

```
In [2]: import numpy as np
import pandas as pd

In [4]: dict1={
    "name":["harry","rohan",'swapnil'],
    "marks":[34,45,67]
    "city":["kolkata",'delhi'],'rampur'
}

converting dictionary data into a dataframe

In [6]: df=pd.DataFrame(dict1)
df

Out[6]:
   name  marks  city
0   harry    34  kolkata
1   rohan    45   delhi
2  swapnil    67  rampur

export to csv/excel

In [10]: df.to_csv("xyz.csv") #file created

In [12]: df.describe() #describes about the whole dataframe

Out[12]:
      marks
count  3.000000
mean   48.666667
std    16.802778
min    34.000000
25%    39.500000
50%    45.000000
75%    56.000000
max    67.000000

In [14]: df["name"]

Out[14]:
0    harry
1    rohan
2  swapnil
Name: name, dtype: object

In [16]: df["name"][0]

Out[16]:
'harry'

In [18]: ser=pd.Series(np.random.rand(10))
ser

Out[18]:
0    0.726980
1    0.687978
2    0.031123
3    0.747492
4    0.358562
5    0.416044
6    0.730750
7    0.633330
8    0.795251
9    0.218001
dtype: float64

In [20]: type(ser)

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

In [27]: newdf=pd.DataFrame(np.random.rand(334,5),index=np.arange(334)) #explicitly mentioning the index is a good prac
newdf.head()

Out[27]:
   0         1         2         3         4
0  0.421288  0.853916  0.704375  0.649743  0.837185
1  0.993801  0.207274  0.444053  0.085572  0.124174
2  0.010187  0.250577  0.024116  0.594619  0.428003
3  0.254682  0.202654  0.426334  0.621735  0.448899
4  0.666455  0.383844  0.304455  0.397567  0.893115

In [29]: type(newdf)

Out[29]:
pandas.core.frame.DataFrame

In [31]: newdf = pd.DataFrame(np.random.rand(334, 5)) #this will work too
newdf.head()

Out[31]:
   0         1         2         3         4
0  0.310280  0.278216  0.103546  0.333126  0.962768
1  0.952521  0.846187  0.518124  0.732422  0.936468
2  0.677125  0.754908  0.421217  0.702812  0.193951
3  0.427790  0.017078  0.375471  0.788276  0.576399
4  0.531119  0.252122  0.830295  0.919345  0.337123

In [35]: newdf[0][0]="head"
newdf.dtypes

Out[35]:
0    object
1    float64
2    float64
3    float64
4    float64
dtype: object

In [37]: newdf.head()

Out[37]:
   0         1         2         3         4
0  head  0.278216  0.103546  0.333126  0.962768
1    2  0.846187  0.518124  0.732422  0.936468
2  0.677125  0.754908  0.421217  0.702812  0.193951
3  0.42779  0.017078  0.375471  0.788276  0.576399
4  0.531119  0.252122  0.830295  0.919345  0.337123

In [39]: newdf[0][1]=2
newdf.head()

Out[39]:
   0         1         2         3         4
0  head  0.278216  0.103546  0.333126  0.962768
1    2  0.846187  0.518124  0.732422  0.936468
2  0.677125  0.754908  0.421217  0.702812  0.193951
3  0.42779  0.017078  0.375471  0.788276  0.576399
4  0.531119  0.252122  0.830295  0.919345  0.337123

In [41]: newdf.index

Out[41]:
RangeIndex(start=0, stop=334, step=1)

In [43]: newdf.columns

Out[43]:
RangeIndex(start=0, stop=5, step=1)

** convert into a numpy array**

In [46]: newdf.to_numpy()

Out[46]:
array([[ 'head', 0.27821561362494773, 0.10354569828951021,
        0.3331261638126414, 0.9627678884867048],
       [2, 0.8461871318039776, 0.5181238922485321, 0.7324218826826583,
        0.9364681797954405],
       [0.6771252083991784, 0.754908482088671, 0.4212171998504376,
        0.7028122414035928, 0.1939514756634706],
       ...,
       [0.4314600981854605, 0.778740430928113, 0.5844662240861598,
        0.05109906437733179, 0.35165219045671672],
       [0.773402444803329, 0.2051386084735574, 0.6300368706304867,
        0.44645137302953264, 0.1390165615151534],
       [0.7324603448251946, 0.7232318642951332, 0.10451357336832534,
        0.6422774826792219, 0.3128783168457124]], dtype=object)

In [48]: newdf.head()

Out[48]:
   0         1         2         3         4
0  head  0.278216  0.103546  0.333126  0.962768
1    2  0.846187  0.518124  0.732422  0.936468
2  0.677125  0.754908  0.421217  0.702812  0.193951
3  0.42779  0.017078  0.375471  0.788276  0.576399
4  0.531119  0.252122  0.830295  0.919345  0.337123

In [50]: newdf.T

Out[50]:
   0         1         2         3         4         5         6         7         8         9  ...   324   325   326   327   328   329   330   331   332   333
0  head  0.278216  0.846187  0.754908  0.017078  0.252122  0.082155  0.498106  0.24201  0.393861  0.73503  ...  0.353224  0.126499  0.626333  0.519084  0.753023  0.422409  0.352365  0.77874  0.205139  0.723232
1  0.278216  0.846187  0.754908  0.017078  0.252122  0.082155  0.498106  0.24201  0.393861  0.73503  ...  0.353224  0.126499  0.626333  0.519084  0.753023  0.422409  0.352365  0.77874  0.205139  0.723232
2  0.105646  0.518124  0.421217  0.375471  0.830295  0.2853  0.845052  0.725312  0.243742  0.370726  ...  0.158267  0.821753  0.111694  0.542335  0.786354  0.301574  0.16808  0.584466  0.630037  0.104514
3  0.333126  0.732422  0.702812  0.788276  0.919345  0.8621  0.451828  0.595144  0.175829  0.481686  ...  0.261253  0.892783  0.848449  0.11854  0.52733  0.772222  0.281917  0.051099  0.446451  0.642277
4  0.962768  0.936468  0.193951  0.576399  0.337123  0.776204  0.719871  0.784996  0.874227  0.602566  ...  0.432456  0.232043  0.732791  0.331616  0.773407  0.20263  0.145513  0.556522  0.139017  0.312878
5 rows × 334 columns

In [52]: newdf.sort_index(axis=0,ascending=False) #axis=0 is for row and axis=1 is for cols

Out[52]:
   0         1         2         3         4
333  0.73246  0.723232  0.104514  0.642277  0.312878
332  0.773402  0.205139  0.630037  0.446451  0.139017
331  0.43146  0.778740  0.584466  0.051099  0.566522
330  0.077725  0.352365  0.168080  0.281917  0.145513
329  0.27942  0.422409  0.301574  0.772222  0.202830
...
...
4  0.531119  0.252122  0.830295  0.919345  0.337123
3  0.42779  0.017078  0.375471  0.788276  0.576399
2  0.677125  0.754908  0.421217  0.702812  0.193951
1  2  0.846187  0.518124  0.732422  0.936468
0  head  0.278216  0.103546  0.333126  0.962768
334 rows × 5 columns

In [54]: newdf[0]

Out[54]:
0    head
1         2
2    0.677125
3    0.42779
4    0.531119
...
329    0.27942
330    0.077725
331    0.43146
332    0.773402
333    0.73246
Name: 0, Length: 334, dtype: object

In [56]: type(newdf[0])

Out[56]:
pandas.core.series.Series

In [60]: newdf.columns=list("abcde")
newdf.head(2)

Out[60]:
   a         b         c         d         e
0  head  0.278216  0.103546  0.333126  0.962768
1  2  0.846187  0.518124  0.732422  0.936468

In [62]: newdf.loc[0,"a"]=6554
newdf.head()

Out[62]:
   a         b         c         d         e
0  6554  0.278216  0.103546  0.333126  0.962768
1  2  0.846187  0.518124  0.732422  0.936468
2  0.677125  0.754908  0.421217  0.702812  0.193951
3  0.42779  0.017078  0.375471  0.788276  0.576399
4  0.531119  0.252122  0.830295  0.919345  0.337123

In [64]: newdf.drop("a",axis=1)
newdf.head()

Out[64]:
   a         b         c         d         e
0  6554  0.278216  0.103546  0.333126  0.962768
1  2  0.846187  0.518124  0.732422  0.936468
2  0.677125  0.754908  0.421217  0.702812  0.193951
3  0.42779  0.017078  0.375471  0.788276  0.576399
4  0.531119  0.252122  0.830295  0.919345  0.337123

In [84]: newdf.drop(["b",axis=1, inplace=True) # original dataframe e drop
newdf.head()

Out[84]:
   a         c         d
0  6554  0.103546  0.333126
1  2  0.518124  0.732422
2  0.677125  0.421217  0.702812
3  0.42779  0.375471  0.788276
4  0.531119  0.830295  0.919345

In [74]: newdf.loc[newdf["a"]<0.3] # 'newdf["a"]<0.3)' ~ This expression creates a boolean mask by evaluating the condition that the values in column 'a' of 'newdf' are less than 0.3. It returns a Series of 'True' or 'False' values, where 'T

Out[74]:
   a         b         c         d
6  0.118977  0.498106  0.845052  0.451828
14 0.061583  0.263718  0.342365  0.526382
16 0.204399  0.701807  0.608853  0.996648
20 0.248264  0.847446  0.687500  0.170266
21 0.119313  0.703611  0.316451  0.086710
...
...
318 0.147585  0.071785  0.011158  0.840929
320 0.271614  0.585468  0.287787  0.000399
326 0.059678  0.626333  0.111694  0.648449
329 0.27942  0.422409  0.301574  0.772222
330 0.077725  0.352365  0.168080  0.281917
107 rows × 4 columns

In [80]: newdf.drop(["a","b",axis=1) # juts copy ta drop holo

Out[80]:
   c         d
0  0.103546  0.333126
1  0.518124  0.732422
2  0.421217  0.702812
3  0.375471  0.788276
4  0.830295  0.919345
...
...
329 0.301574  0.772222
330 0.168080  0.281917
331 0.584466  0.051099
332 0.630037  0.446451
333 0.104514  0.642277
334 rows × 2 columns

In [107]: data=pd.read_excel("InsuranceDataset.xlsx") # reading excel data

In [109]: data

Out[109]:
   Column1  Column2
0    Age  Premium
1    25    1800
2    30    3200
3    35    4200
4    40    4700
5    45    5500

In [115]:

In [117]: import pandas as pd

# Create a sample DataFrame with some data
data = {
    "Name": ['Alice', 'Bob', 'Charlie', 'David'],
    "Age": [24, 30, 22, 35],
    "City": ['New York', 'Los Angeles', 'Chicago', 'Houston']
}

df = pd.DataFrame(data)

# Display the DataFrame
print(df)

# Save the DataFrame to an Excel file
df.to_excel("InsuranceDataset.xlsx", index=False)

   Name  Age  City
0  Alice  24  New York
1   Bob   30  Los Angeles
```

