import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import matplotlib as mpl  
  
# %matplotlib inline  
mpl.style.use("ggplot")

car = pd.read\_csv("./car\_resale\_prices.csv")  
car.head()

Unnamed: 0 full\_name resale\_price registered\_year \  
0 0 2017 Maruti Baleno 1.2 Alpha ₹ 5.45 Lakh 2017   
1 1 2018 Tata Hexa XTA ₹ 10 Lakh 2018   
2 2 2015 Maruti Swift Dzire VXI ₹ 4.50 Lakh 2015   
3 3 2015 Maruti Swift Dzire VXI ₹ 4.50 Lakh 2015   
4 4 2009 Hyundai i10 Magna 1.1 ₹ 1.60 Lakh 2009   
  
 engine\_capacity insurance transmission\_type kms\_driven \  
0 1197 cc Third Party insurance Manual 40,000 Kms   
1 2179 cc Third Party insurance Automatic 70,000 Kms   
2 1197 cc Third Party insurance Manual 70,000 Kms   
3 1197 cc Third Party insurance Manual 70,000 Kms   
4 1086 cc Third Party insurance Manual 80,000 Kms   
  
 owner\_type fuel\_type max\_power seats mileage body\_type city   
0 First Owner Petrol 83.1bhp 5.0 21.4 kmpl Hatchback Agra   
1 First Owner Diesel 153.86bhp 7.0 17.6 kmpl MUV Agra   
2 Second Owner Petrol 83.14bhp 5.0 20.85 kmpl Sedan Agra   
3 Second Owner Petrol 83.14bhp 5.0 20.85 kmpl Sedan Agra   
4 First Owner Petrol 68.05bhp 5.0 19.81 kmpl Hatchback Agra

print("Shape: ", car.shape)  
car.info()

Shape: (17446, 15)  
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 17446 entries, 0 to 17445  
Data columns (total 15 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 Unnamed: 0 17446 non-null int64   
 1 full\_name 17446 non-null object   
 2 resale\_price 17446 non-null object   
 3 registered\_year 17377 non-null object   
 4 engine\_capacity 17432 non-null object   
 5 insurance 17439 non-null object   
 6 transmission\_type 17446 non-null object   
 7 kms\_driven 17443 non-null object   
 8 owner\_type 17401 non-null object   
 9 fuel\_type 17446 non-null object   
 10 max\_power 17344 non-null object   
 11 seats 17436 non-null float64  
 12 mileage 16938 non-null object   
 13 body\_type 17446 non-null object   
 14 city 17446 non-null object   
dtypes: float64(1), int64(1), object(13)  
memory usage: 2.0+ MB

cars\_new = pd.DataFrame()  
# car[  
# [  
# "full\_name",  
# "resale\_price",  
# "registered\_year",  
# "engine\_capacity",  
# "transmission\_type",  
# "kms\_driven",  
# "owner\_type",  
# "fuel\_type",  
# "max\_power",  
# "seats",  
# "mileage",  
# "body\_type",  
# ]  
# ]

## Cleaning data

### Removing year from full\_name

print(car["full\_name"].isnull().sum())  
  
cars\_new["full\_name"] = car["full\_name"].map(lambda x: x[5:])  
  
cars\_new["full\_name"]

0

0 Maruti Baleno 1.2 Alpha  
1 Tata Hexa XTA  
2 Maruti Swift Dzire VXI  
3 Maruti Swift Dzire VXI  
4 Hyundai i10 Magna 1.1  
 ...   
17441 Honda Amaze VX i-Vtech  
17442 Toyota Camry 2.5 Hybrid  
17443 Toyota Corolla Altis GL MT  
17444 Hyundai Creta 1.6 CRDi AT SX Plus  
17445 Maruti Swift Dzire VDi  
Name: full\_name, Length: 17446, dtype: object

import re  
  
print(car["resale\_price"].isnull().sum())  
  
cars\_new["resale\_price"] = car["resale\_price"].map(  
 lambda x: float(re.search(r"\d+(\.\d+)?", x).group(0)) \* 100000  
)  
  
cars\_new["resale\_price"]

0

0 545000.0  
1 1000000.0  
2 450000.0  
3 450000.0  
4 160000.0  
 ...   
17441 325000.0  
17442 2075000.0  
17443 835000.0  
17444 1395000.0  
17445 650000.0  
Name: resale\_price, Length: 17446, dtype: float64

print(car["registered\_year"].isnull().sum())  
  
cars\_new["registered\_year"] = car["full\_name"].map(  
 lambda x: int(re.search(r"\d{4}", str(x)).group(0))  
)  
  
cars\_new["registered\_year"]

69

0 2017  
1 2018  
2 2015  
3 2015  
4 2009  
 ...   
17441 2013  
17442 2016  
17443 2016  
17444 2019  
17445 2017  
Name: registered\_year, Length: 17446, dtype: int64

# car["engine\_capacity"].info()  
# check for null values  
print(car["engine\_capacity"].isnull().sum())  
  
cars\_new["engine\_capacity"] = car["engine\_capacity"].map(  
 lambda x: pd.NA  
 if re.search(r"\d+", str(x)) is None  
 else int(re.search(r"\d+", str(x)).group(0))  
)  
  
cars\_new["engine\_capacity"]

14

0 1197  
1 2179  
2 1197  
3 1197  
4 1086  
 ...   
17441 1198  
17442 2494  
17443 1798  
17444 1582  
17445 1248  
Name: engine\_capacity, Length: 17446, dtype: object

print(car["transmission\_type"].isnull().sum())  
  
cars\_new["transmission\_type"] = car["transmission\_type"]  
  
cars\_new["transmission\_type"]

0

0 Manual  
1 Automatic  
2 Manual  
3 Manual  
4 Manual  
 ...   
17441 Manual  
17442 Automatic  
17443 Manual  
17444 Automatic  
17445 Manual  
Name: transmission\_type, Length: 17446, dtype: object

print(car["kms\_driven"].isnull().sum())  
  
cars\_new["kms\_driven"] = car["kms\_driven"].map(  
 lambda x: pd.NA  
 if re.search(r"\d+", str(x).replace(",", "")) is None  
 else int(re.search(r"\d+", str(x).replace(",", "")).group(0))  
)  
  
cars\_new["kms\_driven"]

3

0 40000  
1 70000  
2 70000  
3 70000  
4 80000  
 ...   
17441 89000  
17442 68000  
17443 81000  
17444 20000  
17445 32000  
Name: kms\_driven, Length: 17446, dtype: object

print(car["owner\_type"].isnull().sum())  
  
cars\_new["owner\_type"] = car["owner\_type"]  
  
cars\_new["owner\_type"]

45

0 First Owner  
1 First Owner  
2 Second Owner  
3 Second Owner  
4 First Owner  
 ...   
17441 Second Owner  
17442 First Owner  
17443 First Owner  
17444 First Owner  
17445 First Owner  
Name: owner\_type, Length: 17446, dtype: object

print(car["fuel\_type"].isnull().sum())  
  
cars\_new["fuel\_type"] = car["fuel\_type"]  
  
cars\_new["fuel\_type"]

0

0 Petrol  
1 Diesel  
2 Petrol  
3 Petrol  
4 Petrol  
 ...   
17441 Petrol  
17442 Petrol  
17443 Petrol  
17444 Diesel  
17445 Diesel  
Name: fuel\_type, Length: 17446, dtype: object

print(car["seats"].isnull().sum())  
  
cars\_new["seats"] = car["seats"].map(  
 lambda x: pd.NA  
 if re.search(r"\d+", str(x)) is None  
 else int(re.search(r"\d+", str(x)).group(0))  
)  
  
cars\_new["seats"]

10

0 5  
1 7  
2 5  
3 5  
4 5  
 ..  
17441 5  
17442 5  
17443 5  
17444 5  
17445 5  
Name: seats, Length: 17446, dtype: object

print(car["mileage"].isnull().sum())  
  
cars\_new["mileage"] = car["mileage"].map(  
 lambda x: pd.NA  
 if re.search(r"\d+(\.\d+)?", str(x)) is None  
 else float(re.search(r"\d+(\.\d+)?", str(x)).group(0))  
)  
  
cars\_new["mileage"]

508

0 21.4  
1 17.6  
2 20.85  
3 20.85  
4 19.81  
 ...   
17441 18.0  
17442 19.16  
17443 14.28  
17444 17.01  
17445 19.3  
Name: mileage, Length: 17446, dtype: object

print(car["body\_type"].isnull().sum())  
cars\_new["body\_type"] = car["body\_type"]  
  
cars\_new["body\_type"]

0

0 Hatchback  
1 MUV  
2 Sedan  
3 Sedan  
4 Hatchback  
 ...   
17441 Sedan  
17442 Sedan  
17443 Sedan  
17444 SUV  
17445 Sedan  
Name: body\_type, Length: 17446, dtype: object

cars\_new.dropna(how="any", inplace=True)  
  
cleaned\_cars = pd.DataFrame()  
  
cleaned\_cars["full\_name"] = cars\_new["full\_name"]  
cleaned\_cars["resale\_price"] = cars\_new["resale\_price"]  
cleaned\_cars["registered\_year"] = cars\_new["registered\_year"]  
cleaned\_cars["engine\_capacity"] = cars\_new["engine\_capacity"].astype(float)  
cleaned\_cars["kms\_driven"] = cars\_new["kms\_driven"].astype(int)  
cleaned\_cars["owner\_type"] = cars\_new["owner\_type"]  
cleaned\_cars["transmission\_type"] = cars\_new["transmission\_type"]  
cleaned\_cars["fuel\_type"] = cars\_new["fuel\_type"]  
cleaned\_cars["mileage"] = cars\_new["mileage"].astype(float)  
cleaned\_cars["body\_type"] = cars\_new["body\_type"]  
cleaned\_cars["seats"] = cars\_new["seats"].astype(int)  
  
cleaned\_cars = cleaned\_cars.reset\_index(drop=True)  
  
cleaned\_cars.info()  
  
# export to csv  
cleaned\_cars.to\_csv("./cleaned\_cars.csv")

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 16880 entries, 0 to 16879  
Data columns (total 11 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 full\_name 16880 non-null object   
 1 resale\_price 16880 non-null float64  
 2 registered\_year 16880 non-null int64   
 3 engine\_capacity 16880 non-null float64  
 4 kms\_driven 16880 non-null int64   
 5 owner\_type 16880 non-null object   
 6 transmission\_type 16880 non-null object   
 7 fuel\_type 16880 non-null object   
 8 mileage 16880 non-null float64  
 9 body\_type 16880 non-null object   
 10 seats 16880 non-null int64   
dtypes: float64(3), int64(3), object(5)  
memory usage: 1.4+ MB

cleaned\_cars.describe(include="all")

full\_name resale\_price registered\_year engine\_capacity \  
count 16880 1.688000e+04 16880.000000 16880.000000   
unique 3197 NaN NaN NaN   
top Maruti Swift VXI NaN NaN NaN   
freq 230 NaN NaN NaN   
mean NaN 9.080702e+05 2016.235545 1419.378318   
std NaN 1.145647e+06 3.645119 465.852730   
min NaN 1.000000e+05 1985.000000 0.000000   
25% NaN 3.847500e+05 2014.000000 1197.000000   
50% NaN 5.850000e+05 2017.000000 1248.000000   
75% NaN 9.000000e+05 2019.000000 1498.000000   
max NaN 9.900000e+06 2023.000000 5998.000000   
  
 kms\_driven owner\_type transmission\_type fuel\_type mileage \  
count 1.688000e+04 16880 16880 16880 16880.000000   
unique NaN 5 2 5 NaN   
top NaN First Owner Manual Petrol NaN   
freq NaN 11851 12277 11068 NaN   
mean 5.934513e+04 NaN NaN NaN 19.379589   
std 6.428737e+04 NaN NaN NaN 4.247416   
min 3.000000e+02 NaN NaN NaN 6.700000   
25% 3.309250e+04 NaN NaN NaN 17.000000   
50% 5.549600e+04 NaN NaN NaN 18.900000   
75% 8.000000e+04 NaN NaN NaN 21.630000   
max 6.275000e+06 NaN NaN NaN 140.000000   
  
 body\_type seats   
count 16880 16880.000000   
unique 25 NaN   
top Hatchback NaN   
freq 7242 NaN   
mean NaN 5.205036   
std NaN 0.660398   
min NaN 2.000000   
25% NaN 5.000000   
50% NaN 5.000000   
75% NaN 5.000000   
max NaN 14.000000