pandas

September 24, 2024

[1]:

```
import numpy as np
import pandas as pd
```

[2]:

```
dict ={
    "name":['pratik','debi','manoj','prabhu'],
    "marks":[92,98,98,69],
    "city":['balasore','bhubans','bhadrak','cuttack']
}
```

[3]:

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

[4]:

```
df
```

[4]:

```
     name  marks      city
0  pratik     92  balasore
1    debi     98   bhubans
2   manoj     98   bhadrak
3  prabhu     69   cuttack
```

[5]:

```
df.to_csv('pratik.csv')
```

[6]:

```
df.to_csv('pratikWithoutindex.csv',index=False)
```

[7]:

```
df.head(2)
```

[7]:

```
     name  marks      city
0  pratik     92  balasore
1    debi     98   bhubans
```

[8]:

```
df.tail(2)
```

[8]:

```
     name  marks     city
2   manoj     98  bhadrak
3  prabhu     69  cuttack
```

[9]:

```
df.describe()
```

```
[9]:             marks
     count   4.000000
     mean   89.250000
     std    13.793114
     min    69.000000
     25%    86.250000
     50%    95.000000
     75%    98.000000
     max    98.000000
```

```
[10]: train =pd.read_csv('train.csv')
```

```
[11]: train
```

```
[11]:    Unnamed: 0  train  speed      city
      0           0  23455     50  balasore
      1           1  83984     98   bhubans
      2           2  34903     34   bhadrak
      3           3  47387     57   cuttack
```

```
[12]: train['speed']
```

```
[12]: 0    50
      1    98
      2    34
      3    57
      Name: speed, dtype: int64
```

```
[13]: train['speed'][1]
```

```
[13]: np.int64(98)
```

```
[14]: train['speed'][0]=50
```

```
C:\Users\Asus\AppData\Local\Temp\ipykernel_32128\566548151.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
  train['speed'][0]=50
```

```
[15]: train
```

```
[15]:    Unnamed: 0  train  speed      city
      0           0  23455     50  balasore
      1           1  83984     98   bhubans
```

```
2               2  34903     34  bhadrak
3               3  47387     57  cuttack
```

```
[16]: train.to_csv('train.csv',index=False)
```

```
[17]: train.index=['first','2nd','third','4th']
```

```
[18]: train
```

```
[18]:        Unnamed: 0  train  speed      city
      first           0  23455     50  balasore
      2nd             1  83984     98   bhubans
      third           2  34903     34   bhadrak
      4th             3  47387     57   cuttack
```

```
[19]: train.to_csv('train_index.csv')
```

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

```
[21]: ser
```

```
[21]: 0     0.351174
      1     0.651403
      2     0.778194
      3     0.788267
      4     0.242746
      5     0.587586
      6     0.978587
      7     0.288774
      8     0.658795
      9     0.461646
      10    0.407176
      11    0.819481
      12    0.759530
      13    0.480361
      14    0.079158
      15    0.267537
      16    0.275200
      17    0.095396
      18    0.559202
      19    0.981194
      dtype: float64
```

```
[22]: type(ser)
```

```
[22]: pandas.core.series.Series
```

```
[23]: newdf = pd.DataFrame(np.random.rand(100,5),index=np.arange(100))
```

```
[24]: newdf
```

```
[24]:            0         1         2         3         4
      0   0.021659  0.194278  0.489157  0.684807  0.157660
      1   0.826964  0.980641  0.241879  0.830032  0.116351
      2   0.991456  0.689004  0.199466  0.478640  0.041413
      3   0.424973  0.200529  0.456661  0.654169  0.420190
      4   0.747753  0.469301  0.599305  0.112292  0.252662
      ..       …         …         …         …         …
      95  0.557259  0.269137  0.187552  0.924462  0.884606
      96  0.665992  0.727035  0.252451  0.583246  0.718507
      97  0.597315  0.942574  0.713916  0.916324  0.999684
      98  0.838130  0.089792  0.556095  0.013597  0.181854
      99  0.270048  0.312865  0.956253  0.636719  0.604408

      [100 rows x 5 columns]
```

```
[25]: newdf.head()
```

```
[25]:          0         1         2         3         4
      0  0.021659  0.194278  0.489157  0.684807  0.157660
      1  0.826964  0.980641  0.241879  0.830032  0.116351
      2  0.991456  0.689004  0.199466  0.478640  0.041413
      3  0.424973  0.200529  0.456661  0.654169  0.420190
      4  0.747753  0.469301  0.599305  0.112292  0.252662
```

```
[26]: newdf
```

```
[26]:            0         1         2         3         4
      0   0.021659  0.194278  0.489157  0.684807  0.157660
      1   0.826964  0.980641  0.241879  0.830032  0.116351
      2   0.991456  0.689004  0.199466  0.478640  0.041413
      3   0.424973  0.200529  0.456661  0.654169  0.420190
      4   0.747753  0.469301  0.599305  0.112292  0.252662
      ..       …         …         …         …         …
      95  0.557259  0.269137  0.187552  0.924462  0.884606
      96  0.665992  0.727035  0.252451  0.583246  0.718507
      97  0.597315  0.942574  0.713916  0.916324  0.999684
      98  0.838130  0.089792  0.556095  0.013597  0.181854
      99  0.270048  0.312865  0.956253  0.636719  0.604408

      [100 rows x 5 columns]
```

```
[27]: newdf.tail()
```

[27]:
```
           0         1         2         3         4
95  0.557259  0.269137  0.187552  0.924462  0.884606
96  0.665992  0.727035  0.252451  0.583246  0.718507
97  0.597315  0.942574  0.713916  0.916324  0.999684
98  0.838130  0.089792  0.556095  0.013597  0.181854
99  0.270048  0.312865  0.956253  0.636719  0.604408
```

[28]:
```
type(newdf)
```

[28]:
```
pandas.core.frame.DataFrame
```

[29]:
```
newdf.head()
```

[29]:
```
          0         1         2         3         4
0  0.021659  0.194278  0.489157  0.684807  0.157660
1  0.826964  0.980641  0.241879  0.830032  0.116351
2  0.991456  0.689004  0.199466  0.478640  0.041413
3  0.424973  0.200529  0.456661  0.654169  0.420190
4  0.747753  0.469301  0.599305  0.112292  0.252662
```

[30]:
```
newdf.describe()
```

[30]:
```
                0           1           2           3           4
count  100.000000  100.000000  100.000000  100.000000  100.000000
mean     0.538079    0.522261    0.492193    0.542379    0.512849
std      0.279806    0.298681    0.291086    0.302127    0.289500
min      0.010101    0.017664    0.001173    0.007964    0.013642
25%      0.325161    0.254093    0.243481    0.286557    0.259371
50%      0.550310    0.517307    0.467070    0.600798    0.496563
75%      0.784579    0.803988    0.721103    0.778028    0.790883
max      0.991456    0.999086    0.990134    0.982641    0.999684
```

[31]:
```
newdf.dtypes
```

[31]:
```
0    float64
1    float64
2    float64
3    float64
4    float64
dtype: object
```

[32]:
```
newdf[0][0]="pratik"
```

```
C:\Users\Asus\AppData\Local\Temp\ipykernel_32128\400600176.py:1: FutureWarning:
ChainedAssignmentError: behaviour will change in pandas 3.0!
You are setting values through chained assignment. Currently this works in
certain cases, but when using Copy-on-Write (which will become the default
behaviour in pandas 3.0) this will never work to update the original DataFrame
```

```
or Series, because the intermediate object on which we are setting values will
behave as a copy.
A typical example is when you are setting values in a column of a DataFrame,
like:

df["col"][row_indexer] = value

Use `df.loc[row_indexer, "col"] = values` instead, to perform the assignment in
a single step and ensure this keeps updating the original `df`.

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

  newdf[0][0]="pratik"
C:\Users\Asus\AppData\Local\Temp\ipykernel_32128\400600176.py:1: FutureWarning:
Setting an item of incompatible dtype is deprecated and will raise an error in a
future version of pandas. Value 'pratik' has dtype incompatible with float64,
please explicitly cast to a compatible dtype first.
  newdf[0][0]="pratik"
```

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

[33]:
```
          0         1         2         3         4
0    pratik  0.194278  0.489157  0.684807  0.157660
1  0.826964  0.980641  0.241879  0.830032  0.116351
2  0.991456  0.689004  0.199466  0.478640  0.041413
3  0.424973  0.200529  0.456661  0.654169  0.420190
4  0.747753  0.469301  0.599305  0.112292  0.252662
```

[34]:
```
newdf.dtypes
```

[34]:
```
0     object
1    float64
2    float64
3    float64
4    float64
dtype: object
```

[35]:
```
newdf.to_csv('DataFrame.csv')
```

[36]:
```
newdf
```

[36]:
```
          0         1         2         3         4
0    pratik  0.194278  0.489157  0.684807  0.157660
1  0.826964  0.980641  0.241879  0.830032  0.116351
2  0.991456  0.689004  0.199466  0.478640  0.041413
3  0.424973  0.200529  0.456661  0.654169  0.420190
```

```
4   0.747753  0.469301  0.599305  0.112292  0.252662
..       …         …         …         …         …
95  0.557259  0.269137  0.187552  0.924462  0.884606
96  0.665992  0.727035  0.252451  0.583246  0.718507
97  0.597315  0.942574  0.713916  0.916324  0.999684
98   0.83813  0.089792  0.556095  0.013597  0.181854
99  0.270048  0.312865  0.956253  0.636719  0.604408

[100 rows x 5 columns]
```

[37]:
```
ser.to_csv('series.csv')
```

[38]:
```
ser
```

[38]:
```
0     0.351174
1     0.651403
2     0.778194
3     0.788267
4     0.242746
5     0.587586
6     0.978587
7     0.288774
8     0.658795
9     0.461646
10    0.407176
11    0.819481
12    0.759530
13    0.480361
14    0.079158
15    0.267537
16    0.275200
17    0.095396
18    0.559202
19    0.981194
dtype: float64
```

[39]:
```
newdf
```

[39]:
```
           0         1         2         3         4
0     pratik  0.194278  0.489157  0.684807  0.157660
1   0.826964  0.980641  0.241879  0.830032  0.116351
2   0.991456  0.689004  0.199466  0.478640  0.041413
3   0.424973  0.200529  0.456661  0.654169  0.420190
4   0.747753  0.469301  0.599305  0.112292  0.252662
..       …         …         …         …         …
95  0.557259  0.269137  0.187552  0.924462  0.884606
96  0.665992  0.727035  0.252451  0.583246  0.718507
```

```
97  0.597315  0.942574  0.713916  0.916324  0.999684
98   0.83813  0.089792  0.556095  0.013597  0.181854
99  0.270048  0.312865  0.956253  0.636719  0.604408

[100 rows x 5 columns]
```

[40]:
```
newdf.head()
```

[40]:
```
          0         1         2         3         4
0    pratik  0.194278  0.489157  0.684807  0.157660
1  0.826964  0.980641  0.241879  0.830032  0.116351
2  0.991456  0.689004  0.199466  0.478640  0.041413
3  0.424973  0.200529  0.456661  0.654169  0.420190
4  0.747753  0.469301  0.599305  0.112292  0.252662
```

[41]:
```
newdf.index
```

[41]:
```
Index([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17,
       18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35,
       36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53,
       54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71,
       72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89,
       90, 91, 92, 93, 94, 95, 96, 97, 98, 99],
      dtype='int64')
```

[42]:
```
newdf.columns
```

[42]:
```
RangeIndex(start=0, stop=5, step=1)
```

[43]:
```
newdf.to_numpy()
```

[43]:
```
array([['pratik', 0.1942777788635659, 0.4891572796078608,
        0.6848072815098287, 0.15765966718896263],
       [0.8269640525872092, 0.9806410740498308, 0.24187928225101185,
        0.8300319201950969, 0.11635145713900363],
       [0.9914561067958125, 0.6890042169954613, 0.1994664741671619,
        0.4786396567693496, 0.04141310123906572],
       [0.4249725790797323, 0.20052899817909486, 0.4566608894191725,
        0.6541692349073818, 0.420189765530247],
       [0.7477527905183773, 0.4693011062142466, 0.5993046664676415,
        0.11229248561279503, 0.25266170092455187],
       [0.507951697199343, 0.9372258226271368, 0.45836040370960907,
        0.3930575535094847, 0.9478197086868269],
       [0.25839221756388464, 0.26135655546813885, 0.0011729112474587744,
        0.6526575203926213, 0.9056995552635041],
       [0.6698865023675451, 0.972192385978283, 0.624571253984406,
        0.5191906934699676, 0.16077549890864384],
       [0.3723951729549362, 0.1746889038791355, 0.38313582317776274,
        0.5241222292230453, 0.09952705689697539],
       [0.9686343416969815, 0.9088751216899055, 0.47011639702897823,
        0.7147140402996837, 0.9968334004272946],
       [0.4113442561093731, 0.88819411167676, 0.13295932717191028,
        0.8018407160374847, 0.20310814632923757],
       [0.6514228915524566, 0.7551755242973917, 0.9847692296121343,
        0.23536088866671034, 0.26250525819811366],
       [0.36119949662317197, 0.9466572679799607, 0.45450994201440786,
        0.6799575779042831, 0.22057987277330116],
       [0.21235656911493883, 0.5246457501809446, 0.9858181811832328,
        0.9474838632002288, 0.8489132539629114],
       [0.907322090352746, 0.06741641049590985, 0.3810711865025215,
        0.7724556438937268, 0.11571889549754832],
       [0.6691655275955503, 0.2151385409770824, 0.3405970801939343,
        0.3373430295127551, 0.3599216612197874],
       [0.8855813693362602, 0.1586062546946806, 0.7426666926283672,
        0.9609909196326468, 0.5177906391372921],
       [0.10156108397736663, 0.8187308402708887, 0.3706746317066608,
        0.8345824374200667, 0.6798670356731275],
       [0.9052515860934014, 0.9809937267328175, 0.5019781406261149,
        0.6364778830520974, 0.9326458613141129],
       [0.5784322403641879, 0.8819207998599409, 0.5049697069331524,
        0.49748396942951323, 0.5821743752523317],
       [0.4736468897736671, 0.9990864779008698, 0.9858025417330464,
        0.962535772720943, 0.444962219843124],
       [0.9336825106199576, 0.826435435229793, 0.7681636912075289,
        0.27979704883598877, 0.4353730719800274],
       [0.8447583954590281, 0.07987047193326324, 0.46402335978606046,
        0.9480856495021658, 0.4015360139358938],
       [0.9711539611706229, 0.7187901767728225, 0.36260118715234835,
        0.020589517286836667, 0.5680469382756526],
       [0.6971205672341145, 0.48571415143361363, 0.6185796101425539,
        0.9140144249442194, 0.25735998859836695],
       [0.5109915344152941, 0.8636955537241493, 0.1502889526765261,
        0.22013762284016536, 0.0719394369670876],
       [0.038986207894519675, 0.1935519597354728, 0.04728408017893437,
        0.08094258408223054, 0.4465220116927052],
       [0.29025388425632415, 0.8433400911932378, 0.8729393521730259,
        0.5863255633941525, 0.7907169091097828],
       [0.8637025317109098, 0.18935779269411468, 0.5876123932629506,
        0.5258043036213544, 0.8330868870332293],
       [0.010100892292269736, 0.6160185959583829, 0.4162174430232509,
        0.1210271902211486, 0.0705029207688268],
       [0.877135637013347, 0.7381415137082076, 0.38560151383694063,
        0.22571504327814884, 0.8124127265394223],
       [0.32992622443912534, 0.5232352404766188, 0.4389689605622106,
        0.8115672301211087, 0.08203820449154064],
       [0.6048870470019608, 0.7401457026358532, 0.3657531923652475,
        0.35172206658554084, 0.8024039214516927],
       [0.49120767851458735, 0.8731128985779638, 0.8845498154960095,
        0.8552895796590315, 0.7717510732359477],
       [0.7485152121498821, 0.3507440221119238, 0.520413932218543,
        0.9415567132205326, 0.4180973627771788],
       [0.4631652415741494, 0.550109182333358, 0.9391604152481113,
        0.5776377733560497, 0.2600411623807649],
       [0.942318216782634, 0.5219819009011722, 0.2106808533329816,
        0.6252592622136599, 0.8787664544241482],
       [0.44337104931429594, 0.2972750558704368, 0.674727232655976,
        0.39521580133172385, 0.37245968738602575],
       [0.8760181950761025, 0.7990735355668472, 0.6266976457626433,
        0.22043273969616728, 0.7913828007286487],
       [0.37236133911490776, 0.31818966620815584, 0.049721030939433586,
        0.06942362811618552, 0.35382762320787575],
       [0.16727160883262948, 0.4242468325694482, 0.29033154162675145,
        0.9396488825763931, 0.9899313584118],
       [0.5611968390701916, 0.5874968644110928, 0.7571241108447279,
        0.7343172887441876, 0.6629299640826769],
       [0.49498333028203745, 0.1683291409305202, 0.07688862592341583,
        0.6142825890981317, 0.2699946520098425],
       [0.23439293878195733, 0.36529203317752745, 0.3297051155052745,
        0.45894499936178346, 0.08098972151426631],
       [0.7974684574705937, 0.711588057712207, 0.3776486564651581,
        0.7497961119260393, 0.954301867370623],
       [0.7074864140781626, 0.9243858583209735, 0.13057927737357733,
        0.7252596233879719, 0.12794609541223023],
       [0.3570624749688307, 0.7041579409424606, 0.10528103590121496,
        0.770172617449857, 0.6737618279997201],
       [0.18234476555591894, 0.2982282157428401, 0.07739939571427579,
        0.6996717705836341, 0.8802838230658465],
       [0.7444445955783477, 0.37786132195404487, 0.8682889443055115,
        0.5873133468684398, 0.6516532671417905],
       [0.025196787957347366, 0.07202135554453515, 0.6256213482413684,
        0.28881050598742364, 0.13697731735486685],
       [0.9753376689654785, 0.48013706222260955, 0.5904293683975694,
        0.6593891269011458, 0.7866175144601693],
       [0.7802831674816852, 0.246058233168716, 0.9431646078892385,
        0.1547331099629461, 0.3290206529720152],
       [0.3458153000567762, 0.6936662479306869, 0.48560896883708615,
        0.014187979704132658, 0.6841165831786653],
       [0.0717341633136277, 0.9029570626552105, 0.8038813076484347,
        0.8312079157635713, 0.2685375453266067],
       [0.34015359639957554, 0.7218173513820487, 0.8329511318948813,
        0.4957391032789641, 0.5175889408248089],
       [0.10715201138592056, 0.38906541244086623, 0.9316319536492694,
        0.10860598524834653, 0.40926748412066793],
       [0.9719789395622693, 0.7895923053863384, 0.8785791702293425,
        0.01108660519669602, 0.701204188083733],
       [0.8891145447011006, 0.24068191386067617, 0.5779682069078104,
        0.11609679643000381, 0.8933161551293752],
       [0.6002193533230992, 0.4754349831501652, 0.13313807363495822,
        0.3147869622788325, 0.9555091284962305],
       [0.8533382124453214, 0.21909492329707614, 0.2599851759699716,
        0.12890107173758347, 0.6175483704305498],
       [0.24878729465843852, 0.46412515506087826, 0.11174924762627969,
        0.49447486608641045, 0.6281787724250528],
       [0.43929497428802733, 0.8573400294202217, 0.05466734264658912,
        0.10247431518285821, 0.28457798018580693],
       [0.48803064808351604, 0.01766369405606083, 0.9040816150392459,
        0.12395030029667098, 0.8552290188772976],
       [0.5815358079540635, 0.6577023119820332, 0.9312366703478583,
        0.8490636302625659, 0.9250114112444303],
       [0.6121139410972322, 0.9939119806678938, 0.9587606136779321,
        0.6875413156181828, 0.8826270655717203],
       [0.43312565109930423, 0.08122767137857634, 0.1550923476962257,
        0.47185804277325993, 0.8633048232535965],
       [0.2918216397128287, 0.25677147521756616, 0.11005066393901242,
        0.691229372566863, 0.18786621840206053],
       [0.3398708544953015, 0.1551188710158774, 0.5781186883037597,
        0.007964286070742954, 0.5859048790867717],
       [0.8403400059292072, 0.8783991082157464, 0.4315693004124893,
        0.9622831706128856, 0.24935042992197642],
       [0.7654367652526965, 0.4914561710659284, 0.42940054260807403,
        0.6911481636986253, 0.20359024072297094],
       [0.6498485793336618, 0.30708784473287054, 0.9142039223842489,
        0.7813671426754686, 0.47553669502031926],
       [0.3338862217336064, 0.16616847851926275, 0.13146550875652163,
        0.7769148172424587, 0.15441482521696615],
       [0.8059072709009452, 0.8796595809442305, 0.9901341988482473,
        0.322753338751132, 0.7285337962742353],
       [0.2539947179697456, 0.8615110010583611, 0.0755737547620311,
        0.879969470069771, 0.8873388159741634],
       [0.043890713625203914, 0.8196690510368108, 0.5089360046242792,
        0.5769998740888921, 0.13024698909474897],
       [0.6491115907200714, 0.5187966917114794, 0.49399834451923574,
        0.7678937875230211, 0.224028581682707],
       [0.18870054597737118, 0.4670008423233859, 0.5505853029007466,
        0.07895694278183696, 0.8404231934099279],
       [0.18979125645760397, 0.7737179725530521, 0.6771046541042605,
        0.7535122445905802, 0.635951729789485],
       [0.9350895779894548, 0.5250845053985096, 0.0842333775672841,
        0.8146797862534763, 0.388308953900391],
       [0.31086435545213176, 0.8297453581429037, 0.8246339320036246,
        0.7245704148413025, 0.5464739518179368],
       [0.9734476127836379, 0.5679845721008621, 0.6972557183804398,
        0.543653977391779, 0.36705497489354744],
       [0.541892256497287, 0.28244321657228744, 0.3486521789937249,
        0.8740724993921274, 0.37632073392914767],
       [0.17423705328305206, 0.12329642587259448, 0.4347795118074467,
        0.8708701579579411, 0.2816886922263657],
       [0.5433611074475276, 0.32896750648389494, 0.025529358901092647,
        0.6862133675082447, 0.8601766108358764],
       [0.3736748204276469, 0.7904542194892099, 0.16806586955208014,
        0.35278649379322646, 0.3786476056836876],
       [0.26085050037964364, 0.4065822058541355, 0.9612033946617836,
        0.9826406118054773, 0.429150019890267],
       [0.8598105259278213, 0.9269964181814986, 0.6838647643633751,
        0.9783379284743963, 0.43045751826603773],
       [0.8090420040995848, 0.3439384495804695, 0.24401424079759282,
        0.010167518293353828, 0.6197784919673057],
       [0.7336315290021075, 0.7370282851309371, 0.7454557356507662,
        0.7155667107905914, 0.7065388038472864],
       [0.24242453995803392, 0.1733694312753885, 0.22013317554225909,
        0.06703466013163395, 0.2774427298849268],
       [0.6050063862195787, 0.5158170147794645, 0.7999723712007253,
        0.036735986449347724, 0.190659062286604],
       [0.639593863241268, 0.15947904512611677, 0.15790934623839237,
        0.5186412918765663, 0.013642063049906561],
       [0.3618586140041541, 0.08575987691537557, 0.34470905760756765,
        0.5440726890120972, 0.5711427288967562],
       [0.1423803287901353, 0.03939546543146433, 0.6959047442437757,
        0.14660057088972356, 0.8136810542378289],
       [0.7524960323083747, 0.055521542639556665, 0.34017271523762316,
        0.8728936911019793, 0.6256366309186344],
       [0.557259222314606, 0.26913681439250303, 0.1875523591624071,
        0.9244615193589909, 0.8846061950491672],
       [0.665992286029553, 0.7270346222013517, 0.2524505723945063,
        0.583246267230746, 0.7185072566976214],
       [0.5973153335252408, 0.9425735143342278, 0.7139156130428396,
        0.9163236540458695, 0.9996835235979742],
       [0.838130089218403, 0.08979212779915624, 0.5560947377295381,
        0.013597316338695209, 0.18185419211915355],
       [0.27004766172019956, 0.3128649253694017, 0.9562531031420977,
        0.6367194260707076, 0.6044080487209217]], dtype=object)
```

[44]:

```
newdf.T
```

[44]:
```
          0         1         2         3         4         5         6  \
0    pratik  0.826964  0.991456  0.424973  0.747753  0.507952  0.258392
1  0.194278  0.980641  0.689004  0.200529  0.469301  0.937226  0.261357
2  0.489157  0.241879  0.199466  0.456661  0.599305   0.45836  0.001173
3  0.684807  0.830032   0.47864  0.654169  0.112292  0.393058  0.652658
4   0.15766  0.116351  0.041413   0.42019  0.252662   0.94782    0.9057

          7         8         9  …        90        91        92        93  \
0  0.669887  0.372395  0.968634  …  0.605006  0.639594  0.361859   0.14238
1  0.972192  0.174689  0.908875  …  0.515817  0.159479   0.08576  0.039395
2  0.624571  0.383136  0.470116  …  0.799972  0.157909  0.344709  0.695905
3  0.519191  0.524122  0.714714  …  0.036736  0.518641  0.544073  0.146601
4  0.160775  0.099527  0.996833  …  0.190659  0.013642  0.571143  0.813681

         94        95        96        97        98        99
0  0.752496  0.557259  0.665992  0.597315   0.83813  0.270048
1  0.055522  0.269137  0.727035  0.942574  0.089792  0.312865
2  0.340173  0.187552  0.252451  0.713916  0.556095  0.956253
3  0.872894  0.924462  0.583246  0.916324  0.013597  0.636719
4  0.625637  0.884606  0.718507  0.999684  0.181854  0.604408

[5 rows x 100 columns]
```

[45]:
```
newdf[0][0]=50.67
```

[46]:
```
newdf
```

[46]:
```
           0         1         2         3         4
0      50.67  0.194278  0.489157  0.684807  0.157660
1   0.826964  0.980641  0.241879  0.830032  0.116351
2   0.991456  0.689004  0.199466  0.478640  0.041413
3   0.424973  0.200529  0.456661  0.654169  0.420190
4   0.747753  0.469301  0.599305  0.112292  0.252662
..       …         …         …         …         …
95  0.557259  0.269137  0.187552  0.924462  0.884606
96  0.665992  0.727035  0.252451  0.583246  0.718507
97  0.597315  0.942574  0.713916  0.916324  0.999684
98   0.83813  0.089792  0.556095  0.013597  0.181854
99  0.270048  0.312865  0.956253  0.636719  0.604408

[100 rows x 5 columns]
```

[47]:
```
newdf.head
```

[47]:
```
<bound method NDFrame.head of            0         1         2         3
4
0      50.67  0.194278  0.489157  0.684807  0.157660
```

```
1   0.826964  0.980641  0.241879  0.830032  0.116351
2   0.991456  0.689004  0.199466  0.478640  0.041413
3   0.424973  0.200529  0.456661  0.654169  0.420190
4   0.747753  0.469301  0.599305  0.112292  0.252662
..       …         …         …         …         …
95  0.557259  0.269137  0.187552  0.924462  0.884606
96  0.665992  0.727035  0.252451  0.583246  0.718507
97  0.597315  0.942574  0.713916  0.916324  0.999684
98   0.83813  0.089792  0.556095  0.013597  0.181854
99  0.270048  0.312865  0.956253  0.636719  0.604408

[100 rows x 5 columns]>
```

```
[48]: newdf.head()
```

```
[48]:          0         1         2         3         4
      0     50.67  0.194278  0.489157  0.684807  0.157660
      1  0.826964  0.980641  0.241879  0.830032  0.116351
      2  0.991456  0.689004  0.199466  0.478640  0.041413
      3  0.424973  0.200529  0.456661  0.654169  0.420190
      4  0.747753  0.469301  0.599305  0.112292  0.252662
```

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

```
[49]:           0         1         2         3         4
      99  0.270048  0.312865  0.956253  0.636719  0.604408
      98   0.83813  0.089792  0.556095  0.013597  0.181854
      97  0.597315  0.942574  0.713916  0.916324  0.999684
      96  0.665992  0.727035  0.252451  0.583246  0.718507
      95  0.557259  0.269137  0.187552  0.924462  0.884606
      ..       …         …         …         …         …
      4   0.747753  0.469301  0.599305  0.112292  0.252662
      3   0.424973  0.200529  0.456661  0.654169  0.420190
      2   0.991456  0.689004  0.199466  0.478640  0.041413
      1   0.826964  0.980641  0.241879  0.830032  0.116351
      0      50.67  0.194278  0.489157  0.684807  0.157660

      [100 rows x 5 columns]
```

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

```
[50]:           0         1         2         3         4
      99  0.270048  0.312865  0.956253  0.636719  0.604408
      98   0.83813  0.089792  0.556095  0.013597  0.181854
      97  0.597315  0.942574  0.713916  0.916324  0.999684
      96  0.665992  0.727035  0.252451  0.583246  0.718507
      95  0.557259  0.269137  0.187552  0.924462  0.884606
```

```
..        …         …         …         …         …
4   0.747753  0.469301  0.599305  0.112292  0.252662
3   0.424973  0.200529  0.456661  0.654169  0.420190
2   0.991456  0.689004  0.199466  0.478640  0.041413
1   0.826964  0.980641  0.241879  0.830032  0.116351
0      50.67  0.194278  0.489157  0.684807  0.157660

[100 rows x 5 columns]
```

```
[51]: newdf[0]
```

```
[51]: 0        50.67
      1     0.826964
      2     0.991456
      3     0.424973
      4     0.747753
              …
      95    0.557259
      96    0.665992
      97    0.597315
      98     0.83813
      99    0.270048
      Name: 0, Length: 100, dtype: object
```

```
[52]: newdf.sort_index(axis=1,ascending=False)
```

```
[52]:            4         3         2         1         0
      0   0.157660  0.684807  0.489157  0.194278     50.67
      1   0.116351  0.830032  0.241879  0.980641  0.826964
      2   0.041413  0.478640  0.199466  0.689004  0.991456
      3   0.420190  0.654169  0.456661  0.200529  0.424973
      4   0.252662  0.112292  0.599305  0.469301  0.747753
      ..        …         …         …         …         …
      95  0.884606  0.924462  0.187552  0.269137  0.557259
      96  0.718507  0.583246  0.252451  0.727035  0.665992
      97  0.999684  0.916324  0.713916  0.942574  0.597315
      98  0.181854  0.013597  0.556095  0.089792   0.83813
      99  0.604408  0.636719  0.956253  0.312865  0.270048

      [100 rows x 5 columns]
```

```
[53]: type(newdf[0])
```

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

```
[54]: newdf2=newdf
```

```
[55]: newdf2[0][0]=50
```

```
[56]: newdf
```

```
[56]:           0         1         2         3         4
      0        50  0.194278  0.489157  0.684807  0.157660
      1  0.826964  0.980641  0.241879  0.830032  0.116351
      2  0.991456  0.689004  0.199466  0.478640  0.041413
      3  0.424973  0.200529  0.456661  0.654169  0.420190
      4  0.747753  0.469301  0.599305  0.112292  0.252662
      ..      …         …         …         …         …
      95 0.557259  0.269137  0.187552  0.924462  0.884606
      96 0.665992  0.727035  0.252451  0.583246  0.718507
      97 0.597315  0.942574  0.713916  0.916324  0.999684
      98  0.83813  0.089792  0.556095  0.013597  0.181854
      99 0.270048  0.312865  0.956253  0.636719  0.604408

      [100 rows x 5 columns]
```

```
[57]: newdf2=newdf.copy()
```

```
[58]: newdf2[0][0]=100
```

```
C:\Users\Asus\AppData\Local\Temp\ipykernel_32128\1947304481.py:1: FutureWarning:
ChainedAssignmentError: behaviour will change in pandas 3.0!
You are setting values through chained assignment. Currently this works in
certain cases, but when using Copy-on-Write (which will become the default
behaviour in pandas 3.0) this will never work to update the original DataFrame
or Series, because the intermediate object on which we are setting values will
behave as a copy.
A typical example is when you are setting values in a column of a DataFrame,
like:

df["col"][row_indexer] = value

Use `df.loc[row_indexer, "col"] = values` instead, to perform the assignment in
a single step and ensure this keeps updating the original `df`.

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

  newdf2[0][0]=100
C:\Users\Asus\AppData\Local\Temp\ipykernel_32128\1947304481.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
  newdf2[0][0]=100
```

```
[59]: newdf
```

```
[59]:           0         1         2         3         4
      0        50  0.194278  0.489157  0.684807  0.157660
      1  0.826964  0.980641  0.241879  0.830032  0.116351
      2  0.991456  0.689004  0.199466  0.478640  0.041413
      3  0.424973  0.200529  0.456661  0.654169  0.420190
      4  0.747753  0.469301  0.599305  0.112292  0.252662
      ..      …         …         …         …         …
      95 0.557259  0.269137  0.187552  0.924462  0.884606
      96 0.665992  0.727035  0.252451  0.583246  0.718507
      97 0.597315  0.942574  0.713916  0.916324  0.999684
      98  0.83813  0.089792  0.556095  0.013597  0.181854
      99 0.270048  0.312865  0.956253  0.636719  0.604408

      [100 rows x 5 columns]
```

```
[60]: newdf2.head()
```

```
[60]:          0         1         2         3         4
      0       100  0.194278  0.489157  0.684807  0.157660
      1  0.826964  0.980641  0.241879  0.830032  0.116351
      2  0.991456  0.689004  0.199466  0.478640  0.041413
      3  0.424973  0.200529  0.456661  0.654169  0.420190
      4  0.747753  0.469301  0.599305  0.112292  0.252662
```

```
[61]: newdf.loc[0,0] = 345
```

```
[62]: newdf
```

```
[62]:           0         1         2         3         4
      0       345  0.194278  0.489157  0.684807  0.157660
      1  0.826964  0.980641  0.241879  0.830032  0.116351
      2  0.991456  0.689004  0.199466  0.478640  0.041413
      3  0.424973  0.200529  0.456661  0.654169  0.420190
      4  0.747753  0.469301  0.599305  0.112292  0.252662
      ..      …         …         …         …         …
      95 0.557259  0.269137  0.187552  0.924462  0.884606
      96 0.665992  0.727035  0.252451  0.583246  0.718507
      97 0.597315  0.942574  0.713916  0.916324  0.999684
      98  0.83813  0.089792  0.556095  0.013597  0.181854
      99 0.270048  0.312865  0.956253  0.636719  0.604408

      [100 rows x 5 columns]
```

```
[63]: newdf.head()
```

```
[63]:          0         1         2         3         4
      0        345  0.194278  0.489157  0.684807  0.157660
      1  0.826964  0.980641  0.241879  0.830032  0.116351
      2  0.991456  0.689004  0.199466  0.478640  0.041413
      3  0.424973  0.200529  0.456661  0.654169  0.420190
      4  0.747753  0.469301  0.599305  0.112292  0.252662
```

```
[64]: newdf.columns= list("ABCDE")
```

```
[65]: newdf
```

```
[65]:           A         B         C         D         E
      0         345  0.194278  0.489157  0.684807  0.157660
      1   0.826964  0.980641  0.241879  0.830032  0.116351
      2   0.991456  0.689004  0.199466  0.478640  0.041413
      3   0.424973  0.200529  0.456661  0.654169  0.420190
      4   0.747753  0.469301  0.599305  0.112292  0.252662
      ..        …         …         …         …         …
      95  0.557259  0.269137  0.187552  0.924462  0.884606
      96  0.665992  0.727035  0.252451  0.583246  0.718507
      97  0.597315  0.942574  0.713916  0.916324  0.999684
      98   0.83813  0.089792  0.556095  0.013597  0.181854
      99  0.270048  0.312865  0.956253  0.636719  0.604408

      [100 rows x 5 columns]
```

```
[66]: newdf
```

```
[66]:           A         B         C         D         E
      0         345  0.194278  0.489157  0.684807  0.157660
      1   0.826964  0.980641  0.241879  0.830032  0.116351
      2   0.991456  0.689004  0.199466  0.478640  0.041413
      3   0.424973  0.200529  0.456661  0.654169  0.420190
      4   0.747753  0.469301  0.599305  0.112292  0.252662
      ..        …         …         …         …         …
      95  0.557259  0.269137  0.187552  0.924462  0.884606
      96  0.665992  0.727035  0.252451  0.583246  0.718507
      97  0.597315  0.942574  0.713916  0.916324  0.999684
      98   0.83813  0.089792  0.556095  0.013597  0.181854
      99  0.270048  0.312865  0.956253  0.636719  0.604408

      [100 rows x 5 columns]
```

```
[67]: newdf.loc[1 ,"C"]= 700
```

[68]:
```
newdf
```

[68]:
```
           A         B           C         D         E
0         345  0.194278    0.489157  0.684807  0.157660
1   0.826964  0.980641  700.000000  0.830032  0.116351
2   0.991456  0.689004    0.199466  0.478640  0.041413
3   0.424973  0.200529    0.456661  0.654169  0.420190
4   0.747753  0.469301    0.599305  0.112292  0.252662
..       …       …         …       …       …
95  0.557259  0.269137    0.187552  0.924462  0.884606
96  0.665992  0.727035    0.252451  0.583246  0.718507
97  0.597315  0.942574    0.713916  0.916324  0.999684
98   0.83813  0.089792    0.556095  0.013597  0.181854
99  0.270048  0.312865    0.956253  0.636719  0.604408

[100 rows x 5 columns]
```

[69]:
```
newdf.loc[0,0]=10
```

[70]:
```
newdf
```

[70]:
```
           A         B           C         D         E     0
0         345  0.194278    0.489157  0.684807  0.157660  10.0
1   0.826964  0.980641  700.000000  0.830032  0.116351   NaN
2   0.991456  0.689004    0.199466  0.478640  0.041413   NaN
3   0.424973  0.200529    0.456661  0.654169  0.420190   NaN
4   0.747753  0.469301    0.599305  0.112292  0.252662   NaN
..       …       …         …       …       …   …
95  0.557259  0.269137    0.187552  0.924462  0.884606   NaN
96  0.665992  0.727035    0.252451  0.583246  0.718507   NaN
97  0.597315  0.942574    0.713916  0.916324  0.999684   NaN
98   0.83813  0.089792    0.556095  0.013597  0.181854   NaN
99  0.270048  0.312865    0.956253  0.636719  0.604408   NaN

[100 rows x 6 columns]
```

[71]:
```
newdf.drop(0,axis=1)
```

[71]:
```
           A         B           C         D         E
0         345  0.194278    0.489157  0.684807  0.157660
1   0.826964  0.980641  700.000000  0.830032  0.116351
2   0.991456  0.689004    0.199466  0.478640  0.041413
3   0.424973  0.200529    0.456661  0.654169  0.420190
4   0.747753  0.469301    0.599305  0.112292  0.252662
..       …       …         …       …       …
95  0.557259  0.269137    0.187552  0.924462  0.884606
96  0.665992  0.727035    0.252451  0.583246  0.718507
```

```
97  0.597315  0.942574    0.713916  0.916324  0.999684
98   0.83813  0.089792    0.556095  0.013597  0.181854
99  0.270048  0.312865    0.956253  0.636719  0.604408

[100 rows x 5 columns]
```

[72]:
```
newdf = newdf.drop(0,axis=1)
```

[73]:
```
newdf
```

[73]:
```
           A         B           C         D         E
0        345  0.194278    0.489157  0.684807  0.157660
1   0.826964  0.980641  700.000000  0.830032  0.116351
2   0.991456  0.689004    0.199466  0.478640  0.041413
3   0.424973  0.200529    0.456661  0.654169  0.420190
4   0.747753  0.469301    0.599305  0.112292  0.252662
..       …        …           …        …        …
95  0.557259  0.269137    0.187552  0.924462  0.884606
96  0.665992  0.727035    0.252451  0.583246  0.718507
97  0.597315  0.942574    0.713916  0.916324  0.999684
98   0.83813  0.089792    0.556095  0.013597  0.181854
99  0.270048  0.312865    0.956253  0.636719  0.604408

[100 rows x 5 columns]
```

[74]:
```
newdf.loc[2,'A']=34
```

[75]:
```
newdf
```

[75]:
```
           A         B           C         D         E
0        345  0.194278    0.489157  0.684807  0.157660
1   0.826964  0.980641  700.000000  0.830032  0.116351
2         34  0.689004    0.199466  0.478640  0.041413
3   0.424973  0.200529    0.456661  0.654169  0.420190
4   0.747753  0.469301    0.599305  0.112292  0.252662
..       …        …           …        …        …
95  0.557259  0.269137    0.187552  0.924462  0.884606
96  0.665992  0.727035    0.252451  0.583246  0.718507
97  0.597315  0.942574    0.713916  0.916324  0.999684
98   0.83813  0.089792    0.556095  0.013597  0.181854
99  0.270048  0.312865    0.956253  0.636719  0.604408

[100 rows x 5 columns]
```

[76]:
```
newdf.loc[0,0]=3467
```

[77]:
```
newdf
```

```
[77]:          A         B           C         D         E       0
      0        345  0.194278    0.489157  0.684807  0.157660  3467.0
      1   0.826964  0.980641  700.000000  0.830032  0.116351     NaN
      2         34  0.689004    0.199466  0.478640  0.041413     NaN
      3   0.424973  0.200529    0.456661  0.654169  0.420190     NaN
      4   0.747753  0.469301    0.599305  0.112292  0.252662     NaN
      ..       …         …          …         …        …       …
      95  0.557259  0.269137    0.187552  0.924462  0.884606     NaN
      96  0.665992  0.727035    0.252451  0.583246  0.718507     NaN
      97  0.597315  0.942574    0.713916  0.916324  0.999684     NaN
      98   0.83813  0.089792    0.556095  0.013597  0.181854     NaN
      99  0.270048  0.312865    0.956253  0.636719  0.604408     NaN

      [100 rows x 6 columns]
```

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

```
[78]:          A         B           C         D         E
      0        345  0.194278    0.489157  0.684807  0.157660
      1   0.826964  0.980641  700.000000  0.830032  0.116351
      2         34  0.689004    0.199466  0.478640  0.041413
      3   0.424973  0.200529    0.456661  0.654169  0.420190
      4   0.747753  0.469301    0.599305  0.112292  0.252662
      ..       …         …          …         …        …
      95  0.557259  0.269137    0.187552  0.924462  0.884606
      96  0.665992  0.727035    0.252451  0.583246  0.718507
      97  0.597315  0.942574    0.713916  0.916324  0.999684
      98   0.83813  0.089792    0.556095  0.013597  0.181854
      99  0.270048  0.312865    0.956253  0.636719  0.604408

      [100 rows x 5 columns]
```

```
[79]: newdf.loc[[1,2],['A','C']]
```

```
[79]:         A           C
      1  0.826964  700.000000
      2        34    0.199466
```

```
[80]: newdf.loc[[1,2],:]
```

```
[80]:         A         B           C         D         E    0
      1  0.826964  0.980641  700.000000  0.830032  0.116351 NaN
      2        34  0.689004    0.199466  0.478640  0.041413 NaN
```

```
[81]: newdf.loc[:,['A','B','C']]
```

```
[81]:            A         B           C
      0        345  0.194278    0.489157
      1   0.826964  0.980641  700.000000
      2         34  0.689004    0.199466
      3   0.424973  0.200529    0.456661
      4   0.747753  0.469301    0.599305
      ..       …         …           …
      95  0.557259  0.269137    0.187552
      96  0.665992  0.727035    0.252451
      97  0.597315  0.942574    0.713916
      98   0.83813  0.089792    0.556095
      99  0.270048  0.312865    0.956253

      [100 rows x 3 columns]
```

```
[82]: newdf.loc[(newdf['A']<0.3)&(newdf['C']>0.4)]
```

```
[82]:            A         B         C         D         E   0
      13  0.212357  0.524646  0.985818  0.947484  0.848913 NaN
      27  0.290254  0.843340  0.872939  0.586326  0.790717 NaN
      29  0.010101  0.616019  0.416217  0.121027  0.070503 NaN
      49  0.025197  0.072021  0.625621  0.288811  0.136977 NaN
      53  0.071734  0.902957  0.803881  0.831208  0.268538 NaN
      55  0.107152  0.389065  0.931632  0.108606  0.409267 NaN
      74  0.043891  0.819669  0.508936  0.577000  0.130247 NaN
      76  0.188701  0.467001  0.550585  0.078957  0.840423 NaN
      77  0.189791  0.773718  0.677105  0.753512  0.635952 NaN
      82  0.174237  0.123296  0.434780  0.870870  0.281689 NaN
      85  0.260851  0.406582  0.961203  0.982641  0.429150 NaN
      93   0.14238  0.039395  0.695905  0.146601  0.813681 NaN
      99  0.270048  0.312865  0.956253  0.636719  0.604408 NaN
```

```
[83]: newdf.iloc[1,3]
```

```
[83]: np.float64(0.8300319201950969)
```

```
[84]: newdf.iloc[0,3]=23
```

```
[85]: newdf
```

```
[85]:            A         B           C          D         E       0
      0        345  0.194278    0.489157  23.000000  0.157660  3467.0
      1   0.826964  0.980641  700.000000   0.830032  0.116351     NaN
      2         34  0.689004    0.199466   0.478640  0.041413     NaN
      3   0.424973  0.200529    0.456661   0.654169  0.420190     NaN
      4   0.747753  0.469301    0.599305   0.112292  0.252662     NaN
      ..       …         …           …          …         …       …
```

```
95  0.557259  0.269137    0.187552   0.924462  0.884606     NaN
96  0.665992  0.727035    0.252451   0.583246  0.718507     NaN
97  0.597315  0.942574    0.713916   0.916324  0.999684     NaN
98   0.83813  0.089792    0.556095   0.013597  0.181854     NaN
99  0.270048  0.312865    0.956253   0.636719  0.604408     NaN

[100 rows x 6 columns]
```

```
[86]: newdf.drop([0])
```

```
[86]:
           A         B           C         D         E   0
1   0.826964  0.980641  700.000000  0.830032  0.116351 NaN
2         34  0.689004    0.199466  0.478640  0.041413 NaN
3   0.424973  0.200529    0.456661  0.654169  0.420190 NaN
4   0.747753  0.469301    0.599305  0.112292  0.252662 NaN
5   0.507952  0.937226    0.458360  0.393058  0.947820 NaN
..        …       …         …       …       …  ..
95  0.557259  0.269137    0.187552  0.924462  0.884606 NaN
96  0.665992  0.727035    0.252451  0.583246  0.718507 NaN
97  0.597315  0.942574    0.713916  0.916324  0.999684 NaN
98   0.83813  0.089792    0.556095  0.013597  0.181854 NaN
99  0.270048  0.312865    0.956253  0.636719  0.604408 NaN

[99 rows x 6 columns]
```

```
[87]: newdf.drop(['A','C'],axis=1)
```

```
[87]:
           B          D         E       0
0   0.194278  23.000000  0.157660  3467.0
1   0.980641   0.830032  0.116351     NaN
2   0.689004   0.478640  0.041413     NaN
3   0.200529   0.654169  0.420190     NaN
4   0.469301   0.112292  0.252662     NaN
..       …        …       …     …
95  0.269137   0.924462  0.884606     NaN
96  0.727035   0.583246  0.718507     NaN
97  0.942574   0.916324  0.999684     NaN
98  0.089792   0.013597  0.181854     NaN
99  0.312865   0.636719  0.604408     NaN

[100 rows x 4 columns]
```

```
[88]: newdf.drop(['A','D'],axis=1)
```

```
[88]:
           B           C         E       0
0   0.194278    0.489157  0.157660  3467.0
1   0.980641  700.000000  0.116351     NaN
```

```
2   0.689004    0.199466  0.041413     NaN
3   0.200529    0.456661  0.420190     NaN
4   0.469301    0.599305  0.252662     NaN
..       …         …       …     …
95  0.269137    0.187552  0.884606     NaN
96  0.727035    0.252451  0.718507     NaN
97  0.942574    0.713916  0.999684     NaN
98  0.089792    0.556095  0.181854     NaN
99  0.312865    0.956253  0.604408     NaN

[100 rows x 4 columns]
```

[89]:

```
newdf.drop(['A','B'],axis=1,inplace=True)
```

[90]:

```
newdf
```

[90]:

```
             C          D         E       0
0     0.489157  23.000000  0.157660  3467.0
1   700.000000   0.830032  0.116351     NaN
2     0.199466   0.478640  0.041413     NaN
3     0.456661   0.654169  0.420190     NaN
4     0.599305   0.112292  0.252662     NaN
..         …        …       …     …
95    0.187552   0.924462  0.884606     NaN
96    0.252451   0.583246  0.718507     NaN
97    0.713916   0.916324  0.999684     NaN
98    0.556095   0.013597  0.181854     NaN
99    0.956253   0.636719  0.604408     NaN

[100 rows x 4 columns]
```

[91]:

```
newdf.drop([1,2],inplace=True)
```

[92]:

```
newdf
```

[92]:

```
           C          D         E       0
0   0.489157  23.000000  0.157660  3467.0
3   0.456661   0.654169  0.420190     NaN
4   0.599305   0.112292  0.252662     NaN
5   0.458360   0.393058  0.947820     NaN
6   0.001173   0.652658  0.905700     NaN
..       …        …       …     …
95  0.187552   0.924462  0.884606     NaN
96  0.252451   0.583246  0.718507     NaN
97  0.713916   0.916324  0.999684     NaN
98  0.556095   0.013597  0.181854     NaN
99  0.956253   0.636719  0.604408     NaN
```

```
[98 rows x 4 columns]
```

[93]:

```
newdf.reset_index()
```

[93]:

```
    index         C          D         E       0
0       0  0.489157  23.000000  0.157660  3467.0
1       3  0.456661   0.654169  0.420190     NaN
2       4  0.599305   0.112292  0.252662     NaN
3       5  0.458360   0.393058  0.947820     NaN
4       6  0.001173   0.652658  0.905700     NaN
..    …       …          …         …      …
93     95  0.187552   0.924462  0.884606     NaN
94     96  0.252451   0.583246  0.718507     NaN
95     97  0.713916   0.916324  0.999684     NaN
96     98  0.556095   0.013597  0.181854     NaN
97     99  0.956253   0.636719  0.604408     NaN

[98 rows x 5 columns]
```

[94]:

```
newdf.reset_index(drop=True)
```

[94]:

```
           C          D         E       0
0   0.489157  23.000000  0.157660  3467.0
1   0.456661   0.654169  0.420190     NaN
2   0.599305   0.112292  0.252662     NaN
3   0.458360   0.393058  0.947820     NaN
4   0.001173   0.652658  0.905700     NaN
..       …          …         …      …
93  0.187552   0.924462  0.884606     NaN
94  0.252451   0.583246  0.718507     NaN
95  0.713916   0.916324  0.999684     NaN
96  0.556095   0.013597  0.181854     NaN
97  0.956253   0.636719  0.604408     NaN

[98 rows x 4 columns]
```

[95]:

```
newdf.reset_index(drop=True,inplace=True)
```

[96]:

```
newdf
```

[96]:

```
           C          D         E       0
0   0.489157  23.000000  0.157660  3467.0
1   0.456661   0.654169  0.420190     NaN
2   0.599305   0.112292  0.252662     NaN
3   0.458360   0.393058  0.947820     NaN
4   0.001173   0.652658  0.905700     NaN
```

```
..        …         …        …     …
93  0.187552   0.924462  0.884606     NaN
94  0.252451   0.583246  0.718507     NaN
95  0.713916   0.916324  0.999684     NaN
96  0.556095   0.013597  0.181854     NaN
97  0.956253   0.636719  0.604408     NaN

[98 rows x 4 columns]
```

[97]:
```
newdf['C'].isnull()
```

[97]:
```
0     False
1     False
2     False
3     False
4     False
      …
93    False
94    False
95    False
96    False
97    False
Name: C, Length: 98, dtype: bool
```

[98]:
```
newdf['D'].isnull()
```

[98]:
```
0     False
1     False
2     False
3     False
4     False
      …
93    False
94    False
95    False
96    False
97    False
Name: D, Length: 98, dtype: bool
```

[99]:
```
newdf['C']=None
```

[100]:
```
newdf
```

[100]:
```
       C          D         E       0
0   None  23.000000  0.157660  3467.0
1   None   0.654169  0.420190     NaN
2   None   0.112292  0.252662     NaN
```

```
3   None   0.393058  0.947820     NaN
4   None   0.652658  0.905700     NaN
..   …         …        …      …
93  None   0.924462  0.884606     NaN
94  None   0.583246  0.718507     NaN
95  None   0.916324  0.999684     NaN
96  None   0.013597  0.181854     NaN
97  None   0.636719  0.604408     NaN

[98 rows x 4 columns]
```

```
[101]: newdf['C'].isnull()
```

```
[101]: 0     True
       1     True
       2     True
       3     True
       4     True
             …
       93    True
       94    True
       95    True
       96    True
       97    True
       Name: C, Length: 98, dtype: bool
```

```
[102]: newdf.iloc[1,2]=None
```

```
[103]: newdf
```

```
[103]:       C          D         E       0
       0   None  23.000000  0.157660  3467.0
       1   None   0.654169       NaN     NaN
       2   None   0.112292  0.252662     NaN
       3   None   0.393058  0.947820     NaN
       4   None   0.652658  0.905700     NaN
       ..   …         …        …      …
       93  None   0.924462  0.884606     NaN
       94  None   0.583246  0.718507     NaN
       95  None   0.916324  0.999684     NaN
       96  None   0.013597  0.181854     NaN
       97  None   0.636719  0.604408     NaN

       [98 rows x 4 columns]
```

```
[104]: newdf.loc[[1,2],:]=None
```

```
[105]: newdf
```

```
[105]:       C          D         E       0
       0   None  23.000000  0.157660  3467.0
       1   None        NaN       NaN     NaN
       2   None        NaN       NaN     NaN
       3   None   0.393058  0.947820     NaN
       4   None   0.652658  0.905700     NaN
       ..   …          …         …      …
       93  None   0.924462  0.884606     NaN
       94  None   0.583246  0.718507     NaN
       95  None   0.916324  0.999684     NaN
       96  None   0.013597  0.181854     NaN
       97  None   0.636719  0.604408     NaN

       [98 rows x 4 columns]
```

```
[106]: newdf.loc[:,['C',0]]=None
```

```
C:\Users\Asus\AppData\Local\Temp\ipykernel_32128\4173185361.py:1: FutureWarning:
Setting an item of incompatible dtype is deprecated and will raise in a future
error of pandas. Value 'None' has dtype incompatible with float64, please
explicitly cast to a compatible dtype first.
  newdf.loc[:,['C',0]]=None
```

```
[107]: newdf
```

```
[107]:       C          D         E     0
       0   None  23.000000  0.157660  None
       1   None        NaN       NaN  None
       2   None        NaN       NaN  None
       3   None   0.393058  0.947820  None
       4   None   0.652658  0.905700  None
       ..   …          …         …    …
       93  None   0.924462  0.884606  None
       94  None   0.583246  0.718507  None
       95  None   0.916324  0.999684  None
       96  None   0.013597  0.181854  None
       97  None   0.636719  0.604408  None

       [98 rows x 4 columns]
```

```
[108]: newdf.loc[:,['C',0]]=23
```

```
[109]: newdf
```

[109]:
```
     C          D         E   0
0   23  23.000000  0.157660  23
1   23        NaN       NaN  23
2   23        NaN       NaN  23
3   23   0.393058  0.947820  23
4   23   0.652658  0.905700  23
..  ..        …         …  ..
93  23   0.924462  0.884606  23
94  23   0.583246  0.718507  23
95  23   0.916324  0.999684  23
96  23   0.013597  0.181854  23
97  23   0.636719  0.604408  23

[98 rows x 4 columns]
```

[110]:
```
newdf.drop(0,axis=1)
```

[110]:
```
     C          D         E
0   23  23.000000  0.157660
1   23        NaN       NaN
2   23        NaN       NaN
3   23   0.393058  0.947820
4   23   0.652658  0.905700
..  ..        …         …
93  23   0.924462  0.884606
94  23   0.583246  0.718507
95  23   0.916324  0.999684
96  23   0.013597  0.181854
97  23   0.636719  0.604408

[98 rows x 3 columns]
```

[111]:
```
newdf.drop(0,axis=1,inplace=True)
```

[112]:
```
newdf
```

[112]:
```
     C          D         E
0   23  23.000000  0.157660
1   23        NaN       NaN
2   23        NaN       NaN
3   23   0.393058  0.947820
4   23   0.652658  0.905700
..  ..        …         …
93  23   0.924462  0.884606
94  23   0.583246  0.718507
95  23   0.916324  0.999684
96  23   0.013597  0.181854
```

```
97  23   0.636719  0.604408

[98 rows x 3 columns]
```

[113]:

```
newdf.to_csv('newdf.csv',index=False)
```

[114]:

```
newdf
```

[114]:

```
     C          D         E
0   23  23.000000  0.157660
1   23        NaN       NaN
2   23        NaN       NaN
3   23   0.393058  0.947820
4   23   0.652658  0.905700
..  ..        …         …
93  23   0.924462  0.884606
94  23   0.583246  0.718507
95  23   0.916324  0.999684
96  23   0.013597  0.181854
97  23   0.636719  0.604408

[98 rows x 3 columns]
```

[115]:

```
newdf.reset_index(drop=True)
```

[115]:

```
     C          D         E
0   23  23.000000  0.157660
1   23        NaN       NaN
2   23        NaN       NaN
3   23   0.393058  0.947820
4   23   0.652658  0.905700
..  ..        …         …
93  23   0.924462  0.884606
94  23   0.583246  0.718507
95  23   0.916324  0.999684
96  23   0.013597  0.181854
97  23   0.636719  0.604408

[98 rows x 3 columns]
```

[116]:

```
df
```

[116]:

```
     name  marks      city
0  pratik     92  balasore
1    debi     98   bhubans
2   manoj     98   bhadrak
3  prabhu     69   cuttack
```

```
[117]: df.dropna()
```

```
[117]:      name  marks      city
       0  pratik     92  balasore
       1    debi     98   bhubans
       2   manoj     98   bhadrak
       3  prabhu     69   cuttack
```

```
[118]: df.dropna(how='all')
```

```
[118]:      name  marks      city
       0  pratik     92  balasore
       1    debi     98   bhubans
       2   manoj     98   bhadrak
       3  prabhu     69   cuttack
```

```
[119]: df.dropna(how='all')
```

```
[119]:      name  marks      city
       0  pratik     92  balasore
       1    debi     98   bhubans
       2   manoj     98   bhadrak
       3  prabhu     69   cuttack
```

```
[120]: df.dropna(how='all',axis=1)
```

```
[120]:      name  marks      city
       0  pratik     92  balasore
       1    debi     98   bhubans
       2   manoj     98   bhadrak
       3  prabhu     69   cuttack
```

```
[121]: df
```

```
[121]:      name  marks      city
       0  pratik     92  balasore
       1    debi     98   bhubans
       2   manoj     98   bhadrak
       3  prabhu     69   cuttack
```

```
[122]: df.dropna()
```

```
[122]:      name  marks      city
       0  pratik     92  balasore
       1    debi     98   bhubans
       2   manoj     98   bhadrak
       3  prabhu     69   cuttack
```

[123]:
```
df
```

[123]:
```
     name  marks      city
0  pratik     92  balasore
1    debi     98   bhubans
2   manoj     98   bhadrak
3  prabhu     69   cuttack
```

[124]:
```
df.to_csv('dataframe.csv',index=False)
```

[125]:
```
df = pd.DataFrame({"Name":['pratik','prabhu','pratik'],
                   "toy":[np.nan,'batmobile','bullwhip'],
                   "born":[pd.NaT,pd.Timestamp("1940-04-25"),pd.NaT]})
```

[126]:
```
df
```

[126]:
```
     Name        toy       born
0  pratik        NaN        NaT
1  prabhu  batmobile 1940-04-25
2  pratik   bullwhip        NaT
```

[127]:
```
df.drop_duplicates(subset=['Name'],keep='first')
```

[127]:
```
     Name        toy       born
0  pratik        NaN        NaT
1  prabhu  batmobile 1940-04-25
```

[128]:
```
df.drop_duplicates(subset=['Name'] , keep='last')
```

[128]:
```
     Name        toy       born
1  prabhu  batmobile 1940-04-25
2  pratik   bullwhip        NaT
```

[129]:
```
df.drop_duplicates(subset=['Name'] , keep=False)
```

[129]:
```
     Name        toy       born
1  prabhu  batmobile 1940-04-25
```

[130]:
```
df.shape
```

[130]:
```
(3, 3)
```

[131]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3 entries, 0 to 2
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
```

```
---  ------  --------------  -----
 0   Name    3 non-null      object
 1   toy     2 non-null      object
 2   born    1 non-null      datetime64[ns]
dtypes: datetime64[ns](1), object(2)
memory usage: 204.0+ bytes
```

```
[132]: df['Name'].value_counts(dropna=True)
```

```
[132]: Name
       pratik    2
       prabhu    1
       Name: count, dtype: int64
```

```
[133]: df['Name'].value_counts(dropna=False)
```

```
[133]: Name
       pratik    2
       prabhu    1
       Name: count, dtype: int64
```

```
[134]: df.isnull()
```

```
[134]:     Name    toy   born
       0  False   True   True
       1  False  False  False
       2  False  False   True
```

```
[143]: data = pd.read_excel('data.xlsx')
```

```
[142]: data
```

```
[142]:    Unnamed: 0    name  marks      city
       0           0  pratik     92  balasore
       1           1    debi     98   bhubans
       2           2   manoj     98   bhadrak
       3           3  prabhu     69   cuttack
```

```
[144]: data = pd.read_excel('data.xlsx',sheet_name='data')
```

```
[145]: data
```

```
[145]:    Unnamed: 0     name  marks      city
       0           0   pratik     92  balasore
       1           1  sdhfhsj     98   bhubans
       2           2    manoj     98   bhadrak
       3           3   prabhu     69   cuttack
```

```
[146]: data.iloc[0,0]=50
```

```
[147]: data
```

```
[147]:    Unnamed: 0     name  marks      city
       0          50   pratik     92  balasore
       1           1  sdhfhsj     98   bhubans
       2           2    manoj     98   bhadrak
       3           3   prabhu     69   cuttack
```

```
[150]: data.to_excel('data.xlsx',sheet_name='pratik')
```

```
[151]: data
```

```
[151]:    Unnamed: 0     name  marks      city
       0          50   pratik     92  balasore
       1           1  sdhfhsj     98   bhubans
       2           2    manoj     98   bhadrak
       3           3   prabhu     69   cuttack
```

```
  [ ]:
```