② programs involving pandas, Numpy and Scipy libraries

pip => package installer for python.

[1] pip install pandas

pandas
└→ python data analysis data
↓
used for working with datasets
└→ it has functions for ① analyzing
② cleaning
③ Exploring
④ manipulating data

=> pandas is python library. pandas is used to analyze data.
pandas is a python library used for working with data sets.
it has function for analyzing, cleaning, exploring & manipulating data.

[2]: import pandas as pd → alias - alternate name for refering to some think
[3]: pd.__version__ → displays the version of python

=> A pandas series is like a column in a table. it is a one-dimensional array holding data of any type.

labels if nothing else is specified, the values are labeled with their index number.

first value has index 0, Second value has index 1 etc.., This label can be used to access a specified value.

[illegible]

series = pd.series(["BMW", "Toyota", "Honda"])

series

```
0      BMW
1      Toyota
2      Honda
dtype: object
```

#create labels with the index argument, you can name your own labels

sinha = pd.Series(["BMW", "Toyota", "Honda"], index=["first", "Second", "third"])

Sinha

```
first    BMW
Second   Toyota
third    Honda
dtype: Object
```

#access individual element using index

Sinha["second"]

'Toyota'

#series-1 dimensional

colors = pd.Series(["Red", "Blue", "white"])

colors

```
0    Red
1    Blue
2    White
dtype:Object
```

```
# 2 data types in pandas
series = pd.series(["BMW", "Toyota", "Honda"])
series
```

o/p

```
    BMW
0
    Toyota
1
    Honda
2
dtype: object
```

create labels with the index argument, you can name your own labels

```
. Sindex = pd. Series (["BMW", "Toyota", "Honda"], index = ["first", "second", "third"])
Sindex.
```

o/p

```
first    BMW
second   Toyota
third    Honda
dtype: object
```

```
#access individual element using index
Sindex ["second"]
```

o/p

```
'Toyota'
```

```
#series-1 dimensional
colors = pd.Series (["Red", "Blue", "white"])
colors
```

o/p

```
0   Red
1   Blue
2   White
dtype: object
```

```
# Dataframe = 2-dimensional
# row - axis = 0
# column - axis = 1
```

=> Dataframes Datasets in pandas are usually multi-dimensional tables, called Dataframes.

Series is like a column, a Dataframe is the whole table.

- car_data = pd.DataFrame ({"car make": series, "color": colors})
  car_data

O/P

```
      carmake    color
  0    BMW       Red
  1    Toyota    Blue
  2    Honda     white
```

```
#import data
car_sales = pd.read_csv ("car-sales.csv")
car_sales
```

table : 1

O/P

```
     make                colour   Odometer (KM)   Doors    price
 0   Toyota              white    150043          4        $ 4,000.00
 1   Honda               Red      87899           4        $ 5,000,00
 2   ~~BMW~~ Toyota      Black    32549           3        $ 7,000.00
     BMW
 3   Nissan              Black    11179           5        $ 22,000.00
 4   Toyota              white    213095          4        $ 3,500.00
 5   Toyota              Green    99213           4        $ 4,500.00
 6   Honda               Blue     45698           4        $ 7,500.00
 7   Honda               Blue     54738           4        $ 7,000.00
 8   Toyota              white    60000           4        $ 6,250.00
 9   Nissan              white    31600           4        $ 9,700.00
```

#export dataframe

```
car-sales.to_csv("export_cars.csv", index=False)

export-cars=pd.read_csv(" export_cars.csv")

export-cars.
```

o/p

table:1

-
```
heart-disease = pd.redd_csv(" url:  ")
heart-disease
```

o/p

table:

303 rows x 14 columns

* describe data

```
export_cars.describe()
```

→ used to view some basic statistical details like percentile, mean, std etc. of a data frame or a N series of numeric values.

o/p

```
        Odometer(KM)      Doors

count   10.000000         10.000000

mean    78601.400000      4.000000

std     61983.471735      0.471405

min     111179.000000     3.000000

25%     35836.250000      4.000000

50%     57369.000000      4.000000

75%     96384.5000        4.000000

max     213095.000000     5.000000
```

- export_cars.dtypes → data types of data in table

O/P

```
make              object
color             object
Odometer (KM)     ~~object~~ int64
Doors             int64
price             object
dtype: object
```

- export_cars.shape

O/P

```
(10, 5)
```

- export_cars.info()

O/P

```
<class 'pandas.core.frame.Dataframe'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 5 columns):
 #    Column          Non-Null        Dtype
---   --------        ----------      ------
 0    Make            10 non-null     Object
 1    color           10 non-null     Object
 2    odometer(KM)    10 non-null     int64
 3    Doors           10 non-null     int64
 4    price           10 non-null     object
dtypes: int64 (2) , object (3)
memory usage: 532.0+ bytes.
```

- export_cars.columns

o/p

Index(['make', 'color', 'Odometer(KM)', 'Doors', 'price'], dtype=object)

- export-cars.index → used to retrieve the row labels or index labels of data frame

o/p

RangeIndex(start=0, stop=10, step=1)

start → Start index value

stop → stop index value

step → step size

0 to 9

- export-cars.describe

- car-sales.describe()

car-sales.info()

car-sales([“odometer (KM)”, “Doors”]].mean()

→ list of columns passed as index

→ calculates means of each column

o/p

Odometer (KM)   78601.4

Doors   4.0

o/p

dtype: float64

- car-sales.sum()

o/p

- car sales[“Doors”].sum()

o/p

40

- len(car-sales) → returns no of rows in a data frame

o/p

10

- car-sales.head ( ) → print the top 5 rows in table.

car-sales.tail( ) → print bottom 5 rows in table

- car-sales.loc[9] → is used to access group of rows and columns by labels if there is no row label

o/p

```
make             Nissan
colour           white
odometer (KM)    31600
Doors            4
price            $9,700.00
Name: 9, dtype: object
```

- car-sales.iloc[2] → access row by integer index

loc. iloc

- animals=pd.Series(["cat", "puppy", "dog"], index=[0, 1, 2])

animal.loc[0]

o/p

```
'cat'
```

- animals.iloc[0]

o/p

```
'cat'
```

- fruits=pd.Series(["apple", "banana", "mango", "berry"], index[0,3,4,3

fruits

```
0    apple
3    banana
4    mango
3    berry
dtype: object
```

- fruits.iloc[2] → considers actual index

'mango'

fruits.iloc[2] → searches based on labels

#

pip install matplotlib → popular plotting lib in python

import matplotlib.pyplot as plt → provides varity of plots
↳ Graphs
— histogram

pyplot → module for plots

car-sales

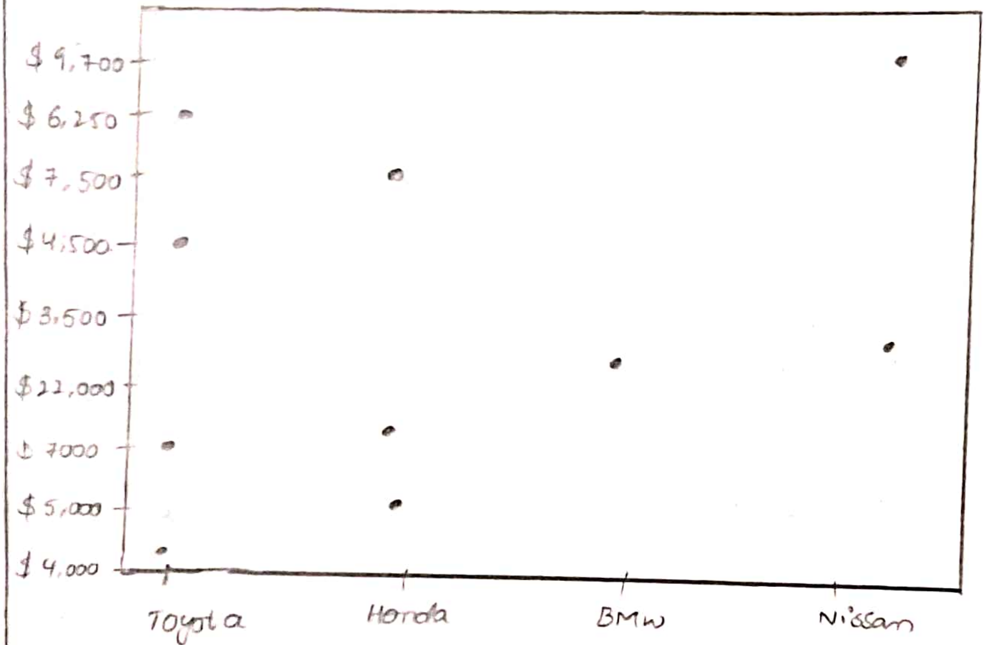
=> pandas uses the plot() method to create diagrams

we can use pyplot, a submodule of the matplotlib library to visualize the diagram on the screen. Specifies that you want a scatter plot with the kind argument

- car-sales.plot (kind = 'scatter', x = 'make', Y = 'price')

O/P

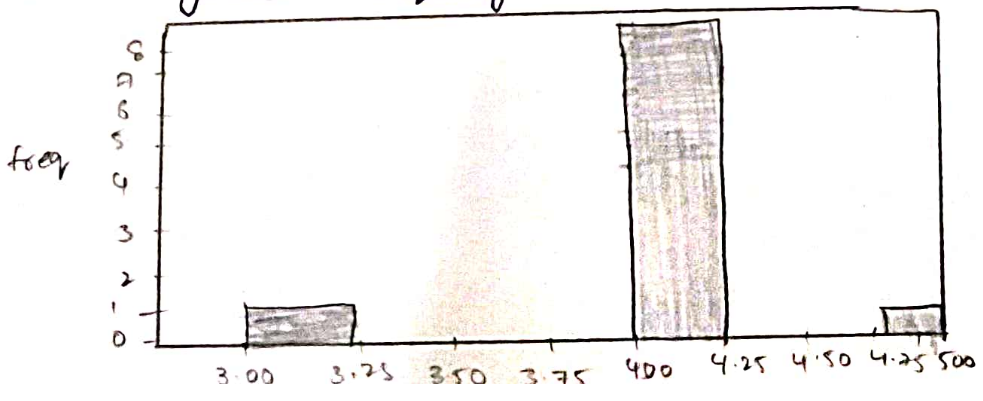
```
<Axes: xlabel = 'Make' , Ylabel = 'price'>
```



- plt. show ()
- car-sales ["Doors"]. plot (kind = 'hist')

O/P

```
<Axes: ylabel = 'frequency'
```


freq