

Name: Vatsal Arya
Roll No.:12
Lab3

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df = pd.read_csv("https://raw.githubusercontent.com/selva86/datasets/master/Cars93_miss.csv")
df.shape
```

(93, 27)

```
print(df.describe())
```

	Min.Price	Price	Max.Price	MPG.city	MPG.highway	EngineSize	\
count	86.000000	91.000000	88.000000	84.000000	91.000000	91.000000	
mean	17.118605	19.616484	21.459091	22.404762	29.065934	2.658242	
std	8.828290	9.724280	10.696563	5.841520	5.370293	1.045845	
min	6.700000	7.400000	7.900000	15.000000	20.000000	1.000000	
25%	10.825000	12.350000	14.575000	18.000000	26.000000	1.800000	
50%	14.600000	17.700000	19.150000	21.000000	28.000000	2.300000	
75%	20.250000	23.500000	24.825000	25.000000	31.000000	3.250000	
max	45.400000	61.900000	80.000000	46.000000	50.000000	5.700000	

	Horsepower	RPM	Rev.per.mile	Fuel.tank.capacity	Passengers	\
count	86.000000	90.000000	87.000000	85.000000	91.000000	
mean	144.000000	5276.666667	2355.000000	16.683529	5.076923	
std	53.455204	605.554811	486.916616	3.375748	1.045953	
min	55.000000	3800.000000	1320.000000	9.200000	2.000000	
25%	100.750000	4800.000000	2017.500000	14.500000	4.000000	
50%	140.000000	5200.000000	2360.000000	16.500000	5.000000	
75%	170.000000	5787.500000	2565.000000	19.000000	6.000000	
max	300.000000	6500.000000	3755.000000	27.000000	8.000000	

	Length	Wheelbase	Width	Turn.circle	Rear.seat.room	\
count	89.000000	92.000000	87.000000	88.000000	89.000000	
mean	182.865169	103.956522	69.448276	38.954545	27.853933	
std	14.792651	6.856317	3.778023	3.304157	3.018129	
min	141.000000	90.000000	60.000000	32.000000	19.000000	
25%	174.000000	98.000000	67.000000	36.000000	26.000000	
50%	181.000000	103.000000	69.000000	39.000000	27.500000	
75%	192.000000	110.000000	72.000000	42.000000	30.000000	
max	219.000000	119.000000	78.000000	45.000000	36.000000	

	Luggage.room	Weight
count	74.000000	86.000000
mean	13.986486	3104.593023
std	3.120824	600.129993
min	6.000000	1695.000000
25%	12.000000	2647.500000
50%	14.000000	3085.000000
75%	16.000000	3567.500000
max	22.000000	4105.000000

```
df.head()
```

	Manufacturer	Model	Type	Min.Price	Price	Max.Price	MPG.city	MPG.highway
0	Acura	Integra	Small	12.9	15.9	18.8	25.0	31.0
1	NaN	Legend	Midsize	29.2	33.9	38.7	18.0	25.0
2	Audi	90	Compact	25.9	29.1	32.3	20.0	26.0
3	Audi	100	Midsize	NaN	37.7	44.6	19.0	26.0
4	BMW	535i	Midsize	NaN	30.0	NaN	22.0	30.0

5 rows × 27 columns

```
df.dtypes
```

Manufacturer	object
Model	object
Type	object

```

Min.Price      float64
Price          float64
Max.Price      float64
MPG.city       float64
MPG.highway    float64
AirBags        object
DriveTrain     object
Cylinders      object
EngineSize     float64
Horsepower     float64
RPM            float64
Rev.per.mile   float64
Man.trans.avail object
Fuel.tank.capacity float64
Passengers     float64
Length         float64
Wheelbase      float64
Width          float64
Turn.circle    float64
Rear.seat.room float64
Luggage.room   float64
Weight         float64
Origin         object
Make           object
dtype: object

```

```
df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 93 entries, 0 to 92
Data columns (total 27 columns):
 #   Column              Non-Null Count  Dtype
---  -
 0   Manufacturer        89 non-null    object
 1   Model               92 non-null    object
 2   Type               90 non-null    object
 3   Min.Price          86 non-null    float64
 4   Price              91 non-null    float64
 5   Max.Price          88 non-null    float64
 6   MPG.city           84 non-null    float64
 7   MPG.highway        91 non-null    float64
 8   AirBags            87 non-null    object
 9   DriveTrain         86 non-null    object
10   Cylinders           88 non-null    object
11   EngineSize         91 non-null    float64
12   Horsepower         86 non-null    float64
13   RPM                90 non-null    float64
14   Rev.per.mile       87 non-null    float64
15   Man.trans.avail    88 non-null    object
16   Fuel.tank.capacity 85 non-null    float64
17   Passengers         91 non-null    float64
18   Length             89 non-null    float64
19   Wheelbase          92 non-null    float64
20   Width              87 non-null    float64
21   Turn.circle        88 non-null    float64
22   Rear.seat.room     89 non-null    float64
23   Luggage.room       74 non-null    float64
24   Weight             86 non-null    float64
25   Origin             88 non-null    object
26   Make              90 non-null    object
dtypes: float64(18), object(9)
memory usage: 19.7+ KB

```

```
df = df.drop_duplicates()
```

```
df.shape
```

```
(93, 27)
```

```
print(df.isnull().sum())
```

```

Manufacturer    4
Model           1
Type            3
Min.Price       7
Price           2
Max.Price       5
MPG.city        9
MPG.highway     2
AirBags         6
DriveTrain      7
Cylinders       5
EngineSize      2
Horsepower      7
RPM             3
Rev.per.mile    6
Man.trans.avail 5
Fuel.tank.capacity 8

```

```
Passengers      2
Length          4
Wheelbase       1
Width           6
Turn.circle     5
Rear.seat.room  4
Luggage.room    19
Weight          7
Origin          5
Make            3
dtype: int64
```

df.isnull()

	Manufacturer	Model	Type	Min.Price	Price	Max.Price	MPG.city	MPG.highway	Ai
0	False	False	False	False	False	False	False	False	
1	True	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	
3	False	False	False	True	False	False	False	False	
4	False	False	False	True	False	True	False	False	
...	
88	False	False	False	False	False	False	False	False	
89	False	False	False	False	False	False	False	False	
90	False	False	False	False	False	False	False	False	
91	False	False	False	False	False	False	False	False	
92	True	False	False	False	False	False	False	False	

93 rows × 27 columns

df[80:88]

	Manufacturer	Model	Type	Min.Price	Price	Max.Price	MPG.city	MPG.highway	
80	Subaru	Loyale	Small	10.5	10.9	11.3	25.0	30.0	
81	Subaru	Legacy	Compact	16.3	19.5	22.7	23.0	30.0	
82	Suzuki	Swift	NaN	7.3	8.6	NaN	39.0	43.0	
83	Toyota	Tercel	Small	NaN	9.8	11.8	32.0	37.0	
84	Toyota	Celica	Sporty	14.2	18.4	22.6	25.0	32.0	
85	Toyota	Camry	Midsize	15.2	NaN	21.2	22.0	29.0	
86	Toyota	Previa	Van	NaN	22.7	26.6	18.0	22.0	
87	Volkswagen	Fox	Small	8.7	9.1	9.5	25.0	33.0	

8 rows × 27 columns

```
val = df['Passengers'].mean()
df['Passengers'][83] = round(val)
```

<ipython-input-12-a939814bbb35>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus

```
df['Passengers'][83] = round(val)
```

df[80:88]

	Manufacturer	Model	Type	Min.Price	Price	Max.Price	MPG.city	MPG.highway
80	Subaru	Loyale	Small	10.5	10.9	11.3	25.0	30.0
81	Subaru	Legacy	Compact	16.3	19.5	22.7	23.0	30.0
82	Suzuki	Swift	NaN	7.3	8.6	NaN	39.0	43.0
83	Toyota	Tercel	Small	NaN	9.8	11.8	32.0	37.0
84	Toyota	Celica	Sporty	14.2	18.4	22.6	25.0	32.0
85	Toyota	Camry	Midsize	15.2	NaN	21.2	22.0	29.0
86	Toyota	Previa	Van	NaN	22.7	26.6	18.0	22.0
87	Volkswagen	Fox	Small	8.7	9.1	9.5	25.0	33.0

```
df.dropna(inplace=True)
df.shape
```

```
(20, 5)
```

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
df.shape
```

```
(20, 27)
```

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
df=df[['Min.Price', 'Price', 'Max.Price', 'MPG.city', 'MPG.highway']]
df.head()
```

	Min.Price	Price	Max.Price	MPG.city	MPG.highway
2	25.9	29.1	32.3	20.0	26.0
7	22.6	23.7	24.9	16.0	25.0
17	18.0	18.8	19.6	17.0	26.0
20	14.5	15.8	17.1	23.0	28.0
21	29.5	29.5	29.5	20.0	26.0

```
df = pd.read_csv("https://raw.githubusercontent.com/selva86/datasets/master/Cars93_miss.csv")
```


```
df
```

	Manufacturer	Model	Type	Min.Price	Price	Max.Price	MPG.city	MPG.highway
0	Acura	Integra	Small	12.9	15.9	18.8	25.0	31.1
1	NaN	Legend	Midsize	29.2	33.9	38.7	18.0	25.1

```
df = df[['Manufacturer', 'Price']][df.Price==df['Price'].min()]
df
```

	Manufacturer	Price
30	Ford	7.4

```
car_manufacturers = df.groupby('Manufacturer')
priceDF = car_manufacturers['Manufacturer','Price'].max()
priceDF
```

 <ipython-input-26-16287efaae54>:2: FutureWarning: Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated in a future version of pandas

```
priceDF = car_manufacturers['Manufacturer','Price'].max()
```

	Manufacturer	Price
	Manufacturer	
	Acura	Acura 15.9
	Audi	Audi 37.7
	BMW	BMW 30.0
	Buick	Buick 26.3
	Cadillac	Cadillac 40.1
	Chevrolet	Chevrolet 38.0
	Chrysler	Chrysler 29.5
	Dodge	Dodge 25.8
	Eagle	Eagle 19.3
	Ford	Ford 20.9
	Geo	Geo 12.5
	Honda	Honda 19.8
	Hyundai	Hyundai 13.9
	Infiniti	Infiniti 47.9
	Lexus	Lexus 28.0
	Lincoln	Lincoln 36.1
	Mazda	Mazda 32.5
	Mercedes-Benz	Mercedes-Benz 61.9
	Mercury	Mercury 14.9
	Mitsubishi	Mitsubishi 26.1
	Nissan	Nissan 21.5
	Oldsmobile	Oldsmobile 20.7
	Plymouth	Plymouth 14.4
	Pontiac	Pontiac 24.4
	Saab	Saab 28.7
	Saturn	Saturn NaN
	Subaru	Subaru 19.5
	Suzuki	Suzuki 8.6
	Toyota	Toyota 22.7
	Volkswagen	Volkswagen 23.3
	Volvo	Volvo 22.7

