne' 'X1' 'Rapid' 'Freestyle' 'Superb' 'Nexon' 'XUV300' 'Dzire VXI'
 'S90' 'WR-V' 'XL6' 'Triber' 'ES' 'Wrangler' 'Camry' 'Elantra' 'Yaris'
 'GL-Class' '7' 'S-Presso' 'Dzire LXI' 'Aura' 'XC' 'Ghibli' 'Continental'
 'CR' 'Kicks' 'S-Class' 'Tucson' 'Harrier' 'X3' 'Octavia' 'Compass' 'CLS'
 'redi-GO' 'Glanza' 'Macan' 'X4' 'Dzire ZXI' 'XC90' 'F-PACE' 'A8' 'MUX'
 'GTC4Lusso' 'GLS' 'X-Trail' 'XE' 'XC60' 'Panamera' 'Alturas' 'Altroz'
 'NX' 'Carnival' 'C' 'RX' 'Ghost' 'Quattroporte' 'Gurkha']
--------------------------------


SELLER_TYPE: >> ['Individual' 'Dealer' 'Trustmark Dealer']
--------------------------------
```

```
FUEL_TYPE: >> ['Petrol' 'Diesel' 'CNG' 'LPG' 'Electric']
------------------------------


TRANSMISSION_TYPE: >> ['Manual' 'Automatic']
------------------------------
```

In [49]:
```
for i in object_cols:
    print(f'{i.upper()}:>> {df[i].nunique()}')
    print('------------------------------\n')
```

```
CAR_NAME:>> 121
------------------------------

BRAND:>> 32
------------------------------

MODEL:>> 120
------------------------------

SELLER_TYPE:>> 3
------------------------------

FUEL_TYPE:>> 5
------------------------------

TRANSMISSION_TYPE:>> 2
------------------------------
```

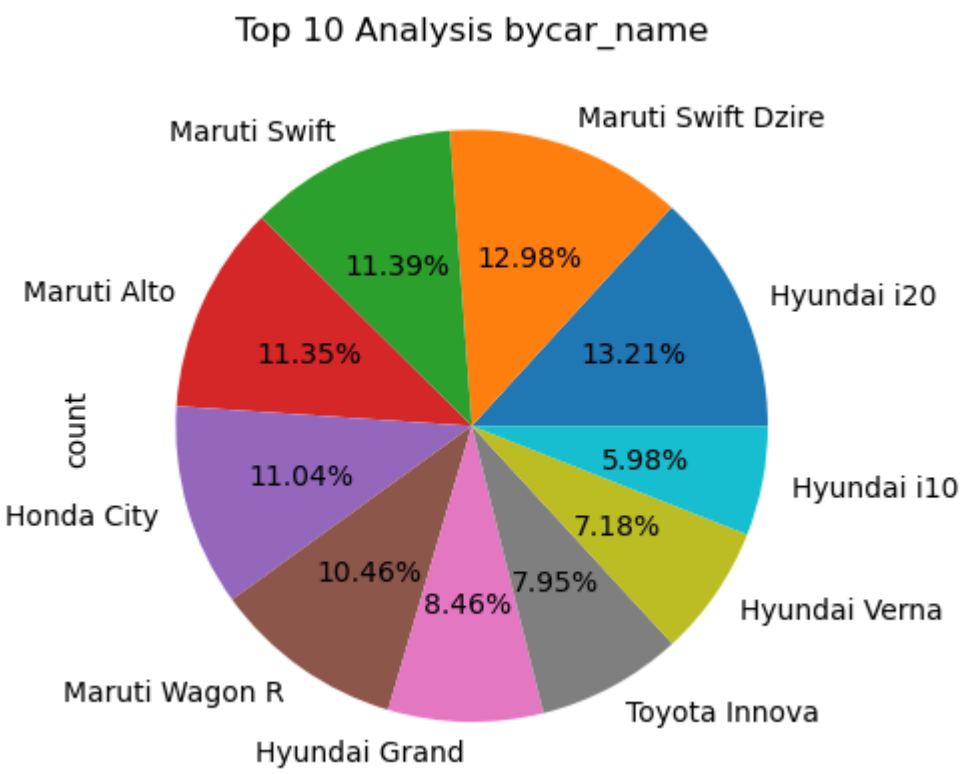
In [50]:
```
for i in object_cols:
    print(f'''{i.upper()}:>>
    {df[i].value_counts().head(10)}''')
    print('------------------------------\n\n')
```

```
CAR_NAME:>>
    car_name
Hyundai i20           906
Maruti Swift Dzire    890
Maruti Swift          781
Maruti Alto           778
Honda City            757
Maruti Wagon R        717
Hyundai Grand         580
Toyota Innova         545
Hyundai Verna         492
Hyundai i10           410
Name: count, dtype: int64
--------------------------------


BRAND:>>
    brand
Maruti        4992
Hyundai       2982
Honda         1485
Mahindra      1011
Toyota         793
Ford           790
Volkswagen     620
Renault        536
BMW            439
Tata           430
Name: count, dtype: int64
--------------------------------


MODEL:>>
    model
i20            906
Swift Dzire    890
Swift          781
Alto           778
City           757
Wagon R        717
Grand          580
Innova         545
Verna          492
i10            410
Name: count, dtype: int64
--------------------------------


SELLER_TYPE:>>
    seller_type
Dealer              9539
Individual          5699
Trustmark Dealer     173
Name: count, dtype: int64
--------------------------------


FUEL_TYPE:>>
    fuel_type
Petrol      7643
```

```
Diesel      7419
CNG          301
LPG           44
Electric       4
Name: count, dtype: int64
--------------------------------


TRANSMISSION_TYPE:>>
    transmission_type
Manual       12225
Automatic     3186
Name: count, dtype: int64
--------------------------------
```

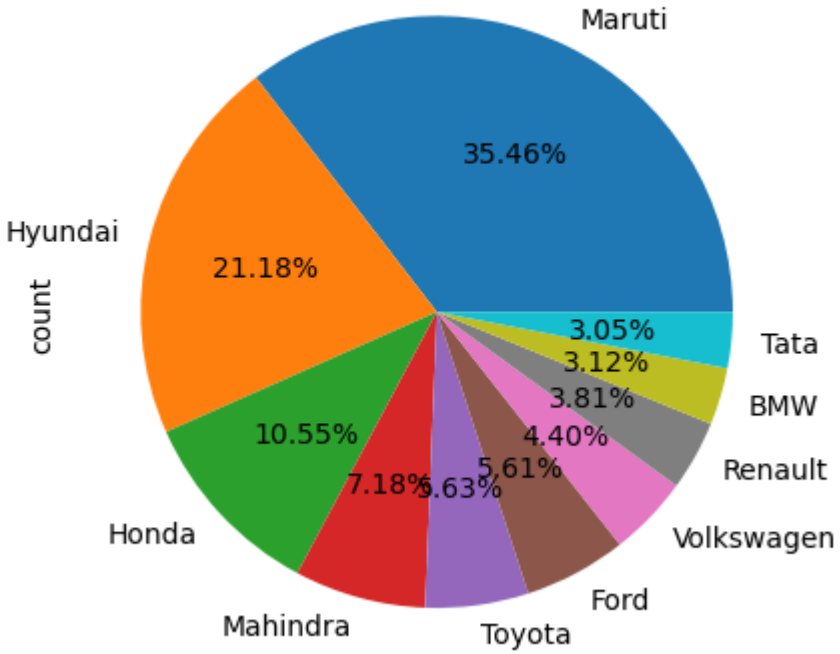
In [51]:
```
import matplotlib.pyplot as plt

for i in object_cols:
    df[i].value_counts().head(10).plot(kind='pie',autopct='%.2f%%')
    print('--------------------------------\n')
    plt.title(f'Top 10 Analysis by{i}')
    plt.show()
```

```
--------------------------------
```

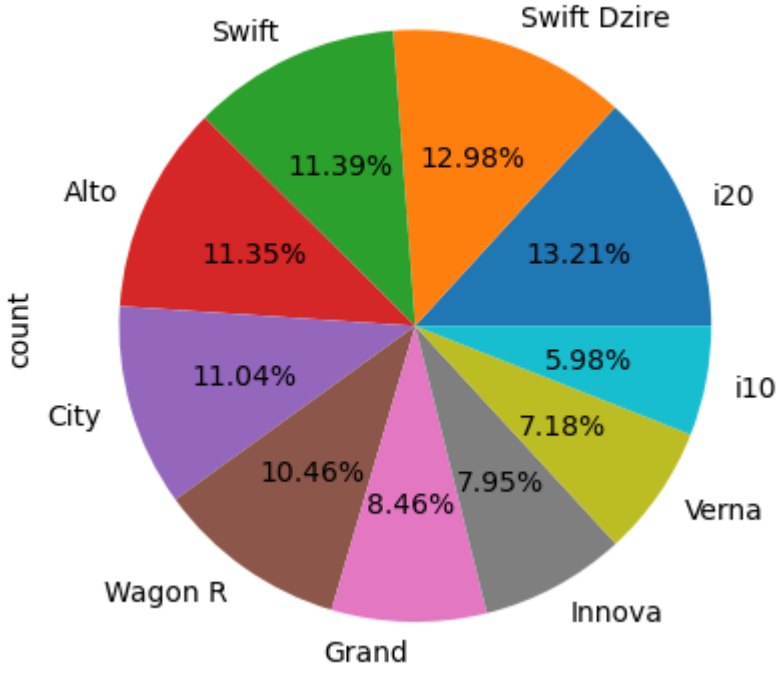


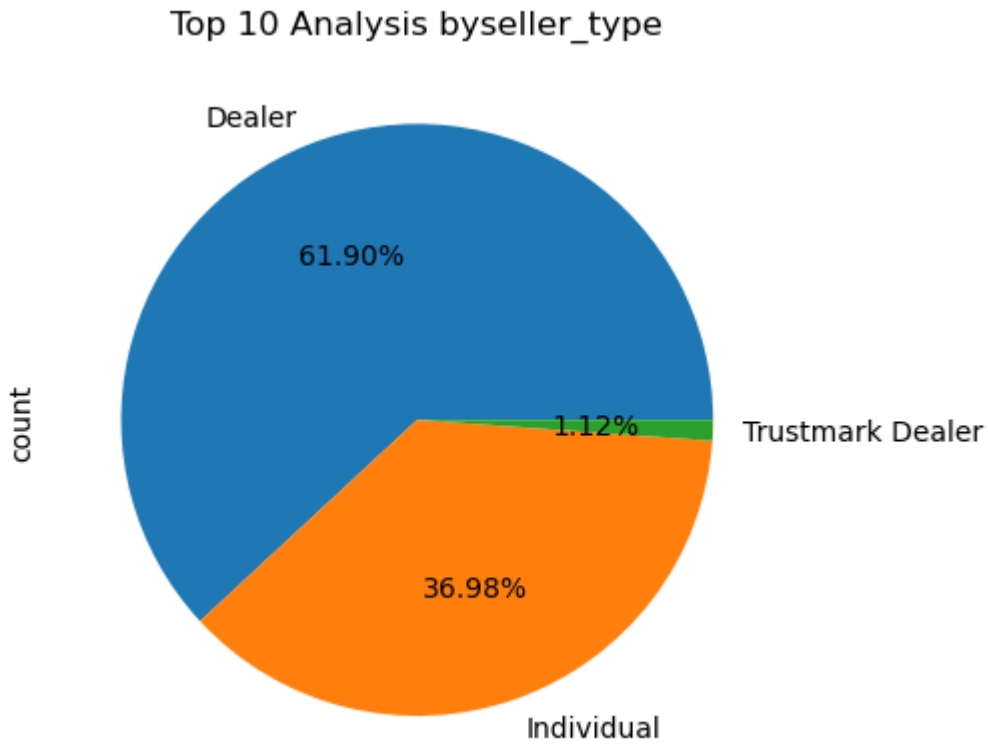

```
--------------------------------
```

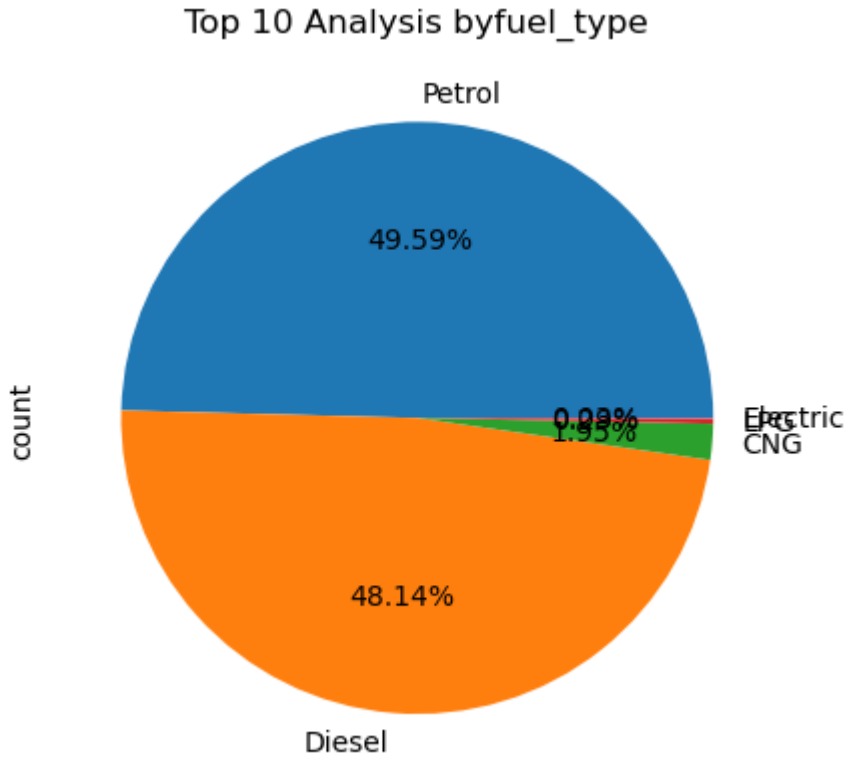
Top 10 Analysis bybrand

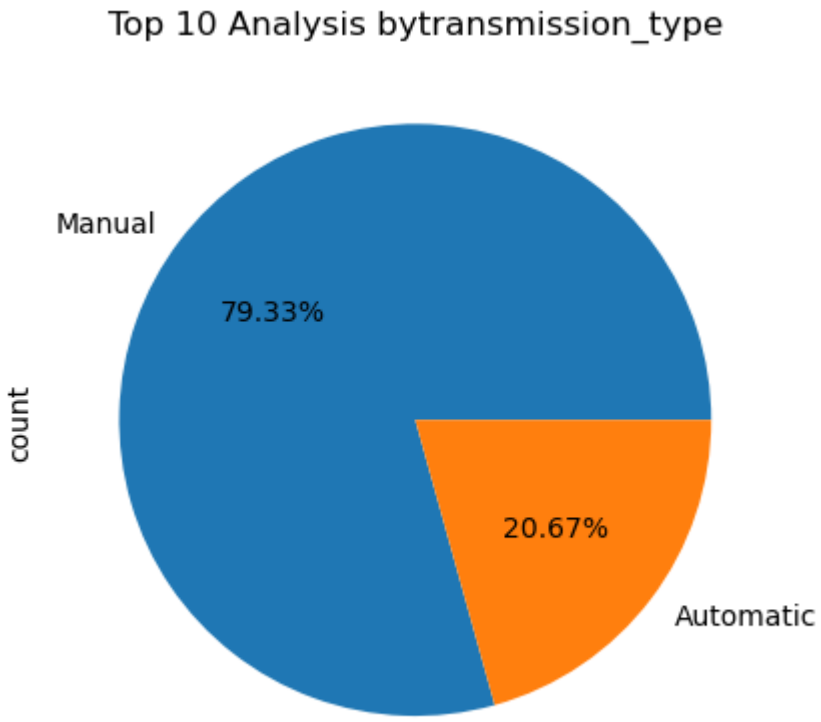


Top 10 Analysis bymodel




Top 10 Analysis byseller_type
Dealer
61.90%
count
1.12%
Trustmark Dealer
36.98%
Individual

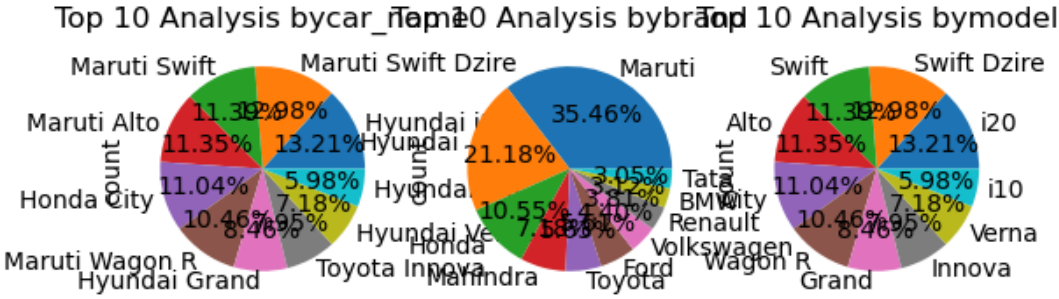

Top 10 Analysis byfuel_type
Petrol
49.59%
count
Electric
CNG
48.14%
Diesel

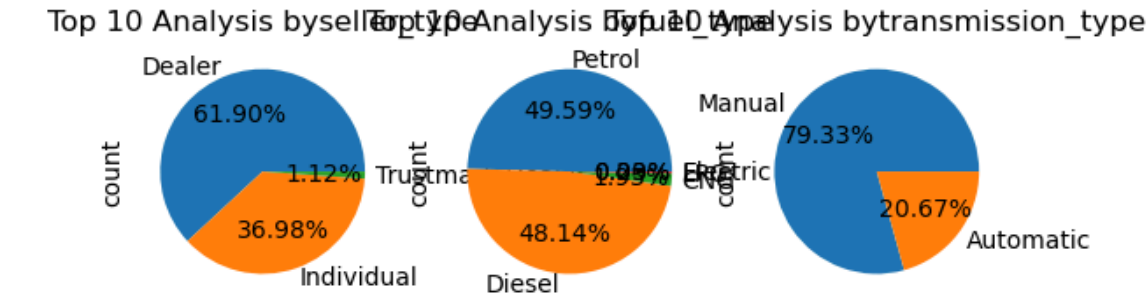


In [52]:
```
print(len(object_cols))
```

6

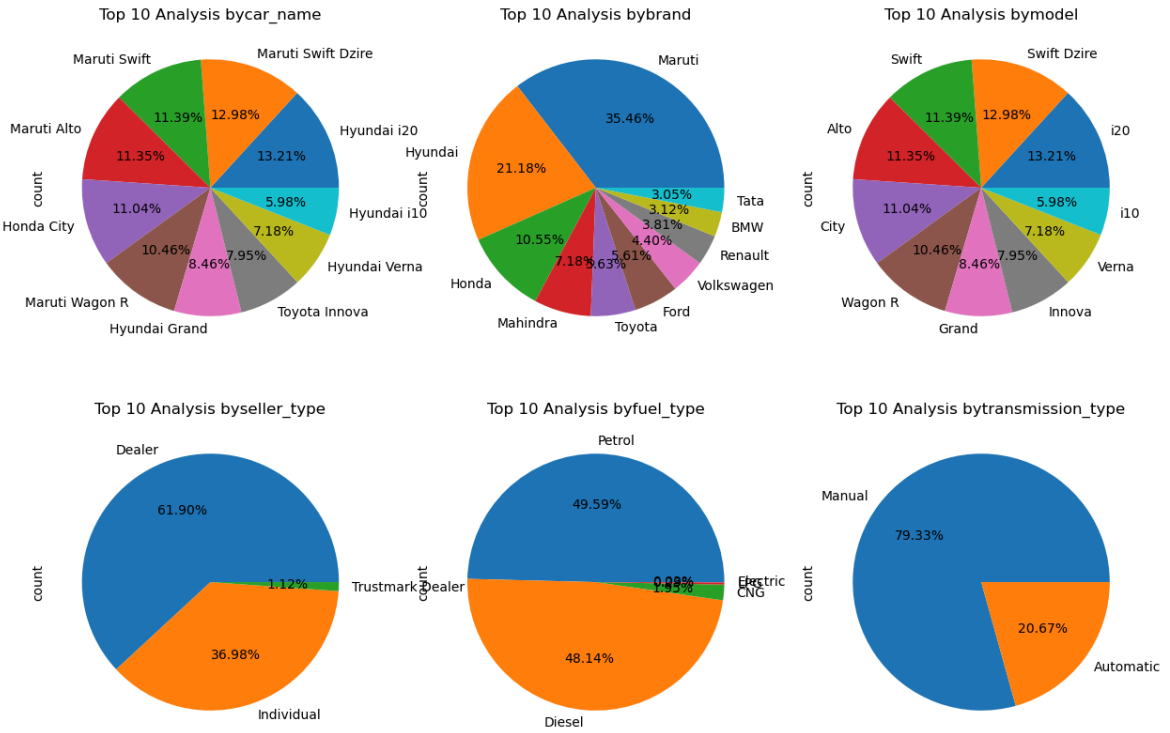
In [53]:
```
import matplotlib.pyplot as plt
for i,j in enumerate(object_cols):
    plt.subplot(2,3,i+1)
    df[j].value_counts().head(10).plot(kind='pie',autopct='%.2f%%')
    plt.title(f'Top 10 Analysis by{j}')
plt.show()
```





In [54]:
```
import matplotlib.pyplot as plt

plt.figure(figsize=(15,10))
```

```
for i,j in enumerate(object_cols):
    plt.subplot(2,3,i+1)
    df[j].value_counts().head(10).plot(kind='pie',autopct='%.2f%%')
    plt.title(f'Top 10 Analysis by{j}')
plt.show()
```



In [55]:
```
print(object_cols)
```

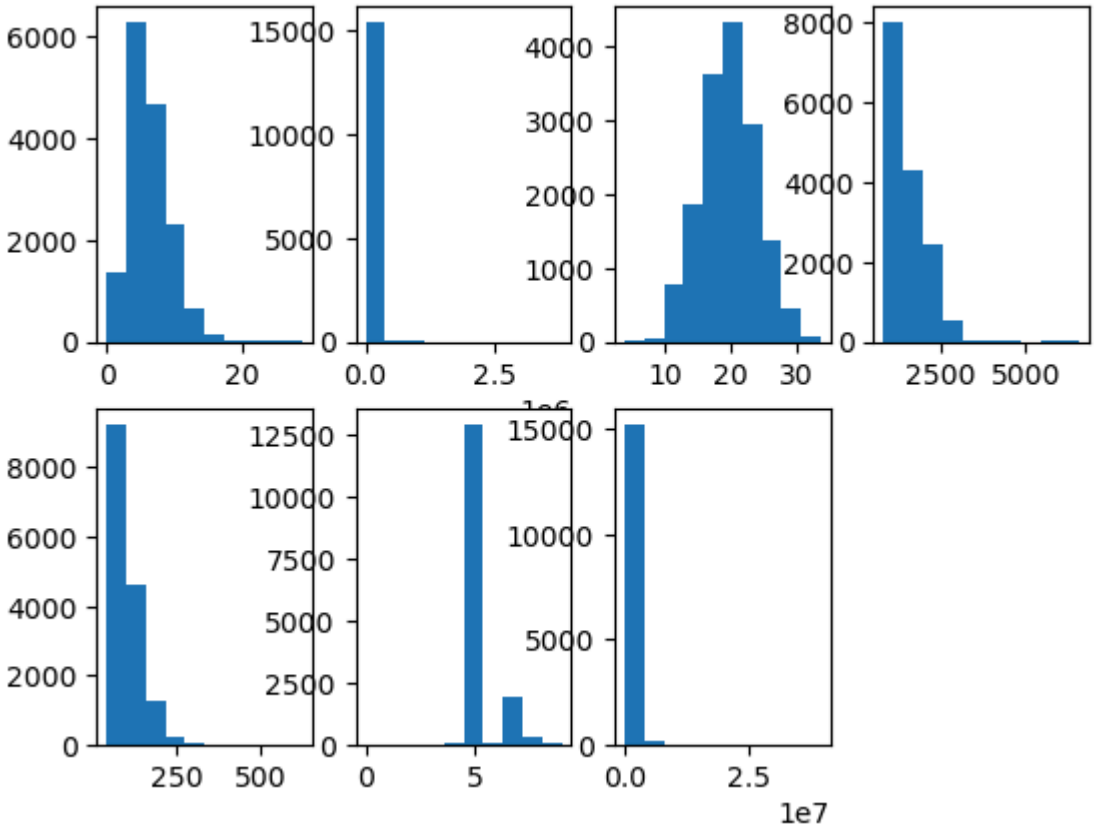
```
Index(['car_name', 'brand', 'model', 'seller_type', 'fuel_type',
       'transmission_type'],
      dtype='object')
```

Number Cols Distribution

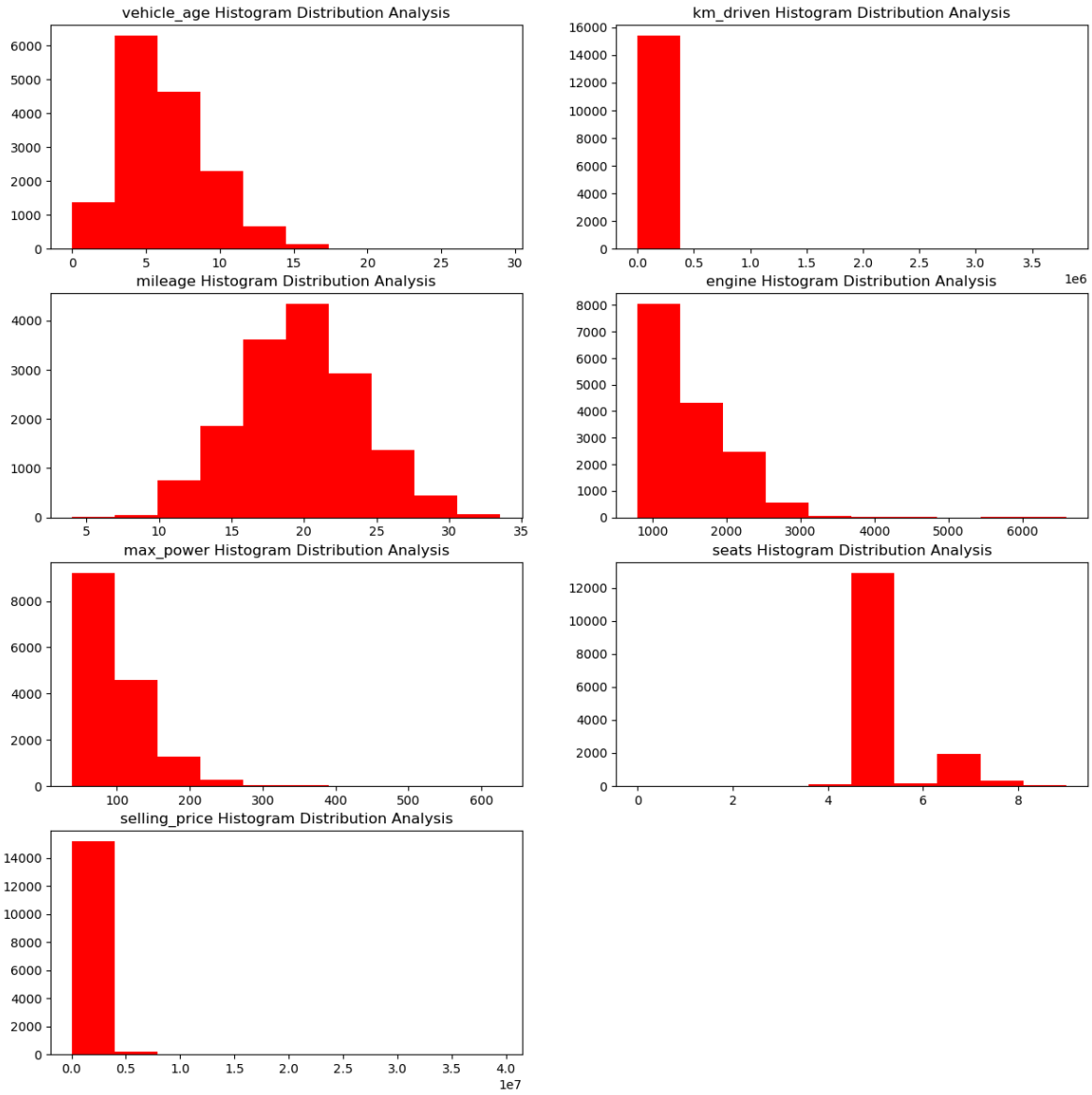
In [56]:
```
number_cols
```

Out[56]:
```
Index(['vehicle_age', 'km_driven', 'mileage', 'engine', 'max_power', 'seats',
       'selling_price'],
      dtype='object')
```

In [57]:
```
for index,value in enumerate(number_cols):
    plt.subplot(2,4,index + 1)
    plt.hist(df[value])
```



In [58]:
```
plt.figure(figsize=(15,15))
for index,value in enumerate(number_cols):
    plt.subplot(4,2,index + 1)
    plt.title(f'{value} Histogram Distribution Analysis')
    plt.hist(df[value], color ='r')
plt.show()
```



Bivariate Analysis

-- Doblue Column Analysis

In [59]:
```
df.columns
```

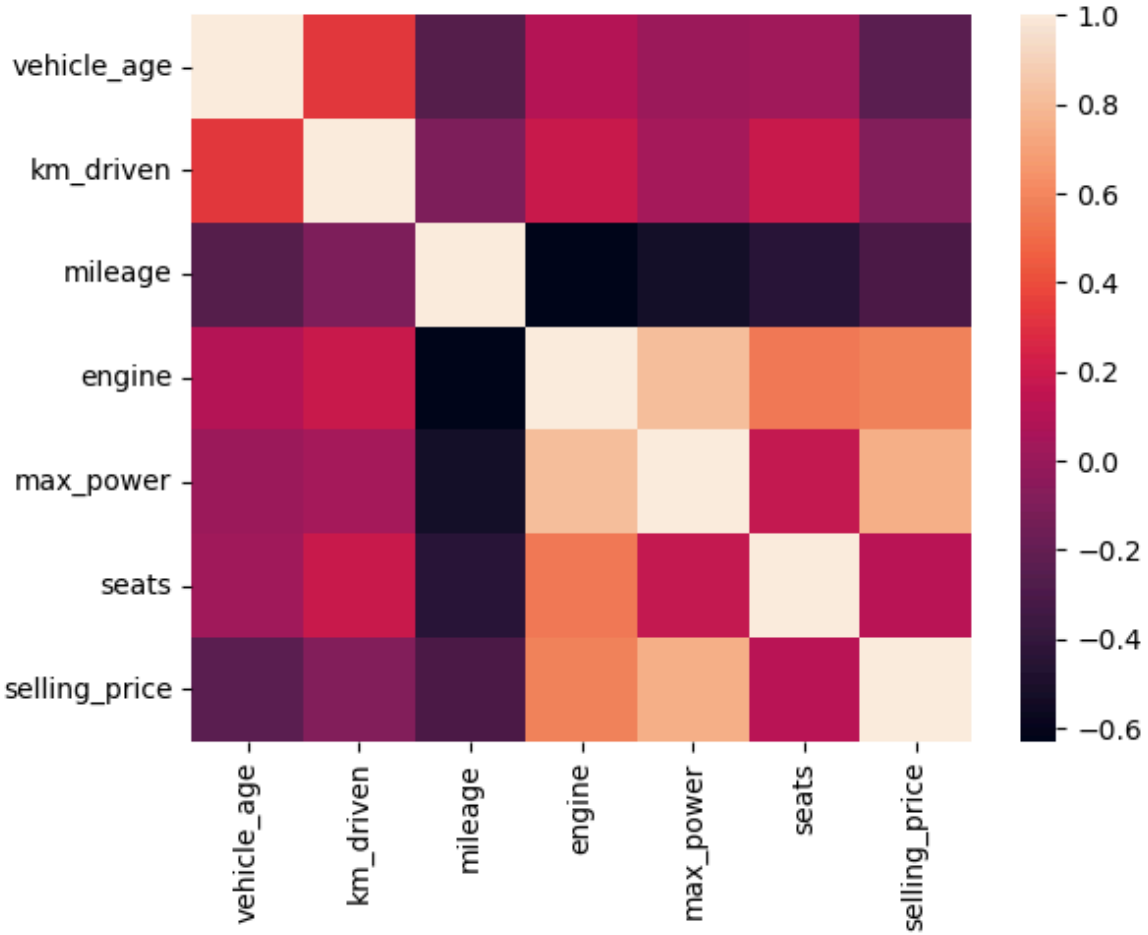
Out[59]:
```
Index(['car_name', 'brand', 'model', 'vehicle_age', 'km_driven', 'seller_type',
       'fuel_type', 'transmission_type', 'mileage', 'engine', 'max_power',
       'seats', 'selling_price'],
      dtype='object')
```

In [60]:
```
'''
-1 to 1: Corr Range
-1 : 100 % negative Correlation
+1 : 100 % Positive Correlation
    .>> Ek Value ke Badhne par dusre 100% se aage Badhega

0 : No Correlation
# Only Works with Numerical Data
'''

corr=df.corr(numeric_only=True).round(2)
```

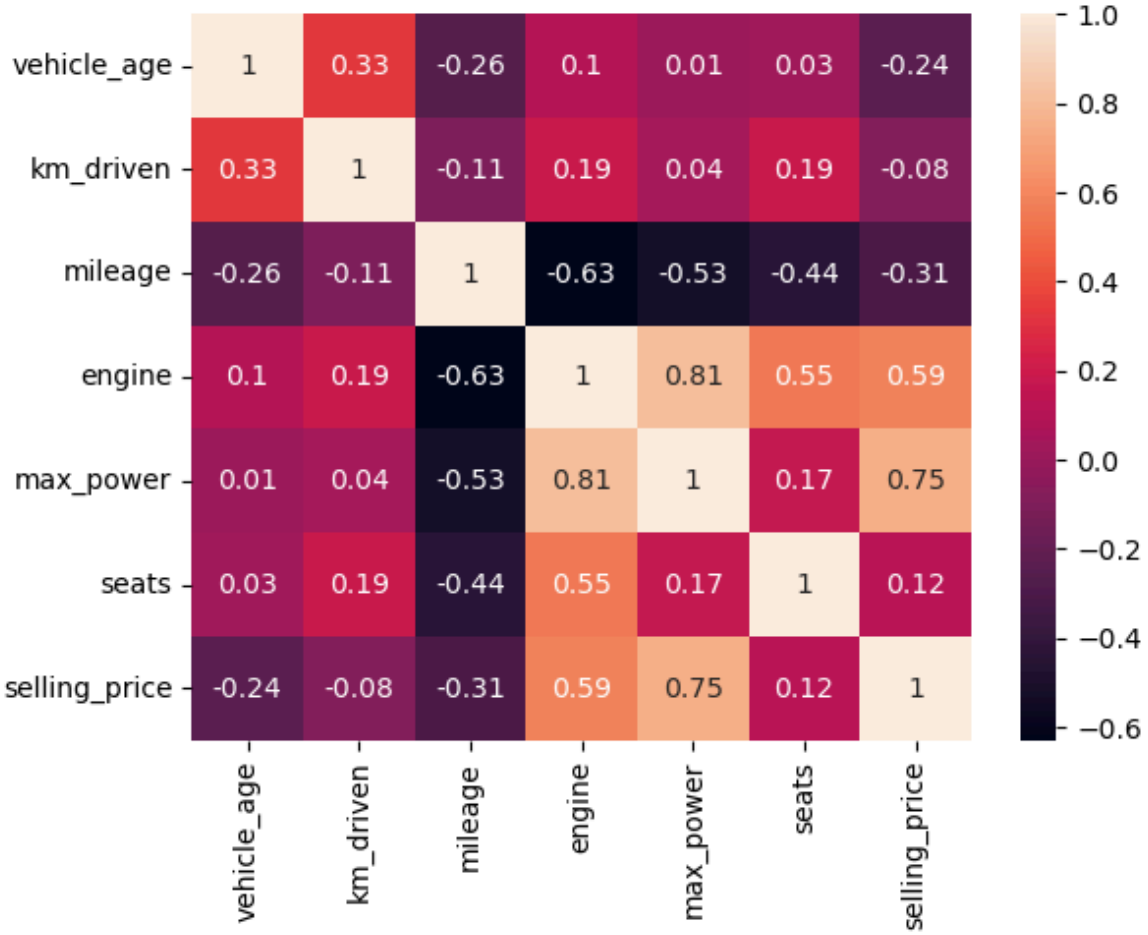
HetaMap

In [61]:
```
import seaborn as sns
```

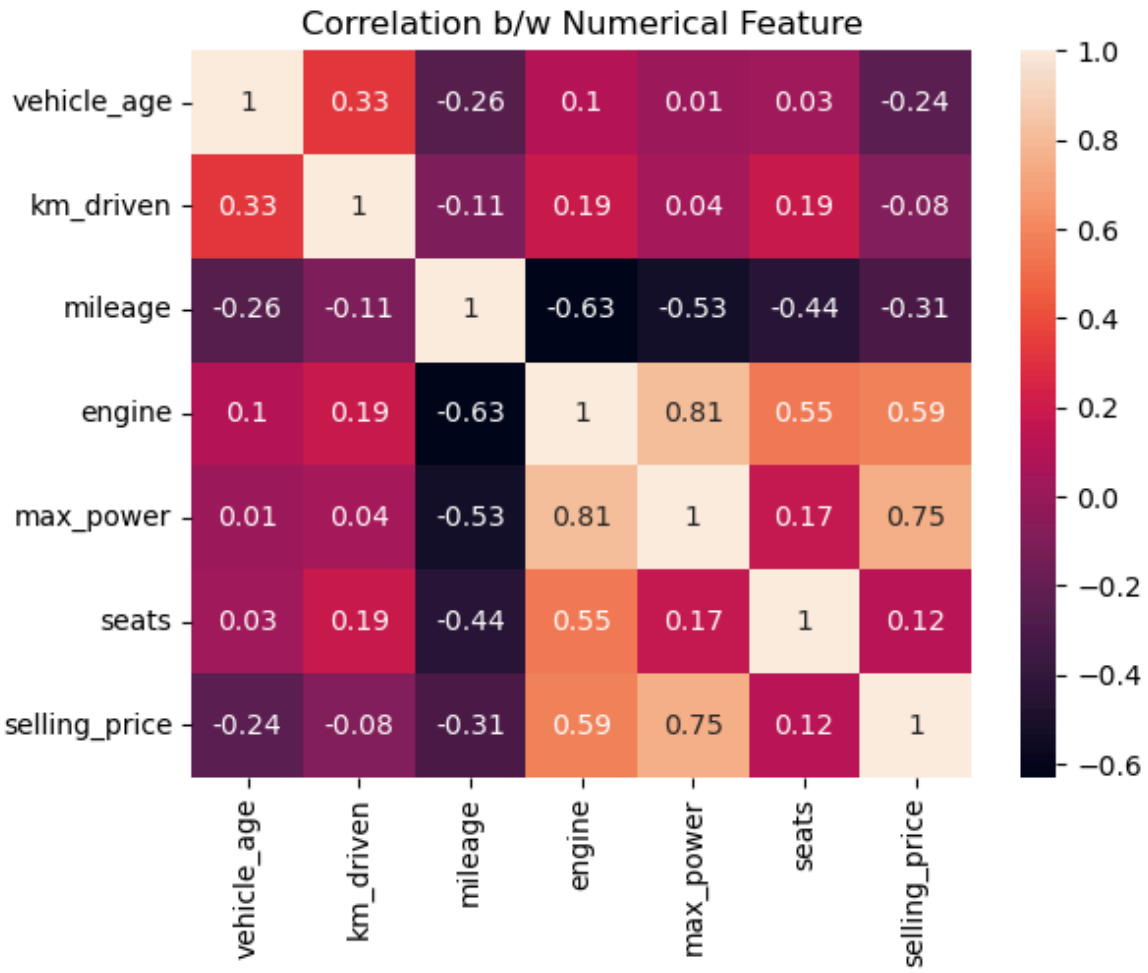
In [62]:
```
sns.heatmap(corr)
plt.show()
```



In [63]:
```
sns.heatmap(corr,annot =True)
plt.show()
```



In [64]:
```
plt.title('Correlation b/w Numerical Feature')
sns.heatmap(corr,annot =True)
plt.show()
```



QNA

Masking: To Filter Record Based On Given Conditon/s

In [65]:
```
df.sample(3)
```

Out[65]:

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	transr
10082	Maruti Alto	Maruti	Alto	6	36390	Dealer	Petrol	
1288	Honda City	Honda	City	11	68000	Dealer	Petrol	
10944	Renault KWID	Renault	KWID	2	14000	Individual	Petrol	

Business & Customers Mindset Questions

-1. Max Driven Car ,Min Driven car -2. Costly Car -3. Cheap Car -4. 5 Seater, Petrol, Automatic, Low Driven, Vehical Age Low, Maruti, Seller type: Individual. -5. 1500 cc> , 7-8

Lakh

In [66]:
```
costly_price=df['selling_price'].max()
```

In [67]:
```
df['selling_price']==costly_price
```

Out[67]:
```
0        False
1        False
2        False
3        False
4        False
         ...
15406    False
15407    False
15408    False
15409    False
15410    False
Name: selling_price, Length: 15411, dtype: bool
```

In [68]:
```
df[df['selling_price']==costly_price]
```

Out[68]:

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	trar
3799	Ferrari GTC4Lusso	Ferrari	GTC4Lusso	2	3800	Dealer	Petrol	

In [69]:
```
df[df['selling_price']==df['selling_price'].min()]
```

Out[69]:

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	transmi
7607	Maruti Wagon R	Maruti	Wagon R	21	80000	Individual	Petrol	

In [70]:
```
df[df['brand']=='Maruti']
```

Out[70]:

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	transn
0	Maruti Alto	Maruti	Alto	9	120000	Individual	Petrol	
3	Maruti Alto	Maruti	Alto	9	37000	Individual	Petrol	
5	Maruti Wagon R	Maruti	Wagon R	8	35000	Individual	Petrol	
7	Maruti Wagon R	Maruti	Wagon R	3	17512	Dealer	Petrol	
9	Maruti Swift	Maruti	Swift	4	28321	Dealer	Petrol	
...	...	...	...	...	...	...	...	
15398	Maruti Alto	Maruti	Alto	7	44000	Dealer	CNG	
15400	Maruti Alto	Maruti	Alto	4	12706	Dealer	Petrol	
15401	Maruti Swift	Maruti	Swift	3	25000	Individual	Petrol	
15404	Maruti Ertiga	Maruti	Ertiga	5	56829	Dealer	Diesel	
15407	Maruti Ertiga	Maruti	Ertiga	2	18000	Dealer	Petrol	

4992 rows × 13 columns

In [71]:
```
df[(df['brand']=='Maruti') & (df['fuel_type']=='Petrol')]
```

Out[71]:

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	transn
0	Maruti Alto	Maruti	Alto	9	120000	Individual	Petrol	
3	Maruti Alto	Maruti	Alto	9	37000	Individual	Petrol	
5	Maruti Wagon R	Maruti	Wagon R	8	35000	Individual	Petrol	
7	Maruti Wagon R	Maruti	Wagon R	3	17512	Dealer	Petrol	
9	Maruti Swift	Maruti	Swift	4	28321	Dealer	Petrol	
...	...	...	...	...	...	...	...	
15388	Maruti Alto	Maruti	Alto	10	41000	Dealer	Petrol	
15390	Maruti Celerio	Maruti	Celerio	6	60000	Individual	Petrol	
15400	Maruti Alto	Maruti	Alto	4	12706	Dealer	Petrol	
15401	Maruti Swift	Maruti	Swift	3	25000	Individual	Petrol	
15407	Maruti Ertiga	Maruti	Ertiga	2	18000	Dealer	Petrol	

2979 rows × 13 columns

In [72]:
```
df[(df['brand']=='Maruti') & (df['fuel_type']=='Petrol') & (df['mileage'] >=20)&
```

Out[72]:

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	transn
3	Maruti Alto	Maruti	Alto	9	37000	Individual	Petrol	
7	Maruti Wagon R	Maruti	Wagon R	3	17512	Dealer	Petrol	
18	Maruti Baleno	Maruti	Baleno	6	20000	Individual	Petrol	
19	Maruti Swift Dzire	Maruti	Swift Dzire	5	40000	Individual	Petrol	
21	Maruti Alto	Maruti	Alto	8	25000	Dealer	Petrol	
...	...	...	...	...	...	...	...	
15370	Maruti Wagon R	Maruti	Wagon R	6	21321	Dealer	Petrol	
15388	Maruti Alto	Maruti	Alto	10	41000	Dealer	Petrol	
15390	Maruti Celerio	Maruti	Celerio	6	60000	Individual	Petrol	
15400	Maruti Alto	Maruti	Alto	4	12706	Dealer	Petrol	
15401	Maruti Swift	Maruti	Swift	3	25000	Individual	Petrol	

2049 rows × 13 columns

In [73]:
```
df[(df['brand']=='Maruti') & (df['fuel_type']=='Petrol') & (df['mileage'] >=25)&
```

Out[73]:

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	transm
14593	Maruti Ciaz	Maruti	Ciaz	4	44771	Dealer	Petrol	

In [74]:
```
df[(df['brand']=='Maruti') &
    (df['fuel_type']=='Petrol')&
    (df['mileage'] >= 20) &
    (df['seats'] == 5) &
    (df['vehicle_age'] <=3)]
```

Out[74]:

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	transn
7	Maruti Wagon R	Maruti	Wagon R	3	17512	Dealer	Petrol	
44	Maruti Wagon R	Maruti	Wagon R	2	10000	Individual	Petrol	
45	Maruti Ignis	Maruti	Ignis	2	20000	Individual	Petrol	
61	Maruti Swift Dzire	Maruti	Swift Dzire	3	30000	Individual	Petrol	
92	Maruti Celerio	Maruti	Celerio	3	5000	Individual	Petrol	
...	...	...	...	...	...	...	...	
15219	Maruti Swift	Maruti	Swift	3	17000	Dealer	Petrol	
15241	Maruti Swift	Maruti	Swift	2	19587	Dealer	Petrol	
15273	Maruti Swift	Maruti	Swift	3	19000	Dealer	Petrol	
15297	Maruti Wagon R	Maruti	Wagon R	3	4500	Dealer	Petrol	
15401	Maruti Swift	Maruti	Swift	3	25000	Individual	Petrol	

656 rows × 13 columns

In [75]:

```
df[(df['brand']=='Maruti') &
    (df['fuel_type']=='Petrol')&
    (df['mileage'] >= 20) &
    (df['seats'] == 5) &
    (df['vehicle_age'] <=3) &
    (df['model'] == 'Swift')
]
```

Out[75]:

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	transm
115	Maruti Swift	Maruti	Swift	3	35000	Dealer	Petrol	
138	Maruti Swift	Maruti	Swift	3	10000	Individual	Petrol	
213	Maruti Swift	Maruti	Swift	3	5000	Individual	Petrol	
382	Maruti Swift	Maruti	Swift	2	8000	Dealer	Petrol	
520	Maruti Swift	Maruti	Swift	1	10000	Individual	Petrol	
...	...	...	...	...	...	...	...	
14825	Maruti Swift	Maruti	Swift	3	15000	Individual	Petrol	
15219	Maruti Swift	Maruti	Swift	3	17000	Dealer	Petrol	
15241	Maruti Swift	Maruti	Swift	2	19587	Dealer	Petrol	
15273	Maruti Swift	Maruti	Swift	3	19000	Dealer	Petrol	
15401	Maruti Swift	Maruti	Swift	3	25000	Individual	Petrol	

106 rows × 13 columns

In [76]:

```
df[(df['brand']=='Maruti') &
    (df['fuel_type']=='Petrol')&
    (df['mileage'] >= 20) &
    (df['seats'] == 5) &
    (df['vehicle_age'] <=3) &
    (df['model'] == 'Swift') &
    (df['km_driven'] <= 15000)
]
```

Out[76]:

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	transm
138	Maruti Swift	Maruti	Swift	3	10000	Individual	Petrol	
213	Maruti Swift	Maruti	Swift	3	5000	Individual	Petrol	
382	Maruti Swift	Maruti	Swift	2	8000	Dealer	Petrol	
520	Maruti Swift	Maruti	Swift	1	10000	Individual	Petrol	
641	Maruti Swift	Maruti	Swift	2	14499	Trustmark Dealer	Petrol	
1003	Maruti Swift	Maruti	Swift	2	8043	Trustmark Dealer	Petrol	
1459	Maruti Swift	Maruti	Swift	3	14000	Dealer	Petrol	
1516	Maruti Swift	Maruti	Swift	2	15000	Dealer	Petrol	
1581	Maruti Swift	Maruti	Swift	3	13000	Dealer	Petrol	
2461	Maruti Swift	Maruti	Swift	3	8260	Dealer	Petrol	
3271	Maruti Swift	Maruti	Swift	2	10000	Individual	Petrol	
3327	Maruti Swift	Maruti	Swift	2	5000	Individual	Petrol	
3432	Maruti Swift	Maruti	Swift	2	14500	Dealer	Petrol	
3909	Maruti Swift	Maruti	Swift	2	8000	Individual	Petrol	
4328	Maruti Swift	Maruti	Swift	3	7000	Dealer	Petrol	
4695	Maruti Swift	Maruti	Swift	3	13300	Individual	Petrol	
4729	Maruti Swift	Maruti	Swift	2	14000	Individual	Petrol	
6343	Maruti Swift	Maruti	Swift	3	13000	Individual	Petrol	
7360	Maruti Swift	Maruti	Swift	2	15000	Individual	Petrol	
7479	Maruti Swift	Maruti	Swift	2	12500	Individual	Petrol	
7519	Maruti Swift	Maruti	Swift	1	5000	Individual	Petrol	

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	transm
7947	Maruti Swift	Maruti	Swift	3	12550	Individual	Petrol	
8023	Maruti Swift	Maruti	Swift	3	10000	Individual	Petrol	
8220	Maruti Swift	Maruti	Swift	2	10000	Individual	Petrol	
8376	Maruti Swift	Maruti	Swift	3	13600	Individual	Petrol	
8829	Maruti Swift	Maruti	Swift	1	1332	Dealer	Petrol	
9124	Maruti Swift	Maruti	Swift	1	10000	Dealer	Petrol	
9181	Maruti Swift	Maruti	Swift	2	6500	Individual	Petrol	
9260	Maruti Swift	Maruti	Swift	2	7200	Individual	Petrol	
10550	Maruti Swift	Maruti	Swift	2	14316	Trustmark Dealer	Petrol	
11514	Maruti Swift	Maruti	Swift	3	4000	Individual	Petrol	
11529	Maruti Swift	Maruti	Swift	1	9000	Individual	Petrol	
12765	Maruti Swift	Maruti	Swift	1	5000	Individual	Petrol	
13036	Maruti Swift	Maruti	Swift	2	7000	Individual	Petrol	
13415	Maruti Swift	Maruti	Swift	2	11320	Individual	Petrol	
13581	Maruti Swift	Maruti	Swift	3	9500	Dealer	Petrol	
13786	Maruti Swift	Maruti	Swift	2	11893	Trustmark Dealer	Petrol	
14129	Maruti Swift	Maruti	Swift	2	5000	Individual	Petrol	
14466	Maruti Swift	Maruti	Swift	2	5000	Individual	Petrol	
14825	Maruti Swift	Maruti	Swift	3	15000	Individual	Petrol	

In [77]:
```
df[(df['brand']=='Maruti') &
    (df['fuel_type']=='Petrol')&
    (df['mileage'] >= 20) &
```

```
    (df['seats'] == 5) &
    (df['vehicle_age'] <=3) &
    (df['model'] == 'Swift') &
    (df['km_driven'] <= 15000) &
    (df['transmission_type'] == 'Automatic')
]
```

Out[77]:

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	transm
138	Maruti Swift	Maruti	Swift	3	10000	Individual	Petrol	
382	Maruti Swift	Maruti	Swift	2	8000	Dealer	Petrol	
1003	Maruti Swift	Maruti	Swift	2	8043	Trustmark Dealer	Petrol	
1516	Maruti Swift	Maruti	Swift	2	15000	Dealer	Petrol	
1581	Maruti Swift	Maruti	Swift	3	13000	Dealer	Petrol	
3432	Maruti Swift	Maruti	Swift	2	14500	Dealer	Petrol	
4729	Maruti Swift	Maruti	Swift	2	14000	Individual	Petrol	
9260	Maruti Swift	Maruti	Swift	2	7200	Individual	Petrol	
13786	Maruti Swift	Maruti	Swift	2	11893	Trustmark Dealer	Petrol	

In [78]:

```
df[(df['brand']=='Maruti') &
    (df['fuel_type']=='Petrol')&
    (df['mileage'] >= 20) &
    (df['seats'] == 5) &
    (df['vehicle_age'] <=3) &
    (df['model'] == 'Swift') &
    (df['km_driven'] <= 15000) &
    (df['transmission_type'] == 'Automatic') &
    (df['seller_type'] == 'Individual')
]
```

Out[78]:

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	transmi
138	Maruti Swift	Maruti	Swift	3	10000	Individual	Petrol	
4729	Maruti Swift	Maruti	Swift	2	14000	Individual	Petrol	
9260	Maruti Swift	Maruti	Swift	2	7200	Individual	Petrol	

5 Seater, Petrol, Automatic, Low Driven, Vehicle Age Low, Maruti, seller type: Idividual, 1500 cc>,7-8 Lakh||

In [79]:
```
result_df=df[(df['brand'] == 'Maruti') &
    (df['seller_type'] == 'Individual') &
    (df['fuel_type'] == 'Petrol') &
    (df['transmission_type'] == 'Automatic') &
    (df['seats'] == 5 ) &
    (df['engine'] >=1200) &
    (df['selling_price'] <= 12_00_000) &
    (df['selling_price'] >= 7_00_000)]
```

In [80]:
```
result_df.sort_values(by ='selling_price').head(1)
```

Out[80]:

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	transm
15188	Maruti Ciaz	Maruti	Ciaz	5	15000	Individual	Petrol	

In [81]:
```
result_df.sort_values(by ='selling_price').tail(1)
```

Out[81]:

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	transmiss
928	Maruti Ciaz	Maruti	Ciaz	3	35000	Individual	Petrol	A

In [82]:
```
result_df.sort_values(by ='selling_price',ascending = False).head(3)
```

Out[82]:

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	transm
928	Maruti Ciaz	Maruti	Ciaz	3	35000	Individual	Petrol	
7270	Maruti Ciaz	Maruti	Ciaz	2	30000	Individual	Petrol	
15038	Maruti Ciaz	Maruti	Ciaz	3	10900	Individual	Petrol	

In [83]:
```
result_df.sort_values(by=['km_driven','vehicle_age'])
```

Out[83]:

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	transm
15038	Maruti Ciaz	Maruti	Ciaz	3	10900	Individual	Petrol	
15188	Maruti Ciaz	Maruti	Ciaz	5	15000	Individual	Petrol	
7270	Maruti Ciaz	Maruti	Ciaz	2	30000	Individual	Petrol	
928	Maruti Ciaz	Maruti	Ciaz	3	35000	Individual	Petrol	
15266	Maruti Ciaz	Maruti	Ciaz	3	47000	Individual	Petrol	
12359	Maruti Ciaz	Maruti	Ciaz	3	50000	Individual	Petrol	
6319	Maruti Ciaz	Maruti	Ciaz	4	50000	Individual	Petrol	

In [84]:
```
renault_duster_df=df[(df['model'] == 'Duster')&
    (df['brand'] == 'Renault')]
```

In [85]:
```
renault_duster_df.sort_values(by='selling_price').head(1)
```

Out[85]:

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	transmis
968	Renault Duster	Renault	Duster	8	95000	Dealer	Diesel	

In [ ]:
```
```

Group by

In [86]:
```
df
```

Out[86]:

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	tr
0	Maruti Alto	Maruti	Alto	9	120000	Individual	Petrol	
1	Hyundai Grand	Hyundai	Grand	5	20000	Individual	Petrol	
2	Hyundai i20	Hyundai	i20	11	60000	Individual	Petrol	
3	Maruti Alto	Maruti	Alto	9	37000	Individual	Petrol	
4	Ford Ecosport	Ford	Ecosport	6	30000	Dealer	Diesel	
...	...	...	...	...	...	...	...	
15406	Hyundai i10	Hyundai	i10	9	10723	Dealer	Petrol	
15407	Maruti Ertiga	Maruti	Ertiga	2	18000	Dealer	Petrol	
15408	Skoda Rapid	Skoda	Rapid	6	67000	Dealer	Diesel	
15409	Mahindra XUV500	Mahindra	XUV500	5	3800000	Dealer	Diesel	
15410	Honda City	Honda	City	2	13000	Dealer	Petrol	

15411 rows × 13 columns

In [87]:
```
df['brand'].value_counts()
```

Out[87]:
```
brand
Maruti           4992
Hyundai          2982
Honda            1485
Mahindra         1011
Toyota            793
Ford              790
Volkswagen        620
Renault           536
BMW               439
Tata              430
Mercedes-Benz     337
Skoda             334
Audi              192
Datsun            170
Jaguar             59
Land Rover         51
Jeep               41
Kia                32
Porsche            21
Volvo              20
MG                 19
Mini               17
Nissan             11
Lexus              10
Isuzu               8
Bentley             3
Maserati            2
ISUZU               2
Ferrari             1
Mercedes-AMG        1
Rolls-Royce         1
Force               1
Name: count, dtype: int64
```

In [88]:
```
grouped_df=df.groupby('brand')
```

In [89]:
```
# min, max, mean, sum.
grouped_df.count()
```

Out[89]:

brand	car_name	model	vehicle_age	km_driven	seller_type	fuel_type	transmiss
Audi	192	192	192	192	192	192	
BMW	439	439	439	439	439	439	
Bentley	3	3	3	3	3	3	
Datsun	170	170	170	170	170	170	
Ferrari	1	1	1	1	1	1	
Force	1	1	1	1	1	1	
Ford	790	790	790	790	790	790	
Honda	1485	1485	1485	1485	1485	1485	
Hyundai	2982	2982	2982	2982	2982	2982	
ISUZU	2	2	2	2	2	2	
Isuzu	8	8	8	8	8	8	
Jaguar	59	59	59	59	59	59	
Jeep	41	41	41	41	41	41	
Kia	32	32	32	32	32	32	
Land Rover	51	51	51	51	51	51	
Lexus	10	10	10	10	10	10	
MG	19	19	19	19	19	19	
Mahindra	1011	1011	1011	1011	1011	1011	
Maruti	4992	4992	4992	4992	4992	4992	
Maserati	2	2	2	2	2	2	
Mercedes-AMG	1	1	1	1	1	1	
Mercedes-Benz	337	337	337	337	337	337	
Mini	17	17	17	17	17	17	
Nissan	11	11	11	11	11	11	
Porsche	21	21	21	21	21	21	
Renault	536	536	536	536	536	536	
Rolls-Royce	1	1	1	1	1	1	
Skoda	334	334	334	334	334	334	
Tata	430	430	430	430	430	430	
Toyota	793	793	793	793	793	793	
Volkswagen	620	620	620	620	620	620	

brand	car_name	model	vehicle_age	km_driven	seller_type	fuel_type	transmiss
Volvo	20	20	20	20	20	20	

In [90]:
```
df.isna().sum()
```

Out[90]:
```
car_name             0
brand                0
model                0
vehicle_age          0
km_driven            0
seller_type          0
fuel_type            0
transmission_type    0
mileage              0
engine               0
max_power            0
seats                0
selling_price        0
dtype: int64
```

loc

In [91]:
```
df
```

Out[91]:

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	tr
0	Maruti Alto	Maruti	Alto	9	120000	Individual	Petrol	
1	Hyundai Grand	Hyundai	Grand	5	20000	Individual	Petrol	
2	Hyundai i20	Hyundai	i20	11	60000	Individual	Petrol	
3	Maruti Alto	Maruti	Alto	9	37000	Individual	Petrol	
4	Ford Ecosport	Ford	Ecosport	6	30000	Dealer	Diesel	
...	...	...	...	...	...	...	...	
15406	Hyundai i10	Hyundai	i10	9	10723	Dealer	Petrol	
15407	Maruti Ertiga	Maruti	Ertiga	2	18000	Dealer	Petrol	
15408	Skoda Rapid	Skoda	Rapid	6	67000	Dealer	Diesel	
15409	Mahindra XUV500	Mahindra	XUV500	5	3800000	Dealer	Diesel	
15410	Honda City	Honda	City	2	13000	Dealer	Petrol	

15411 rows × 13 columns

In [92]:
```
df.loc[4,'vehicle_age']
```

Out[92]:
```
np.int64(6)
```

In [93]:
```
import numpy as np
```

In [94]:
```
np.nan
```

Out[94]:
```
nan
```

In [95]:
```
df.loc[4,'vehicle_age'] =np.nan
```

In [96]:
```
df
```

Out[96]:

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	tr
0	Maruti Alto	Maruti	Alto	9.0	120000	Individual	Petrol	
1	Hyundai Grand	Hyundai	Grand	5.0	20000	Individual	Petrol	
2	Hyundai i20	Hyundai	i20	11.0	60000	Individual	Petrol	
3	Maruti Alto	Maruti	Alto	9.0	37000	Individual	Petrol	
4	Ford Ecosport	Ford	Ecosport	NaN	30000	Dealer	Diesel	
...	...	...	...	...	...	...	...	
15406	Hyundai i10	Hyundai	i10	9.0	10723	Dealer	Petrol	
15407	Maruti Ertiga	Maruti	Ertiga	2.0	18000	Dealer	Petrol	
15408	Skoda Rapid	Skoda	Rapid	6.0	67000	Dealer	Diesel	
15409	Mahindra XUV500	Mahindra	XUV500	5.0	3800000	Dealer	Diesel	
15410	Honda City	Honda	City	2.0	13000	Dealer	Petrol	

15411 rows × 13 columns

In [97]:
```
grouped_df.count()
```

Out[97]:

brand	car_name	model	vehicle_age	km_driven	seller_type	fuel_type	transmiss
Audi	192	192	192	192	192	192	
BMW	439	439	439	439	439	439	
Bentley	3	3	3	3	3	3	
Datsun	170	170	170	170	170	170	
Ferrari	1	1	1	1	1	1	
Force	1	1	1	1	1	1	
Ford	790	790	790	790	790	790	
Honda	1485	1485	1485	1485	1485	1485	
Hyundai	2982	2982	2982	2982	2982	2982	
ISUZU	2	2	2	2	2	2	
Isuzu	8	8	8	8	8	8	
Jaguar	59	59	59	59	59	59	
Jeep	41	41	41	41	41	41	
Kia	32	32	32	32	32	32	
Land Rover	51	51	51	51	51	51	
Lexus	10	10	10	10	10	10	
MG	19	19	19	19	19	19	
Mahindra	1011	1011	1011	1011	1011	1011	
Maruti	4992	4992	4992	4992	4992	4992	
Maserati	2	2	2	2	2	2	
Mercedes-AMG	1	1	1	1	1	1	
Mercedes-Benz	337	337	337	337	337	337	
Mini	17	17	17	17	17	17	
Nissan	11	11	11	11	11	11	
Porsche	21	21	21	21	21	21	
Renault	536	536	536	536	536	536	
Rolls-Royce	1	1	1	1	1	1	
Skoda	334	334	334	334	334	334	
Tata	430	430	430	430	430	430	
Toyota	793	793	793	793	793	793	
Volkswagen	620	620	620	620	620	620	

brand	car_name	model	vehicle_age	km_driven	seller_type	fuel_type	transmiss
Volvo	20	20	20	20	20	20	

In [98]:
```
df.isna().sum()
```

Out[98]:
```
car_name              0
brand                 0
model                 0
vehicle_age           1
km_driven             0
seller_type           0
fuel_type             0
transmission_type     0
mileage               0
engine                0
max_power             0
seats                 0
selling_price         0
dtype: int64
```

In [99]:
```
df.sample()
```

Out[99]:

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	transn
3676	Hyundai i10	Hyundai	i10	10.0	76000	Individual	Petrol	

In [100…
```
df.groupby('brand')['vehicle_age'].min()
```

Out[100…

```
brand
Audi             1.0
BMW              0.0
Bentley          9.0
Datsun           1.0
Ferrari          2.0
Force            5.0
Ford             1.0
Honda            1.0
Hyundai          1.0
ISUZU            4.0
Isuzu            2.0
Jaguar           2.0
Jeep             1.0
Kia              1.0
Land Rover       2.0
Lexus            2.0
MG               1.0
Mahindra         0.0
Maruti           0.0
Maserati         5.0
Mercedes-AMG     4.0
Mercedes-Benz    1.0
Mini             3.0
Nissan           1.0
Porsche          4.0
Renault          1.0
Rolls-Royce      4.0
Skoda            2.0
Tata             1.0
Toyota           1.0
Volkswagen       1.0
Volvo            1.0
Name: vehicle_age, dtype: float64
```

In [101…

```
df.groupby('brand')['vehicle_age'].min().sort_values()
```

Out[101…

```
brand
BMW              0.0
Mahindra         0.0
Maruti           0.0
Audi             1.0
Ford             1.0
Kia              1.0
Datsun           1.0
Honda            1.0
Renault          1.0
Volkswagen       1.0
Toyota           1.0
Nissan           1.0
Mercedes-Benz    1.0
MG               1.0
Jeep             1.0
Hyundai          1.0
Volvo            1.0
Tata             1.0
Skoda            2.0
Lexus            2.0
Isuzu            2.0
Ferrari          2.0
Jaguar           2.0
Land Rover       2.0
Mini             3.0
Rolls-Royce      4.0
ISUZU            4.0
Mercedes-AMG     4.0
Porsche          4.0
Force            5.0
Maserati         5.0
Bentley          9.0
Name: vehicle_age, dtype: float64
```

In [102…

```
df.groupby('brand')['vehicle_age'].min().sort_values(ascending= False).head(5)
```

Out[102…

```
brand
Bentley     9.0
Force       5.0
Maserati    5.0
ISUZU       4.0
Porsche     4.0
Name: vehicle_age, dtype: float64
```

In [103…

```
df.groupby('brand')['vehicle_age'].agg(['max','min'])
```

Out[103…

brand	max	min
Audi	12.0	1.0
BMW	25.0	0.0
Bentley	10.0	9.0
Datsun	7.0	1.0
Ferrari	2.0	2.0
Force	5.0	5.0
Ford	14.0	1.0
Honda	22.0	1.0
Hyundai	15.0	1.0
ISUZU	4.0	4.0
Isuzu	5.0	2.0
Jaguar	10.0	2.0
Jeep	4.0	1.0
Kia	2.0	1.0
Land Rover	11.0	2.0
Lexus	5.0	2.0
MG	2.0	1.0
Mahindra	18.0	0.0
Maruti	29.0	0.0
Maserati	5.0	5.0
Mercedes-AMG	4.0	4.0
Mercedes-Benz	17.0	1.0
Mini	9.0	3.0
Nissan	11.0	1.0
Porsche	12.0	4.0
Renault	9.0	1.0
Rolls-Royce	4.0	4.0
Skoda	18.0	2.0
Tata	15.0	1.0
Toyota	15.0	1.0
Volkswagen	11.0	1.0
Volvo	11.0	1.0

In [104…

```
df.groupby('brand')['vehicle_age'].agg(['max','min','mean'])
```

Out[104...

brand	max	min	mean
Audi	12.0	1.0	7.369792
BMW	25.0	0.0	6.619590
Bentley	10.0	9.0	9.333333
Datsun	7.0	1.0	4.117647
Ferrari	2.0	2.0	2.000000
Force	5.0	5.0	5.000000
Ford	14.0	1.0	5.927757
Honda	22.0	1.0	6.263973
Hyundai	15.0	1.0	6.119048
ISUZU	4.0	4.0	4.000000
Isuzu	5.0	2.0	3.750000
Jaguar	10.0	2.0	6.372881
Jeep	4.0	1.0	3.195122
Kia	2.0	1.0	1.656250
Land Rover	11.0	2.0	6.078431
Lexus	5.0	2.0	3.600000
MG	2.0	1.0	1.684211
Mahindra	18.0	0.0	6.018793
Maruti	29.0	0.0	5.939103
Maserati	5.0	5.0	5.000000
Mercedes-AMG	4.0	4.0	4.000000
Mercedes-Benz	17.0	1.0	7.623145
Mini	9.0	3.0	6.352941
Nissan	11.0	1.0	4.000000
Porsche	12.0	4.0	6.904762
Renault	9.0	1.0	4.837687
Rolls-Royce	4.0	4.0	4.000000
Skoda	18.0	2.0	6.835329
Tata	15.0	1.0	4.283721
Toyota	15.0	1.0	6.532156
Volkswagen	11.0	1.0	6.675806
Volvo	11.0	1.0	5.400000

In [105…

```
df.groupby('brand')['vehicle_age'].agg(['max','min','mean']).round(2)
```

Out[105…

brand	max	min	mean
Audi	12.0	1.0	7.37
BMW	25.0	0.0	6.62
Bentley	10.0	9.0	9.33
Datsun	7.0	1.0	4.12
Ferrari	2.0	2.0	2.00
Force	5.0	5.0	5.00
Ford	14.0	1.0	5.93
Honda	22.0	1.0	6.26
Hyundai	15.0	1.0	6.12
ISUZU	4.0	4.0	4.00
Isuzu	5.0	2.0	3.75
Jaguar	10.0	2.0	6.37
Jeep	4.0	1.0	3.20
Kia	2.0	1.0	1.66
Land Rover	11.0	2.0	6.08
Lexus	5.0	2.0	3.60
MG	2.0	1.0	1.68
Mahindra	18.0	0.0	6.02
Maruti	29.0	0.0	5.94
Maserati	5.0	5.0	5.00
Mercedes-AMG	4.0	4.0	4.00
Mercedes-Benz	17.0	1.0	7.62
Mini	9.0	3.0	6.35
Nissan	11.0	1.0	4.00
Porsche	12.0	4.0	6.90
Renault	9.0	1.0	4.84
Rolls-Royce	4.0	4.0	4.00
Skoda	18.0	2.0	6.84
Tata	15.0	1.0	4.28
Toyota	15.0	1.0	6.53
Volkswagen	11.0	1.0	6.68
Volvo	11.0	1.0	5.40

In [106…
```
grouped_df= df.groupby('brand')['vehicle_age'].agg(['max','min','mean']).round(2
```

In [107…
```
grouped_df.reset_index()
```

Out[107…

	brand	max	min	mean
0	Audi	12.0	1.0	7.37
1	BMW	25.0	0.0	6.62
2	Bentley	10.0	9.0	9.33
3	Datsun	7.0	1.0	4.12
4	Ferrari	2.0	2.0	2.00
5	Force	5.0	5.0	5.00
6	Ford	14.0	1.0	5.93
7	Honda	22.0	1.0	6.26
8	Hyundai	15.0	1.0	6.12
9	ISUZU	4.0	4.0	4.00
10	Isuzu	5.0	2.0	3.75
11	Jaguar	10.0	2.0	6.37
12	Jeep	4.0	1.0	3.20
13	Kia	2.0	1.0	1.66
14	Land Rover	11.0	2.0	6.08
15	Lexus	5.0	2.0	3.60
16	MG	2.0	1.0	1.68
17	Mahindra	18.0	0.0	6.02
18	Maruti	29.0	0.0	5.94
19	Maserati	5.0	5.0	5.00
20	Mercedes-AMG	4.0	4.0	4.00
21	Mercedes-Benz	17.0	1.0	7.62
22	Mini	9.0	3.0	6.35
23	Nissan	11.0	1.0	4.00
24	Porsche	12.0	4.0	6.90
25	Renault	9.0	1.0	4.84
26	Rolls-Royce	4.0	4.0	4.00
27	Skoda	18.0	2.0	6.84
28	Tata	15.0	1.0	4.28
29	Toyota	15.0	1.0	6.53
30	Volkswagen	11.0	1.0	6.68
31	Volvo	11.0	1.0	5.40

In [108…

```
grouped_df.reset_index().sort_values(['brand','min'])
```

Out[108...

	brand	max	min	mean
0	Audi	12.0	1.0	7.37
1	BMW	25.0	0.0	6.62
2	Bentley	10.0	9.0	9.33
3	Datsun	7.0	1.0	4.12
4	Ferrari	2.0	2.0	2.00
5	Force	5.0	5.0	5.00
6	Ford	14.0	1.0	5.93
7	Honda	22.0	1.0	6.26
8	Hyundai	15.0	1.0	6.12
9	ISUZU	4.0	4.0	4.00
10	Isuzu	5.0	2.0	3.75
11	Jaguar	10.0	2.0	6.37
12	Jeep	4.0	1.0	3.20
13	Kia	2.0	1.0	1.66
14	Land Rover	11.0	2.0	6.08
15	Lexus	5.0	2.0	3.60
16	MG	2.0	1.0	1.68
17	Mahindra	18.0	0.0	6.02
18	Maruti	29.0	0.0	5.94
19	Maserati	5.0	5.0	5.00
20	Mercedes-AMG	4.0	4.0	4.00
21	Mercedes-Benz	17.0	1.0	7.62
22	Mini	9.0	3.0	6.35
23	Nissan	11.0	1.0	4.00
24	Porsche	12.0	4.0	6.90
25	Renault	9.0	1.0	4.84
26	Rolls-Royce	4.0	4.0	4.00
27	Skoda	18.0	2.0	6.84
28	Tata	15.0	1.0	4.28
29	Toyota	15.0	1.0	6.53
30	Volkswagen	11.0	1.0	6.68
31	Volvo	11.0	1.0	5.40

In [109…
```
df
```

Out[109…

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	tr
0	Maruti Alto	Maruti	Alto	9.0	120000	Individual	Petrol	
1	Hyundai Grand	Hyundai	Grand	5.0	20000	Individual	Petrol	
2	Hyundai i20	Hyundai	i20	11.0	60000	Individual	Petrol	
3	Maruti Alto	Maruti	Alto	9.0	37000	Individual	Petrol	
4	Ford Ecosport	Ford	Ecosport	NaN	30000	Dealer	Diesel	
…	…	…	…	…	…	…	…	
15406	Hyundai i10	Hyundai	i10	9.0	10723	Dealer	Petrol	
15407	Maruti Ertiga	Maruti	Ertiga	2.0	18000	Dealer	Petrol	
15408	Skoda Rapid	Skoda	Rapid	6.0	67000	Dealer	Diesel	
15409	Mahindra XUV500	Mahindra	XUV500	5.0	3800000	Dealer	Diesel	
15410	Honda City	Honda	City	2.0	13000	Dealer	Petrol	

15411 rows × 13 columns

In [110…
```
grouped_df=df.groupby('model')['selling_price'].aggregate(['min','max','mean','c
```

In [111…
```
grouped_df[grouped_df['model'] =='Alto']
```

Out[111…

	model	min	max	mean	count
7	Alto	45000	485000	245452.44	778

In [112…
```
grouped_df[grouped_df['model'] == '3']
```

Out[112…

	model	min	max	mean	count
0	3	550000	4500000	1786302.63	152

In [113…
```
df['model'].nunique()
```

Out[113…
```
120
```

In [114…
```
grouped_df=df.groupby('car_name')['selling_price'].aggregate(['min','max','mean'
```

In [115…
```
grouped_df
```

Out[115…

	car_name	min	max	mean	count
120	Volvo XC90	4100000	6975000	5231000.00	4
119	Volvo XC60	1400000	1825000	1645000.00	5
118	Volvo XC	1200000	8195000	4099285.71	7
117	Volvo S90	3650000	4750000	4187500.00	4
116	Volkswagen Vento	200000	1250000	521566.80	247
...	...	...	...	...	...
4	BMW 3	550000	4500000	1786302.63	152
3	Audi Q7	1000000	6800000	3037391.30	23
2	Audi A8	2200000	5500000	3291666.67	6
1	Audi A6	857000	4600000	2076875.00	64
0	Audi A4	750000	4200000	1566747.47	99

121 rows × 5 columns

In [116…
```
grouped_df.head(50)
```

Out[116...

	car_name	min	max	mean	count
120	Volvo XC90	4100000	6975000	5231000.00	4
119	Volvo XC60	1400000	1825000	1645000.00	5
118	Volvo XC	1200000	8195000	4099285.71	7
117	Volvo S90	3650000	4750000	4187500.00	4
116	Volkswagen Vento	200000	1250000	521566.80	247
115	Volkswagen Polo	173000	975000	513222.52	373
114	Toyota Yaris	890000	1300000	1031588.24	17
113	Toyota Innova	265000	2350000	1176111.93	545
112	Toyota Glanza	747000	915000	829250.00	8
111	Toyota Fortuner	723000	3650000	1947529.41	187
110	Toyota Camry	345000	2400000	1614277.78	36
109	Tata Tigor	395000	740000	553054.05	37
108	Tata Tiago	285000	665000	452048.28	145
107	Tata Safari	70000	1270000	520840.00	100
106	Tata Nexon	550000	1030000	805364.71	85
105	Tata Hexa	800000	1600000	1284170.73	41
104	Tata Harrier	1228000	1750000	1618523.81	21
103	Tata Altroz	730000	730000	730000.00	1
102	Skoda Superb	235000	3550000	917021.51	93
101	Skoda Rapid	225000	1195000	565895.60	182
100	Skoda Octavia	200000	2375000	1247627.12	59
99	Rolls-Royce Ghost	24200000	24200000	24200000.00	1
98	Renault Triber	550000	800000	654500.00	12
97	Renault KWID	200000	550000	342558.82	306
96	Renault Duster	295000	1155000	567389.91	218
95	Porsche Panamera	3800000	6500000	5058333.33	3
94	Porsche Macan	5975000	5995000	5985000.00	2
93	Porsche Cayenne	2000000	11100000	5077500.00	16
92	Nissan X-Trail	440000	1100000	711666.67	3
91	Nissan Kicks	850000	1450000	1046750.00	8
90	Mini Cooper	1290000	3875000	2182647.06	17
89	Mercedes-Benz S-Class	625000	13000000	3559783.78	37
88	Mercedes-Benz GLS	5375000	8000000	6781083.33	12

	car_name	min	max	mean	count
87	Mercedes-Benz GL-Class	1690000	7595000	3929444.44	36
86	Mercedes-Benz E-Class	315000	5743000	2021856.00	125
85	Mercedes-Benz CLS	700000	7500000	3433333.33	9
84	Mercedes-Benz C-Class	425000	4200000	1676550.85	118
83	Mercedes-AMG C	5100000	5100000	5100000.00	1
82	Maserati Quattroporte	6000000	6000000	6000000.00	1
81	Maserati Ghibli	6200000	6200000	6200000.00	1
80	Maruti XL6	1000000	1200000	1113571.43	7
79	Maruti Wagon R	40000	625000	307390.34	717
78	Maruti Vitara	525000	1225000	830596.61	295
77	Maruti Swift Dzire	165000	925000	525888.76	890
76	Maruti Swift	120000	875000	471736.24	781
75	Maruti S-Presso	400000	550000	463230.77	13
74	Maruti Ignis	415000	700000	532602.74	73
73	Maruti Ertiga	350000	1100000	719860.06	343
72	Maruti Eeco	130000	490000	334872.00	125
71	Maruti Dzire ZXI	440000	760000	550000.00	4

In [117…
```
df['model'].nunique()
```

Out[117…
```
120
```

Double Col Grouping and Single Agg

In [118…
```
df.sample()
```

Out[118…

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	transm
2377	Renault KWID	Renault	KWID	5.0	25000	Individual	Petrol	

In [119…
```
df.groupby(['seller_type','fuel_type'])['car_name'].count().reset_index()
```

Out[119…

	seller_type	fuel_type	car_name
0	Dealer	CNG	216
1	Dealer	Diesel	4687
2	Dealer	Electric	3
3	Dealer	LPG	16
4	Dealer	Petrol	4617
5	Individual	CNG	69
6	Individual	Diesel	2720
7	Individual	Electric	1
8	Individual	LPG	28
9	Individual	Petrol	2881
10	Trustmark Dealer	CNG	16
11	Trustmark Dealer	Diesel	12
12	Trustmark Dealer	Petrol	145

In [120…

```
df.groupby(['seller_type','fuel_type'])['car_name'].count().reset_index().sort_v
```

Out[120…

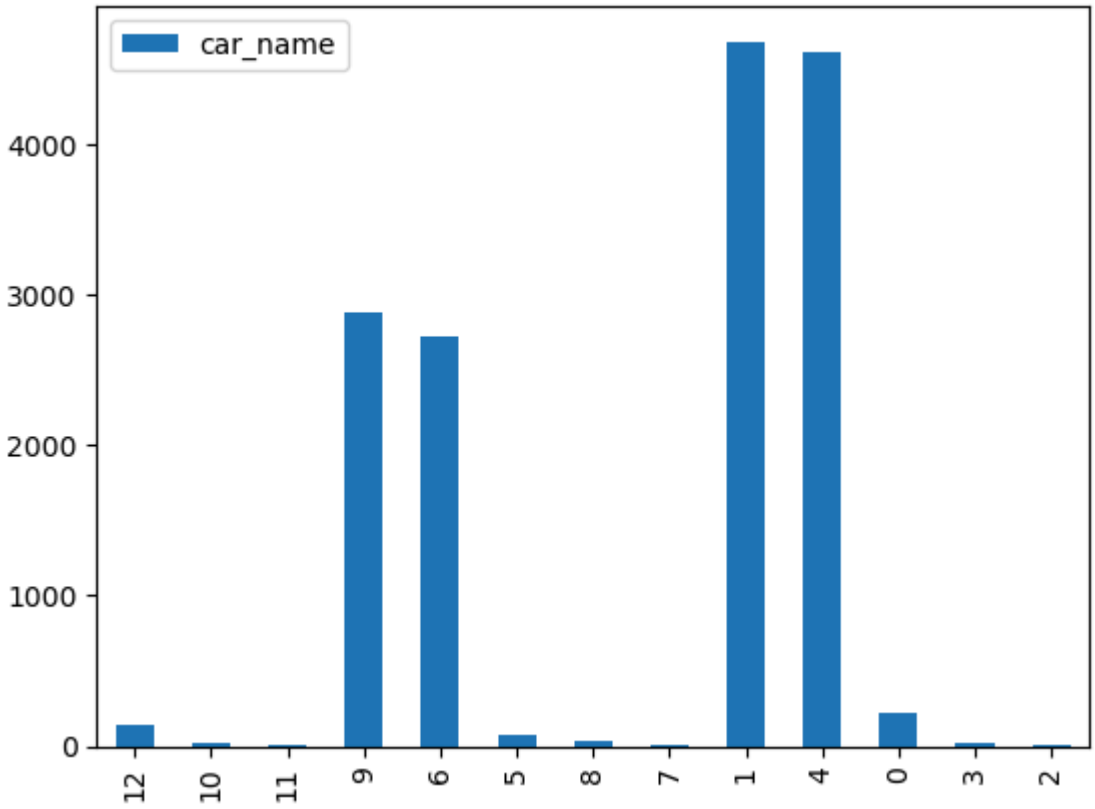
	seller_type	fuel_type	car_name
12	Trustmark Dealer	Petrol	145
10	Trustmark Dealer	CNG	16
11	Trustmark Dealer	Diesel	12
9	Individual	Petrol	2881
6	Individual	Diesel	2720
5	Individual	CNG	69
8	Individual	LPG	28
7	Individual	Electric	1
1	Dealer	Diesel	4687
4	Dealer	Petrol	4617
0	Dealer	CNG	216
3	Dealer	LPG	16
2	Dealer	Electric	3

In [121…

```
df.groupby(['seller_type','fuel_type'])['car_name'].count().reset_index().sort_v
```

Out[121…

```
<Axes: >
```



In [122…

```
df
```

Out[122...

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	tr
0	Maruti Alto	Maruti	Alto	9.0	120000	Individual	Petrol	
1	Hyundai Grand	Hyundai	Grand	5.0	20000	Individual	Petrol	
2	Hyundai i20	Hyundai	i20	11.0	60000	Individual	Petrol	
3	Maruti Alto	Maruti	Alto	9.0	37000	Individual	Petrol	
4	Ford Ecosport	Ford	Ecosport	NaN	30000	Dealer	Diesel	
...	...	...	...	...	...	...	...	
15406	Hyundai i10	Hyundai	i10	9.0	10723	Dealer	Petrol	
15407	Maruti Ertiga	Maruti	Ertiga	2.0	18000	Dealer	Petrol	
15408	Skoda Rapid	Skoda	Rapid	6.0	67000	Dealer	Diesel	
15409	Mahindra XUV500	Mahindra	XUV500	5.0	3800000	Dealer	Diesel	
15410	Honda City	Honda	City	2.0	13000	Dealer	Petrol	

15411 rows × 13 columns

In [123...
```
car_name_grouped_df=df.groupby('car_name')
```

In [124...
```
car_name_grouped_df.get_group('Hyundai i10')
```

Out[124...

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	trans
6	Hyundai i10	Hyundai	i10	8.0	40000	Dealer	Petrol	
108	Hyundai i10	Hyundai	i10	7.0	32111	Trustmark Dealer	Petrol	
241	Hyundai i10	Hyundai	i10	13.0	57215	Individual	Petrol	
261	Hyundai i10	Hyundai	i10	9.0	110699	Dealer	Petrol	
307	Hyundai i10	Hyundai	i10	10.0	58000	Dealer	Petrol	
...	...	...	...	...	...	...	...	
15335	Hyundai i10	Hyundai	i10	12.0	100000	Individual	Petrol	
15344	Hyundai i10	Hyundai	i10	12.0	65000	Dealer	Petrol	
15359	Hyundai i10	Hyundai	i10	12.0	8500	Dealer	Petrol	
15395	Hyundai i10	Hyundai	i10	9.0	64000	Dealer	Petrol	
15406	Hyundai i10	Hyundai	i10	9.0	10723	Dealer	Petrol	

410 rows × 13 columns

In [125...

```
car_name_grouped_df.head(1)
```

Out[125...

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_ty
0	Maruti Alto	Maruti	Alto	9.0	120000	Individual	Petı
1	Hyundai Grand	Hyundai	Grand	5.0	20000	Individual	Petı
2	Hyundai i20	Hyundai	i20	11.0	60000	Individual	Petı
4	Ford Ecosport	Ford	Ecosport	NaN	30000	Dealer	Dies
5	Maruti Wagon R	Maruti	Wagon R	8.0	35000	Individual	Petı
...	...	...	...	...	...	...	
9714	Lexus RX	Lexus	RX	4.0	52000	Dealer	Petı
10969	Rolls-Royce Ghost	Rolls-Royce	Ghost	4.0	5000	Individual	Petı
11420	Maserati Quattroporte	Maserati	Quattroporte	5.0	9500	Dealer	Dies
12564	Isuzu MUX	Isuzu	MUX	2.0	34000	Dealer	Dies
13450	Force Gurkha	Force	Gurkha	5.0	60000	Individual	Dies

121 rows × 13 columns

In [126...

```
car_name_grouped_df['selling_price'].max().reset_index()
```

Out[126...

	car_name	selling_price
0	Audi A4	4200000
1	Audi A6	4600000
2	Audi A8	5500000
3	Audi Q7	6800000
4	BMW 3	4500000
...	...	...
116	Volkswagen Vento	1250000
117	Volvo S90	4750000
118	Volvo XC	8195000
119	Volvo XC60	1825000
120	Volvo XC90	6975000

121 rows × 2 columns

In [127...

```
df['car_name'].nunique()
```

Out[127… 121

In [128…
```
df['model'].nunique()
```

Out[128… 120

In [129…
```
df
```

Out[129…

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	tr
0	Maruti Alto	Maruti	Alto	9.0	120000	Individual	Petrol	
1	Hyundai Grand	Hyundai	Grand	5.0	20000	Individual	Petrol	
2	Hyundai i20	Hyundai	i20	11.0	60000	Individual	Petrol	
3	Maruti Alto	Maruti	Alto	9.0	37000	Individual	Petrol	
4	Ford Ecosport	Ford	Ecosport	NaN	30000	Dealer	Diesel	
...	...	...	...	...	...	...	...	
15406	Hyundai i10	Hyundai	i10	9.0	10723	Dealer	Petrol	
15407	Maruti Ertiga	Maruti	Ertiga	2.0	18000	Dealer	Petrol	
15408	Skoda Rapid	Skoda	Rapid	6.0	67000	Dealer	Diesel	
15409	Mahindra XUV500	Mahindra	XUV500	5.0	3800000	Dealer	Diesel	
15410	Honda City	Honda	City	2.0	13000	Dealer	Petrol	

15411 rows × 13 columns

In [130…
```
df.sample()
```

Out[130…

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	tra
11594	Mahindra XUV500	Mahindra	XUV500	3.0	17750	Dealer	Diesel	

Multivariate Analysis

In [131…
```
df.groupby(['seller_type','fuel_type'])[['mileage','selling_price']].agg(min).re
```

```
C:\Users\Administrator\AppData\Local\Temp\ipykernel_1616\2791010624.py:1: FutureWarning: The provided callable <built-in function min> is currently using DataFrameGroupBy.min. In a future version of pandas, the provided callable will be used directly. To keep current behavior pass the string "min" instead.
  df.groupby(['seller_type','fuel_type'])[['mileage','selling_price']].agg(min).reset_index()
```

Out[131…

	seller_type	fuel_type	mileage	selling_price
0	Dealer	CNG	11.88	150000
1	Dealer	Diesel	7.50	105000
2	Dealer	Electric	19.16	1699000
3	Dealer	LPG	13.45	120000
4	Dealer	Petrol	4.00	55000
5	Individual	CNG	11.88	125000
6	Individual	Diesel	9.00	100000
7	Individual	Electric	19.16	2000000
8	Individual	LPG	13.45	80000
9	Individual	Petrol	8.50	40000
10	Trustmark Dealer	CNG	17.50	210000
11	Trustmark Dealer	Diesel	15.40	390000
12	Trustmark Dealer	Petrol	13.00	230000

In [132…

```
df.groupby(['seller_type','fuel_type'])[['mileage','selling_price']].agg(['min',
```

Out[132...

	seller_type	fuel_type	mileage			selling_price		
			min	max	mean	min	max	mean
0	Dealer	CNG	11.88	33.54	26.01	150000	1080000	426208.33
1	Dealer	Diesel	7.50	28.40	19.86	105000	9200000	1128263.49
2	Dealer	Electric	19.16	19.16	19.16	1699000	1865000	1804666.67
3	Dealer	LPG	13.45	26.20	18.27	120000	403000	232812.50
4	Dealer	Petrol	4.00	28.09	18.71	55000	39500000	635360.41
5	Individual	CNG	11.88	33.54	24.26	125000	825000	381942.03
6	Individual	Diesel	9.00	28.40	20.39	100000	8500000	781109.19
7	Individual	Electric	19.16	19.16	19.16	2000000	2000000	2000000.00
8	Individual	LPG	13.45	26.20	17.59	80000	420000	191107.14
9	Individual	Petrol	8.50	25.17	19.75	40000	24200000	473092.29
10	Trustmark Dealer	CNG	17.50	33.54	29.88	210000	855000	456812.50
11	Trustmark Dealer	Diesel	15.40	28.40	22.11	390000	1575000	807833.33
12	Trustmark Dealer	Petrol	13.00	25.17	19.81	230000	1150000	565144.83

In [133...

```
df[(df['seller_type'] == 'Dealer') &
   (df['fuel_type'] == 'Electric')]
```

Out[133...

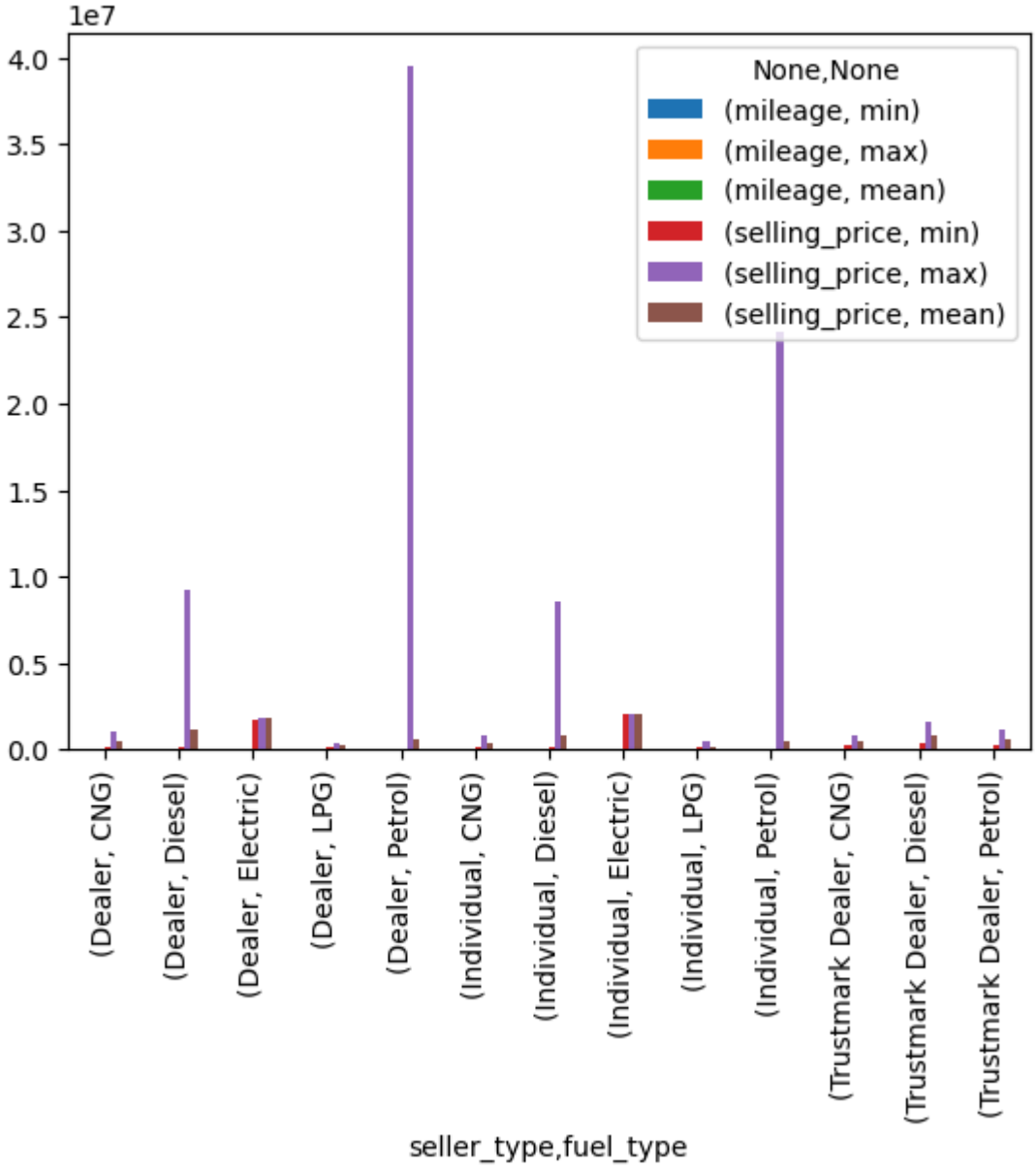
	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	transm
1997	Toyota Camry	Toyota	Camry	6.0	49500	Dealer	Electric	
14323	Toyota Camry	Toyota	Camry	5.0	85000	Dealer	Electric	
14600	Toyota Camry	Toyota	Camry	6.0	64000	Dealer	Electric	

In [134...

```
df.groupby(['seller_type','fuel_type'])[['mileage','selling_price']].agg(['min',
```

Out[134...

```
<Axes: xlabel='seller_type,fuel_type'>
```



LOC vs ILOC

In [135…

```
df
```

Out[135...

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	tr
0	Maruti Alto	Maruti	Alto	9.0	120000	Individual	Petrol	
1	Hyundai Grand	Hyundai	Grand	5.0	20000	Individual	Petrol	
2	Hyundai i20	Hyundai	i20	11.0	60000	Individual	Petrol	
3	Maruti Alto	Maruti	Alto	9.0	37000	Individual	Petrol	
4	Ford Ecosport	Ford	Ecosport	NaN	30000	Dealer	Diesel	
...	...	...	...	...	...	...	...	
15406	Hyundai i10	Hyundai	i10	9.0	10723	Dealer	Petrol	
15407	Maruti Ertiga	Maruti	Ertiga	2.0	18000	Dealer	Petrol	
15408	Skoda Rapid	Skoda	Rapid	6.0	67000	Dealer	Diesel	
15409	Mahindra XUV500	Mahindra	XUV500	5.0	3800000	Dealer	Diesel	
15410	Honda City	Honda	City	2.0	13000	Dealer	Petrol	

15411 rows × 13 columns

Loc : You have to give location value to Search Value in a Given DF

In [136...
```
df.loc[15408,'seller_type'] = 'Mr.Dealer'.upper()
```

In [137...
```
df.loc[15408,'seller_type']
```

Out[137...
```
'MR.DEALER'
```

In [138...
```
df
```

Out[138...

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	tr
0	Maruti Alto	Maruti	Alto	9.0	120000	Individual	Petrol	
1	Hyundai Grand	Hyundai	Grand	5.0	20000	Individual	Petrol	
2	Hyundai i20	Hyundai	i20	11.0	60000	Individual	Petrol	
3	Maruti Alto	Maruti	Alto	9.0	37000	Individual	Petrol	
4	Ford Ecosport	Ford	Ecosport	NaN	30000	Dealer	Diesel	
...	...	...	...	...	...	...	...	
15406	Hyundai i10	Hyundai	i10	9.0	10723	Dealer	Petrol	
15407	Maruti Ertiga	Maruti	Ertiga	2.0	18000	Dealer	Petrol	
15408	Skoda Rapid	Skoda	Rapid	6.0	67000	MR.DEALER	Diesel	
15409	Mahindra XUV500	Mahindra	XUV500	5.0	3800000	Dealer	Diesel	
15410	Honda City	Honda	City	2.0	13000	Dealer	Petrol	

15411 rows × 13 columns

ILOC : Index based Location Search'

In [139...

```
df.iloc[15408,4] = '67000'
```

```
C:\Users\Administrator\AppData\Local\Temp\ipykernel_1616\637406355.py:1: FutureWarning: Setting an item of incompatible dtype is deprecated and will raise an error in a future version of pandas. Value '67000' has dtype incompatible with int64, please explicitly cast to a compatible dtype first.
  df.iloc[15408,4] = '67000'
```

In [140...

```
df
```

Out[140…

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	tr
0	Maruti Alto	Maruti	Alto	9.0	120000	Individual	Petrol	
1	Hyundai Grand	Hyundai	Grand	5.0	20000	Individual	Petrol	
2	Hyundai i20	Hyundai	i20	11.0	60000	Individual	Petrol	
3	Maruti Alto	Maruti	Alto	9.0	37000	Individual	Petrol	
4	Ford Ecosport	Ford	Ecosport	NaN	30000	Dealer	Diesel	
…	…	…	…	…	…	…	…	
15406	Hyundai i10	Hyundai	i10	9.0	10723	Dealer	Petrol	
15407	Maruti Ertiga	Maruti	Ertiga	2.0	18000	Dealer	Petrol	
15408	Skoda Rapid	Skoda	Rapid	6.0	67000	MR.DEALER	Diesel	
15409	Mahindra XUV500	Mahindra	XUV500	5.0	3800000	Dealer	Diesel	
15410	Honda City	Honda	City	2.0	13000	Dealer	Petrol	

15411 rows × 13 columns

In [141…
```
df.iloc[15408,[4,5]] = [67000,'Dealer']
```

In [142…
```
df
```

Out[142...

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	tr
0	Maruti Alto	Maruti	Alto	9.0	120000	Individual	Petrol	
1	Hyundai Grand	Hyundai	Grand	5.0	20000	Individual	Petrol	
2	Hyundai i20	Hyundai	i20	11.0	60000	Individual	Petrol	
3	Maruti Alto	Maruti	Alto	9.0	37000	Individual	Petrol	
4	Ford Ecosport	Ford	Ecosport	NaN	30000	Dealer	Diesel	
...	...	...	...	...	...	...	...	
15406	Hyundai i10	Hyundai	i10	9.0	10723	Dealer	Petrol	
15407	Maruti Ertiga	Maruti	Ertiga	2.0	18000	Dealer	Petrol	
15408	Skoda Rapid	Skoda	Rapid	6.0	67000	Dealer	Diesel	
15409	Mahindra XUV500	Mahindra	XUV500	5.0	3800000	Dealer	Diesel	
15410	Honda City	Honda	City	2.0	13000	Dealer	Petrol	

15411 rows × 13 columns

In [144...

```
  Cell In[144], line 1
    pdf notebook.ipynb
        ^
SyntaxError: invalid syntax
```

In [ ]: