

```
# NAME :- RISHIKESH MAHESH JAMADAR
# BRANCH :- ENTC(ELECTRONICS & TELECOMMUNICATION)
# COLLAGE :- WALCHAND INSTITUTE OF TECHNOLOGY,SOLAPUR
# STD:- THIRD YEAR
# TITLE :- TAKE ANY DATA SET AND PERFORM EXPLORATORY DATA ANALYSIS
```

```
#1. using panda library to display the data set
import pandas as pd
df = pd.read_csv('/content/Car Data.csv')
df
```

↗

	Car ID	Brand	Model	Year	Color	Mileage	Price	Location
0	1	Toyota	Camry	2018	White	45000	18000	Los Angeles
1	2	Honda	Civic	2019	Blue	35000	16000	New York
2	3	Ford	Focus	2017	Silver	55000	14000	Chicago
3	4	Chevrolet	Cruze	2016	Red	60000	12000	Miami
4	5	Hyundai	Elantra	2018	Black	40000	15000	San Francisco
...	...	...	...	...	...	...	...	...
66	67	Honda	Fit	2017	Gray	55000	12000	Atlanta
67	68	Ford	Fusion	2018	White	50000	15000	Phoenix
68	69	Chevrolet	Malibu	2019	Blue	40000	17000	Houston
69	70	Hyundai	Venue	2016	Silver	60000	14000	Seattle
70	71	Toyota	Yaris	2017	Black	55000	12000	Los Angeles

71 rows × 8 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 71 entries, 0 to 70
Data columns (total 8 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Car ID      71 non-null    int64
1   Brand       71 non-null    object
2   Model       71 non-null    object
3   Year        71 non-null    int64
4   Color       71 non-null    object
5   Mileage     71 non-null    int64
6   Price       71 non-null    int64
7   Location    71 non-null    object
dtypes: int64(4), object(4)
memory usage: 4.6+ KB
```

```
df.size # size of the data
```

568

```
df.shape# number of rows and cols
```

(71, 8)

```
#2 # cheaking the null value
df.isnull().sum()
```

```
Car ID      0
Brand       0
Model       0
Year        0
Color       0
Mileage     0
Price       0
Location    0
dtype: int64
```

```
df['Color'] = df['Color'].str.replace('paint','')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 71 entries, 0 to 70
```

```
Data columns (total 9 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Car ID       71 non-null      int64
1   Brand        71 non-null      object
2   Model        71 non-null      object
3   Year         71 non-null      int64
4   Color        71 non-null      object
5   Mileage      71 non-null      int64
6   Price        71 non-null      int64
7   Location     71 non-null      object
8   Color        71 non-null      object
dtypes: int64(4), object(5)
memory usage: 5.1+ KB
```

### #3 PLOTTING THE GRAPH OF PRICE OF CARS

```
import seaborn as sns
sns.distplot(df['Price'])
```

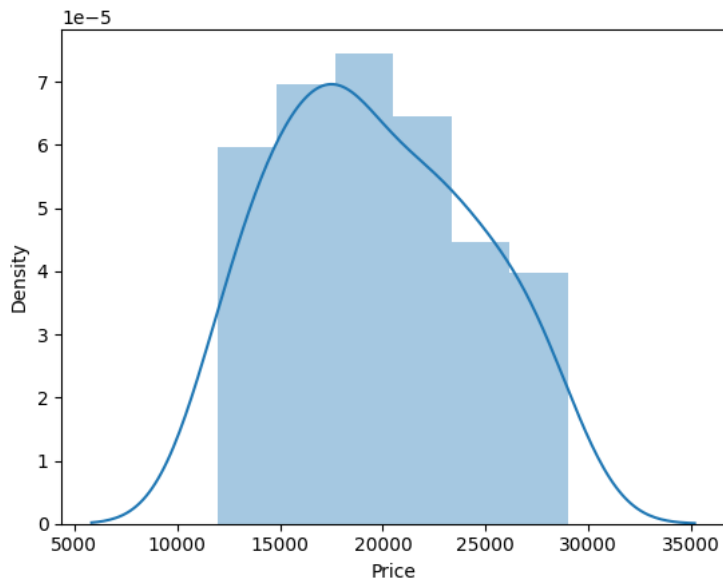
<ipython-input-98-b020f02619da>:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(df['Price'])
<Axes: xlabel='Price', ylabel='Density'>
```



### #4 COUNTING THE BRANDS

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

```
Toyota      15
Honda       14
Ford        14
Chevrolet   14
Hyundai     14
Name: Brand, dtype: int64
```

Double-click (or enter) to edit

### #5 GRAPH OF CAR BRANDS

```
df['Brand'].value_counts().plot(kind='bar')
```

Age Group	Number of People
10-14	15
15-19	14
20-24	14
25-29	14
30-34	14

Car Model	Number of Cars Sold (in millions)
Fusion	3.0
Civic	3.0
Camry	2.0
Pilot	2.0
Yaris	2.0
Venue	2.0
Malibu	2.0
Fit	2.0
Sienna	2.0
Palisade	2.0
Tahoe	2.0
Edge	2.0
Odyssey	2.0
Santa Fe	2.0
Traverse	2.0
Explorer	2.0
4Runner	2.0
Cruze	2.0
Sonata	2.0
Elantra	2.0
Corolla	2.0
Accord	2.0
Impala	2.0
Mustang	2.0
RAV4	2.0
CR-V	2.0
Escape	2.0
Equinox	2.0
Accent	1.0
Avalon	1.0
Genesis	1.0
Camaro	1.0
Fiesta	1.0
Prius	1.0
Highlander	1.0
Spark	1.0
EcoSport	1.0
HR-V	1.0
Focus	1.0
Kona	1.0
Tucson	1.0

A bar chart comparing the average prices of five car brands. The y-axis, labeled 'Price', ranges from 0 to 20,000. The x-axis, labeled 'Brand', lists Toyota, Honda, Ford, Chevrolet, and Hyundai. Each bar is a different color and includes a vertical error bar representing the range of prices.

Brand	Average Price (approx.)
Toyota	20,500
Honda	18,800
Ford	18,900
Chevrolet	20,200
Hyundai	19,500

Fusion	3
Civic	3

```

Camry      2
Pilot      2
Yaris      2
Venue      2
Malibu     2
Fit        2
Sienna     2
Palisade   2
Tahoe      2
Edge       2
Odyssey    2
Santa Fe   2
Traverse   2
Explorer    2
4Runner    2
Cruze      2
Sonata     2
Elantra    2
Corolla    2
Accord     2
Impala     2
Mustang    2
Rav4       2
CR-V       2
Escape     2
Equinox    2
Accent     1
Avalon     1
Genesis    1
Camaro     1
Fiesta     1
Prius      1
Highlander 1
Spark      1
EcoSport   1
HR-V       1
Focus      1
Kona       1
Tucson     1
Name: Model, dtype: int64

```

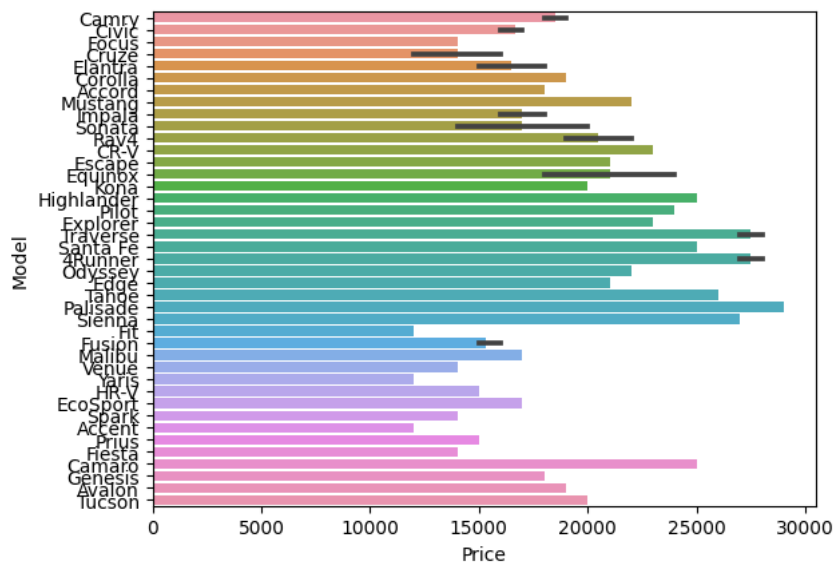
#9 want to find which most expensive model in car

```

import matplotlib.pyplot as plt
sns.barplot(x = df['Price'], y = df['Model'])
plt.xticks

```

<function matplotlib.pyplot.xticks(ticks=None, labels=None, \*, minor=False, \*\*kwargs)>



#10 want to find common price in car

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

```

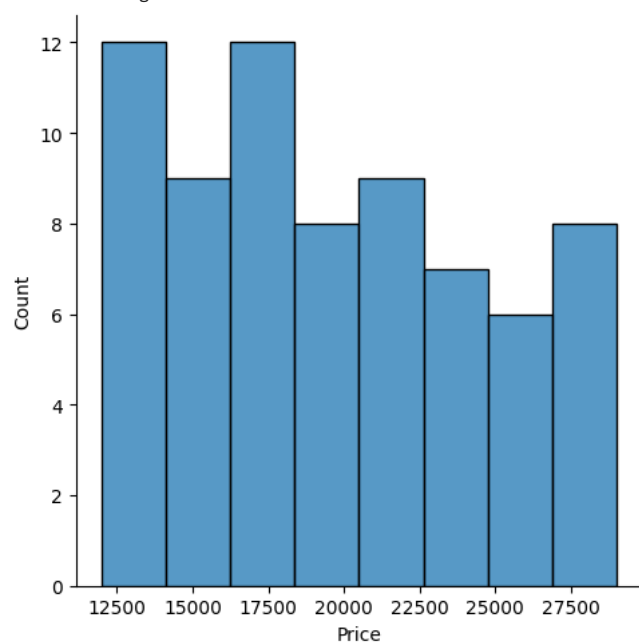
18000    7
14000    6
12000    6
19000    5
22000    5
17000    5
15000    5
23000    4
16000    4
25000    4
27000    4
21000    4

```

```
20000    3
24000    3
28000    2
26000    2
29000    2
Name: Price, dtype: int64
```

```
#11 PLOTTING COMMON PRICE OF CARS
sns.displot(df['Price'])
```

<seaborn.axisgrid.FacetGrid at 0x7a9de7aaa590>



```
# 12 want to know which Location have most counts
df['Location'].value_counts()
```

```
Los Angeles    8
New York       7
Chicago        7
Miami          7
San Francisco  7
Dallas         7
Atlanta        7
Phoenix        7
Houston        7
Seattle        7
Name: Location, dtype: int64
```

```
#13 GRAPH OF LOCATION
sns.displot(df['Location'])
```

<seaborn.axisgrid.FacetGrid at 0x7a9de85168f0>

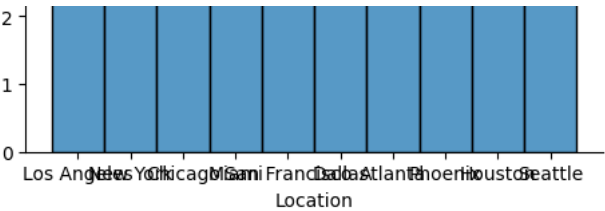
```
#14 want to know most expensive car brand
df[['Brand', 'Model', 'Price']].max()
```

Brand Toyota  
Model Yaris  
Price 29000  
dtype: object



```
# 15 want to know which model car has most expensive
df[['Price', 'Model', 'Location']].max()
```

Price 29000  
Model Yaris  
dtype: object



```
# NAME:- RISHIKESH MAHESH JAMADAR
# COLLAGE:- WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR
# STD:- THIRD YEAR
# BRANCH :- ENTC(ELECTRONICS & TELECOMMUNICATION)
# TITLE :- WRITE A CODE FOR CHEAKER BORD USING FOR LOOP
```

```
import numpy as np
import cv2
```

```
# Checkerboard size
rows, cols = 8, 8
```

```
# Checkerboard square size
square_size = 100
```

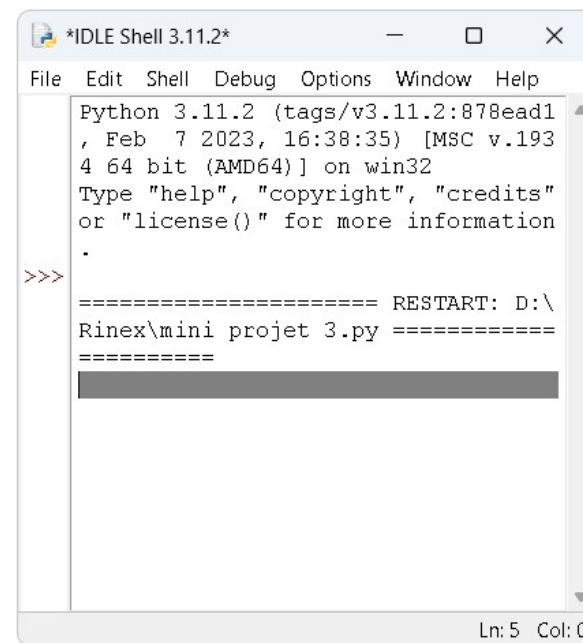
```
# Create an empty checkerboard matrix
checkerboard = np.zeros((rows * square_size, cols * square_size), dtype=np.uint8)
```

```
# Fill the checkerboard matrix with alternating black and white squares
```

```
for i in range(rows):
    for j in range(cols):
        if (i + j) % 2 == 0:
            checkerboard[i * square_size:(i + 1) * square_size, j * square_size:(j + 1) * square_size] = 255
```

```
# Create a window to display the checkerboard
cv2.namedWindow('Checkerboard', cv2.WINDOW_NORMAL)
```

```
# Display the checkerboard
cv2.imshow('Checkerboard', checkerboard)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



```
Python 3.11.2 (tags/v3.11.2:878ead1
, Feb 7 2023, 16:38:35) [MSC v.193
4 64 bit (AMD64)] on win32
Type "help", "copyright", "credits"
or "license()" for more information
.>>>
===== RESTART: D:\
Rinex\mini projet 3.py =====
=====
Ln: 5 Col: 0
```

