

Manufacturing Year

1995

1998

1999

2000

2001

2002

2003

2004

2005

2006

2007

2008

2009

2010

2011

2012

2013

2014

2015

2016

2017

>

company_name



ownership



car_name



All



All



All



1,532.40

Avg Engine Capacity

63.22K

Avg Kms Driven

8.55

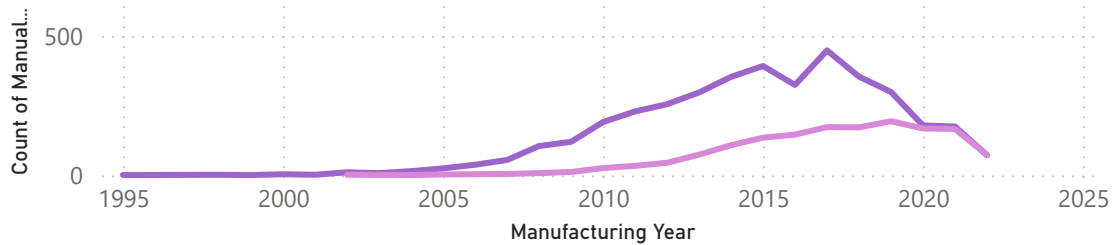
Avg Car Age

5

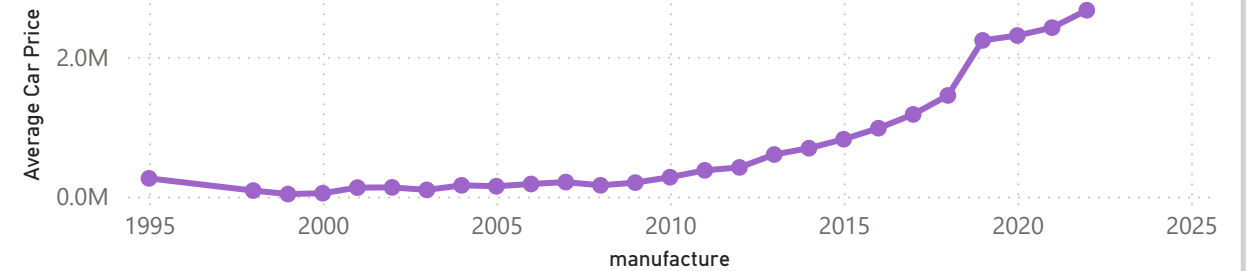
Avg Seats in Car

Count of Manual Cars and Count of Automatic Cars by Manufacturing Year

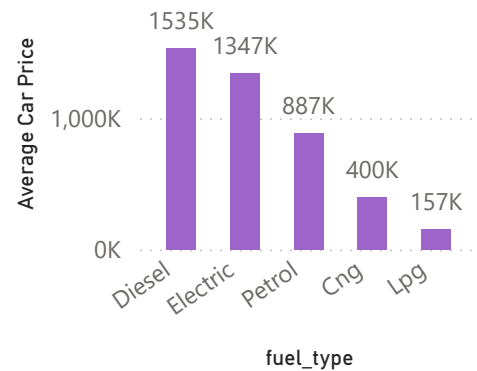
● Count of Manual Cars ● Count of Automatic Cars



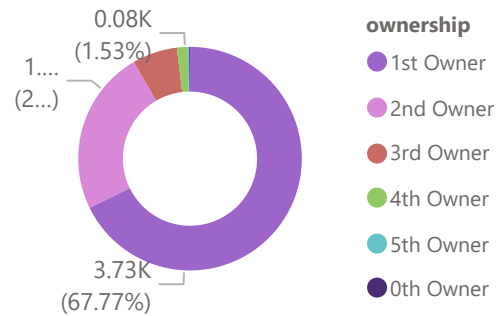
Average Car Price by Manufacturing Year



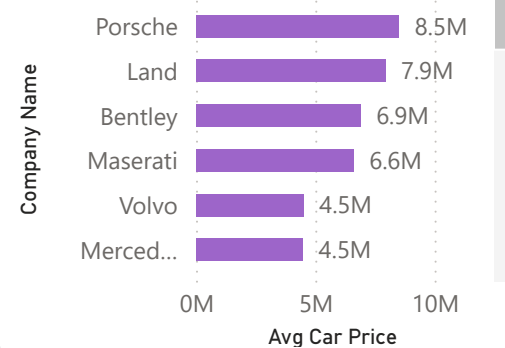
Average Car Price by fuel_type



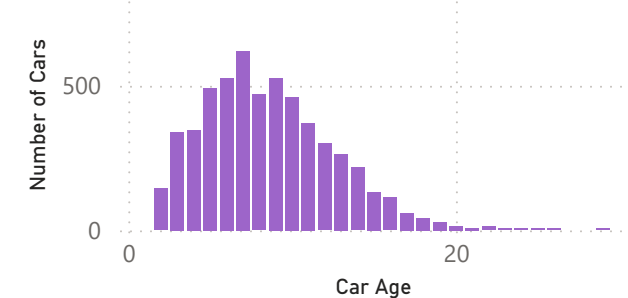
Count of car_name by ownership



Avg Car Price by Company Name



Number of Cars by Car Age



Manufacturing Year

2002	2004	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022
------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------

company_name

Tata

ownership

All

car_name

All

1,551.13

Avg Engine Capacity

65.44K

Avg Kms Driven

7.69

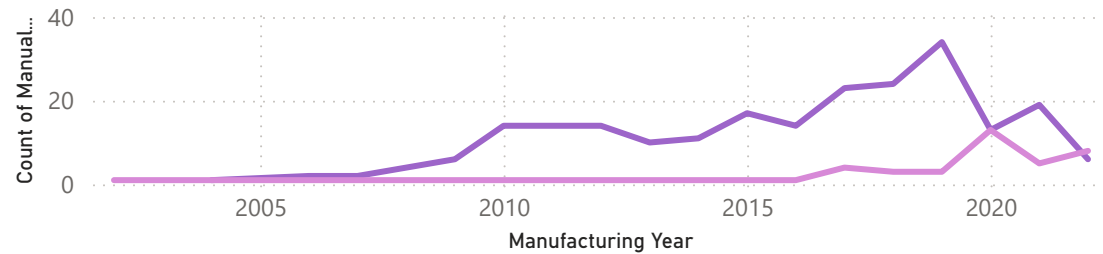
Avg Car Age

5

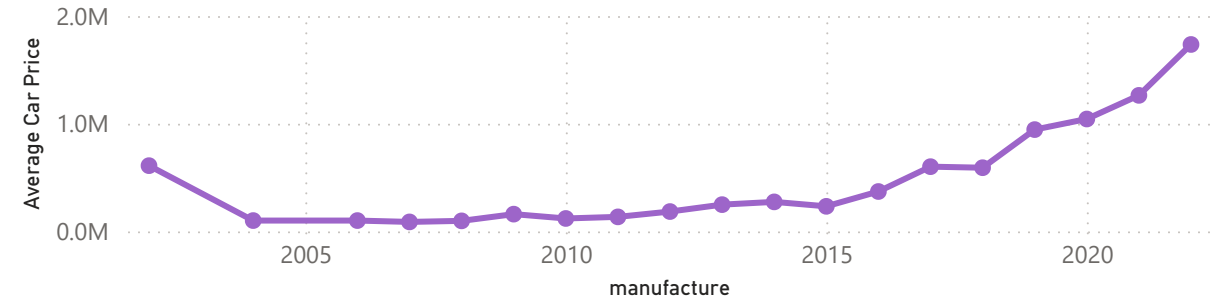
Avg Seats in Car

Count of Manual Cars and Count of Automatic Cars by Manufacturing Year

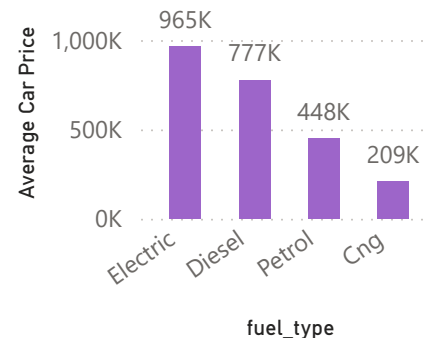
Count of Manual Cars Count of Automatic Cars



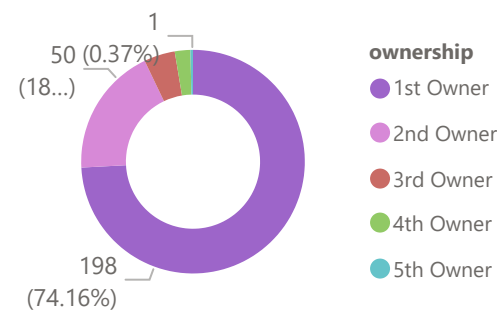
Average Car Price by Manufacturing Year



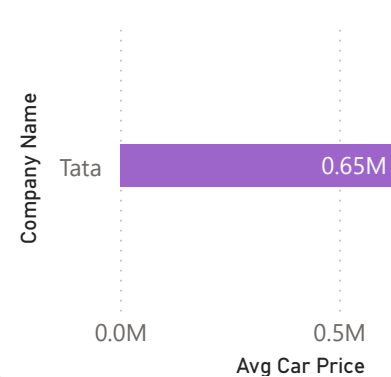
Average Car Price by fuel_type



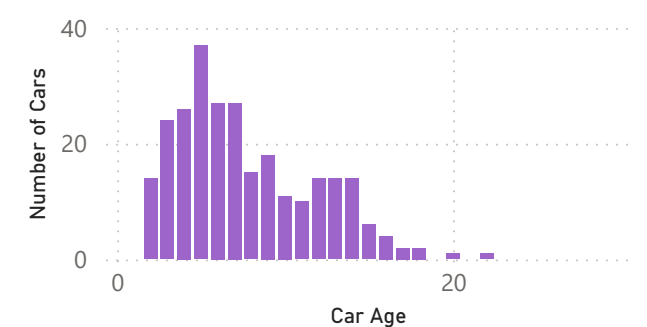
Count of car_name by ownership



Avg Car Price by Company Name



Number of Cars by Car Age



Manufacturing Year

2017

company_name

Audi

ownership

All

car_name

A3 35 TDI Premi...

1,086.00

Avg Engine Capacity

11.00K

Avg Kms Driven

7.00

Avg Car Age

5

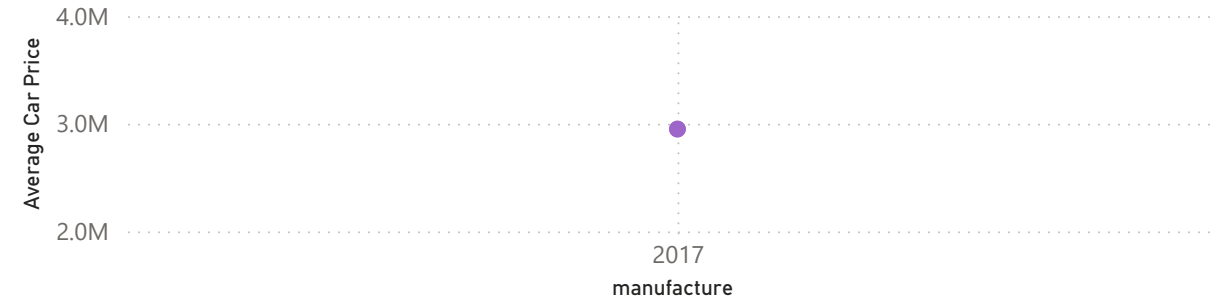
Avg Seats in Car

Count of Manual Cars and Count of Automatic Cars by Manufacturing Year

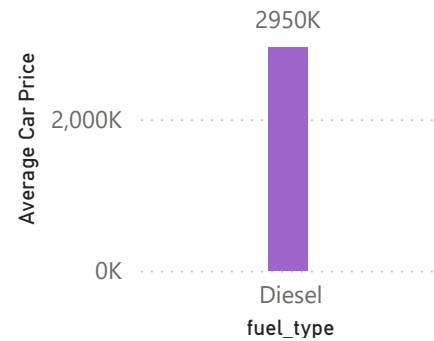
● Count of Manual Cars ● Count of Automatic Cars



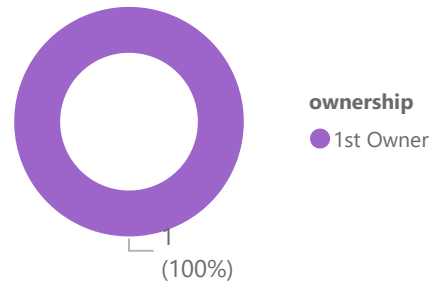
Average Car Price by Manufacturing Year



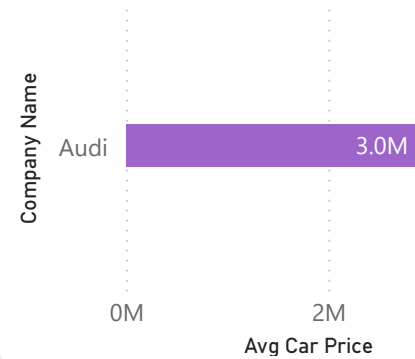
Average Car Price by fuel_type



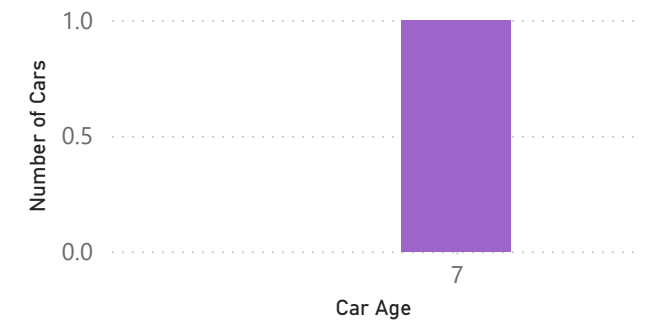
Count of car_name by ownership



Avg Car Price by Company Name



Number of Cars by Car Age



Manufacturing Year

2022

company_name

Hyundai

ownership

All

car_name

Santro Sportz AMT

1,086.00

Avg Engine Capacity

5.27K

Avg Kms Driven

2.00

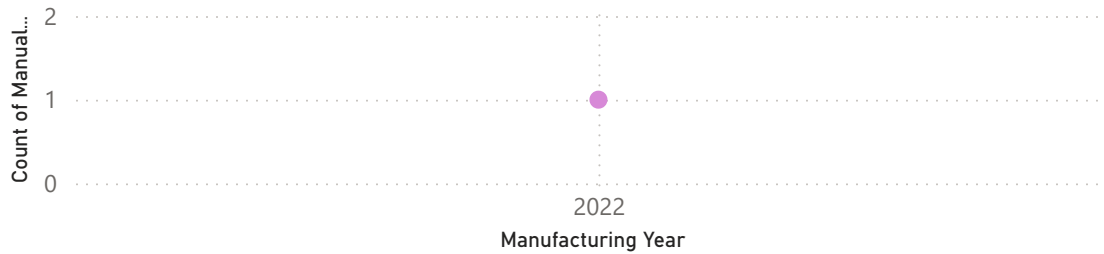
Avg Car Age

5

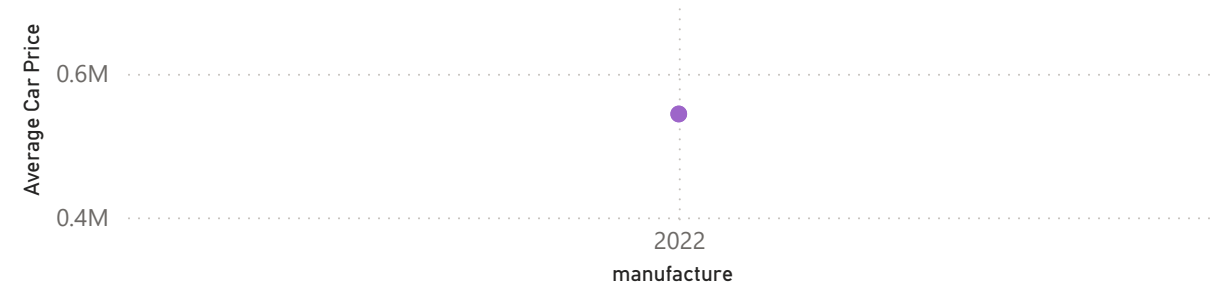
Avg Seats in Car

Count of Manual Cars and Count of Automatic Cars by Manufacturing Year

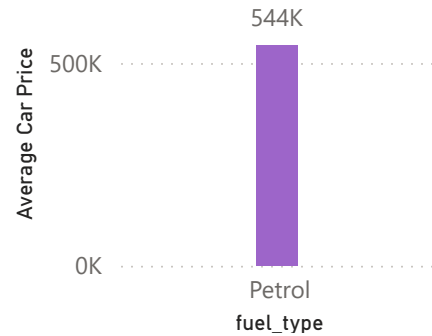
Count of Manual Cars Count of Automatic Cars



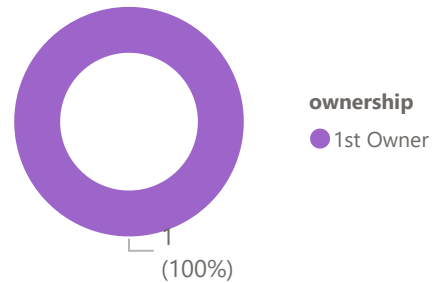
Average Car Price by Manufacturing Year



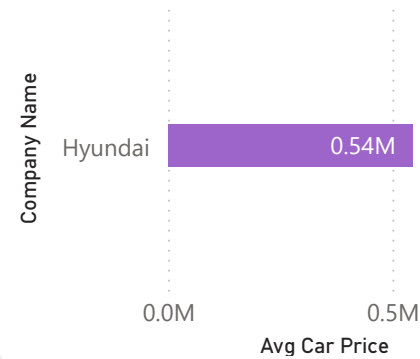
Average Car Price by fuel_type



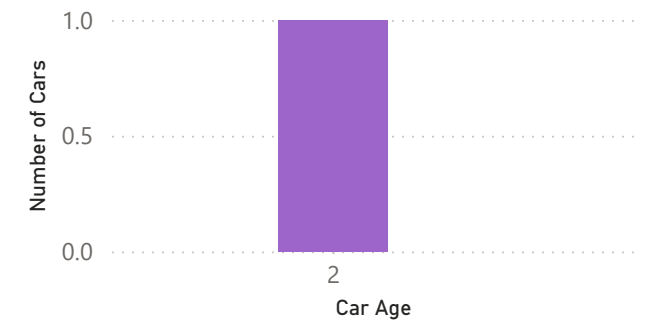
Count of car_name by ownership



Avg Car Price by Company Name



Number of Cars by Car Age



CAR SALES DATA ANALYSIS REPORT

KEY INSIGHTS

1. Key Metrics

- i. Average Engine Capacity for All cars combined: 1532.40 cc, it varies for different car brands.
 - a. Ex: for Audi; Avg Engine Capacity = 1832.20 cc
- ii. Average Km driven for all cars combined: 63.22K km
- iii. Average Car Age: 8.55 years
- iv. Average Seats in Car: 5

2. Data can be filtered based on:

- i. Company Name
- ii. Car Name
- iii. Ownership
- iv. Year Wise – ranging from 1995 to 2022

3. Count of Manual Cars and Count of Automatic Cars by Manufacturing Year

- i. This chart reveals trends in the manufacturing of manual vs. automatic cars over the years.
- ii. A noticeable increase in the production of automatic cars may indicate a shift in consumer preference towards automatic transmissions.
- iii. Peaks and troughs in the counts can show periods of high and low production for each type of transmission.
- iv. Automatic Cars came into Existence in 2002.

4. Average Car Price by Manufacturing Year

- i. This chart illustrates how car prices have changed over time.
- ii. The trend may show an increase in car prices, indicating inflation, advancements in car technology, or changes in consumer demand.
- iii. Sudden spikes or drops in average prices can point to specific events or changes in the automotive industry.
- iv. The car prices have hiked from 0.26 Million in 1995 to 2.67 Million in 2022.

5. Average Car Price by Fuel Type

- i. Shows the average price difference between cars with different fuel types (Diesel, Electric, Petrol, CNG, LPG).
- ii. Electric cars might have higher average prices, reflecting their newer technology and increasing demand for eco-friendly vehicles.
- iii. Diesel and Petrol cars could have varying prices based on their efficiency and fuel costs.

6. Count of Car Name by Ownership

- i. Indicates the distribution of cars based on their ownership status (e.g., 1st Owner, 2nd Owner, etc.).
- ii. A higher count of first-owner cars might suggest a more recent or higher-quality inventory.
- iii. Multiple-owner cars can indicate older vehicles or those with lower reliability.
- iv. Around 67.77% (3.73k) cars are 1st owner cars.
- v. It helps buyers and sellers gauge the value and reliability of cars based on ownership history.

7. Car Price by Company Name

- i. This chart compares the average prices of cars from different manufacturers.
- ii. Identifies which companies produce more expensive vs. more affordable cars.
- iii. Helps to recognize brands with high market value or premium pricing.
- iv. Assists consumers in making brand-based purchasing decisions.
- v. Helps manufacturers understand their market positioning relative to competitors.
- vi. Company Producing the most Expensive car is **PORSCHE** whose Average Car Price is 8.5 Million.
- vii. Company Producing the Cheapest car is **CHEVROLET** whose Average Car Price is 0.2 Million.

8. Number of Cars by Car Age

- i. Shows the distribution of cars based on their age.
- ii. A higher number of newer cars indicates a more current inventory.
- iii. Older cars being more prevalent can suggest a market for second-hand or classic vehicles.
- iv. Helps in assessing the age profile of available cars.

SUMMARY

Each chart on the dashboard provides unique and valuable insights into the car market. From understanding trends in transmission types and fuel preferences to analysing price distributions and ownership patterns, these charts collectively offer a comprehensive view of the automotive landscape. This information is crucial for manufacturers, dealers, and consumers to make informed decisions in a dynamic market.

**DATA CLEANING,
TRANSFORMATION CODES
DONE USING
PYTHON - PANDAS**

DATA ANALYSIS ON CAR SALES DATA

IMPORTING LIBRARIES

```
In [ ]: import pandas as pd
import numpy as np
```

IMPORTING DATA & Analyzing its Structure

```
In [ ]: data = pd.read_csv('car_price.csv')
data.head(1)
```

```
Out[ ]:
```

	car_name	car_prices_in_rupee	kms_driven	fuel_type	transmission	ownership	engine	Seats	manufacture
0	Jeep Compass 2.0 Longitude Option BSIV	10.03 Lakh	86,226 kms	Diesel	Manual	1st Owner	1956 cc	5 Seats	2017.0

```
In [ ]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5512 entries, 0 to 5511
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  -
0   car_name               5512 non-null   object
1   car_prices_in_rupee    5510 non-null   object
2   kms_driven             5511 non-null   object
3   fuel_type              5511 non-null   object
4   transmission           5511 non-null   object
5   ownership              5511 non-null   object
6   engine                 5512 non-null   object
7   Seats                  5511 non-null   object
8   manufacture            5511 non-null   float64
dtypes: float64(1), object(8)
memory usage: 387.7+ KB
```

```
In [ ]: data.shape
```

```
Out[ ]: (5512, 9)
```

Percentage of Data having Null Values

```
In [ ]: (data.isnull().sum() / data.shape[0])
```

```
Out[ ]: car_name          0.000000
car_prices_in_rupee    0.036284
kms_driven             0.018142
fuel_type              0.018142
transmission           0.018142
ownership              0.018142
engine                 0.000000
Seats                  0.018142
manufacture            0.018142
dtype: float64
```

REMOVING ALL NULL VALUES

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

```
In [ ]: (data.isnull().sum() / data.shape[0])
```

```
Out[ ]: car_name          0.0
car_prices_in_rupee    0.0
kms_driven             0.0
fuel_type              0.0
transmission           0.0
ownership              0.0
engine                 0.0
Seats                  0.0
manufacture            0.0
dtype: float64
```

Resolving Column Wise Problems

1. car_name

differentiating company name and car name

```
In [ ]: data['car_name']
```

```
Out[ ]: 0      Jeep Compass 2.0 Longitude Option BSIV
        1      Renault Duster RXZ Turbo CVT
        2      Toyota Camry 2.5 G
        3      Honda Jazz VX CVT
        4      Volkswagen Polo 1.2 MPI Highline
        ...
        5507     BMW X1 sDrive 20d xLine
        5508     BMW M Series M4 Coupe
        5509     Jaguar XF 2.2 Litre Luxury
        5510     BMW 7 Series 730Ld
        5511     BMW 5 Series 520d M Sport
        Name: car_name, Length: 5504, dtype: object
```

```
In [ ]: x = data['car_name'][0]
        print(x)
        x[x.index(' ')]
```

Jeep Compass 2.0 Longitude Option BSIV

```
Out[ ]: 'Jeep'
```

```
In [ ]: def company_name(x):
        return x[x.index(' ')]
        data['car_name'].apply(company_name)
```

```
Out[ ]: 0      Jeep
        1      Renault
        2      Toyota
        3      Honda
        4      Volkswagen
        ...
        5507     BMW
        5508     BMW
        5509     Jaguar
        5510     BMW
        5511     BMW
        Name: car_name, Length: 5504, dtype: object
```

```
In [ ]: data['company_name'] = data['car_name'].apply(company_name)
```

```
In [ ]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 5504 entries, 0 to 5511
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  -
0   car_name              5504 non-null   object
1   car_prices_in_rupee   5504 non-null   object
2   kms_driven            5504 non-null   object
3   fuel_type             5504 non-null   object
4   transmission          5504 non-null   object
5   ownership             5504 non-null   object
6   engine               5504 non-null   object
7   Seats                5504 non-null   object
8   manufacture           5504 non-null   float64
9   xompany_name          5504 non-null   object
10  company_name          5504 non-null   object
dtypes: float64(1), object(10)
memory usage: 645.0+ KB
```

```
In [ ]: data = data.drop('xompany_name', axis=1)
```

```
In [ ]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 5504 entries, 0 to 5511
Data columns (total 10 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   car_name              5504 non-null   object
1   car_prices_in_rupee   5504 non-null   object
2   kms_driven            5504 non-null   object
3   fuel_type             5504 non-null   object
4   transmission          5504 non-null   object
5   ownership             5504 non-null   object
6   engine               5504 non-null   object
7   Seats                5504 non-null   object
8   manufacture          5504 non-null   float64
9   company_name         5504 non-null   object
dtypes: float64(1), object(9)
memory usage: 602.0+ KB
```

```
In [ ]: data.head(1)
```

```
Out[ ]:      car_name  car_prices_in_rupee  kms_driven  fuel_type  transmission  ownership  engine  Seats  manufacture  company_name
0   Jeep Compass  10.03 Lakh    86,226 kms    Diesel      Manual    1st Owner    1956 cc    5 Seats    2017.0    Jeep
```

```
In [ ]: x = data['car_name'][0]
print(x)
x[x.index('')+1:]
```

```
Jeep Compass 2.0 Longitude Option BSIV
```

```
Out[ ]: 'Compass 2.0 Longitude Option BSIV'
```

```
In [ ]: def car_name(x):
        return x[x.index('')+1:]
data['car_name'].apply(car_name)
```

```
Out[ ]: 0      Compass 2.0 Longitude Option BSIV
        1      Duster RXZ Turbo CVT
        2      Camry 2.5 G
        3      Jazz VX CVT
        4      Polo 1.2 MPI Highline
        ...
        5507     X1 sDrive 20d xLine
        5508     M Series M4 Coupe
        5509     XF 2.2 Litre Luxury
        5510     7 Series 730Ld
        5511     5 Series 520d M Sport
Name: car_name, Length: 5504, dtype: object
```

```
In [ ]: data['car_name'] = data['car_name'].apply(car_name)
```

```
In [ ]: data.head(5)
```

```
Out[ ]:
```

	car_name	car_prices_in_rupee	kms_driven	fuel_type	transmission	ownership	engine	Seats	manufacture	company_name
0	Compass 2.0 Longitude Option BSIV	10.03 Lakh	86,226 kms	Diesel	Manual	1st Owner	1956 cc	5 Seats	2017.0	Jeep
1	Duster RXZ Turbo CVT	12.83 Lakh	13,248 kms	Petrol	Automatic	1st Owner	1330 cc	5 Seats	2021.0	Renault
2	Camry 2.5 G	16.40 Lakh	60,343 kms	Petrol	Automatic	1st Owner	2494 cc	5 Seats	2016.0	Toyota
3	Jazz VX CVT	7.77 Lakh	26,696 kms	Petrol	Automatic	1st Owner	1199 cc	5 Seats	2018.0	Honda
4	Polo 1.2 MPI Highline	5.15 Lakh	69,414 kms	Petrol	Manual	1st Owner	1199 cc	5 Seats	2016.0	Volkswagen

2. car_prices_in_rupee

```
In [ ]: data['car_prices_in_rupee'] = data['car_prices_in_rupee'].str.replace(',','')
```

```
In [ ]: x = data['car_prices_in_rupee'][0]
        print(x)
        x[:x.index(' ')]
```

10.03 Lakh

```
Out[ ]: '10.03'
```

```
In [ ]: def rupee_change(x):
        p = x.split(' ')
        try:
            if (p[1] == 'Lakh'):
                return str(round(float(p[0]) * 100000,1))
            elif (p[1] == 'Crore'):
                return str(round(float(p[0]) * 10000000,1))
        except:
            return x
```

```
In [ ]: data['car_prices_in_rupee'] = data['car_prices_in_rupee'].apply(rupee_change)
```

```
In [ ]: data.head(10)
```


Out[]:

	car_name	car_prices_in_rupee	kms_driven	fuel_type	transmission	ownership	engine	Seats	manufacture	company_name
0	Compass 2.0 Longitude Option BSIV	1003000.0	86,226 kms	Diesel	Manual	1st Owner	1956 cc	5 Seats	2017.0	Jeep
1	Duster RXZ Turbo CVT	1283000.0	13,248 kms	Petrol	Automatic	1st Owner	1330 cc	5 Seats	2021.0	Renault
2	Camry 2.5 G	1640000.0	60,343 kms	Petrol	Automatic	1st Owner	2494 cc	5 Seats	2016.0	Toyota
3	Jazz VX CVT	777000.0	26,696 kms	Petrol	Automatic	1st Owner	1199 cc	5 Seats	2018.0	Honda
4	Polo 1.2 MPI Highline	515000.0	69,414 kms	Petrol	Manual	1st Owner	1199 cc	5 Seats	2016.0	Volkswagen
5	Vento 1.2 TSI Highline AT	766000.0	49,719 kms	Petrol	Automatic	1st Owner	1197 cc	5 Seats	2017.0	Volkswagen
6	Vento 1.2 TSI Highline Plus AT	758000.0	43,688 kms	Petrol	Automatic	1st Owner	1197 cc	5 Seats	2017.0	Volkswagen
7	WR-V VX Diesel	1160000.0	14,470 kms	Diesel	Manual	1st Owner	1498 cc	5 Seats	2021.0	Honda
8	City i VTEC CVT SV	699000.0	21,429 kms	Petrol	Automatic	1st Owner	1497 cc	5 Seats	2015.0	Honda
9	Duster Petrol RXS CVT	753000.0	31,750 kms	Petrol	Automatic	1st Owner	1498 cc	5 Seats	2017.0	Renault

In []: data.info()

```

<class 'pandas.core.frame.DataFrame'>
Index: 5504 entries, 0 to 5511
Data columns (total 10 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   car_name              5504 non-null   object
 1   car_prices_in_rupee   5504 non-null   object
 2   kms_driven            5504 non-null   object
 3   fuel_type             5504 non-null   object
 4   transmission          5504 non-null   object
 5   ownership             5504 non-null   object
 6   engine               5504 non-null   object
 7   Seats                5504 non-null   object
 8   manufacture          5504 non-null   float64
 9   company_name         5504 non-null   object
dtypes: float64(1), object(9)
memory usage: 602.0+ KB

```

change data type of car_prices_in_rupee column

```
In [ ]: data['car_prices_in_rupee'] = data['car_prices_in_rupee'].astype('float')
```

```
In [ ]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 5504 entries, 0 to 5511
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ---
 0   car_name            5504 non-null   object
 1   car_prices_in_rupee 5504 non-null   float64
 2   kms_driven          5504 non-null   object
 3   fuel_type           5504 non-null   object
 4   transmission        5504 non-null   object
 5   ownership           5504 non-null   object
 6   engine              5504 non-null   object
 7   Seats              5504 non-null   object
 8   manufacture         5504 non-null   float64
 9   company_name        5504 non-null   object
dtypes: float64(2), object(8)
memory usage: 602.0+ KB
```

3. kms_driven

```
In [ ]: data['kms_driven'] = data['kms_driven'].str.replace(',','')
```

```
In [ ]: data.head(10)
```

Out[]:

	car_name	car_prices_in_rupee	kms_driven	fuel_type	transmission	ownership	engine	Seats	manufacture	company_name
0	Compass 2.0 Longitude Option BSIV	1003000.0	86226 kms	Diesel	Manual	1st Owner	1956 cc	5 Seats	2017.0	Jeep
1	Duster RXZ Turbo CVT	1283000.0	13248 kms	Petrol	Automatic	1st Owner	1330 cc	5 Seats	2021.0	Renault
2	Camry 2.5 G	1640000.0	60343 kms	Petrol	Automatic	1st Owner	2494 cc	5 Seats	2016.0	Toyota
3	Jazz VX CVT	777000.0	26696 kms	Petrol	Automatic	1st Owner	1199 cc	5 Seats	2018.0	Honda
4	Polo 1.2 MPI Highline	515000.0	69414 kms	Petrol	Manual	1st Owner	1199 cc	5 Seats	2016.0	Volkswagen
5	Vento 1.2 TSI Highline AT	766000.0	49719 kms	Petrol	Automatic	1st Owner	1197 cc	5 Seats	2017.0	Volkswagen
6	Vento 1.2 TSI Highline Plus AT	758000.0	43688 kms	Petrol	Automatic	1st Owner	1197 cc	5 Seats	2017.0	Volkswagen
7	WR-V VX Diesel	1160000.0	14470 kms	Diesel	Manual	1st Owner	1498 cc	5 Seats	2021.0	Honda
8	City i VTEC CVT SV	699000.0	21429 kms	Petrol	Automatic	1st Owner	1497 cc	5 Seats	2015.0	Honda
9	Duster Petrol RXS CVT	753000.0	31750 kms	Petrol	Automatic	1st Owner	1498 cc	5 Seats	2017.0	Renault

```
In [ ]: def change_kms_col(x):
        p = x.split(' ')
        return p[0]
```

```
In [ ]: data['kms_driven'] = data['kms_driven'].apply(change_kms_col)
```

```
In [ ]: data.head(10)
```

Out []:

	car_name	car_prices_in_rupee	kms_driven	fuel_type	transmission	ownership	engine	Seats	manufacture	company_name
0	Compass 2.0 Longitude Option BSIV	1003000.0	86226	Diesel	Manual	1st Owner	1956 cc	5 Seats	2017.0	Jeep
1	Duster RXZ Turbo CVT	1283000.0	13248	Petrol	Automatic	1st Owner	1330 cc	5 Seats	2021.0	Renault
2	Camry 2.5 G	1640000.0	60343	Petrol	Automatic	1st Owner	2494 cc	5 Seats	2016.0	Toyota
3	Jazz VX CVT	777000.0	26696	Petrol	Automatic	1st Owner	1199 cc	5 Seats	2018.0	Honda
4	Polo 1.2 MPI Highline	515000.0	69414	Petrol	Manual	1st Owner	1199 cc	5 Seats	2016.0	Volkswagen
5	Vento 1.2 TSI Highline AT	766000.0	49719	Petrol	Automatic	1st Owner	1197 cc	5 Seats	2017.0	Volkswagen
6	Vento 1.2 TSI Highline Plus AT	758000.0	43688	Petrol	Automatic	1st Owner	1197 cc	5 Seats	2017.0	Volkswagen
7	WR-V VX Diesel	1160000.0	14470	Diesel	Manual	1st Owner	1498 cc	5 Seats	2021.0	Honda
8	City i VTEC CVT SV	699000.0	21429	Petrol	Automatic	1st Owner	1497 cc	5 Seats	2015.0	Honda
9	Duster Petrol RXS CVT	753000.0	31750	Petrol	Automatic	1st Owner	1498 cc	5 Seats	2017.0	Renault

In []: `data['kms_driven'] = data['kms_driven'].astype('int')`In []: `data.info()`

```
<class 'pandas.core.frame.DataFrame'>
Index: 5504 entries, 0 to 5511
Data columns (total 10 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   car_name              5504 non-null   object
1   car_prices_in_rupee   5504 non-null   float64
2   kms_driven            5504 non-null   int64
3   fuel_type             5504 non-null   object
4   transmission          5504 non-null   object
5   ownership             5504 non-null   object
6   engine               5504 non-null   object
7   Seats                5504 non-null   object
8   manufacture           5504 non-null   float64
9   company_name          5504 non-null   object
dtypes: float64(2), int64(1), object(7)
memory usage: 602.0+ KB
```

```
In [ ]: data.head(10)
```

Out[]:

	car_name	car_prices_in_rupee	kms_driven	fuel_type	transmission	ownership	engine	Seats	manufacture	company_name
0	Compass 2.0 Longitude Option BSIV	1003000.0	86226	Diesel	Manual	1st Owner	1956 cc	5 Seats	2017.0	Jeep
1	Duster RXZ Turbo CVT	1283000.0	13248	Petrol	Automatic	1st Owner	1330 cc	5 Seats	2021.0	Renault
2	Camry 2.5 G	1640000.0	60343	Petrol	Automatic	1st Owner	2494 cc	5 Seats	2016.0	Toyota
3	Jazz VX CVT	777000.0	26696	Petrol	Automatic	1st Owner	1199 cc	5 Seats	2018.0	Honda
4	Polo 1.2 MPI Highline	515000.0	69414	Petrol	Manual	1st Owner	1199 cc	5 Seats	2016.0	Volkswagen
5	Vento 1.2 TSI Highline AT	766000.0	49719	Petrol	Automatic	1st Owner	1197 cc	5 Seats	2017.0	Volkswagen
6	Vento 1.2 TSI Highline Plus AT	758000.0	43688	Petrol	Automatic	1st Owner	1197 cc	5 Seats	2017.0	Volkswagen
7	WR-V VX Diesel	1160000.0	14470	Diesel	Manual	1st Owner	1498 cc	5 Seats	2021.0	Honda
8	City i VTEC CVT SV	699000.0	21429	Petrol	Automatic	1st Owner	1497 cc	5 Seats	2015.0	Honda
9	Duster Petrol RXS CVT	753000.0	31750	Petrol	Automatic	1st Owner	1498 cc	5 Seats	2017.0	Renault

4. manufacture

In []: data['manufacture'] = data['manufacture'].astype('int')

In []: data.info()

```
<class 'pandas.core.frame.DataFrame'>
Index: 5504 entries, 0 to 5511
Data columns (total 10 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   car_name              5504 non-null   object
1   car_prices_in_rupee   5504 non-null   float64
2   kms_driven            5504 non-null   int64
3   fuel_type             5504 non-null   object
4   transmission          5504 non-null   object
5   ownership             5504 non-null   object
6   engine               5504 non-null   object
7   Seats                5504 non-null   object
8   manufacture           5504 non-null   int64
9   company_name          5504 non-null   object
dtypes: float64(1), int64(2), object(7)
memory usage: 602.0+ KB
```

```
In [ ]: data.head(10)
```


Out[]:

	car_name	car_prices_in_rupee	kms_driven	fuel_type	transmission	ownership	engine	Seats	manufacture	company_name
0	Compass 2.0 Longitude Option BSIV	1003000.0	86226	Diesel	Manual	1st Owner	1956 cc	5 Seats	2017	Jeep
1	Duster RXZ Turbo CVT	1283000.0	13248	Petrol	Automatic	1st Owner	1330 cc	5 Seats	2021	Renault
2	Camry 2.5 G	1640000.0	60343	Petrol	Automatic	1st Owner	2494 cc	5 Seats	2016	Toyota
3	Jazz VX CVT	777000.0	26696	Petrol	Automatic	1st Owner	1199 cc	5 Seats	2018	Honda
4	Polo 1.2 MPI Highline	515000.0	69414	Petrol	Manual	1st Owner	1199 cc	5 Seats	2016	Volkswagen
5	Vento 1.2 TSI Highline AT	766000.0	49719	Petrol	Automatic	1st Owner	1197 cc	5 Seats	2017	Volkswagen
6	Vento 1.2 TSI Highline Plus AT	758000.0	43688	Petrol	Automatic	1st Owner	1197 cc	5 Seats	2017	Volkswagen
7	WR-V VX Diesel	1160000.0	14470	Diesel	Manual	1st Owner	1498 cc	5 Seats	2021	Honda
8	City i VTEC CVT SV	699000.0	21429	Petrol	Automatic	1st Owner	1497 cc	5 Seats	2015	Honda
9	Duster Petrol RXS CVT	753000.0	31750	Petrol	Automatic	1st Owner	1498 cc	5 Seats	2017	Renault

5. seats

In []: data['Seats'] = data['Seats'].str.replace('Seats', '')

In []: data.head(10)

Out []:

	car_name	car_prices_in_rupee	kms_driven	fuel_type	transmission	ownership	engine	Seats	manufacture	company_name
0	Compass 2.0 Longitude Option BSIV	1003000.0	86226	Diesel	Manual	1st Owner	1956 cc	5	2017	Jeep
1	Duster RXZ Turbo CVT	1283000.0	13248	Petrol	Automatic	1st Owner	1330 cc	5	2021	Renault
2	Camry 2.5 G	1640000.0	60343	Petrol	Automatic	1st Owner	2494 cc	5	2016	Toyota
3	Jazz VX CVT	777000.0	26696	Petrol	Automatic	1st Owner	1199 cc	5	2018	Honda
4	Polo 1.2 MPI Highline	515000.0	69414	Petrol	Manual	1st Owner	1199 cc	5	2016	Volkswagen
5	Vento 1.2 TSI Highline AT	766000.0	49719	Petrol	Automatic	1st Owner	1197 cc	5	2017	Volkswagen
6	Vento 1.2 TSI Highline Plus AT	758000.0	43688	Petrol	Automatic	1st Owner	1197 cc	5	2017	Volkswagen
7	WR-V VX Diesel	1160000.0	14470	Diesel	Manual	1st Owner	1498 cc	5	2021	Honda
8	City i VTEC CVT SV	699000.0	21429	Petrol	Automatic	1st Owner	1497 cc	5	2015	Honda
9	Duster Petrol RXS CVT	753000.0	31750	Petrol	Automatic	1st Owner	1498 cc	5	2017	Renault

In []: data['Seats'] = data['Seats'].astype('int')

In []: data.info()

```

<class 'pandas.core.frame.DataFrame'>
Index: 5504 entries, 0 to 5511
Data columns (total 10 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   car_name               5504 non-null   object
1   car_prices_in_rupee    5504 non-null   float64
2   kms_driven             5504 non-null   int64
3   fuel_type              5504 non-null   object
4   transmission           5504 non-null   object
5   ownership              5504 non-null   object
6   engine                 5504 non-null   object
7   Seats                  5504 non-null   int64
8   manufacture            5504 non-null   int64
9   company_name           5504 non-null   object
dtypes: float64(1), int64(3), object(6)
memory usage: 602.0+ KB

```

6. engine

```

In [ ]: data['engine'] = data['engine'].str.replace(' cc','')
        data['engine'] = data['engine'].astype('int')

```

```

In [ ]: data.head()

```

Out[]:

	car_name	car_prices_in_rupee	kms_driven	fuel_type	transmission	ownership	engine	Seats	manufacture	company_name
0	Compass 2.0 Longitude Option BSIV	1003000.0	86226	Diesel	Manual	1st Owner	1956	5	2017	Jeep
1	Duster RXZ Turbo CVT	1283000.0	13248	Petrol	Automatic	1st Owner	1330	5	2021	Renault
2	Camry 2.5 G	1640000.0	60343	Petrol	Automatic	1st Owner	2494	5	2016	Toyota
3	Jazz VX CVT	777000.0	26696	Petrol	Automatic	1st Owner	1199	5	2018	Honda
4	Polo 1.2 MPI Highline	515000.0	69414	Petrol	Manual	1st Owner	1199	5	2016	Volkswagen

In []: data.info()

```
<class 'pandas.core.frame.DataFrame'>
Index: 5504 entries, 0 to 5511
Data columns (total 10 columns):
#   Column                Non-Null Count  Dtype
---  -
0   car_name              5504 non-null  object
1   car_prices_in_rupee   5504 non-null  float64
2   kms_driven            5504 non-null  int64
3   fuel_type             5504 non-null  object
4   transmission          5504 non-null  object
5   ownership             5504 non-null  object
6   engine               5504 non-null  int64
7   Seats                5504 non-null  int64
8   manufacture           5504 non-null  int64
9   company_name          5504 non-null  object
dtypes: float64(1), int64(4), object(5)
memory usage: 602.0+ KB
```

In []: data.rename(columns={'engine': 'engine_in_cc'}, inplace=True)

In []: data.info()

```

<class 'pandas.core.frame.DataFrame'>
Index: 5504 entries, 0 to 5511
Data columns (total 10 columns):
 #   Column                Non-Null Count  Dtype  
---  -
 0   car_name              5504 non-null   object  
 1   car_prices_in_rupee   5504 non-null   float64  
 2   kms_driven            5504 non-null   int64  
 3   fuel_type             5504 non-null   object  
 4   transmission          5504 non-null   object  
 5   ownership             5504 non-null   object  
 6   engine_in_cc          5504 non-null   int64  
 7   Seats                 5504 non-null   int64  
 8   manufacture           5504 non-null   int64  
 9   company_name          5504 non-null   object  
dtypes: float64(1), int64(4), object(5)
memory usage: 602.0+ KB

```

```
In [ ]: data.head()
```

```
Out[ ]:
```

	car_name	car_prices_in_rupee	kms_driven	fuel_type	transmission	ownership	engine_in_cc	Seats	manufacture	company_name
0	Compass 2.0 Longitude Option BSIV	1003000.0	86226	Diesel	Manual	1st Owner	1956	5	2017	Jeep
1	Duster RXZ Turbo CVT	1283000.0	13248	Petrol	Automatic	1st Owner	1330	5	2021	Renault
2	Camry 2.5 G	1640000.0	60343	Petrol	Automatic	1st Owner	2494	5	2016	Toyota
3	Jazz VX CVT	777000.0	26696	Petrol	Automatic	1st Owner	1199	5	2018	Honda
4	Polo 1.2 MPI Highline	515000.0	69414	Petrol	Manual	1st Owner	1199	5	2016	Volkswagen

```
In [ ]: data.to_csv('car_price_transformed_data.csv', index=False)
```

```
In [ ]:
```