Pandas:

- Pandas is a one library for reading the data and process the data and we can analyze the data
- It is one of the best library for ML and datascience
- Pandas can be explore into two types:
  - Series : it is a 1-D array
  - DataFrames : it is a 2-D array

DataFrames

In [2]:
```
# Create a DataFrame

import pandas as pd
import numpy as np

df=pd.DataFrame()
type(df)
```

Out[2]:
```
pandas.core.frame.DataFrame
```

In [3]:
```
# pass the data into dataframe
l=[1,2,3,4,5,6,7,8,9]
df=pd.DataFrame(l)
df
```

...

In [4]:
```
l=[[1,2],[3,4],[5,6]]
df=pd.DataFrame(l)
df
```

Out[4]:
```
   0  1
0  1  2
1  3  4
2  5  6
```

In [5]:
```
l=[[1,2],[3,4],[5,6]]
df=pd.DataFrame(l,columns=["one","two"])
df
```

Out[5]:
```
   one  two
0    1    2
1    3    4
2    5    6
```

In [10]:
```
# pass the data into dataframe using numpy
a=np.array(["Haritha","Vekata sai","pavan kumar","sravya"])
df=pd.DataFrame(a,columns=['Name'])
df
```

Out[10]:
```
          Name
0      Haritha
1   Vekata sai
2  pavan kumar
3       sravya
```

In [12]:
```
# pass the data into dataframe using list of tuples
l=[("sivaram krishna",30),("one",12),("myname",45)]
df=pd.DataFrame(l)
df.columns=["name","marks"]
df
```

Out[12]:
```
              name  marks
0  sivaram krishna     30
1              one     12
2           myname     45
```

In [14]:

```
# pass the data into dataframe using dictionary
d={"sname":["hello","good","siva","naga","manam"],"age":[90,100,45,32,6]}
df=pd.DataFrame(d)
df.columns=["your name","your age"]
df
```

Out[14]:

	your name	your age
0	hello	90
1	good	100
2	siva	45
3	naga	32
4	manam	6

In [16]:

```
# add a new column
df['marks']=[12,1,3,14,25]
df
```

Out[16]:

	your name	your age	marks
0	hello	90	12
1	good	100	1
2	siva	45	3
3	naga	32	14
4	manam	6	25

In [17]:

```
d={"sno":[101,102,103,104,105],"sname":['a','b','c','d','e'],"sage":[10,12,14,11
df=pd.DataFrame(d)
df
```

Out[17]:

	sno	sname	sage
0	101	a	10
1	102	b	12
2	103	c	14
3	104	d	11
4	105	e	17

In [18]:
```
d={"sno":[101,102,103,104,105],"sname":['a','b','c','d','e'],"sage":[10,12,14,11,
df=pd.DataFrame(d,index=[1,2,3,4,5])
df
```

Out[18]:

	sno	sname	sage
1	101	a	10
2	102	b	12
3	103	c	14
4	104	d	11
5	105	e	17

In [30]:
```
df.set_index('sno',inplace=True)
```

In [31]:
```
df
```

Out[31]:

	sname	sage
sno		
101	a	10
102	b	12
103	c	14
104	d	11
105	e	17

In [32]:
```
df.reset_index(inplace=True)
df
```

Out[32]:

	sno	sname	sage
0	101	a	10
1	102	b	12
2	103	c	14
3	104	d	11
4	105	e	17

How to access the data from DF

In [33]:

```
# get the columns name
df.columns
```

Out[33]:

```
Index(['sno', 'sname', 'sage'], dtype='object')
```

In [34]:

```
# get the index range
df.index
```

Out[34]:

```
RangeIndex(start=0, stop=5, step=1)
```

In [35]:

```
# get the columns values
df['sno']
```

Out[35]:

```
0    101
1    102
2    103
3    104
4    105
Name: sno, dtype: int64
```

In [36]:

```
type(df['sno'])
```

Out[36]:

```
pandas.core.series.Series
```

In [37]:

```
df['sno'].ndim
```

Out[37]:

```
1
```

In [41]:

```
df[['sno','sname']]
#type(df[['sno','sname']])  # type of dataframe
#df[['sno','sname']].ndim   # find the number of dim
```

Out[41]:

	sno	sname
0	101	a
1	102	b
2	103	c
3	104	d
4	105	e

In [43]:
```
df[['sno','sage']]
```

Out[43]:

	sno	sage
0	101	10
1	102	12
2	103	14
3	104	11
4	105	17

In [46]:
```
df[df.columns[1:]]
```

Out[46]:

	sname	sage
0	a	10
1	b	12
2	c	14
3	d	11
4	e	17

In [49]:
```
# access the data using rows

#df[:]
#df[2:]
df[:3]
```

Out[49]:

	sno	sname	sage
0	101	a	10
1	102	b	12
2	103	c	14

In [55]:
```
# access the rows usnig two ways
df
```

Out[55]:

	sno	sname	sage
0	101	a	10
1	102	b	12
2	103	c	14
3	104	d	11
4	105	e	17

In [56]:
```
df.loc[3]
```

Out[56]:
```
sno      104
sname      d
sage      11
Name: 3, dtype: object
```

In [57]:
```
df.loc[:3]
```

Out[57]:

	sno	sname	sage
0	101	a	10
1	102	b	12
2	103	c	14
3	104	d	11

In [58]:
```
df.loc[2:]
```

Out[58]:

	sno	sname	sage
2	103	c	14
3	104	d	11
4	105	e	17

In [60]:
```
df.loc[:,"sname":]
```

Out[60]:

	sname	sage
0	a	10
1	b	12
2	c	14
3	d	11
4	e	17

In [62]:
```
df.loc[:2,:"sname"]
```

Out[62]:

	sno	sname
0	101	a
1	102	b
2	103	c

In [63]:
```
df.loc[:2]
```

Out[63]:
```
   sno  sname  sage
0  101      a    10
1  102      b    12
2  103      c    14
```

In [64]:
```
df.set_index('sname',inplace=True)
```

In [65]:
```
df
```

Out[65]:
```
       sno  sage
sname
    a  101    10
    b  102    12
    c  103    14
    d  104    11
    e  105    17
```

In [67]:
```
df.loc[:'c']
```

Out[67]:
```
       sno  sage
sname
    a  101    10
    b  102    12
    c  103    14
```

In [68]:
```
df.iloc[:2]
```

Out[68]:
```
       sno  sage
sname
    a  101    10
    b  102    12
```

In [69]:

```
df.reset_index(inplace=True)
df
```

Out[69]:

	sname	sno	sage
0	a	101	10
1	b	102	12
2	c	103	14
3	d	104	11
4	e	105	17

In [70]:

```
df.iloc[1:4,:1]
```

Out[70]:

	sname
1	b
2	c
3	d

In [71]:

```
k={'fname':['a','b','c','d','e'],"lname":['sai','ganesh','vijay','raju','ravi'],
df=pd.DataFrame(k)
df
```

Out[71]:

	fname	lname	marks
0	a	sai	31
1	b	ganesh	85
2	c	vijay	23
3	d	raju	20
4	e	ravi	55

In [74]:

```
r=[]
for i in df['marks']:
    if i>0 and i<=25:
        r.append("fail")
    elif i>25 and i<60:
        r.append("second")
    else:
        r.append("first")
df['sresults']=r
df
```

Out[74]:

	fname	lname	marks	sresults
0	a	sai	31	second
1	b	ganesh	85	first
2	c	vijay	23	fail
3	d	raju	20	fail
4	e	ravi	55	second

In [76]:

```
# using single condition

#df['sresults']=='fail'

df[df['sresults']=='fail']
```

Out[76]:

	fname	lname	marks	sresults
2	c	vijay	23	fail
3	d	raju	20	fail

In [78]:

```
df[df['marks']>=50]
```

Out[78]:

	fname	lname	marks	sresults
1	b	ganesh	85	first
4	e	ravi	55	second

In [82]:

```
df[(df['marks']<22) & (df['sresults']=='fail')]
```

Out[82]:

	fname	lname	marks	sresults
3	d	raju	20	fail

read the data into dataframe using files

- csv
    - pd.read_csv("filename.csv")

- excel
    - pd.read_excel("filename.xlsx)
- json
    - pd.read_json("filename.json")
- html
    - pd.read_html("fileurl")

In [83]:

```
# if you want to read the csv files
df=pd.read_csv("weather.csv")
df
```

Out[83]:

	s.no	est	temparature	winspeed	humidity	event	city
0	1	1/2/2019	30	12	12.50	rain	guntur
1	1	1/2/2019	33	14	14.23	rain	vijayawada
2	2	1/3/2019	43	13	43.13	rain	vizag
3	3	1/4/2019	55	8	55.80	fullair	guntur
4	4	1/5/2019	66	10	66.10	cold	vijayawada
5	5	1/6/2019	34	15	34.15	cold	vizag
6	6	1/7/2019	76	17	76.17	fullair	guntur
7	7	1/8/2019	89	23	89.23	cold	vijayawada
8	8	1/9/2019	23	9	23.90	fullair	vizag
9	9	1/10/2019	90	11	90.11	cold	guntur
10	10	1/11/2019	65	18	65.18	fullair	vijayawada
11	11	1/12/2019	78	20	78.20	fullair	vizag

In [84]:

```
df.head()
```

Out[84]:

	s.no	est	temparature	winspeed	humidity	event	city
0	1	1/2/2019	30	12	12.50	rain	guntur
1	1	1/2/2019	33	14	14.23	rain	vijayawada
2	2	1/3/2019	43	13	43.13	rain	vizag
3	3	1/4/2019	55	8	55.80	fullair	guntur
4	4	1/5/2019	66	10	66.10	cold	vijayawada

In [85]:

```
df.head(3)
```

Out[85]:

	s.no	est	temparature	winspeed	humidity	event	city
0	1	1/2/2019	30	12	12.50	rain	guntur
1	1	1/2/2019	33	14	14.23	rain	vijayawada
2	2	1/3/2019	43	13	43.13	rain	vizag

In [86]:

```
df.tail()
```

Out[86]:

	s.no	est	temparature	winspeed	humidity	event	city
7	7	1/8/2019	89	23	89.23	cold	vijayawada
8	8	1/9/2019	23	9	23.90	fullair	vizag
9	9	1/10/2019	90	11	90.11	cold	guntur
10	10	1/11/2019	65	18	65.18	fullair	vijayawada
11	11	1/12/2019	78	20	78.20	fullair	vizag

In [87]:

```
df.tail(3)
```

Out[87]:

	s.no	est	temparature	winspeed	humidity	event	city
9	9	1/10/2019	90	11	90.11	cold	guntur
10	10	1/11/2019	65	18	65.18	fullair	vijayawada
11	11	1/12/2019	78	20	78.20	fullair	vizag

In [89]:
```
# conver csv data json

jsondata=df.to_json(orient='records')
jsondata
```

Out[89]:
```
'[{"s.no":1,"est":"1\\/2\\/2019","temparature":30,"winspeed":12,"humidity":12.
5,"event":"rain","city":"guntur"},{"s.no":1,"est":"1\\/2\\/2019","temparature":
33,"winspeed":14,"humidity":14.23,"event":"rain","city":"vijayawada"},{"s.no":
2,"est":"1\\/3\\/2019","temparature":43,"winspeed":13,"humidity":43.13,"even
t":"rain","city":"vizag"},{"s.no":3,"est":"1\\/4\\/2019","temparature":55,"wins
peed":8,"humidity":55.8,"event":"fullair","city":"guntur"},{"s.no":4,"est":"1
\\/5\\/2019","temparature":66,"winspeed":10,"humidity":66.1,"event":"cold","cit
y":"vijayawada"},{"s.no":5,"est":"1\\/6\\/2019","temparature":34,"winspeed":1
5,"humidity":34.15,"event":"cold","city":"vizag"},{"s.no":6,"est":"1\\/7\\/201
9","temparature":76,"winspeed":17,"humidity":76.17,"event":"fullair","city":"gu
ntur"},{"s.no":7,"est":"1\\/8\\/2019","temparature":89,"winspeed":23,"humidit
y":89.23,"event":"cold","city":"vijayawada"},{"s.no":8,"est":"1\\/9\\/2019","te
mparature":23,"winspeed":9,"humidity":23.9,"event":"fullair","city":"vizag"},
{"s.no":9,"est":"1\\/10\\/2019","temparature":90,"winspeed":11,"humidity":90.1
1,"event":"cold","city":"guntur"},{"s.no":10,"est":"1\\/11\\/2019","temparatur
e":65,"winspeed":18,"humidity":65.18,"event":"fullair","city":"vijayawada"},
{"s.no":11,"est":"1\\/12\\/2019","temparature":78,"winspeed":20,"humidity":78.
2,"event":"fullair","city":"vizag"}]'
```

In [90]:
```
# read json data
json_df=pd.read_json(jsondata)
json_df
```

Out[90]:

	city	est	event	humidity	s.no	temparature	winspeed
0	guntur	1/2/2019	rain	12.50	1	30	12
1	vijayawada	1/2/2019	rain	14.23	1	33	14
2	vizag	1/3/2019	rain	43.13	2	43	13
3	guntur	1/4/2019	fullair	55.80	3	55	8
4	vijayawada	1/5/2019	cold	66.10	4	66	10
5	vizag	1/6/2019	cold	34.15	5	34	15
6	guntur	1/7/2019	fullair	76.17	6	76	17
7	vijayawada	1/8/2019	cold	89.23	7	89	23
8	vizag	1/9/2019	fullair	23.90	8	23	9
9	guntur	1/10/2019	cold	90.11	9	90	11
10	vijayawada	1/11/2019	fullair	65.18	10	65	18
11	vizag	1/12/2019	fullair	78.20	11	78	20

In [101]:

```
# read data using html
import requests

u="https://www.worldcoinindex.com/"

curl=requests.get(u)

curl
```

Out[101]:

```
<Response [200]>
```

In [94]:

```
cd=pd.read_html(curl.text)
```

In [96]:

```
type(cd)
```

Out[96]:

```
list
```

In [98]:

```
cd2=cd[0]
```

In [100]:

```
df=cd2.iloc[:,2:]
df.head()
```

Out[100]:

	Name	Ticker	Last price	%	24 high	24 low	Price Charts 7d	24 volume	# Coins	Market cap
0	Bitcoin	BTC	$ 10,845	-1.02%	$ 11,046	$ 10,802	NaN	$ 5.33B	18.49M	$ 200.55B
1	Ethereum	ETH	$ 380.53	+4.09%	$ 384.68	$ 364.35	NaN	$ 4.03B	112.63M	$ 42.86B
2	Tron	TRX	$ 0.027113	+3.10%	$ 0.027537	$ 0.026288	NaN	$ 739.00M	66.68B	$ 1.80B
3	Chainlink	LINK	$ 10.82	+0.57%	$ 11.70	$ 10.53	NaN	$ 684.96M	350.00M	$ 3.78B
4	Uniswap	UNI	$ 2.58	-31.12%	$ 3.84	$ 2.50	NaN	$ 654.11M	NaN	NaN

In [ ]:

```

```