In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
dict1 = {
    "name" :['harry', 'rohan', 'skillf' ,'shub'],
    "marks":[92, 34, 24, 17],
    "city":['rampur', 'kolkata', 'bareilly', 'kasrawad']
}
```

In [3]:
```
df = pd.DataFrame(dict1)
```

In [4]:
```
df
```

Out[4]:

	name	marks	city
0	harry	92	rampur
1	rohan	34	kolkata
2	skillf	24	bareilly
3	shub	17	kasrawad

if we want an excel sheet which contain above info

In [5]:
```
df.to_csv('friends.csv')
```

if we donot want index to be saved in csv file then give index='false'

In [6]:
```
df.to_csv('friends_indFalse.csv', index=False)
```

if we want to sea start some rows use head(no.of rows)

In [7]:
```
df.head(2)
```

Out[7]:

	name	marks	city
0	harry	92	rampur
1	rohan	34	kolkata

In [8]:
```
df.tail(2)
```

Out[8]:

	name	marks	city
2	skillf	24	bareilly
3	shub	17	kasrawad

In [9]:
```
df.describe
```

Out[9]:
```
<bound method NDFrame.describe of      name  marks      city
0   harry     92    rampur
1   rohan     34   kolkata
2  skillf     24  bareilly
3    shub     17  kasrawad>
```

In [10]:
```
df.describe()
```

Out[10]:

	marks
count	4.00000
mean	41.75000
std	34.21866
min	17.00000
25%	22.25000
50%	29.00000
75%	48.50000
max	92.00000

how to read file csv

In [11]:
```
z = pd.read_csv('arshad.csv')
```

In [12]:
```
z
```

Out[12]:

	Unnamed: 0	name	marks	city	place
0	0	harry	92	rampur	karnataka
1	1	rohan	34	kolkata	delhi
2	2	skillf	24	bareilly	kanpur
3	3	shub	17	kasrawad	dhlokpur

In [13]:
```
z.describe()
```

Out[13]:

	Unnamed: 0	marks
count	4.000000	4.00000
mean	1.500000	41.75000
std	1.290994	34.21866
min	0.000000	17.00000
25%	0.750000	22.25000
50%	1.500000	29.00000
75%	2.250000	48.50000
max	3.000000	92.00000

In [14]:
```
z['city']
```

Out[14]:
```
0      rampur
1     kolkata
2    bareilly
3    kasrawad
Name: city, dtype: object
```

In [15]:
```
z['city'][0]
```

Out[15]:
```
'rampur'
```

In [16]:
```
z.index = ['first','second','third','fourth']
```

In [17]:
```
z
```

Out[17]:

	Unnamed: 0	name	marks	city	place
first	0	harry	92	rampur	karnataka
second	1	rohan	34	kolkata	delhi
third	2	skillf	24	bareilly	kanpur
fourth	3	shub	17	kasrawad	dhlokpur

pandas is an open source data analysis library written in python

it provides rich and highly robust data operation

**Pandas the 2 types of data structure a) series--1D array b) Dataframe-- tabular spreadsheet like structure

In [18]:
```
ser =pd.Series(np.random.rand(34))
```

In [19]:
```
ser
```

Out[19]:
```
0     0.659948
1     0.257416
2     0.839821
3     0.795284
4     0.018219
5     0.255862
6     0.809243
7     0.775737
8     0.602007
9     0.569659
10    0.242806
11    0.154368
12    0.950697
13    0.515644
14    0.788616
15    0.377165
16    0.493464
17    0.436935
18    0.728692
19    0.456798
20    0.369801
21    0.301072
22    0.495114
23    0.966958
24    0.142774
25    0.107962
26    0.822321
27    0.868726
28    0.779655
29    0.228320
30    0.197103
31    0.210602
32    0.372436
33    0.498191
dtype: float64
```

In [20]:
```
type(ser)
```

Out[20]:
```
pandas.core.series.Series
```

In [21]:
```
newdf= pd.DataFrame(np.random.rand(334,5) , index=np.arange(334))
```

In [22]:
```
newdf.head()
```

Out[22]:

	0	1	2	3	4
0	0.981387	0.047178	0.410469	0.906853	0.569243
1	0.016836	0.734050	0.126945	0.263769	0.488318
2	0.387189	0.012013	0.944811	0.358023	0.603031
3	0.111808	0.556741	0.320828	0.609528	0.088663
4	0.626778	0.604772	0.422851	0.016174	0.726360

In [23]:
```
type(newdf)
```

Out[23]:
```
pandas.core.frame.DataFrame
```

In [24]:
```
newdf.describe()
```

Out[24]:

	0	1	2	3	4
count	334.000000	334.000000	334.000000	334.000000	334.000000
mean	0.498764	0.501234	0.515180	0.503483	0.500053
std	0.291968	0.282288	0.296740	0.283687	0.300982
min	0.000752	0.003625	0.000174	0.000526	0.001896
25%	0.241529	0.280933	0.251925	0.253257	0.242412
50%	0.507454	0.505142	0.515287	0.511504	0.495729
75%	0.748460	0.730432	0.774709	0.740317	0.771569
max	0.994820	0.998786	0.998571	0.999933	0.999906

In [25]:
```
newdf.dtypes
```

Out[25]:
```
0    float64
1    float64
2    float64
3    float64
4    float64
dtype: object
```

In [26]:
```
newdf.index
```

Out[26]:
```
Index([  0,   1,   2,   3,   4,   5,   6,   7,   8,   9,
       ...
       324, 325, 326, 327, 328, 329, 330, 331, 332, 333],
      dtype='int32', length=334)
```

In [27]:
```
newdf.columns
```

Out[27]:
```
RangeIndex(start=0, stop=5, step=1)
```

**To create datafram to numpy array

In [28]:
```
newdf.to_numpy()
```

Out[28]:
```
array([[9.81386614e-01, 4.71775852e-02, 4.10468587e-01, 9.06853182e-01,
        5.69243408e-01],
       [1.68356788e-02, 7.34049632e-01, 1.26944839e-01, 2.63768796e-01,
        4.88317778e-01],
       [3.87189485e-01, 1.20133733e-02, 9.44810570e-01, 3.58022988e-01,
        6.03031457e-01],
       ...,
       [8.08197834e-01, 9.53497436e-01, 7.59403969e-01, 5.41219961e-01,
        4.45339486e-01],
       [4.58430484e-01, 5.66239485e-01, 7.93475485e-01, 5.25988066e-04,
        9.58334084e-01],
       [8.60946424e-01, 2.12859356e-01, 6.23613951e-01, 9.40913591e-01,
        6.17476565e-01]])
```

Transpose the given matrix

In [29]:
```
newdf.T
```

Out[29]:

	0	1	2	3	4	5	6	7	8	9	...	324	325	326	327
0	0.981387	0.016836	0.387189	0.111808	0.626778	0.933420	0.092981	0.244177	0.541221	0.724183	...	0.958799	0.244848	0.857413	0.041146
1	0.047178	0.734050	0.012013	0.556741	0.604772	0.915104	0.813800	0.417075	0.328432	0.005997	...	0.668989	0.065161	0.966726	0.600579
2	0.410469	0.126945	0.944811	0.320828	0.422851	0.568491	0.085199	0.322943	0.274705	0.998571	...	0.231979	0.951983	0.325132	0.323497
3	0.906853	0.263769	0.358023	0.609528	0.016174	0.379565	0.986276	0.057308	0.425801	0.508729	...	0.564567	0.389500	0.722942	0.777024
4	0.569243	0.488318	0.603031	0.088663	0.726360	0.908891	0.675064	0.086660	0.262430	0.872643	...	0.927983	0.890916	0.177547	0.401692

5 rows × 334 columns

In [30]:
```
newdf.sort_index(axis=0)
```

Out[30]:

	0	1	2	3	4
0	0.981387	0.047178	0.410469	0.906853	0.569243
1	0.016836	0.734050	0.126945	0.263769	0.488318
2	0.387189	0.012013	0.944811	0.358023	0.603031
3	0.111808	0.556741	0.320828	0.609528	0.088663
4	0.626778	0.604772	0.422851	0.016174	0.726360
...	...	...	...	...	...
329	0.511623	0.155061	0.402782	0.758123	0.063201
330	0.021982	0.909521	0.208313	0.609759	0.189407
331	0.808198	0.953497	0.759404	0.541220	0.445339
332	0.458430	0.566239	0.793475	0.000526	0.958334
333	0.860946	0.212859	0.623614	0.940914	0.617477

334 rows × 5 columns

In [31]:
```
newdf.sort_index(axis=0, ascending=False)
```

Out[31]:

	0	1	2	3	4
333	0.860946	0.212859	0.623614	0.940914	0.617477
332	0.458430	0.566239	0.793475	0.000526	0.958334
331	0.808198	0.953497	0.759404	0.541220	0.445339
330	0.021982	0.909521	0.208313	0.609759	0.189407
329	0.511623	0.155061	0.402782	0.758123	0.063201
...	...	...	...	...	...
4	0.626778	0.604772	0.422851	0.016174	0.726360
3	0.111808	0.556741	0.320828	0.609528	0.088663
2	0.387189	0.012013	0.944811	0.358023	0.603031
1	0.016836	0.734050	0.126945	0.263769	0.488318
0	0.981387	0.047178	0.410469	0.906853	0.569243

334 rows × 5 columns

In [32]:
```
newdf.sort_index(axis=1)
```

Out[32]:

	0	1	2	3	4
0	0.981387	0.047178	0.410469	0.906853	0.569243
1	0.016836	0.734050	0.126945	0.263769	0.488318
2	0.387189	0.012013	0.944811	0.358023	0.603031
3	0.111808	0.556741	0.320828	0.609528	0.088663
4	0.626778	0.604772	0.422851	0.016174	0.726360
...	...	...	...	...	...
329	0.511623	0.155061	0.402782	0.758123	0.063201
330	0.021982	0.909521	0.208313	0.609759	0.189407
331	0.808198	0.953497	0.759404	0.541220	0.445339
332	0.458430	0.566239	0.793475	0.000526	0.958334
333	0.860946	0.212859	0.623614	0.940914	0.617477

334 rows × 5 columns

In [33]:
```
newdf.head()
```

Out[33]:

	0	1	2	3	4
0	0.981387	0.047178	0.410469	0.906853	0.569243
1	0.016836	0.734050	0.126945	0.263769	0.488318
2	0.387189	0.012013	0.944811	0.358023	0.603031
3	0.111808	0.556741	0.320828	0.609528	0.088663
4	0.626778	0.604772	0.422851	0.016174	0.726360

In [34]:
```
newdf[0]
```

Out[34]:
```
0      0.981387
1      0.016836
2      0.387189
3      0.111808
4      0.626778
         ...
329    0.511623
330    0.021982
331    0.808198
332    0.458430
333    0.860946
Name: 0, Length: 334, dtype: float64
```

In [35]:
```
type(newdf)
```

Out[35]:
```
pandas.core.frame.DataFrame
```

In [36]:
```
type(newdf[0])
```

Out[36]:
```
pandas.core.series.Series
```

newdf2 = newdf i.e we have created a view of newdf if we change in newdf2 then it will change in newdf also

**To create copy we use newdf2 = newdf.copy()

In [37]:
```
newdf
```

Out[37]:

	0	1	2	3	4
0	0.981387	0.047178	0.410469	0.906853	0.569243
1	0.016836	0.734050	0.126945	0.263769	0.488318
2	0.387189	0.012013	0.944811	0.358023	0.603031
3	0.111808	0.556741	0.320828	0.609528	0.088663
4	0.626778	0.604772	0.422851	0.016174	0.726360
...	...	...	...	...	...
329	0.511623	0.155061	0.402782	0.758123	0.063201
330	0.021982	0.909521	0.208313	0.609759	0.189407
331	0.808198	0.953497	0.759404	0.541220	0.445339
332	0.458430	0.566239	0.793475	0.000526	0.958334
333	0.860946	0.212859	0.623614	0.940914	0.617477

334 rows × 5 columns

In [38]:
```
newdf.drop(0, axis=1)
```

Out[38]:

	1	2	3	4
0	0.047178	0.410469	0.906853	0.569243
1	0.734050	0.126945	0.263769	0.488318
2	0.012013	0.944811	0.358023	0.603031
3	0.556741	0.320828	0.609528	0.088663
4	0.604772	0.422851	0.016174	0.726360
...	...	...	...	...
329	0.155061	0.402782	0.758123	0.063201
330	0.909521	0.208313	0.609759	0.189407
331	0.953497	0.759404	0.541220	0.445339
332	0.566239	0.793475	0.000526	0.958334
333	0.212859	0.623614	0.940914	0.617477

334 rows × 4 columns

In [39]:
```
newdf.loc[[1,2],[3,4]]
```

Out[39]:

	3	4
1	0.263769	0.488318
2	0.358023	0.603031

In [40]:
```
newdf.loc[[1,2], :]
```

Out[40]:

	0	1	2	3	4
1	0.016836	0.734050	0.126945	0.263769	0.488318
2	0.387189	0.012013	0.944811	0.358023	0.603031

In [41]:
```
newdf.loc[(newdf[2]<0.3)]
```

Out[41]:

	0	1	2	3	4
1	0.016836	0.734050	0.126945	0.263769	0.488318
6	0.092981	0.813800	0.085199	0.986276	0.675064
8	0.541221	0.328432	0.274705	0.425801	0.262430
10	0.730977	0.908674	0.209980	0.762736	0.924916
11	0.573006	0.792138	0.180557	0.869337	0.630173
...	...	...	...	...	...
314	0.251282	0.479647	0.253538	0.473303	0.695182
315	0.638531	0.348324	0.225180	0.214779	0.437052
317	0.613529	0.678587	0.225373	0.313584	0.001896
324	0.958799	0.668989	0.231979	0.564567	0.927983
330	0.021982	0.909521	0.208313	0.609759	0.189407

98 rows × 5 columns

In [42]:
```
newdf.loc[ (newdf[2]<0.3) & (newdf[3]>0.1)]
```

Out[42]:

	0	1	2	3	4
1	0.016836	0.734050	0.126945	0.263769	0.488318
6	0.092981	0.813800	0.085199	0.986276	0.675064
8	0.541221	0.328432	0.274705	0.425801	0.262430
10	0.730977	0.908674	0.209980	0.762736	0.924916
11	0.573006	0.792138	0.180557	0.869337	0.630173
...	...	...	...	...	...
314	0.251282	0.479647	0.253538	0.473303	0.695182
315	0.638531	0.348324	0.225180	0.214779	0.437052
317	0.613529	0.678587	0.225373	0.313584	0.001896
324	0.958799	0.668989	0.231979	0.564567	0.927983
330	0.021982	0.909521	0.208313	0.609759	0.189407

91 rows × 5 columns

In [43]:
```
newdf.iloc[0,4]
```

Out[43]:
```
0.5692434077437327
```

In [44]:
```
newdf.head(2)
```

Out[44]:

	0	1	2	3	4
0	0.981387	0.047178	0.410469	0.906853	0.569243
1	0.016836	0.734050	0.126945	0.263769	0.488318

delete any row

In [45]:
```
newdf.drop([0] ,axis=1)
```

Out[45]:

	1	2	3	4
0	0.047178	0.410469	0.906853	0.569243
1	0.734050	0.126945	0.263769	0.488318
2	0.012013	0.944811	0.358023	0.603031
3	0.556741	0.320828	0.609528	0.088663
4	0.604772	0.422851	0.016174	0.726360
...	...	...	...	...
329	0.155061	0.402782	0.758123	0.063201
330	0.909521	0.208313	0.609759	0.189407
331	0.953497	0.759404	0.541220	0.445339
332	0.566239	0.793475	0.000526	0.958334
333	0.212859	0.623614	0.940914	0.617477

334 rows × 4 columns

In [46]:
```
newdf
```

Out[46]:

	0	1	2	3	4
0	0.981387	0.047178	0.410469	0.906853	0.569243
1	0.016836	0.734050	0.126945	0.263769	0.488318
2	0.387189	0.012013	0.944811	0.358023	0.603031
3	0.111808	0.556741	0.320828	0.609528	0.088663
4	0.626778	0.604772	0.422851	0.016174	0.726360
...	...	...	...	...	...
329	0.511623	0.155061	0.402782	0.758123	0.063201
330	0.021982	0.909521	0.208313	0.609759	0.189407
331	0.808198	0.953497	0.759404	0.541220	0.445339
332	0.458430	0.566239	0.793475	0.000526	0.958334
333	0.860946	0.212859	0.623614	0.940914	0.617477

334 rows × 5 columns

In [47]:
```
newdf.reset_index()
```

Out[47]:

	index	0	1	2	3	4
0	0	0.981387	0.047178	0.410469	0.906853	0.569243
1	1	0.016836	0.734050	0.126945	0.263769	0.488318
2	2	0.387189	0.012013	0.944811	0.358023	0.603031
3	3	0.111808	0.556741	0.320828	0.609528	0.088663
4	4	0.626778	0.604772	0.422851	0.016174	0.726360
...	...	...	...	...	...	...
329	329	0.511623	0.155061	0.402782	0.758123	0.063201
330	330	0.021982	0.909521	0.208313	0.609759	0.189407
331	331	0.808198	0.953497	0.759404	0.541220	0.445339
332	332	0.458430	0.566239	0.793475	0.000526	0.958334
333	333	0.860946	0.212859	0.623614	0.940914	0.617477

334 rows × 6 columns

In [50]:
```
newdf.reset_index(drop=True)
```

Out[50]:

	0	1	2	3	4
0	0.981387	0.047178	0.410469	0.906853	0.569243
1	0.016836	0.734050	0.126945	0.263769	0.488318
2	0.387189	0.012013	0.944811	0.358023	0.603031
3	0.111808	0.556741	0.320828	0.609528	0.088663
4	0.626778	0.604772	0.422851	0.016174	0.726360
...	...	...	...	...	...
329	0.511623	0.155061	0.402782	0.758123	0.063201
330	0.021982	0.909521	0.208313	0.609759	0.189407
331	0.808198	0.953497	0.759404	0.541220	0.445339
332	0.458430	0.566239	0.793475	0.000526	0.958334
333	0.860946	0.212859	0.623614	0.940914	0.617477

334 rows × 5 columns

In [52]:
```
newdf[3].isnull()
```

Out[52]:
```
0      False
1      False
2      False
3      False
4      False
       ...
329    False
330    False
331    False
332    False
333    False
Name: 3, Length: 334, dtype: bool
```

In [53]:
```
newdf[2]=None
```

In [54]:
```
newdf
```

Out[54]:

	0	1	2	3	4
0	0.981387	0.047178	None	0.906853	0.569243
1	0.016836	0.734050	None	0.263769	0.488318
2	0.387189	0.012013	None	0.358023	0.603031
3	0.111808	0.556741	None	0.609528	0.088663
4	0.626778	0.604772	None	0.016174	0.726360
...	...	...	...	...	...
329	0.511623	0.155061	None	0.758123	0.063201
330	0.021982	0.909521	None	0.609759	0.189407
331	0.808198	0.953497	None	0.541220	0.445339
332	0.458430	0.566239	None	0.000526	0.958334
333	0.860946	0.212859	None	0.940914	0.617477

334 rows × 5 columns

In [56]:
```
newdf[2].isnull()
```

Out[56]:
```
0      True
1      True
2      True
3      True
4      True
       ...
329    True
330    True
331    True
332    True
333    True
Name: 2, Length: 334, dtype: bool
```

In [57]:
```
newdf.loc[:, [2]]=56
```

In [58]:
```
newdf
```

Out[58]:

	0	1	2	3	4
0	0.981387	0.047178	56	0.906853	0.569243
1	0.016836	0.734050	56	0.263769	0.488318
2	0.387189	0.012013	56	0.358023	0.603031
3	0.111808	0.556741	56	0.609528	0.088663
4	0.626778	0.604772	56	0.016174	0.726360
...	...	...	...	...	...
329	0.511623	0.155061	56	0.758123	0.063201
330	0.021982	0.909521	56	0.609759	0.189407
331	0.808198	0.953497	56	0.541220	0.445339
332	0.458430	0.566239	56	0.000526	0.958334
333	0.860946	0.212859	56	0.940914	0.617477

334 rows × 5 columns

for inspect we can use

1)df.head 2)df.tail 3)df.shape 4)df.describe() 5)df.info() 6)df['name'].value_count() 7)df.notnull() 8)df.isnull()

In [ ]:
```

```