



Search or jump to...

/

Pulls Issues Marketplace Explore



kawshikbuet17 / Python

Unwatch

1

Star

1

Fork

0

Code

Issues

Pull requests

Actions

Projects

Wiki

Security

...

master

Python / pandas / Pandas_Handbook.ipynb

Go to file

...



kawshikbuet17 pandas rerun and requirements.txt added

Latest commit 9263d32 25 days ago

History

1 contributor

7683 lines (7683 sloc) | 224 KB



Raw

Blame

In [1]: import numpy as np
import pandas as pdIn [2]: dict1 = {
 "name":['kawshik', 'mim', 'mohana', 'fahim'],
 "marks": [92, 34, 24, 17],
 "city": ['naogaon', 'barishal', 'mymensingh', 'dhaka']
}

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

In [4]: df

Out[4]:

	name	marks	city
0	kawshik	92	naogaon
1	mim	34	barishal
2	mohana	24	mymensingh
3	fahim	17	dhaka

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

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

In [7]: df.head(2)

Out[7]:

	name	marks	city
0	kawshik	92	naogaon
1	mim	34	barishal

In [8]: df.tail(2)

Out[8]:

	name	marks	city
2	mohana	24	mymensingh
3	fahim	17	dhaka

In [9]: df.describe()

Out[9]:

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

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

```
In [11]: kawshik
```

	Train No	Speed	City
0	96859	56	naogaon
1	49494	234	barishal
2	14567	31	mymensingh
3	49678	76	dhaka

```
In [12]: kawshik['Speed']
```

```
Out[12]: 0      56
          1     234
          2      31
          3      76
Name: Speed, dtype: int64
```

```
In [13]: kawshik['Speed'][0]
```

```
Out[13]: 56
```

```
In [14]: kawshik['Speed'][0] = 50
```

```
d:\coding\python\venv\lib\site-packages\ipykernel_launcher.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
    """Entry point for launching an IPython kernel.
```

```
In [15]: kawshik
```

	Train No	Speed	City
0	96859	50	naogaon
1	49494	234	barishal
2	14567	31	mymensingh
3	49678	76	dhaka

```
In [16]: kawshik.to_csv('kawshik.csv')
```

```
In [17]: kawshik
```

	Train No	Speed	City
0	96859	50	naogaon
1	49494	234	barishal
2	14567	31	mymensingh
3	49678	76	dhaka

In [18]: `kawshik.index = ['first', 'second', 'third', 'forth']`

In [19]: `kawshik`

Out[19]:

	Train No	Speed	City
first	96859	50	naogaon
second	49494	234	barishal
third	14567	31	mymensingh
forth	49678	76	dhaka

In [20]: `kawshik['Speed']`

Out[20]:

first	50
second	234
third	31
forth	76

Name: Speed, dtype: int64

In [21]: `type(kawshik['Speed'])`

Out[21]: `pandas.core.series.Series`

In [22]: `type(kawshik)`

Out[22]: `pandas.core.frame.DataFrame`

In [23]: `type(kawshik['City'])`

Out[23]: `pandas.core.series.Series`

In [24]: `ser = pd.Series(np.random.rand(34))`

In [25]: `ser`

Out[25]:

0	0.408625
1	0.958209
2	0.050102
3	0.943148
4	0.988070
5	0.201819
6	0.021301
7	0.209862
8	0.786548
9	0.685465
10	0.662113
11	0.131019
12	0.879929
13	0.241299
14	0.652830
15	0.736738
16	0.623727
17	0.293467
18	0.554056
19	0.912506
20	0.665680
21	0.118875
22	0.519187
23	0.187980
24	0.261654
25	0.996156
26	0.728173
27	0.505267
28	0.324265
29	0.096287
30	0.449520
31	0.154427
32	0.672149
33	0.902211

```
... v.0.22.1
dtype: float64
```

```
In [26]: type(ser)
```

```
Out[26]: pandas.core.series.Series
```

```
In [27]: newdf = pd.DataFrame(np.random.rand(334, 5), index=np.arange(334))
```

```
In [28]: newdf.head
```

```
Out[28]: <bound method NDFrame.head of
          0      1      2      3      4
0  0.799405  0.746828  0.119205  0.608367  0.786477
1  0.795904  0.457078  0.432732  0.753908  0.220771
2  0.355519  0.660129  0.693084  0.058004  0.326732
3  0.278972  0.760987  0.535669  0.533717  0.379935
4  0.029968  0.483665  0.278039  0.511621  0.288220
...
329 0.706344  0.611424  0.103477  0.843983  0.877947
330 0.499120  0.638131  0.957900  0.284939  0.221631
331 0.690365  0.246217  0.394901  0.448833  0.517266
332 0.572044  0.055095  0.311525  0.994674  0.135890
333 0.720756  0.880090  0.365951  0.525556  0.764522
[334 rows x 5 columns]>
```

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

```
Out[29]: <bound method NDFrame.tail of
          0      1      2      3      4
0  0.799405  0.746828  0.119205  0.608367  0.786477
1  0.795904  0.457078  0.432732  0.753908  0.220771
2  0.355519  0.660129  0.693084  0.058004  0.326732
3  0.278972  0.760987  0.535669  0.533717  0.379935
4  0.029968  0.483665  0.278039  0.511621  0.288220
...
329 0.706344  0.611424  0.103477  0.843983  0.877947
330 0.499120  0.638131  0.957900  0.284939  0.221631
331 0.690365  0.246217  0.394901  0.448833  0.517266
332 0.572044  0.055095  0.311525  0.994674  0.135890
333 0.720756  0.880090  0.365951  0.525556  0.764522
[334 rows x 5 columns]>
```

```
In [30]: newdf
```

```
Out[30]:
```

	0	1	2	3	4
0	0.799405	0.746828	0.119205	0.608367	0.786477
1	0.795904	0.457078	0.432732	0.753908	0.220771
2	0.355519	0.660129	0.693084	0.058004	0.326732
3	0.278972	0.760987	0.535669	0.533717	0.379935
4	0.029968	0.483665	0.278039	0.511621	0.288220
...
329	0.706344	0.611424	0.103477	0.843983	0.877947
330	0.499120	0.638131	0.957900	0.284939	0.221631
331	0.690365	0.246217	0.394901	0.448833	0.517266
332	0.572044	0.055095	0.311525	0.994674	0.135890
333	0.720756	0.880090	0.365951	0.525556	0.764522

```
334 rows x 5 columns
```

```
In [31]: type(newdf)
```

```
Out[31]: pandas.core.frame.DataFrame
```

```
In [32]: newdf.describe()
```

```
Out[32]:
```

	0	1	2	3	4
--	---	---	---	---	---

	count	334.000000	334.000000	334.000000	334.000000	334.000000
	mean	0.513096	0.493616	0.477430	0.480567	0.510105
	std	0.286507	0.288004	0.288410	0.287394	0.287044
	min	0.009405	0.000872	0.000328	0.002398	0.002146
	25%	0.273721	0.246986	0.217492	0.230989	0.279995
	50%	0.509766	0.481127	0.471246	0.472999	0.493294
	75%	0.769157	0.747412	0.726509	0.729428	0.767157
	max	0.996351	0.998968	0.998639	0.999197	0.998879

In [33]: newdf.dtypes

Out[33]: 0 float64
1 float64
2 float64
3 float64
4 float64
dtype: object

In [34]: newdf[0][0] = "kawshik"

In [35]: newdf

	0	1	2	3	4
0	kawshik	0.746828	0.119205	0.608367	0.786477
1	0.795904	0.457078	0.432732	0.753908	0.220771
2	0.355519	0.660129	0.693084	0.058004	0.326732
3	0.278972	0.760987	0.535669	0.533717	0.379935
4	0.0299681	0.483665	0.278039	0.511621	0.288220
...
329	0.706344	0.611424	0.103477	0.843983	0.877947
330	0.49912	0.638131	0.957900	0.284939	0.221631
331	0.690365	0.246217	0.394901	0.448833	0.517266
332	0.572044	0.055095	0.311525	0.994674	0.135890
333	0.720756	0.880090	0.365951	0.525556	0.764522

334 rows × 5 columns

In [36]: newdf.dtypes

Out[36]: 0 object
1 float64
2 float64
3 float64
4 float64
dtype: object

In [37]: newdf.index

Out[37]: Int64Index([0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
...,
324, 325, 326, 327, 328, 329, 330, 331, 332, 333],
dtype='int64', length=334)

In [38]: newdf.columns

Out[38]: RangeIndex(start=0, stop=5, step=1)

In [39]: newdf.to_numpy()

Out[39]: array([['kawshik', 0.7468280270640056, 0.11920502426756574,
0.608367111772112, 0.7864772610002004],

```
0.88856/4611/35413, 0./864//5618905984],  
[0.7959037030131311, 0.45707764926212713, 0.4327318960508558,  
0.7539077119089841, 0.22077067322180044],  
[0.35551882438767723, 0.6601287028473476, 0.6930844521588839,  
0.058003737174239234, 0.326731727999359],  
...,  
[0.6903648875868753, 0.24621745410772, 0.39490078854755184,  
0.4488328297446289, 0.5172662921371345],  
[0.5720436956060753, 0.05509470369302227, 0.31152468771458264,  
0.9946744429792433, 0.13588963508998675],  
[0.7207556460259394, 0.8800899770480043, 0.3659510118066531,  
0.5255562368227233, 0.7645221099695815]], dtype=object)
```

In [40]: newdf[0][0] = 0.3

```
d:\coding\python\venv\lib\site-packages\ipykernel_launcher.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  
    """Entry point for launching an IPython kernel.
```

In [41]: newdf.head()

Out[41]:

	0	1	2	3	4
0	0.3	0.746828	0.119205	0.608367	0.786477
1	0.795904	0.457078	0.432732	0.753908	0.220771
2	0.355519	0.660129	0.693084	0.058004	0.326732
3	0.278972	0.760987	0.535669	0.533717	0.379935
4	0.0299681	0.483665	0.278039	0.511621	0.288220

In [42]: newdf.to_numpy()

Out[42]: array([[0.3, 0.7468280270640056, 0.11920502426756574, 0.6083674611733413,
0.7864773618903984],
[0.7959037030131311, 0.45707764926212713, 0.4327318960508558,
0.7539077119089841, 0.22077067322180044],
[0.35551882438767723, 0.6601287028473476, 0.6930844521588839,
0.058003737174239234, 0.326731727999359],
...,
[0.6903648875868753, 0.24621745410772, 0.39490078854755184,
0.4488328297446289, 0.5172662921371345],
[0.5720436956060753, 0.05509470369302227, 0.31152468771458264,
0.9946744429792433, 0.13588963508998675],
[0.7207556460259394, 0.8800899770480043, 0.3659510118066531,
0.5255562368227233, 0.7645221099695815]], dtype=object)

In [43]: newdf.T

Out[43]:

	0	1	2	3	4	5	6	7	8	9
0	0.3	0.795904	0.355519	0.278972	0.0299681	0.291862	0.834531	0.924205	0.129359	0.63
1	0.746828	0.457078	0.660129	0.760987	0.483665	0.599023	0.431699	0.857907	0.88125	0.46
2	0.119205	0.432732	0.693084	0.535669	0.278039	0.974973	0.474514	0.110374	0.459516	0.65
3	0.608367	0.753908	0.0580037	0.533717	0.511621	0.311611	0.0612383	0.762608	0.957592	0.94
4	0.786477	0.220771	0.326732	0.379935	0.28822	0.935842	0.900006	0.919642	0.560121	0.04

5 rows × 334 columns

In [44]: newdf.head()

Out[44]:

	0	1	2	3	4
0	0.3	0.746828	0.119205	0.608367	0.786477
1	0.795904	0.457078	0.432732	0.753908	0.220771
2	0.355519	0.660129	0.693084	0.058004	0.326732

3	0.278972	0.760987	0.535669	0.533717	0.379935
4	0.0299681	0.483665	0.278039	0.511621	0.288220

```
In [45]: newdf.sort_index(axis=0, ascending=False)
```

Out[45]:

	0	1	2	3	4
333	0.720756	0.880090	0.365951	0.525556	0.764522
332	0.572044	0.055095	0.311525	0.994674	0.135890
331	0.690365	0.246217	0.394901	0.448833	0.517266
330	0.49912	0.638131	0.957900	0.284939	0.221631
329	0.706344	0.611424	0.103477	0.843983	0.877947
...
4	0.0299681	0.483665	0.278039	0.511621	0.288220
3	0.278972	0.760987	0.535669	0.533717	0.379935
2	0.355519	0.660129	0.693084	0.058004	0.326732
1	0.795904	0.457078	0.432732	0.753908	0.220771
0	0.3	0.746828	0.119205	0.608367	0.786477

334 rows × 5 columns

```
In [46]: newdf.sort_index(axis=1, ascending=False)
```

Out[46]:

	4	3	2	1	0
0	0.786477	0.608367	0.119205	0.746828	0.3
1	0.220771	0.753908	0.432732	0.457078	0.795904
2	0.326732	0.058004	0.693084	0.660129	0.355519
3	0.379935	0.533717	0.535669	0.760987	0.278972
4	0.288220	0.511621	0.278039	0.483665	0.0299681
...
329	0.877947	0.843983	0.103477	0.611424	0.706344
330	0.221631	0.284939	0.957900	0.638131	0.49912
331	0.517266	0.448833	0.394901	0.246217	0.690365
332	0.135890	0.994674	0.311525	0.055095	0.572044
333	0.764522	0.525556	0.365951	0.880090	0.720756

334 rows × 5 columns

```
In [47]: type(newdf)
```

Out[47]: pandas.core.frame.DataFrame

```
In [48]: newdf[0]
```

```
Out[48]: 0      0.3
1      0.795904
2      0.355519
3      0.278972
4      0.0299681
...
329    0.706344
330    0.49912
331    0.690365
332    0.572044
333    0.720756
Name: 0, Length: 334, dtype: object
```

```
In [49]: type(newdf[0])
```

```
Out[49]: pandas.core.series.Series
```

```
In [50]: newdf2 = newdf
```

```
In [51]: newdf2[0][0] = 1705043
```

```
d:\coding\python\venv\lib\site-packages\ipykernel_launcher.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
```

```
    """Entry point for launching an IPython kernel.
```

```
In [52]: newdf
```

```
Out[52]:
```

	0	1	2	3	4
0	1705043	0.746828	0.119205	0.608367	0.786477
1	0.795904	0.457078	0.432732	0.753908	0.220771
2	0.355519	0.660129	0.693084	0.058004	0.326732
3	0.278972	0.760987	0.535669	0.533717	0.379935
4	0.0299681	0.483665	0.278039	0.511621	0.288220
...
329	0.706344	0.611424	0.103477	0.843983	0.877947
330	0.49912	0.638131	0.957900	0.284939	0.221631
331	0.690365	0.246217	0.394901	0.448833	0.517266
332	0.572044	0.055095	0.311525	0.994674	0.135890
333	0.720756	0.880090	0.365951	0.525556	0.764522

334 rows × 5 columns

```
In [53]: newdf2 = newdf.copy()
```

```
In [54]: newdf2.head()
```

```
Out[54]:
```

	0	1	2	3	4
0	1705043	0.746828	0.119205	0.608367	0.786477
1	0.795904	0.457078	0.432732	0.753908	0.220771
2	0.355519	0.660129	0.693084	0.058004	0.326732
3	0.278972	0.760987	0.535669	0.533717	0.379935
4	0.0299681	0.483665	0.278039	0.511621	0.288220

```
In [55]: newdf.head()
```

```
Out[55]:
```

	0	1	2	3	4
0	1705043	0.746828	0.119205	0.608367	0.786477
1	0.795904	0.457078	0.432732	0.753908	0.220771
2	0.355519	0.660129	0.693084	0.058004	0.326732
3	0.278972	0.760987	0.535669	0.533717	0.379935
4	0.0299681	0.483665	0.278039	0.511621	0.288220

```
In [56]: newdf2[0][0] = 100
```

```
d:\coding\python\venv\lib\site-packages\ipykernel_launcher.py:1: SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame
```

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
"""\nEntry point for launching an IPython kernel.

In [57]: newdf.head()

Out[57]:

	0	1	2	3	4
0	1705043	0.746828	0.119205	0.608367	0.786477
1	0.795904	0.457078	0.432732	0.753908	0.220771
2	0.355519	0.660129	0.693084	0.058004	0.326732
3	0.278972	0.760987	0.535669	0.533717	0.379935
4	0.0299681	0.483665	0.278039	0.511621	0.288220

In [58]: newdf.loc[0,0] = 654

In [59]: newdf.head(2)

Out[59]:

	0	1	2	3	4
0	654	0.746828	0.119205	0.608367	0.786477
1	0.795904	0.457078	0.432732	0.753908	0.220771

In [60]: newdf.columns = list("ABCDE")

In [61]: newdf

Out[61]:

	A	B	C	D	E
0	654	0.746828	0.119205	0.608367	0.786477
1	0.795904	0.457078	0.432732	0.753908	0.220771
2	0.355519	0.660129	0.693084	0.058004	0.326732
3	0.278972	0.760987	0.535669	0.533717	0.379935
4	0.0299681	0.483665	0.278039	0.511621	0.288220
...
329	0.706344	0.611424	0.103477	0.843983	0.877947
330	0.49912	0.638131	0.957900	0.284939	0.221631
331	0.690365	0.246217	0.394901	0.448833	0.517266
332	0.572044	0.055095	0.311525	0.994674	0.135890
333	0.720756	0.880090	0.365951	0.525556	0.764522

334 rows × 5 columns

In [62]: newdf.head(2)

Out[62]:

	A	B	C	D	E
0	654	0.746828	0.119205	0.608367	0.786477
1	0.795904	0.457078	0.432732	0.753908	0.220771

In [63]: newdf.loc[0,0] = 1705

In [64]: newdf.head(2)

Out[64]:

	A	B	C	D	E	0
0	654	0.746828	0.119205	0.608367	0.786477	1705.0
1	0.795904	0.457078	0.432732	0.753908	0.220771	NaN

```
In [65]: newdf.head()
```

```
Out[65]:
```

	A	B	C	D	E	0
0	654	0.746828	0.119205	0.608367	0.786477	1705.0
1	0.795904	0.457078	0.432732	0.753908	0.220771	NaN
2	0.355519	0.660129	0.693084	0.058004	0.326732	NaN
3	0.278972	0.760987	0.535669	0.533717	0.379935	NaN
4	0.0299681	0.483665	0.278039	0.511621	0.288220	NaN

```
In [66]: newdf.loc[0, 'A'] = 1705
```

```
In [67]: newdf.head()
```

```
Out[67]:
```

	A	B	C	D	E	0
0	1705	0.746828	0.119205	0.608367	0.786477	1705.0
1	0.795904	0.457078	0.432732	0.753908	0.220771	NaN
2	0.355519	0.660129	0.693084	0.058004	0.326732	NaN
3	0.278972	0.760987	0.535669	0.533717	0.379935	NaN
4	0.0299681	0.483665	0.278039	0.511621	0.288220	NaN

```
In [68]: newdf.drop(0, axis=1)
```

```
Out[68]:
```

	A	B	C	D	E
0	1705	0.746828	0.119205	0.608367	0.786477
1	0.795904	0.457078	0.432732	0.753908	0.220771
2	0.355519	0.660129	0.693084	0.058004	0.326732
3	0.278972	0.760987	0.535669	0.533717	0.379935
4	0.0299681	0.483665	0.278039	0.511621	0.288220
...
329	0.706344	0.611424	0.103477	0.843983	0.877947
330	0.49912	0.638131	0.957900	0.284939	0.221631
331	0.690365	0.246217	0.394901	0.448833	0.517266
332	0.572044	0.055095	0.311525	0.994674	0.135890
333	0.720756	0.880090	0.365951	0.525556	0.764522

334 rows × 5 columns

```
In [69]: newdf = newdf.drop(0, axis=1)
```

```
In [70]: newdf.head()
```

```
Out[70]:
```

	A	B	C	D	E
0	1705	0.746828	0.119205	0.608367	0.786477
1	0.795904	0.457078	0.432732	0.753908	0.220771
2	0.355519	0.660129	0.693084	0.058004	0.326732
3	0.278972	0.760987	0.535669	0.533717	0.379935
4	0.0299681	0.483665	0.278039	0.511621	0.288220

```
In [71]: newdf.loc[[1,2], ['C', 'D']]
```

```
Out[71]:
```

	C	D
1	0.457078	0.753908
2	0.432732	0.058004

1	0.432732	0.753908
2	0.693084	0.058004

In [72]: newdf.head()

Out[72]:

	A	B	C	D	E
0	1705	0.746828	0.119205	0.608367	0.786477
1	0.795904	0.457078	0.432732	0.753908	0.220771
2	0.355519	0.660129	0.693084	0.058004	0.326732
3	0.278972	0.760987	0.535669	0.533717	0.379935
4	0.0299681	0.483665	0.278039	0.511621	0.288220

In [73]: newdf.loc[:, ['C', 'D']]

Out[73]:

	C	D
0	0.119205	0.608367
1	0.432732	0.753908
2	0.693084	0.058004
3	0.535669	0.533717
4	0.278039	0.511621
...
329	0.103477	0.843983
330	0.957900	0.284939
331	0.394901	0.448833
332	0.311525	0.994674
333	0.365951	0.525556

334 rows × 2 columns

In [74]: newdf.loc[[1,2], :]

Out[74]:

	A	B	C	D	E
1	0.795904	0.457078	0.432732	0.753908	0.220771
2	0.355519	0.660129	0.693084	0.058004	0.326732

In [75]: newdf.loc[newdf['A'] < 0.3]

Out[75]:

	A	B	C	D	E
3	0.278972	0.760987	0.535669	0.533717	0.379935
4	0.0299681	0.483665	0.278039	0.511621	0.288220
5	0.291862	0.599023	0.974973	0.311611	0.935842
8	0.129359	0.881250	0.459516	0.957592	0.560121
14	0.100699	0.932347	0.297595	0.477471	0.224610
...
312	0.221173	0.602139	0.209530	0.193026	0.202523
315	0.142574	0.594299	0.062924	0.867649	0.335347
320	0.186702	0.350780	0.509517	0.094040	0.295364
321	0.237899	0.345210	0.001442	0.100539	0.416182
328	0.0518385	0.103758	0.222138	0.463990	0.213876

93 rows × 5 columns

```
In [76]: newdf.loc[(newdf['A']<0.3) & (newdf['C']>0.1)]
```

Out[76]:

	A	B	C	D	E
3	0.278972	0.760987	0.535669	0.533717	0.379935
4	0.0299681	0.483665	0.278039	0.511621	0.288220
5	0.291862	0.599023	0.974973	0.311611	0.935842
8	0.129359	0.881250	0.459516	0.957592	0.560121
14	0.100699	0.932347	0.297595	0.477471	0.224610
...
309	0.19228	0.357623	0.964034	0.040266	0.553172
311	0.0103135	0.776357	0.665863	0.914261	0.589711
312	0.221173	0.602139	0.209530	0.193026	0.202523
320	0.186702	0.350780	0.509517	0.094040	0.295364
328	0.0518385	0.103758	0.222138	0.463990	0.213876

82 rows × 5 columns

```
In [77]: newdf.head(2)
```

Out[77]:

	A	B	C	D	E
0	1705	0.746828	0.119205	0.608367	0.786477
1	0.795904	0.457078	0.432732	0.753908	0.220771

```
In [78]: newdf.iloc[0,4]
```

Out[78]: 0.7864773618903984

```
In [79]: newdf.iloc[[0,5], [1,2]]
```

Out[79]:

	B	C
0	0.746828	0.119205
5	0.599023	0.974973

```
In [80]: newdf.head(3)
```

Out[80]:

	A	B	C	D	E
0	1705	0.746828	0.119205	0.608367	0.786477
1	0.795904	0.457078	0.432732	0.753908	0.220771
2	0.355519	0.660129	0.693084	0.058004	0.326732

```
In [81]: newdf.drop([0])
```

Out[81]:

	A	B	C	D	E
1	0.795904	0.457078	0.432732	0.753908	0.220771
2	0.355519	0.660129	0.693084	0.058004	0.326732
3	0.278972	0.760987	0.535669	0.533717	0.379935
4	0.0299681	0.483665	0.278039	0.511621	0.288220
5	0.291862	0.599023	0.974973	0.311611	0.935842
...
329	0.706344	0.611424	0.103477	0.843983	0.877947
330	0.49912	0.638131	0.957900	0.284939	0.221631

331	0.690365	0.246217	0.394901	0.448833	0.517266
332	0.572044	0.055095	0.311525	0.994674	0.135890
333	0.720756	0.880090	0.365951	0.525556	0.764522

333 rows × 5 columns

```
In [82]: newdf.drop(['A', 'C'], axis=1)
```

Out[82]:

	B	D	E
0	0.746828	0.608367	0.786477
1	0.457078	0.753908	0.220771
2	0.660129	0.058004	0.326732
3	0.760987	0.533717	0.379935
4	0.483665	0.511621	0.288220
...
329	0.611424	0.843983	0.877947
330	0.638131	0.284939	0.221631
331	0.246217	0.448833	0.517266
332	0.055095	0.994674	0.135890
333	0.880090	0.525556	0.764522

334 rows × 3 columns

```
In [83]: newdf
```

Out[83]:

	A	B	C	D	E
0	1705	0.746828	0.119205	0.608367	0.786477
1	0.795904	0.457078	0.432732	0.753908	0.220771
2	0.355519	0.660129	0.693084	0.058004	0.326732
3	0.278972	0.760987	0.535669	0.533717	0.379935
4	0.0299681	0.483665	0.278039	0.511621	0.288220
...
329	0.706344	0.611424	0.103477	0.843983	0.877947
330	0.49912	0.638131	0.957900	0.284939	0.221631
331	0.690365	0.246217	0.394901	0.448833	0.517266
332	0.572044	0.055095	0.311525	0.994674	0.135890
333	0.720756	0.880090	0.365951	0.525556	0.764522

334 rows × 5 columns

```
In [84]: newdf.drop(['A', 'C'], axis=1, inplace=True)
```

```
In [85]: newdf
```

Out[85]:

	B	D	E
0	0.746828	0.608367	0.786477
1	0.457078	0.753908	0.220771
2	0.660129	0.058004	0.326732
3	0.760987	0.533717	0.379935
4	0.483665	0.511621	0.288220
...

329	0.611424	0.843983	0.877947
330	0.638131	0.284939	0.221631
331	0.246217	0.448833	0.517266
332	0.055095	0.994674	0.135890
333	0.880090	0.525556	0.764522

334 rows × 3 columns

```
In [86]: newdf.drop([1, 5], axis=0, inplace=True)
```

```
In [87]: newdf
```

Out[87]:

	B	D	E
0	0.746828	0.608367	0.786477
2	0.660129	0.058004	0.326732
3	0.760987	0.533717	0.379935
4	0.483665	0.511621	0.288220
6	0.431699	0.061238	0.900006
...
329	0.611424	0.843983	0.877947
330	0.638131	0.284939	0.221631
331	0.246217	0.448833	0.517266
332	0.055095	0.994674	0.135890
333	0.880090	0.525556	0.764522

332 rows × 3 columns

```
In [88]: newdf.head(3)
```

Out[88]:

	B	D	E
0	0.746828	0.608367	0.786477
2	0.660129	0.058004	0.326732
3	0.760987	0.533717	0.379935

```
In [89]: newdf.reset_index()
```

Out[89]:

	index	B	D	E
0	0	0.746828	0.608367	0.786477
1	2	0.660129	0.058004	0.326732
2	3	0.760987	0.533717	0.379935
3	4	0.483665	0.511621	0.288220
4	6	0.431699	0.061238	0.900006
...
327	329	0.611424	0.843983	0.877947
328	330	0.638131	0.284939	0.221631
329	331	0.246217	0.448833	0.517266
330	332	0.055095	0.994674	0.135890
331	333	0.880090	0.525556	0.764522

332 rows × 4 columns

```
In [90]: newdf.reset_index(drop=True, inplace=True)
```

```
In [91]: newdf.head()
```

```
Out[91]:
```

	B	D	E
0	0.746828	0.608367	0.786477
1	0.660129	0.058004	0.326732
2	0.760987	0.533717	0.379935
3	0.483665	0.511621	0.288220
4	0.431699	0.061238	0.900006

```
In [92]: newdf['B'].isnull()
```

```
Out[92]: 0    False
```

```
1    False
```

```
2    False
```

```
3    False
```

```
4    False
```

```
...
```

```
327   False
```

```
328   False
```

```
329   False
```

```
330   False
```

```
331   False
```

```
Name: B, Length: 332, dtype: bool
```

```
In [93]: newdf['B'] = None
```

```
In [94]: newdf['B'].isnull()
```

```
Out[94]: 0    True
```

```
1    True
```

```
2    True
```

```
3    True
```

```
4    True
```

```
...
```

```
327   True
```

```
328   True
```

```
329   True
```

```
330   True
```

```
331   True
```

```
Name: B, Length: 332, dtype: bool
```

```
In [95]: newdf.loc[:, ['B']] = None
```

```
In [96]: newdf
```

```
Out[96]:
```

	B	D	E
0	None	0.608367	0.786477
1	None	0.058004	0.326732
2	None	0.533717	0.379935
3	None	0.511621	0.288220
4	None	0.061238	0.900006
...
327	None	0.843983	0.877947
328	None	0.284939	0.221631
329	None	0.448833	0.517266
330	None	0.994674	0.135890
331	None	0.525556	0.764522

```
332 rows × 3 columns
```

```
In [97]: newdf.loc[:, ['B']] = 56
```

```
In [98]: newdf
```

```
Out[98]:
```

	B	D	E
0	56	0.608367	0.786477
1	56	0.058004	0.326732
2	56	0.533717	0.379935
3	56	0.511621	0.288220
4	56	0.061238	0.900006
...
327	56	0.843983	0.877947
328	56	0.284939	0.221631
329	56	0.448833	0.517266
330	56	0.994674	0.135890
331	56	0.525556	0.764522

332 rows × 3 columns

```
In [99]: df = pd.DataFrame({'name': ['Alfred', 'Batman', 'Catwoman'],
                           "toy": [np.nan, 'Batmobile', 'Bullwhip'],
                           "born": [pd.NaT, pd.Timestamp("1940-04-25"), pd.NaT]
                           })
```

```
In [100]: df
```

```
Out[100]:
```

	name	toy	born
0	Alfred	NaN	NaT
1	Batman	Batmobile	1940-04-25
2	Catwoman	Bullwhip	NaT

```
In [101]: df.dropna()
```

```
Out[101]:
```

	name	toy	born
1	Batman	Batmobile	1940-04-25

```
In [102]: df.dropna(how='all')
```

```
Out[102]:
```

	name	toy	born
0	Alfred	NaN	NaT
1	Batman	Batmobile	1940-04-25
2	Catwoman	Bullwhip	NaT

```
In [103]: df = pd.DataFrame({'name': ['Alfred', 'Batman', 'Catwoman'],
                           "toy": [np.nan, np.nan, np.nan],
                           "born": [pd.NaT, pd.Timestamp("1940-04-25"), pd.NaT]
                           })
```

```
In [104]: df
```

```
Out[104]:
```

	name	toy	born
0	Alfred	NaN	NaT
1	Batman	NaN	1940-04-25
2	Catwoman	NaN	NaT

```
In [105]: df.dropna(how='all', axis=1)
```

```
Out[105]:
```

	name	born
0	Alfred	NaT
1	Batman	1940-04-25
2	Catwoman	NaT

```
In [106]: df
```

```
Out[106]:
```

	name	toy	born
0	Alfred	NaN	NaT
1	Batman	NaN	1940-04-25
2	Catwoman	NaN	NaT

```
In [107]: df = pd.DataFrame({"name":['Alfred', 'Batman', 'Alfred'],
                           "toy": [np.nan, 'Batmobile', 'Bullwhip'],
                           "born": [pd.NaT, pd.Timestamp("1940-04-25"), pd.NaT]
                           })
```

```
In [108]: df
```

```
Out[108]:
```

	name	toy	born
0	Alfred	NaN	NaT
1	Batman	Batmobile	1940-04-25
2	Alfred	Bullwhip	NaT

```
In [109]: df.drop_duplicates()
```

```
Out[109]:
```

	name	toy	born
0	Alfred	NaN	NaT
1	Batman	Batmobile	1940-04-25
2	Alfred	Bullwhip	NaT

```
In [110]: df.drop_duplicates(axis=1)
```

```
-----  
TypeError                                                 Traceback (most recent call last)  
<ipython-input-110-a01ca93282de> in <module>  
----> 1 df.drop_duplicates(axis=1)
```

```
TypeError: drop_duplicates() got an unexpected keyword argument 'axis'
```

```
In [111]: df.drop_duplicates(subset=['name'])
```

```
Out[111]:
```

	name	toy	born
0	Alfred	NaN	NaT
1	Batman	Batmobile	1940-04-25

```
In [112]: df
```

```
Out[112]:
```

	name	toy	born
0	Alfred	NaN	NaT
1	Batman	Batmobile	1940-04-25
2	Alfred	Bullwhip	NaT

```
In [113]: df.drop_duplicates(subset=['name'], keep='first')
```

Out[113]:

	name	toy	born
0	Alfred	NaN	NaT
1	Batman	Batmobile	1940-04-25

```
In [114]: df.drop_duplicates(subset=['name'], keep='last')
```

Out[114]:

	name	toy	born
1	Batman	Batmobile	1940-04-25
2	Alfred	Bullwhip	NaT

```
In [115]: df.drop_duplicates(subset=['name'], keep=False)
```

Out[115]:

	name	toy	born
1	Batman	Batmobile	1940-04-25

```
In [116]: df.shape
```

Out[116]: (3, 3)

```
In [117]: df.info
```

Out[117]: <bound method DataFrame.info of
0 Alfred NaN NaT
1 Batman Batmobile 1940-04-25
2 Alfred Bullwhip NaT>

```
In [118]: 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: 200.0+ bytes
```

```
In [119]: df['name'].value_counts(dropna=False)
```

Out[119]: Alfred 2
Batman 1
Name: name, dtype: int64

```
In [120]: df['toy'].value_counts(dropna=False)
```

Out[120]: Batmobile 1
Bullwhip 1
NaN 1
Name: toy, dtype: int64

```
In [121]: df['toy'].value_counts(dropna=True)
```

Out[121]: Batmobile 1
Bullwhip 1
Name: toy, dtype: int64

```
In [122]: df.notnull()
```

Out[122]:

	name	toy	born
0	True	False	False
1	True	True	True
2	True	True	False

```
In [123]: df.isnull()
```

Out[123]:

	name	toy	born
0	False	True	True
1	False	False	False
2	False	False	True

Excel Sheet Handling

```
In [124]: #pip install xlrd
```

```
In [125]: data = pd.read_excel('data.xlsx')
```

```
-----  
XLRError                                     Traceback (most recent call last)  
<ipython-input-125-d4b807c641e9> in <module>  
----> 1 data = pd.read_excel('data.xlsx')  
  
d:\coding\python\venv\lib\site-packages\pandas\util\_decorators.py in wrapper(*args, **  
kwargs)  
    294         )  
    295             warnings.warn(msg, FutureWarning, stacklevel=stacklevel)  
--> 296         return func(*args, **kwargs)  
    297  
    298     return wrapper  
  
d:\coding\python\venv\lib\site-packages\pandas\io\excel\_base.py in read_excel(io, shee  
t_name, header, names, index_col, usecols, squeeze, dtype, engine, converters, true_<br>  
values, false_values, skiprows, nrows, na_values, keep_default_na, na_filter, verbose, pa  
re_se_dates, date_parser, thousands, comment, skipfooter, convert_float, mangle_dupe_cols)  
    302     if not isinstance(io, ExcelFile):  
--> 304         io = ExcelFile(io, engine=engine)  
    305     elif engine and engine != io.engine:  
    306         raise ValueError()  
  
d:\coding\python\venv\lib\site-packages\pandas\io\excel\_base.py in __init__(self, path  
_or_buffer, engine)  
    865         self._io = stringify_path(path_or_buffer)  
    866  
--> 867         self._reader = self._engines[engine](self._io)  
    868  

```

```
d, ragged_rows, ignore_workbook_corruption)
    168      # files that xlrd can parse don't start with the expected signature.
    169      if file_format and file_format != 'xls':
--> 170          raise XLRDError(FILE_FORMAT_DESCRIPTIONS[file_format]+'; not supported')
d')
171
172      bk = open_workbook_xls()

XLRDError: Excel xlsx file; not supported
```

In [126]: `data = pd.read_excel('data.xls')`

In [127]: `data`

Out[127]:

	Unnamed: 0	Train No S2	Speed S2	City S2
0	0	1705043	56	naogaon
1	1	49494	234	barishal
2	2	14567	31	mymensingh
3	3	49678	76	dhaka

In [128]: `data = pd.read_excel('data.xls', sheet_name='Sheet1')`

```
-----
ValueError                                                 Traceback (most recent call last)
d:\coding\python\venv\lib\site-packages\xlrd\book.py in sheet_by_name(self, sheet_name)
    465         try:
--> 466             sheetx = self._sheet_names.index(sheet_name)
    467         except ValueError:
```

```
ValueError: 'Sheet1' is not in list
```

During handling of the above exception, another exception occurred:

```
XLRDError                                                 Traceback (most recent call last)
<ipython-input-128-4bb00da9abf3> in <module>
----> 1 data = pd.read_excel('data.xls', sheet_name='Sheet1')

d:\coding\python\venv\lib\site-packages\pandas\util\_decorators.py in wrapper(*args, **kwargs)
    294         )
    295         warnings.warn(msg, FutureWarning, stacklevel=stacklevel)
--> 296     return func(*args, **kwargs)
    297
    298     return wrapper

d:\coding\python\venv\lib\site-packages\pandas\io\excel\_base.py in read_excel(io, sheet_name, header, names, index_col, usecols, squeeze, dtype, engine, converters, true_values, false_values, skiprows, nrows, na_values, keep_default_na, na_filter, verbose, parse_dates, date_parser, thousands, comment, skipfooter, convert_float, mangle_dupe_cols)
    332         skipfooter=skipfooter,
    333         convert_float=convert_float,
--> 334         mangle_dupe_cols=mangle_dupe_cols,
    335     )
    336


```

```
d:\coding\python\venv\lib\site-packages\pandas\io\excel\_base.py in parse(self, sheet_name, header, names, index_col, usecols, squeeze, converters, true_values, false_values, skiprows, nrows, na_values, parse_dates, date_parser, thousands, comment, skipfooter, convert_float, mangle_dupe_cols, **kwds)
    924         convert_float=convert_float,
    925         mangle_dupe_cols=mangle_dupe_cols,
--> 926         **kwds,
    927     )
    928


```

```
d:\coding\python\venv\lib\site-packages\pandas\io\excel\_base.py in parse(self, sheet_name, header, names, index_col, usecols, squeeze, dtype, true_values, false_values, skiprows, nrows, na_values, verbose, parse_dates, date_parser, thousands, comment, skipfooter, convert_float, mangle_dupe_cols, **kwds)
    437
    438         if isinstance(asheetname, str):
```

```
--> 439             sheet = self.get_sheet_by_name(asheetname)
440         else: # assume an integer if not a string
441             sheet = self.get_sheet_by_index(asheetname)

d:\coding\python\venv\lib\site-packages\pandas\io\xlrd.py in get_sheet_by_name(self, name)
    42
    43     def get_sheet_by_name(self, name):
--> 44         return self.book.sheet_by_name(name)
    45
    46     def get_sheet_by_index(self, index):

d:\coding\python\venv\lib\site-packages\xlrd\book.py in sheet_by_name(self, sheet_name)
    46       sheetx = self._sheet_names.index(sheet_name)
    47   except ValueError:
--> 48       raise XLRDError('No sheet named <%r>' % sheet_name)
    49   return self.sheet_by_index(sheetx)
    50

XLRDError: No sheet named <'Sheet1'>
```

In [129]: data

Out[129]:

	Unnamed: 0	Train No S2	Speed S2	City S2
0	0	1705043	56	naogaon
1	1	49494	234	barishal
2	2	14567	31	mymensingh
3	3	49678	76	dhaka

In [130]: data = pd.read_excel('data.xls', sheet_name='Sheet2')

In [131]: data

Out[131]:

	Unnamed: 0	Train No S2	Speed S2	City S2
0	0	1705043	56	naogaon
1	1	49494	234	barishal
2	2	14567	31	mymensingh
3	3	49678	76	dhaka

In [138]: data.iloc[0, 0] = 1705

In [139]: data

Out[139]:

	Unnamed: 0	Train No S2	Speed S2	City S2
0	1705	1705043	56	naogaon
1	1	49494	234	barishal
2	2	14567	31	mymensingh
3	3	49678	76	dhaka

In [140]: data.to_excel('data.xls', sheet_name='Sheet2')

In [141]: #pip install xlwt

In [142]: data.to_excel('data.xls', sheet_name='Sheet2')

In [143]: #Sheet1 is no more. It has been deleted for this.

In []:

© 2021 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)

[Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)