More Important Functions

```
# value_counts
# sort_values
# rank
# sort index
# set index
# rename index -> rename
# reset index
# unique & nunique
# isnull/notnull/hasnans
# dropna
# fillna
# drop_duplicates
# drop
# apply
# isin
# corr
# nlargest -> nsmallest
# insert
# copy
```

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

```
a = pd.Series([1,1,1,2,2,3])
a.value_counts()
```

```
1    3
2    2
3    1
dtype: int64
```

```
# value_counts(series and dataframe)
```

```
marks = pd.DataFrame([
    [100,80,10],
    [90,70,7],
    [120,100,14],
    [80,70,14],
    [80,70,14]
],columns=['iq','marks','package'])

marks
```

	iq	marks	package
0	100	80	10
1	90	70	7
2	120	100	14
3	80	70	14
4	80	70	14

```
marks.value_counts()
```

```
iq   marks  package
80   70     14         2
90   70     7          1
100  80     10         1
120  100    14         1
dtype: int64
```

```
ipl = pd.read_csv('ipl-matches.csv')
ipl[~ipl['MatchNumber'].str.isdigit()]['Player_of_Match'].value_counts()
```

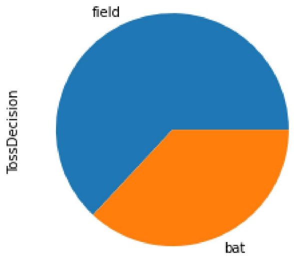
```
KA Pollard           3
F du Plessis         3
SK Raina             3
```

```
A Kumble            2
MK Pandey           2
YK Pathan           2
M Vijay             2
JJ Bumrah           2
AB de Villiers      2
SR Watson           2
HH Pandya           1
Harbhajan Singh     1
A Nehra             1
V Sehwag            1
UT Yadav            1
MS Bisla            1
BJ Hodge            1
MEK Hussey          1
MS Dhoni            1
CH Gayle            1
MM Patel            1
DE Bollinger        1
AC Gilchrist        1
RG Sharma           1
DA Warner           1
MC Henriques        1
JC Buttler          1
RM Patidar          1
DA Miller           1
VR Iyer             1
SP Narine           1
RD Gaikwad          1
TA Boult            1
MP Stoinis          1
KS Williamson       1
RR Pant             1
SA Yadav            1
Rashid Khan         1
AD Russell          1
KH Pandya           1
KV Sharma           1
NM Coulter-Nile     1
Washington Sundar   1
BCJ Cutting         1
M Ntini             1
Name: Player_of_Match, dtype: int64
```

```
# find which player has won most potm -> in finals and qualifiers
```

```
# Toss decision plot
ipl['TossDecision'].value_counts().plot(kind='pie')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f034efd49d0>
```



```
# how many matches each team has played
(ipl['Team2'].value_counts() + ipl['Team1'].value_counts()).sort_values(ascending=False)
```

```
Mumbai Indians                 231
Royal Challengers Bangalore    226
Kolkata Knight Riders          223
Chennai Super Kings            208
Rajasthan Royals               192
Kings XI Punjab                190
Delhi Daredevils               161
Sunrisers Hyderabad            152
Deccan Chargers                 75
Delhi Capitals                  63
Pune Warriors                   46
Gujarat Lions                   30
Punjab Kings                    28
Gujarat Titans                  16
```

```
Rising Pune Supergiant          16
Lucknow Super Giants            15
Kochi Tuskers Kerala            14
Rising Pune Supergiants         14
dtype: int64
```

```
# sort_values(series and dataframe) -> ascending -> na_position -> inplace -> multiple cols
```

```
x = pd.Series([12,14,1,56,89])
x
```

```
0    12
1    14
2     1
3    56
4    89
dtype: int64
```

```
x.sort_values(ascending=False)
```

```
4    89
3    56
1    14
0    12
2     1
dtype: int64
```

```
movies = pd.read_csv('movies.csv')
movies.head(4)
```

	title_x	imdb_id	poster_path	wiki_link	title_y	original_title	is_a
0	Uri: The Surgical Strike	tt8291224	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wiki/Uri:_The_Surgica...	Uri: The Surgical Strike	Uri: The Surgical Strike	
1	Battalion 609	tt9472208	NaN	https://en.wikipedia.org/wiki/Battalion_609	Battalion 609	Battalion 609	
2	The Accidental Prime Minister (film)	tt6986710	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wiki/The_Accidental_P...	The Accidental Prime Minister	The Accidental Prime Minister	
3	Why Cheat India	tt8108208	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wiki/Why_Cheat_India	Why Cheat India	Why Cheat India	

```
movies.sort_values('title_x',ascending=False)
```

	title_x	imdb_id	poster_path	wiki_link	title_y	original_titl
1623	Zubeidaa	tt0255713	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wiki/Zubeidaa	Zubeidaa	Zubeida
939	Zor Lagaa Ke...Haiya!	tt1479857	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wiki/Zor_Lagaa_Ke...H...	Zor Lagaa Ke... Haiya!	Zor Lagaa Ke.. Haiya
756	Zokkomon	tt1605790	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wiki/Zokkomon	Zokkomon	Zokkomo
670	Zindagi Tere Naam	tt2164702	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wiki/Zindagi_Tere_Naam	Zindagi Tere Naam	Zindagi Ter Naan
778	Zindagi Na Milegi Dobara	tt1562872	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wiki/Zindagi_Na_Mileg...	Zindagi Na Milegi Dobara	Zindagi N Milegi Dobar
...	...	...	...	...	...	..

```
students = pd.DataFrame(
    {
        'name':['nitish','ankit','rupesh',np.nan,'mrityunjay',np.nan,'rishabh',np.nan,'aditya',np.nan],
        'college':['bit','iit','vit',np.nan,np.nan,'vlsi','ssit',np.nan,np.nan,'git'],
        'branch':['eee','it','cse',np.nan,'me','ce','civ','cse','bio',np.nan],
        'cgpa':[6.66,8.25,6.41,np.nan,5.6,9.0,7.4,10,7.4,np.nan],
        'package':[4,5,6,np.nan,6,7,8,9,np.nan,np.nan]

    }
)
```

```
students
```

	name	college	branch	cgpa	package
0	nitish	bit	eee	6.66	4.0
1	ankit	iit	it	8.25	5.0
2	rupesh	vit	cse	6.41	6.0
3	NaN	NaN	NaN	NaN	NaN
4	mrityunjay	NaN	me	5.60	6.0
5	NaN	vlsi	ce	9.00	7.0
6	rishabh	ssit	civ	7.40	8.0
7	NaN	NaN	cse	10.00	9.0
8	aditya	NaN	bio	7.40	NaN
9	NaN	git	NaN	NaN	NaN

```
students.sort_values('name',na_position='first',ascending=False,inplace=True)
```

```
students
```

	name	college	branch	cgpa	package
3	NaN	NaN	NaN	NaN	NaN
5	NaN	vlsi	ce	9.00	7.0
7	NaN	NaN	cse	10.00	9.0
9	NaN	git	NaN	NaN	NaN
2	rupesh	vit	cse	6.41	6.0
6	rishabh	ssit	civ	7.40	8.0
0	nitish	bit	eee	6.66	4.0
4	mrityunjay	NaN	me	5.60	6.0
1	ankit	iit	it	8.25	5.0
8	aditya	NaN	bio	7.40	NaN

```
movies.sort_values(['year_of_release','title_x'],ascending=[True,False])
```

	title_x	imdb_id	poster_path	wiki_link	title_y	original_titl
1623	Zubeidaa	tt0255713	https://upload.wikimedia.org/wikipedia/en/thum	https://en.wikipedia.org/wiki/Zubeidaa	Zubeidaa	Zubeidaa

```
# rank(series)
batsman = pd.read_csv('batsman_runs_ipl.csv')
batsman.head()
```

	batter	batsman_run
0	A Ashish Reddy	280
1	A Badoni	161
2	A Chandila	4
3	A Chopra	53
4	A Choudhary	25

Raaste Raaste Yeh Raaste Hai

```
batsman['batting_rank'] = batsman['batsman_run'].rank(ascending=False)
batsman.sort_values('batting_rank')
```

	batter	batsman_run	batting_rank
569	V Kohli	6634	1.0
462	S Dhawan	6244	2.0
130	DA Warner	5883	3.0
430	RG Sharma	5881	4.0
493	SK Raina	5536	5.0
...	...	...	...
512	SS Cottrell	0	594.0
466	S Kaushik	0	594.0
203	IC Pandey	0	594.0
467	S Ladda	0	594.0
468	S Lamichhane	0	594.0

605 rows × 3 columns

Albert Albert

```
# sort_index(series and dataframe)
```

Kyun

```
marks = {
    'maths':67,
    'english':57,
    'science':89,
    'hindi':100
}

marks_series = pd.Series(marks)
marks_series
```

```
maths       67
english     57
science     89
hindi      100
dtype: int64
```

```
marks_series.sort_index(ascending=False)
```

```
science     89
maths       67
hindi      100
english     57
dtype: int64
```

```
movies.sort_index(ascending=False)
```

	title_x	imdb_id	poster_path	wiki_link	title_y	original_titl
1628	Humsafar	tt2403201	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wiki/Humsafar	Humsafar	Humsaf
1627	Daaka	tt10833860	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wiki/Daaka	Daaka	Daal
1626	Sabse Bada Sukh	tt0069204	NaN	https://en.wikipedia.org/wiki/Sabse_Bada_Sukh	Sabse Bada Sukh	Sabse Bad Sul
1625	Yeh Zindagi Ka Safar	tt0298607	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wiki/Yeh_Zindagi_Ka_S...	Yeh Zindagi Ka Safar	Yeh Zindagi K Saf
1624	Tera Mera Saath Rahen	tt0301250	https://upload.wikimedia.org/wikipedia/en/2/2b...	https://en.wikipedia.org/wiki/Tera_Mera_Saath_...	Tera Mera Saath Rahen	Tera Mera Saa Rahe
...	...	...	...	...	...	
4	Evening Shadows	tt6028796	NaN	https://en.wikipedia.org/wiki/Evening_Shadows	Evening Shadows	Evenin Shadov
3	Why Cheat India	tt8108208	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wiki/Why_Cheat_India	Why Cheat India	Why Cheat Ind
2	The Accidental Prime Minister (film)	tt6986710	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wiki/The_Accidental_P...	The Accidental Prime Minister	The Acciden Prime Minist
1	Battalion 609	tt9472208	NaN	https://en.wikipedia.org/wiki/Battalion_609	Battalion 609	Battalion 60

```
# set_index(dataframe) -> inplace
batsman.set_index('batter',inplace=True)
```

```
batsman
```

	batsman_run	batting_rank
batter		
A Ashish Reddy	280	166.5
A Badoni	161	226.0
A Chandila	4	535.0
A Chopra	53	329.0
A Choudhary	25	402.5

```
# reset_index(series + dataframe) -> drop parameter
batsman.reset_index(inplace=True)
```

	batter	batsman_run	batting_rank
0	A Ashish Reddy	280	166.5
1	A Badoni	161	226.0
2	A Chandila	4	535.0
3	A Chopra	53	329.0
4	A Choudhary	25	402.5
...	...	...	...
600	Yash Dayal	0	594.0
601	Yashpal Singh	47	343.0
602	Younis Khan	3	547.5
603	Yuvraj Singh	2754	27.0
604	Z Khan	117	256.0

605 rows × 3 columns

```
batsman
```

	batsman_run	batting_rank
batter		
A Ashish Reddy	280	166.5
A Badoni	161	226.0
A Chandila	4	535.0
A Chopra	53	329.0
A Choudhary	25	402.5
...	...	...
Yash Dayal	0	594.0
Yashpal Singh	47	343.0
Younis Khan	3	547.5
Yuvraj Singh	2754	27.0
Z Khan	117	256.0

605 rows × 2 columns

```
# how to replace existing index without loosing
batsman.reset_index().set_index('batting_rank')
```

	batter	batsman_run
batting_rank		
166.5	A Ashish Reddy	280
226.0	A Badoni	161
535.0	A Chandila	4
329.0	A Chopra	53
402.5	A Choudhary	25
...	...	...
594.0	Yash Dayal	0

```
# series to dataframe using reset_index
marks_series.reset_index()
```

	index	0
0	maths	67
1	english	57
2	science	89
3	hindi	100

```
# rename(dataframe) -> index
```

```
movies.set_index('title_x',inplace=True)
```

```
movies.rename(columns={'imdb_id':'imdb','poster_path':'link'},inplace=True)
```

```
movies.rename(index={'Uri: The Surgical Strike':'Uri','Battalion 609':'Battalion'})
```

	imdb	link	wiki_lin
title_x			
Uri	tt8291224	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wiki/Uri:_The_Surgica.
Battalion	tt9472208	NaN	https://en.wikipedia.org/wiki/Battalion_60
...	...	...	.
Rahen			
Zindagi	tt0298607	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wiki/Yeh_Zindagi_Ka_S.
Sabse Bada	[illegible]	NaN	[illegible]
Daaka	tt10833860	https://upload.wikimedia.org/wikipedia/en/thum	https://en.wikipedia.org/wiki/Daak

1629 rows × 17 columns

```
# unique(series)
temp = pd.Series([1,1,2,2,3,3,4,4,5,5,np.nan,np.nan])
print(temp)
```

```
0     1.0
1     1.0
2     2.0
3     2.0
4     3.0
5     3.0
6     4.0
7     4.0
8     5.0
9     5.0
10    NaN
11    NaN
dtype: float64
```

```
len(temp.unique())
```

```
6
```

```
temp.nunique()
```

```
5
```

```
len(ipl['Season'].unique())
```

```
15
```

```
# nunique(series + dataframe) -> does not count nan -> dropna parameter
ipl['Season'].nunique()
```

```
15
```

```
# isnull(series + dataframe)
students['name'][students['name'].isnull()]
```

```
3    NaN
5    NaN
7    NaN
9    NaN
Name: name, dtype: object
```

```
# notnull(series + dataframe)
students['name'][students['name'].notnull()]
```

```
0        nitish
1         ankit
2        rupesh
4    mrityunjay
6       rishabh
8        aditya
Name: name, dtype: object
```

```
# hasnans(series)
students['name'].hasnans
```

```
True
```

```
students
```

	name	college	branch	cgpa	package
0	nitish	bit	eee	6.66	4.0
1	ankit	iit	it	8.25	5.0
2	rupesh	vit	cse	6.41	6.0
3	NaN	NaN	NaN	NaN	NaN
4	mrityunjay	NaN	me	5.60	6.0
5	NaN	vlsi	ce	9.00	7.0
6	rishabh	ssit	civ	7.40	8.0
7	NaN	NaN	cse	10.00	9.0
8	aditya	NaN	bio	7.40	NaN
9	NaN	git	NaN	NaN	NaN

```
students.isnull()
```

	name	college	branch	cgpa	package
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	True	True	True	True	True
4	False	True	False	False	False
5	True	False	False	False	False
6	False	False	False	False	False
7	True	True	False	False	False
8	False	True	False	False	True
9	True	False	True	True	True

```
students.notnull()
```

	name	college	branch	cgpa	package
0	True	True	True	True	True
1	True	True	True	True	True
2	True	True	True	True	True
3	False	False	False	False	False
4	True	False	True	True	True
5	False	True	True	True	True
6	True	True	True	True	True
7	False	False	True	True	True
8	True	False	True	True	False
9	False	True	False	False	False

```
# dropna(series + dataframe) -> how parameter -> works like or
students['name'].dropna()
```

```
0        nitish
1         ankit
2        rupesh
4    mrityunjay
6       rishabh
8        aditya
Name: name, dtype: object
```

```
students
```

```
students
```

	name	college	branch	cgpa	package
0	nitish	bit	eee	6.66	4.0
1	ankit	iit	it	8.25	5.0
2	rupesh	vit	cse	6.41	6.0
3	NaN	NaN	NaN	NaN	NaN
4	mrityunjay	NaN	me	5.60	6.0
5	NaN	vlsi	ce	9.00	7.0
6	rishabh	ssit	civ	7.40	8.0
7	NaN	NaN	cse	10.00	9.0
8	aditya	NaN	bio	7.40	NaN
9	NaN	git	NaN	NaN	NaN

```
students.dropna(how='any')
```

	name	college	branch	cgpa	package
0	nitish	bit	eee	6.66	4.0
1	ankit	iit	it	8.25	5.0
2	rupesh	vit	cse	6.41	6.0
6	rishabh	ssit	civ	7.40	8.0

```
students.dropna(how='all')
```

	name	college	branch	cgpa	package
0	nitish	bit	eee	6.66	4.0
1	ankit	iit	it	8.25	5.0
2	rupesh	vit	cse	6.41	6.0
4	mrityunjay	NaN	me	5.60	6.0
5	NaN	vlsi	ce	9.00	7.0
6	rishabh	ssit	civ	7.40	8.0
7	NaN	NaN	cse	10.00	9.0
8	aditya	NaN	bio	7.40	NaN
9	NaN	git	NaN	NaN	NaN

```
students.dropna(subset=['name'])
```

	name	college	branch	cgpa	package
0	nitish	bit	eee	6.66	4.0
1	ankit	iit	it	8.25	5.0
2	rupesh	vit	cse	6.41	6.0
4	mrityunjay	NaN	me	5.60	6.0
6	rishabh	ssit	civ	7.40	8.0
8	aditya	NaN	bio	7.40	NaN

```
students.dropna(subset=['name','college'])
```

	name	college	branch	cgpa	package
0	nitish	bit	eee	6.66	4.0

```
students.dropna(inplace=True)
```

	name	college	branch	cgpa	package
0	nitish	bit	eee	6.66	4.0
1	ankit	iit	it	8.25	5.0
2	rupesh	vit	cse	6.41	6.0
3	NaN	NaN	NaN	NaN	NaN
4	mrityunjay	NaN	me	5.60	6.0
5	NaN	vlsi	ce	9.00	7.0
6	rishabh	ssit	civ	7.40	8.0
7	NaN	NaN	cse	10.00	9.0
8	aditya	NaN	bio	7.40	NaN
9	NaN	git	NaN	NaN	NaN

```
# fillna(series + dataframe)
students['name'].fillna('unknown')
```

```
0        nitish
1         ankit
2        rupesh
3       unknown
4    mrityunjay
5       unknown
6       rishabh
7       unknown
8        aditya
9       unknown
Name: name, dtype: object
```

```
students
```

	name	college	branch	cgpa	package
0	nitish	bit	eee	6.66	4.0
1	ankit	iit	it	8.25	5.0
2	rupesh	vit	cse	6.41	6.0
3	NaN	NaN	NaN	NaN	NaN
4	mrityunjay	NaN	me	5.60	6.0
5	NaN	vlsi	ce	9.00	7.0
6	rishabh	ssit	civ	7.40	8.0
7	NaN	NaN	cse	10.00	9.0
8	aditya	NaN	bio	7.40	NaN
9	NaN	git	NaN	NaN	NaN

```
students['package'].fillna(students['package'].mean())
```

```
0    4.000000
1    5.000000
2    6.000000
3    6.428571
4    6.000000
5    7.000000
6    8.000000
7    9.000000
8    6.428571
9    6.428571
Name: package, dtype: float64
```

```
students['name'].fillna(method='bfill')
```

```
0        nitish
1         ankit
2        rupesh
3    mrityunjay
4    mrityunjay
5       rishabh
6       rishabh
7        aditya
8        aditya
9           NaN
Name: name, dtype: object
```

```
# drop_duplicates(series + dataframe) -> works like and -> duplicated()
```

```
temp = pd.Series([1,1,1,2,3,3,4,4])
temp.drop_duplicates()
```

```
0    1
3    2
4    3
6    4
dtype: int64
```

```
marks.drop_duplicates(keep='last')
```

	iq	marks	package
0	100	80	10
1	90	70	7
2	120	100	14
4	80	70	14

```
# find the last match played by virat kohli in Delhi
ipl['all_players'] = ipl['Team1Players'] + ipl['Team2Players']
ipl.head()
```

	ID	City	Date	Season	MatchNumber	Team1	Team2	Venue	TossWinner	TossDecision	...	WinningTeam	WonBy
0	1312200	Ahmedabad	2022-05-29	2022	Final	Rajasthan Royals	Gujarat Titans	Narendra Modi Stadium, Ahmedabad	Rajasthan Royals	bat	...	Gujarat Titans	Wickets
1	1312199	Ahmedabad	2022-05-27	2022	Qualifier 2	Royal Challengers Bangalore	Rajasthan Royals	Narendra Modi Stadium, Ahmedabad	Rajasthan Royals	field	...	Rajasthan Royals	Wickets
2	1312198	Kolkata	2022-05-25	2022	Eliminator	Royal Challengers Bangalore	Lucknow Super Giants	Eden Gardens, Kolkata	Lucknow Super Giants	field	...	Royal Challengers Bangalore	Runs
3	1312197	Kolkata	2022-05-24	2022	Qualifier 1	Rajasthan Royals	Gujarat Titans	Eden Gardens, Kolkata	Gujarat Titans	field	...	Gujarat Titans	Wickets
4	1304116	Mumbai	2022-05-22	2022	70	Sunrisers Hyderabad	Punjab Kings	Wankhede Stadium, Mumbai	Sunrisers Hyderabad	bat	...	Punjab Kings	Wickets

5 rows × 21 columns

```
def did_kohli_play(players_list):
  return 'V Kohli' in players_list
```

```
ipl['did_kohli_play'] = ipl['all_players'].apply(did_kohli_play)
ipl[(ipl['City'] == 'Delhi') & (ipl['did_kohli_play'] == True)].drop_duplicates(subset=['City','did_kohli_play'],keep='first')
```

	ID	City	Date	Season	MatchNumber	Team1	Team2	Venue	TossWinner	TossDecision	...	WonBy	Margin	method	Player_
208	1178421	Delhi	2019-04-28	2019	46	Delhi Capitals	Royal Challengers Bangalore	Arun Jaitley Stadium	Delhi Capitals	bat	...	Runs	16.0	NaN	

1 rows × 22 columns

```
students.drop_duplicates()
```

```
# drop(series + dataframe)
temp = pd.Series([10,2,3,16,45,78,10])
temp
```

```
0    10
1     2
2     3
3    16
4    45
5    78
6    10
dtype: int64
```

```
temp.drop(index=[0,6])
```

```
1     2
2     3
3    16
4    45
5    78
dtype: int64
```

```
students
```

	name	college	branch	cgpa	package
0	nitish	bit	eee	6.66	4.0
1	ankit	iit	it	8.25	5.0
2	rupesh	vit	cse	6.41	6.0
3	NaN	NaN	NaN	NaN	NaN
4	mrityunjay	NaN	me	5.60	6.0
5	NaN	vlsi	ce	9.00	7.0
6	rishabh	ssit	civ	7.40	8.0
7	NaN	NaN	cse	10.00	9.0
8	aditya	NaN	bio	7.40	NaN
9	NaN	git	NaN	NaN	NaN

```
students.drop(columns=['branch','cgpa'],inplace=True)
```

	name	college	package
0	nitish	bit	4.0
1	ankit	iit	5.0
2	rupesh	vit	6.0

```
students.set_index('name').drop(index=['nitish','aditya'])
```

	college	branch	cgpa	package
name				
ankit	iit	it	8.25	5.0
rupesh	vit	cse	6.41	6.0
NaN	NaN	NaN	NaN	NaN
mrityunjay	NaN	me	5.60	6.0
NaN	vlsi	ce	9.00	7.0
rishabh	ssit	civ	7.40	8.0
NaN	NaN	cse	10.00	9.0
NaN	git	NaN	NaN	NaN

```
# apply(series + dataframe)
temp = pd.Series([10,20,30,40,50])
```

```
temp
```

```
0    10
1    20
2    30
3    40
4    50
dtype: int64
```

```
def sigmoid(value):
  return 1/1+np.exp(-value)
```

```
temp.apply(sigmoid)
```

```
0    1.000045
1    1.000000
2    1.000000
3    1.000000
4    1.000000
dtype: float64
```

```
points_df = pd.DataFrame(
    {
        '1st point':[(3,4),(-6,5),(0,0),(-10,1),(4,5)],
        '2nd point':[(-3,4),(0,0),(2,2),(10,10),(1,1)]
    }
)
```

```
points_df
```

	1st point	2nd point
0	(3, 4)	(-3, 4)
1	(-6, 5)	(0, 0)
2	(0, 0)	(2, 2)
3	(-10, 1)	(10, 10)
4	(4, 5)	(1, 1)

```
def euclidean(row):
  pt_A = row['1st point']
  pt_B = row['2nd point']
```

```
points_df['distance'] = points_df.apply(euclidean,axis=1)
points_df
```

	1st point	2nd point	distance
0	(3, 4)	(-3, 4)	6.000000
1	(-6, 5)	(0, 0)	7.810250
2	(0, 0)	(2, 2)	2.828427
3	(-10, 1)	(10, 10)	21.931712
4	(4, 5)	(1, 1)	5.000000

```
# isin(series)
```

```
# corr
```

```
# nlargest and nsmallest(series and dataframe)
```

```
# insert(dataframe)
```

```
# copy(series + dataframe)
```