

Pandas Notebook

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

```
In [2]: dict1 = {
    "name": ['Nishant', 'Rohan', 'Sunny', 'Mickey'],
    "Age": [25, 35, 36, 26],
    "City": ['Kolkata', 'Mumbai', 'Dehradun', 'Lucknow']
}
```

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

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

```
Out[4]:
```

	name	Age	City
0	Nishant	25	Kolkata
1	Rohan	35	Mumbai
2	Sunny	36	Dehradun
3	Mickey	26	Lucknow

```
In [5]: #How to create a CSV file from Jupyter notebook
```

```
In [6]: df.to_csv('students.csv')
```

```
In [7]: # If do not want to show the index then
```

```
In [8]: df.to_csv('students_index_False.csv', index = False)
```

```
In [9]: df.head(2)
```

```
Out[9]:
```

	name	Age	City
0	Nishant	25	Kolkata
1	Rohan	35	Mumbai

```
In [10]: df.tail(2)
```

```
Out[10]:
```

	name	Age	City
2	Sunny	36	Dehradun
3	Mickey	26	Lucknow

```
In [11]: #To check the statistical analysis
```

```
In [12]: df.describe()
```

Out[12]:

Age	
count	4.000000
mean	30.500000
std	5.802298
min	25.000000
25%	25.750000
50%	30.500000
75%	35.250000
max	36.000000

In [13]: *#If we create CSV file in our local folder for import in jupyter note book*

In [14]: `Nishant = pd.read_csv('Pandas.csv')`

In [15]: `Nishant`

Out[15]:

	Train	Train no	City	Destination
0	Purshottam	2345	Gaya	Mumbai
1	Rajdhani	5467	Mumbai	Puri
2	Nilanchal	8754	Puri	Gwalior
3	Poorva	9009	Dehradun	Gaya
4	Mahabodhi	1020	Gwalior	Pune
5	LTT Exp	7890	Pune	Dehradun

In [16]: `Nishant['City']`

Out[16]:

```
0      Gaya
1     Mumbai
2      Puri
3  Dehradun
4     Gwalior
5      Pune
Name: City, dtype: object
```

In [17]: `Nishant['Destination']`

Out[17]:

```
0      Mumbai
1      Puri
2     Gwalior
3      Gaya
4      Pune
5  Dehradun
Name: Destination, dtype: object
```

In [18]: `Nishant['City'] [0]` *#here 0 is index*

Out[18]: 'Gaya'

In [19]: `Nishant.head()`

```
Out[19]:
```

	Train	Train no	City	Destination
0	Purshottam	2345	Gaya	Mumbai
1	Rajdhani	5467	Mumbai	Puri
2	Nilanchal	8754	Puri	Gwalior
3	Poorva	9009	Dehradun	Gaya
4	Mahabodhi	1020	Gwalior	Pune

```
In [20]: Nishant['City'] [0] = 'Patna' #here we can chage the value
```

C:\Users\Nishant Nandkeolyar\AppData\Local\Temp\ipykernel_8140\3508710218.py:1: 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-a-copy

Nishant['City'] [0] = 'Patna' #here we can chage the value

```
In [21]: Nishant.head()
```

```
Out[21]:
```

	Train	Train no	City	Destination
0	Purshottam	2345	Patna	Mumbai
1	Rajdhani	5467	Mumbai	Puri
2	Nilanchal	8754	Puri	Gwalior
3	Poorva	9009	Dehradun	Gaya
4	Mahabodhi	1020	Gwalior	Pune

```
In [22]: #Create series in pandas
```

```
In [23]: ser = pd.Series(np.random.rand)
```

```
In [24]: ser
```

```
Out[24]: 0    <built-in method rand of numpy.random.mtrand.R...
dtype: object
```

```
In [25]: #If we write 20 here in series
```

```
In [26]: ser = pd.Series(np.random.rand(20))
```

```
In [27]: ser
```

```
Out[27]: 0    0.787451
1    0.419274
2    0.952585
3    0.670536
4    0.643045
5    0.680459
6    0.058115
7    0.100156
8    0.147472
9    0.168076
10   0.593672
11   0.421787
12   0.271467
13   0.426238
14   0.777745
15   0.997814
16   0.457040
17   0.162759
18   0.981348
19   0.700551
dtype: float64
```

```
In [28]: type(ser)
```

```
Out[28]: pandas.core.series.Series
```

```
In [29]: #Create new dataframe
```

```
In [30]: Newdf = pd.DataFrame(np.random.rand(224,10), index = np.arange(224))
```

```
In [31]: Newdf
```

```
Out[31]:
```

	0	1	2	3	4	5	6	7	8	9
0	0.939886	0.786601	0.956868	0.258590	0.334036	0.870225	0.633365	0.816873	0.811798	0.04
1	0.158507	0.520862	0.679760	0.102497	0.113052	0.726117	0.315770	0.719951	0.922452	0.11
2	0.049013	0.828629	0.914315	0.260196	0.926152	0.150249	0.933029	0.913121	0.905099	0.59
3	0.549031	0.540318	0.029625	0.599735	0.378303	0.905322	0.487571	0.247969	0.038808	0.84
4	0.839633	0.086155	0.210866	0.451999	0.392206	0.826961	0.124213	0.996235	0.651352	0.69
...
219	0.310491	0.703000	0.499299	0.127928	0.071682	0.447261	0.841341	0.651170	0.447458	0.00
220	0.808144	0.514275	0.870238	0.803786	0.234934	0.100896	0.001808	0.007777	0.452187	0.70
221	0.228910	0.341982	0.560597	0.175265	0.499093	0.079968	0.665550	0.890360	0.856866	0.28
222	0.266593	0.376427	0.597294	0.504088	0.907032	0.804976	0.171475	0.906328	0.013591	0.85
223	0.861299	0.579981	0.625031	0.427594	0.796474	0.406625	0.553957	0.287599	0.803210	0.70

224 rows × 10 columns

```
In [32]: Newdf.head()
```

```
Out[32]:
```

	0	1	2	3	4	5	6	7	8	
0	0.939886	0.786601	0.956868	0.258590	0.334036	0.870225	0.633365	0.816873	0.811798	0.0417
1	0.158507	0.520862	0.679760	0.102497	0.113052	0.726117	0.315770	0.719951	0.922452	0.1155
2	0.049013	0.828629	0.914315	0.260196	0.926152	0.150249	0.933029	0.913121	0.905099	0.5927
3	0.549031	0.540318	0.029625	0.599735	0.378303	0.905322	0.487571	0.247969	0.038808	0.8447
4	0.839633	0.086155	0.210866	0.451999	0.392206	0.826961	0.124213	0.996235	0.651352	0.6947

```
In [33]: type(Newdf)
```

```
Out[33]: pandas.core.frame.DataFrame
```

```
In [34]: Newdf.describe()
```

```
Out[34]:
```

	0	1	2	3	4	5	6
count	224.000000	224.000000	224.000000	224.000000	224.000000	224.000000	224.000000
mean	0.503642	0.473807	0.507056	0.490418	0.485518	0.487501	0.484080
std	0.289616	0.283514	0.293597	0.285726	0.288842	0.294964	0.294706
min	0.002155	0.007269	0.006454	0.000967	0.000731	0.002260	0.001808
25%	0.257708	0.242987	0.278246	0.251151	0.234016	0.224280	0.226968
50%	0.505333	0.441551	0.503445	0.481457	0.489508	0.484186	0.486299
75%	0.763239	0.715537	0.751853	0.761632	0.745627	0.763284	0.764377
max	0.997724	0.996305	0.999708	0.994333	0.993286	0.985289	0.999669

```
In [35]: #Changes the index
```

```
In [36]: Newdf[0][0] = "Nishant" #so datatype cahges in object, Lets see
```

```
In [37]: Newdf
```

```
Out[37]:
```

	0	1	2	3	4	5	6	7	8	
0	Nishant	0.786601	0.956868	0.258590	0.334036	0.870225	0.633365	0.816873	0.811798	0.04
1	0.158507	0.520862	0.679760	0.102497	0.113052	0.726117	0.315770	0.719951	0.922452	0.11
2	0.049013	0.828629	0.914315	0.260196	0.926152	0.150249	0.933029	0.913121	0.905099	0.59
3	0.549031	0.540318	0.029625	0.599735	0.378303	0.905322	0.487571	0.247969	0.038808	0.84
4	0.839633	0.086155	0.210866	0.451999	0.392206	0.826961	0.124213	0.996235	0.651352	0.69
...
219	0.310491	0.703000	0.499299	0.127928	0.071682	0.447261	0.841341	0.651170	0.447458	0.00
220	0.808144	0.514275	0.870238	0.803786	0.234934	0.100896	0.001808	0.007777	0.452187	0.70
221	0.22891	0.341982	0.560597	0.175265	0.499093	0.079968	0.665550	0.890360	0.856866	0.28
222	0.266593	0.376427	0.597294	0.504088	0.907032	0.804976	0.171475	0.906328	0.013591	0.85
223	0.861299	0.579981	0.625031	0.427594	0.796474	0.406625	0.553957	0.287599	0.803210	0.70

224 rows × 10 columns

```
In [38]: Newdf.dtypes
```

```
Out[38]: 0    object
1    float64
2    float64
3    float64
4    float64
5    float64
6    float64
7    float64
8    float64
9    float64
dtype: object
```

```
In [39]: Newdf.head()
```

```
Out[39]:
```

	0	1	2	3	4	5	6	7	8	
0	Nishant	0.786601	0.956868	0.258590	0.334036	0.870225	0.633365	0.816873	0.811798	0.0417
1	0.158507	0.520862	0.679760	0.102497	0.113052	0.726117	0.315770	0.719951	0.922452	0.1155
2	0.049013	0.828629	0.914315	0.260196	0.926152	0.150249	0.933029	0.913121	0.905099	0.5927
3	0.549031	0.540318	0.029625	0.599735	0.378303	0.905322	0.487571	0.247969	0.038808	0.8447
4	0.839633	0.086155	0.210866	0.451999	0.392206	0.826961	0.124213	0.996235	0.651352	0.6947

```
In [40]: #Now print the Index
```

```
In [41]: Newdf.index
```

```
Out[41]: Index([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9,
...
214, 215, 216, 217, 218, 219, 220, 221, 222, 223],
dtype='int32', length=224)
```

```
In [42]: Newdf.columns
```

```
Out[42]: RangeIndex(start=0, stop=10, step=1)
```

```
In [43]: #Convert into numpy array so
```

```
In [44]: Newdf.to_numpy()
```

```
Out[44]: array([[ 'Nishant', 0.7866014991269611, 0.9568678736250221, ...,  
                0.8168734221269348, 0.8117978719842629, 0.041700052840509705],  
                [0.1585069019691283, 0.5208618005211417, 0.6797602953965162, ...,  
                0.7199508418943447, 0.9224522035191955, 0.11556844304896174],  
                [0.04901297453635789, 0.8286290620417543, 0.914314556056611, ...,  
                0.913120699497796, 0.9050988488250544, 0.5927566694071933],  
                ...,  
                [0.22890973998518493, 0.3419821385616538, 0.5605971558978635, ...,  
                0.8903601099620417, 0.8568661656876306, 0.2851089638952957],  
                [0.2665927143874033, 0.3764270711282032, 0.5972940923681915, ...,  
                0.9063281772361724, 0.013591383770761412, 0.8536079014101885],  
                [0.8612987742824519, 0.5799813868189065, 0.6250309603020344, ...,  
                0.28759877417912927, 0.8032103663995345, 0.7053884032123618]],  
                dtype=object)
```

```
In [45]: #For Transpose
```

```
In [46]: Newdf.T
```

```
Out[46]:
```

	0	1	2	3	4	5	6	7	8	
0	Nishant	0.158507	0.049013	0.549031	0.839633	0.379833	0.957077	0.966062	0.341733	0.7323
1	0.786601	0.520862	0.828629	0.540318	0.086155	0.356152	0.877792	0.992817	0.065424	0.410
2	0.956868	0.67976	0.914315	0.029625	0.210866	0.016603	0.219664	0.880957	0.651852	0.0786
3	0.25859	0.102497	0.260196	0.599735	0.451999	0.217805	0.546625	0.101933	0.285352	0.7239
4	0.334036	0.113052	0.926152	0.378303	0.392206	0.849607	0.757603	0.622176	0.62389	0.5290
5	0.870225	0.726117	0.150249	0.905322	0.826961	0.093775	0.079295	0.066924	0.883058	0.5254
6	0.633365	0.31577	0.933029	0.487571	0.124213	0.574011	0.791152	0.672912	0.49782	0.7995
7	0.816873	0.719951	0.913121	0.247969	0.996235	0.546228	0.850478	0.745784	0.479898	0.3793
8	0.811798	0.922452	0.905099	0.038808	0.651352	0.127571	0.247325	0.244734	0.165445	0.6118
9	0.0417	0.115568	0.592757	0.844798	0.694787	0.978502	0.193767	0.63629	0.265926	0.1062

10 rows × 224 columns

```
In [47]: #Use sort index function
```

```
In [48]: Newdf.sort_index(axis = 0, ascending = False)
```

```
Out[48]:
```

	0	1	2	3	4	5	6	7	8	
223	0.861299	0.579981	0.625031	0.427594	0.796474	0.406625	0.553957	0.287599	0.803210	0.70
222	0.266593	0.376427	0.597294	0.504088	0.907032	0.804976	0.171475	0.906328	0.013591	0.85
221	0.22891	0.341982	0.560597	0.175265	0.499093	0.079968	0.665550	0.890360	0.856866	0.28
220	0.808144	0.514275	0.870238	0.803786	0.234934	0.100896	0.001808	0.007777	0.452187	0.70
219	0.310491	0.703000	0.499299	0.127928	0.071682	0.447261	0.841341	0.651170	0.447458	0.00
...
4	0.839633	0.086155	0.210866	0.451999	0.392206	0.826961	0.124213	0.996235	0.651352	0.69
3	0.549031	0.540318	0.029625	0.599735	0.378303	0.905322	0.487571	0.247969	0.038808	0.84
2	0.049013	0.828629	0.914315	0.260196	0.926152	0.150249	0.933029	0.913121	0.905099	0.59
1	0.158507	0.520862	0.679760	0.102497	0.113052	0.726117	0.315770	0.719951	0.922452	0.11
0	Nishant	0.786601	0.956868	0.258590	0.334036	0.870225	0.633365	0.816873	0.811798	0.04

224 rows × 10 columns

```
In [49]: Newdf.sort_index(axis = 1, ascending = False)
```

```
Out[49]:
```

	9	8	7	6	5	4	3	2	1	
0	0.041700	0.811798	0.816873	0.633365	0.870225	0.334036	0.258590	0.956868	0.786601	Ni
1	0.115568	0.922452	0.719951	0.315770	0.726117	0.113052	0.102497	0.679760	0.520862	0.15
2	0.592757	0.905099	0.913121	0.933029	0.150249	0.926152	0.260196	0.914315	0.828629	0.04
3	0.844798	0.038808	0.247969	0.487571	0.905322	0.378303	0.599735	0.029625	0.540318	0.54
4	0.694787	0.651352	0.996235	0.124213	0.826961	0.392206	0.451999	0.210866	0.086155	0.83
...
219	0.008918	0.447458	0.651170	0.841341	0.447261	0.071682	0.127928	0.499299	0.703000	0.31
220	0.706434	0.452187	0.007777	0.001808	0.100896	0.234934	0.803786	0.870238	0.514275	0.80
221	0.285109	0.856866	0.890360	0.665550	0.079968	0.499093	0.175265	0.560597	0.341982	0.2
222	0.853608	0.013591	0.906328	0.171475	0.804976	0.907032	0.504088	0.597294	0.376427	0.26
223	0.705388	0.803210	0.287599	0.553957	0.406625	0.796474	0.427594	0.625031	0.579981	0.86

224 rows × 10 columns

```
In [50]: Newdf.head()
```



```
Out[50]:
```

	0	1	2	3	4	5	6	7	8	
0	Nishant	0.786601	0.956868	0.258590	0.334036	0.870225	0.633365	0.816873	0.811798	0.0417
1	0.158507	0.520862	0.679760	0.102497	0.113052	0.726117	0.315770	0.719951	0.922452	0.1155
2	0.049013	0.828629	0.914315	0.260196	0.926152	0.150249	0.933029	0.913121	0.905099	0.5927
3	0.549031	0.540318	0.029625	0.599735	0.378303	0.905322	0.487571	0.247969	0.038808	0.8447
4	0.839633	0.086155	0.210866	0.451999	0.392206	0.826961	0.124213	0.996235	0.651352	0.6947

```
In [51]: Newdf[0]
```

```
Out[51]:
```

0	Nishant
1	0.158507
2	0.049013
3	0.549031
4	0.839633
...	
219	0.310491
220	0.808144
221	0.22891
222	0.266593
223	0.861299

Name: 0, Length: 224, dtype: object

```
In [52]: type(Newdf[0]) # for check the Type
```

```
Out[52]: pandas.core.series.Series
```

```
In [53]: type(Newdf[0])
```

```
Out[53]: pandas.core.series.Series
```

```
In [54]: #If we set the 0 in rows and colum so we use here
```

```
In [55]: Newdf.loc[0,0] = 765
```

```
In [56]: Newdf.head()
```

```
Out[56]:
```

	0	1	2	3	4	5	6	7	8	
0	765	0.786601	0.956868	0.258590	0.334036	0.870225	0.633365	0.816873	0.811798	0.0417
1	0.158507	0.520862	0.679760	0.102497	0.113052	0.726117	0.315770	0.719951	0.922452	0.1155
2	0.049013	0.828629	0.914315	0.260196	0.926152	0.150249	0.933029	0.913121	0.905099	0.5927
3	0.549031	0.540318	0.029625	0.599735	0.378303	0.905322	0.487571	0.247969	0.038808	0.8447
4	0.839633	0.086155	0.210866	0.451999	0.392206	0.826961	0.124213	0.996235	0.651352	0.6947

```
In [57]: Newdf.loc[0,'A'] = 76566
```

```
In [58]: Newdf.head()
```

```
Out[58]:
```

	0	1	2	3	4	5	6	7	8	
0	765	0.786601	0.956868	0.258590	0.334036	0.870225	0.633365	0.816873	0.811798	0.0417
1	0.158507	0.520862	0.679760	0.102497	0.113052	0.726117	0.315770	0.719951	0.922452	0.1155
2	0.049013	0.828629	0.914315	0.260196	0.926152	0.150249	0.933029	0.913121	0.905099	0.5927
3	0.549031	0.540318	0.029625	0.599735	0.378303	0.905322	0.487571	0.247969	0.038808	0.8447
4	0.839633	0.086155	0.210866	0.451999	0.392206	0.826961	0.124213	0.996235	0.651352	0.6947

```
In [59]: #Use Drop here for remove the column
```

```
In [60]: Newdf.drop(0, axis = 1)
```

```
Out[60]:
```

	1	2	3	4	5	6	7	8	9	
0	0.786601	0.956868	0.258590	0.334036	0.870225	0.633365	0.816873	0.811798	0.041700	765
1	0.520862	0.679760	0.102497	0.113052	0.726117	0.315770	0.719951	0.922452	0.115568	I
2	0.828629	0.914315	0.260196	0.926152	0.150249	0.933029	0.913121	0.905099	0.592757	I
3	0.540318	0.029625	0.599735	0.378303	0.905322	0.487571	0.247969	0.038808	0.844798	I
4	0.086155	0.210866	0.451999	0.392206	0.826961	0.124213	0.996235	0.651352	0.694787	I
...
219	0.703000	0.499299	0.127928	0.071682	0.447261	0.841341	0.651170	0.447458	0.008918	I
220	0.514275	0.870238	0.803786	0.234934	0.100896	0.001808	0.007777	0.452187	0.706434	I
221	0.341982	0.560597	0.175265	0.499093	0.079968	0.665550	0.890360	0.856866	0.285109	I
222	0.376427	0.597294	0.504088	0.907032	0.804976	0.171475	0.906328	0.013591	0.853608	I
223	0.579981	0.625031	0.427594	0.796474	0.406625	0.553957	0.287599	0.803210	0.705388	I

224 rows × 10 columns

```
In [61]: #axis1 = column  
#axis0 = row
```

```
In [62]: Newdf.loc[[1,2],:]
```

```
Out[62]:
```

	0	1	2	3	4	5	6	7	8	
1	0.158507	0.520862	0.679760	0.102497	0.113052	0.726117	0.315770	0.719951	0.922452	0.1155
2	0.049013	0.828629	0.914315	0.260196	0.926152	0.150249	0.933029	0.913121	0.905099	0.5927

```
In [63]: #Now run complex querries in pandas
```

```
In [64]: Newdf.loc[(Newdf['A']<0.3)]
```

```
Out[64]:
```

0	1	2	3	4	5	6	7	8	9	A
---	---	---	---	---	---	---	---	---	---	---

```
In [65]: Newdf.iloc[0,5]
```

```
Out[65]: 0.8702253013947464
```

```
In [66]: Newdf.iloc[[0,5], [1,2]]
```

```
Out[66]:
```

	1	2
0	0.786601	0.956868
5	0.356152	0.016603

```
In [67]: #If we drop the rows
```

```
In [68]: Newdf.drop([0])
```

```
Out[68]:
```

	0	1	2	3	4	5	6	7	8	
1	0.158507	0.520862	0.679760	0.102497	0.113052	0.726117	0.315770	0.719951	0.922452	0.11
2	0.049013	0.828629	0.914315	0.260196	0.926152	0.150249	0.933029	0.913121	0.905099	0.59
3	0.549031	0.540318	0.029625	0.599735	0.378303	0.905322	0.487571	0.247969	0.038808	0.84
4	0.839633	0.086155	0.210866	0.451999	0.392206	0.826961	0.124213	0.996235	0.651352	0.69
5	0.379833	0.356152	0.016603	0.217805	0.849607	0.093775	0.574011	0.546228	0.127571	0.97
...
219	0.310491	0.703000	0.499299	0.127928	0.071682	0.447261	0.841341	0.651170	0.447458	0.00
220	0.808144	0.514275	0.870238	0.803786	0.234934	0.100896	0.001808	0.007777	0.452187	0.70
221	0.22891	0.341982	0.560597	0.175265	0.499093	0.079968	0.665550	0.890360	0.856866	0.28
222	0.266593	0.376427	0.597294	0.504088	0.907032	0.804976	0.171475	0.906328	0.013591	0.85
223	0.861299	0.579981	0.625031	0.427594	0.796474	0.406625	0.553957	0.287599	0.803210	0.70

223 rows × 11 columns

```
In [69]: #If drop column
```

```
In [70]: Newdf.drop(['A'], axis = 1)  #(but our datafram is same because we didnt put here Ne
```

```
Out[70]:
```

	0	1	2	3	4	5	6	7	8	
0	765	0.786601	0.956868	0.258590	0.334036	0.870225	0.633365	0.816873	0.811798	0.04
1	0.158507	0.520862	0.679760	0.102497	0.113052	0.726117	0.315770	0.719951	0.922452	0.11
2	0.049013	0.828629	0.914315	0.260196	0.926152	0.150249	0.933029	0.913121	0.905099	0.59
3	0.549031	0.540318	0.029625	0.599735	0.378303	0.905322	0.487571	0.247969	0.038808	0.84
4	0.839633	0.086155	0.210866	0.451999	0.392206	0.826961	0.124213	0.996235	0.651352	0.69
...
219	0.310491	0.703000	0.499299	0.127928	0.071682	0.447261	0.841341	0.651170	0.447458	0.00
220	0.808144	0.514275	0.870238	0.803786	0.234934	0.100896	0.001808	0.007777	0.452187	0.70
221	0.22891	0.341982	0.560597	0.175265	0.499093	0.079968	0.665550	0.890360	0.856866	0.28
222	0.266593	0.376427	0.597294	0.504088	0.907032	0.804976	0.171475	0.906328	0.013591	0.85
223	0.861299	0.579981	0.625031	0.427594	0.796474	0.406625	0.553957	0.287599	0.803210	0.70

224 rows × 10 columns

```
In [71]: #If see again the dataframe
```

```
In [72]: Newdf.head()
```

```
Out[72]:
```

	0	1	2	3	4	5	6	7	8	
0	765	0.786601	0.956868	0.258590	0.334036	0.870225	0.633365	0.816873	0.811798	0.0417
1	0.158507	0.520862	0.679760	0.102497	0.113052	0.726117	0.315770	0.719951	0.922452	0.1155
2	0.049013	0.828629	0.914315	0.260196	0.926152	0.150249	0.933029	0.913121	0.905099	0.5927
3	0.549031	0.540318	0.029625	0.599735	0.378303	0.905322	0.487571	0.247969	0.038808	0.8447
4	0.839633	0.086155	0.210866	0.451999	0.392206	0.826961	0.124213	0.996235	0.651352	0.6947

```
In [73]: #Inplace function
```

```
In [74]: Newdf.drop(['A'], axis = 1, inplace = True)
```

```
In [75]: Newdf
```

```
Out[75]:
```

	0	1	2	3	4	5	6	7	8	
0	765	0.786601	0.956868	0.258590	0.334036	0.870225	0.633365	0.816873	0.811798	0.04
1	0.158507	0.520862	0.679760	0.102497	0.113052	0.726117	0.315770	0.719951	0.922452	0.11
2	0.049013	0.828629	0.914315	0.260196	0.926152	0.150249	0.933029	0.913121	0.905099	0.59
3	0.549031	0.540318	0.029625	0.599735	0.378303	0.905322	0.487571	0.247969	0.038808	0.84
4	0.839633	0.086155	0.210866	0.451999	0.392206	0.826961	0.124213	0.996235	0.651352	0.69
...
219	0.310491	0.703000	0.499299	0.127928	0.071682	0.447261	0.841341	0.651170	0.447458	0.00
220	0.808144	0.514275	0.870238	0.803786	0.234934	0.100896	0.001808	0.007777	0.452187	0.70
221	0.22891	0.341982	0.560597	0.175265	0.499093	0.079968	0.665550	0.890360	0.856866	0.28
222	0.266593	0.376427	0.597294	0.504088	0.907032	0.804976	0.171475	0.906328	0.013591	0.85
223	0.861299	0.579981	0.625031	0.427594	0.796474	0.406625	0.553957	0.287599	0.803210	0.70

224 rows × 10 columns

```
In [76]: Newdf.drop([1]) #so here rows 0 will be removed
```

```
Out[76]:
```

	0	1	2	3	4	5	6	7	8	
0	765	0.786601	0.956868	0.258590	0.334036	0.870225	0.633365	0.816873	0.811798	0.04
2	0.049013	0.828629	0.914315	0.260196	0.926152	0.150249	0.933029	0.913121	0.905099	0.59
3	0.549031	0.540318	0.029625	0.599735	0.378303	0.905322	0.487571	0.247969	0.038808	0.84
4	0.839633	0.086155	0.210866	0.451999	0.392206	0.826961	0.124213	0.996235	0.651352	0.69
5	0.379833	0.356152	0.016603	0.217805	0.849607	0.093775	0.574011	0.546228	0.127571	0.97
...
219	0.310491	0.703000	0.499299	0.127928	0.071682	0.447261	0.841341	0.651170	0.447458	0.00
220	0.808144	0.514275	0.870238	0.803786	0.234934	0.100896	0.001808	0.007777	0.452187	0.70
221	0.22891	0.341982	0.560597	0.175265	0.499093	0.079968	0.665550	0.890360	0.856866	0.28
222	0.266593	0.376427	0.597294	0.504088	0.907032	0.804976	0.171475	0.906328	0.013591	0.85
223	0.861299	0.579981	0.625031	0.427594	0.796474	0.406625	0.553957	0.287599	0.803210	0.70

223 rows × 10 columns

```
In [78]: Newdf
```

Out[78]:

	0	1	2	3	4	5	6	7	8	
0	765	0.786601	0.956868	0.258590	0.334036	0.870225	0.633365	0.816873	0.811798	0.04
1	0.158507	0.520862	0.679760	0.102497	0.113052	0.726117	0.315770	0.719951	0.922452	0.11
2	0.049013	0.828629	0.914315	0.260196	0.926152	0.150249	0.933029	0.913121	0.905099	0.59
3	0.549031	0.540318	0.029625	0.599735	0.378303	0.905322	0.487571	0.247969	0.038808	0.84
4	0.839633	0.086155	0.210866	0.451999	0.392206	0.826961	0.124213	0.996235	0.651352	0.69
...
219	0.310491	0.703000	0.499299	0.127928	0.071682	0.447261	0.841341	0.651170	0.447458	0.00
220	0.808144	0.514275	0.870238	0.803786	0.234934	0.100896	0.001808	0.007777	0.452187	0.70
221	0.22891	0.341982	0.560597	0.175265	0.499093	0.079968	0.665550	0.890360	0.856866	0.28
222	0.266593	0.376427	0.597294	0.504088	0.907032	0.804976	0.171475	0.906328	0.013591	0.85
223	0.861299	0.579981	0.625031	0.427594	0.796474	0.406625	0.553957	0.287599	0.803210	0.70

224 rows × 10 columns

In [79]: Newdf.drop([0])

Out[79]:

	0	1	2	3	4	5	6	7	8	
1	0.158507	0.520862	0.679760	0.102497	0.113052	0.726117	0.315770	0.719951	0.922452	0.11
2	0.049013	0.828629	0.914315	0.260196	0.926152	0.150249	0.933029	0.913121	0.905099	0.59
3	0.549031	0.540318	0.029625	0.599735	0.378303	0.905322	0.487571	0.247969	0.038808	0.84
4	0.839633	0.086155	0.210866	0.451999	0.392206	0.826961	0.124213	0.996235	0.651352	0.69
5	0.379833	0.356152	0.016603	0.217805	0.849607	0.093775	0.574011	0.546228	0.127571	0.97
...
219	0.310491	0.703000	0.499299	0.127928	0.071682	0.447261	0.841341	0.651170	0.447458	0.00
220	0.808144	0.514275	0.870238	0.803786	0.234934	0.100896	0.001808	0.007777	0.452187	0.70
221	0.22891	0.341982	0.560597	0.175265	0.499093	0.079968	0.665550	0.890360	0.856866	0.28
222	0.266593	0.376427	0.597294	0.504088	0.907032	0.804976	0.171475	0.906328	0.013591	0.85
223	0.861299	0.579981	0.625031	0.427594	0.796474	0.406625	0.553957	0.287599	0.803210	0.70

223 rows × 10 columns

In [81]: Newdf.head(3)

Out[81]:

	0	1	2	3	4	5	6	7	8	
0	765	0.786601	0.956868	0.258590	0.334036	0.870225	0.633365	0.816873	0.811798	0.0417
1	0.158507	0.520862	0.679760	0.102497	0.113052	0.726117	0.315770	0.719951	0.922452	0.1155
2	0.049013	0.828629	0.914315	0.260196	0.926152	0.150249	0.933029	0.913121	0.905099	0.5927

```
In [83]: Newdf.drop([1], axis = 1)
```

```
Out[83]:
```

	0	2	3	4	5	6	7	8	9
0	765	0.956868	0.258590	0.334036	0.870225	0.633365	0.816873	0.811798	0.041700
1	0.158507	0.679760	0.102497	0.113052	0.726117	0.315770	0.719951	0.922452	0.115568
2	0.049013	0.914315	0.260196	0.926152	0.150249	0.933029	0.913121	0.905099	0.592757
3	0.549031	0.029625	0.599735	0.378303	0.905322	0.487571	0.247969	0.038808	0.844798
4	0.839633	0.210866	0.451999	0.392206	0.826961	0.124213	0.996235	0.651352	0.694787
...
219	0.310491	0.499299	0.127928	0.071682	0.447261	0.841341	0.651170	0.447458	0.008918
220	0.808144	0.870238	0.803786	0.234934	0.100896	0.001808	0.007777	0.452187	0.706434
221	0.22891	0.560597	0.175265	0.499093	0.079968	0.665550	0.890360	0.856866	0.285109
222	0.266593	0.597294	0.504088	0.907032	0.804976	0.171475	0.906328	0.013591	0.853608
223	0.861299	0.625031	0.427594	0.796474	0.406625	0.553957	0.287599	0.803210	0.705388

224 rows × 9 columns

```
In [84]: Newdf
```

```
Out[84]:
```

	0	1	2	3	4	5	6	7	8	
0	765	0.786601	0.956868	0.258590	0.334036	0.870225	0.633365	0.816873	0.811798	0.041700
1	0.158507	0.520862	0.679760	0.102497	0.113052	0.726117	0.315770	0.719951	0.922452	0.115568
2	0.049013	0.828629	0.914315	0.260196	0.926152	0.150249	0.933029	0.913121	0.905099	0.592757
3	0.549031	0.540318	0.029625	0.599735	0.378303	0.905322	0.487571	0.247969	0.038808	0.844798
4	0.839633	0.086155	0.210866	0.451999	0.392206	0.826961	0.124213	0.996235	0.651352	0.694787
...
219	0.310491	0.703000	0.499299	0.127928	0.071682	0.447261	0.841341	0.651170	0.447458	0.008918
220	0.808144	0.514275	0.870238	0.803786	0.234934	0.100896	0.001808	0.007777	0.452187	0.706434
221	0.22891	0.341982	0.560597	0.175265	0.499093	0.079968	0.665550	0.890360	0.856866	0.285109
222	0.266593	0.376427	0.597294	0.504088	0.907032	0.804976	0.171475	0.906328	0.013591	0.853608
223	0.861299	0.579981	0.625031	0.427594	0.796474	0.406625	0.553957	0.287599	0.803210	0.705388

224 rows × 10 columns

```
In [85]: #If set the values use loc funtion
```

```
In [86]: Newdf.loc[:, ['B']] = 56
```

```
In [87]: Newdf
```

```
Out[87]:
```

	0	1	2	3	4	5	6	7	8	
0	765	0.786601	0.956868	0.258590	0.334036	0.870225	0.633365	0.816873	0.811798	0.04
1	0.158507	0.520862	0.679760	0.102497	0.113052	0.726117	0.315770	0.719951	0.922452	0.11
2	0.049013	0.828629	0.914315	0.260196	0.926152	0.150249	0.933029	0.913121	0.905099	0.59
3	0.549031	0.540318	0.029625	0.599735	0.378303	0.905322	0.487571	0.247969	0.038808	0.84
4	0.839633	0.086155	0.210866	0.451999	0.392206	0.826961	0.124213	0.996235	0.651352	0.69
...
219	0.310491	0.703000	0.499299	0.127928	0.071682	0.447261	0.841341	0.651170	0.447458	0.00
220	0.808144	0.514275	0.870238	0.803786	0.234934	0.100896	0.001808	0.007777	0.452187	0.70
221	0.22891	0.341982	0.560597	0.175265	0.499093	0.079968	0.665550	0.890360	0.856866	0.28
222	0.266593	0.376427	0.597294	0.504088	0.907032	0.804976	0.171475	0.906328	0.013591	0.85
223	0.861299	0.579981	0.625031	0.427594	0.796474	0.406625	0.553957	0.287599	0.803210	0.70

224 rows × 11 columns

```
In [88]: Newdf.shape
```

```
Out[88]: (224, 11)
```

```
In [89]: Newdf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 224 entries, 0 to 223
Data columns (total 11 columns):
#   Column  Non-Null Count  Dtype
---  -
0    0      224 non-null    object
1    1      224 non-null    float64
2    2      224 non-null    float64
3    3      224 non-null    float64
4    4      224 non-null    float64
5    5      224 non-null    float64
6    6      224 non-null    float64
7    7      224 non-null    float64
8    8      224 non-null    float64
9    9      224 non-null    float64
10   B       224 non-null    int64
dtypes: float64(9), int64(1), object(1)
memory usage: 26.2+ KB
```

```
In [90]: Newdf.notnull()
```



```
Out[90]:
```

	0	1	2	3	4	5	6	7	8	9	B
0	True	True	True	True	True	True	True	True	True	True	True
1	True	True	True	True	True	True	True	True	True	True	True
2	True	True	True	True	True	True	True	True	True	True	True
3	True	True	True	True	True	True	True	True	True	True	True
4	True	True	True	True	True	True	True	True	True	True	True
...
219	True	True	True	True	True	True	True	True	True	True	True
220	True	True	True	True	True	True	True	True	True	True	True
221	True	True	True	True	True	True	True	True	True	True	True
222	True	True	True	True	True	True	True	True	True	True	True
223	True	True	True	True	True	True	True	True	True	True	True

224 rows × 11 columns

```
In [91]: Newdf.isnull()
```

```
Out[91]:
```

	0	1	2	3	4	5	6	7	8	9	B
0	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False
...
219	False	False	False	False	False	False	False	False	False	False	False
220	False	False	False	False	False	False	False	False	False	False	False
221	False	False	False	False	False	False	False	False	False	False	False
222	False	False	False	False	False	False	False	False	False	False	False
223	False	False	False	False	False	False	False	False	False	False	False

224 rows × 11 columns

```
In [92]: Newdf.min()
```

```
Out[92]: 0    0.002155
          1    0.007269
          2    0.006454
          3    0.000967
          4    0.000731
          5    0.00226
          6    0.001808
          7    0.007448
          8    0.000834
          9    0.001633
          B      56
          dtype: object
```

```
In [93]: Newdf.max()
```

```
Out[93]: 0      765
          1    0.996305
          2    0.999708
          3    0.994333
          4    0.993286
          5    0.985289
          6    0.999669
          7    0.99627
          8    0.997999
          9    0.990654
          B      56
          dtype: object
```

```
In [94]: Newdf.mean()
```

```
Out[94]: 0    3.914625
          1    0.473807
          2    0.507056
          3    0.490418
          4    0.485518
          5    0.487501
          6    0.48408
          7    0.52221
          8    0.49024
          9    0.489231
          B      56.0
          dtype: object
```

```
In [ ]:
```