

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

▼ Creating DataFrame

```
# using lists
student_data = [
    [100,80,10],
    [90,70,7],
    [120,100,14],
    [80,50,2]
]

pd.DataFrame(student_data,columns=['iq','marks','package'])
```



	iq	marks	package
0	100	80	10
1	90	70	7
2	120	100	14
3	80	50	2

```
# using dicts

student_dict = {
    'name':['nitish','ankit','rupesh','rishabh','amit','ankita'],
    'iq':[100,90,120,80,0,0],
    'marks':[80,70,100,50,0,0],
    'package':[10,7,14,2,0,0]
}

students = pd.DataFrame(student_dict)
students.set_index('name',inplace=True)
students
```

	iq	marks	package
name			
nitish	100	80	10
ankit	90	70	7
rupesh	120	100	14
rishabh	80	50	2
amit	0	0	0
ankita	0	0	0

```
# using read_csv
movies = pd.read_csv('movies.csv')
movies
```

	title_x	imdb_id	poster_path	wiki_link	title_y	original_title
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
4	Evening Shadows	tt6028796	NaN	https://en.wikipedia.org/wiki/Evening_Shadows	Evening Shadows	Evening Shadows
...
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 Saath Rahen
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 Ka Safar
1626	Sabse Bada Sukh	tt0069204	NaN	https://en.wikipedia.org/wiki/Sabse_Bada_Sukh	Sabse Bada Sukh	Sabse Bada Sukh
1627	Daaka	tt10833860	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wiki/Daaka	Daaka	Daaka

```
ipl = pd.read_csv('ipl-matches.csv')
ipl
```

	ID	City	Date	Season	MatchNumber	Team1	Team2	Venue	TossWinner	TossDecision	SuperOver	Winning
0	1312200	Ahmedabad	2022-05-29	2022	Final	Rajasthan Royals	Gujarat Titans	Narendra Modi Stadium, Ahmedabad	Rajasthan Royals	bat	N	Gujarat Titans
1	1312199	Ahmedabad	2022-05-27	2022	Qualifier 2	Royal Challengers Bangalore	Rajasthan Royals	Narendra Modi Stadium, Ahmedabad	Rajasthan Royals	field	N	Rajasthan Royals
2	1312198	Kolkata	2022-05-25	2022	Eliminator	Royal Challengers Bangalore	Lucknow Super Giants	Eden Gardens, Kolkata	Lucknow Super Giants	field	N	Chennai Super Kings
3	1312197	Kolkata	2022-05-24	2022	Qualifier 1	Rajasthan Royals	Gujarat Titans	Eden Gardens, Kolkata	Gujarat Titans	field	N	Gujarat Titans
4	1304116	Mumbai	2022-05-22	2022	70	Sunrisers Hyderabad	Punjab Kings	Wankhede Stadium, Mumbai	Sunrisers Hyderabad	bat	N	Punjab Kings
...
945	335986	Kolkata	2008-04-20	2007/08	4	Kolkata Knight Riders	Deccan Chargers	Eden Gardens	Deccan Chargers	bat	N	Kolkata Knight Riders
946	335985	Mumbai	2008-04-20	2007/08	5	Mumbai Indians	Royal Challengers Bangalore	Wankhede Stadium	Mumbai Indians	bat	N	Chennai Super Kings
947	335984	Delhi	2008-04-19	2007/08	3	Delhi Daredevils	Rajasthan Royals	Feroz Shah Kotla	Rajasthan Royals	bat	N	Delhi Daredevils
948	335983	Chandigarh	2008-04-19	2007/08	2	Kings XI Punjab	Chennai Super Kings	Punjab Cricket Association Stadium, Mohali	Chennai Super Kings	bat	N	Chennai Super Kings
949	335982	Bangalore	2008-04-18	2007/08	1	Royal Challengers Bangalore	Kolkata Knight Riders	M Chinnaswamy Stadium	Royal Challengers Bangalore	field	N	Kolkata Knight Riders

▼ DataFrame Attributes and Methods

```
# shape
movies.shape
ipl.shape

(950, 20)
```

```
# dtypes
movies.dtypes
ipl.dtypes
```

```
ID          int64
City        object
Date        object
Season      object
MatchNumber object
Team1       object
Team2       object
Venue       object
TossWinner  object
TossDecision object
```

```
SuperOver          object
WinningTeam        object
WonBy              object
Margin            float64
method            object
Player_of_Match    object
Team1Players       object
Team2Players       object
Umpire1            object
Umpire2            object
dtype: object

# index
movies.index
ipl.index

RangeIndex(start=0, stop=950, step=1)

# columns
movies.columns
ipl.columns
student.columns

Index(['iq', 'marks', 'package'], dtype='object')

# values
student.values
ipl.values

array([[1312200, 'Ahmedabad', '2022-05-29', ...,
        "['WP Saha', 'Shubman Gill', 'MS Wade', 'HH Pandya', 'DA Miller', 'R Tewatia', 'Rashid Khan', 'R Sai Kishore', 'LH Ferguson',
        'Yash Dayal', 'Mohammed Shami']]",
        'CB Gaffaney', 'Nitin Menon'],
       [1312199, 'Ahmedabad', '2022-05-27', ...,
        "['YBK Jaiswal', 'JC Buttler', 'SV Samson', 'D Padikkal', 'SO Hetmyer', 'R Parag', 'R Ashwin', 'TA Boult', 'YS Chahal', 'M
        Prasidh Krishna', 'OC McCoy']]",
        'CB Gaffaney', 'Nitin Menon'],
       [1312198, 'Kolkata', '2022-05-25', ...,
        "['Q de Kock', 'KL Rahul', 'M Vohra', 'DJ Hooda', 'MP Stoinis', 'E Lewis', 'KH Pandya', 'PVD Chameera', 'Mohsin Khan', 'Avesh
        Khan', 'Ravi Bishnoi']]",
        'J Madanagopal', 'MA Gough'],
       ...,
       [335984, 'Delhi', '2008-04-19', ...,
        "['T Kohli', 'YK Pathan', 'SR Watson', 'M Kaif', 'DS Lehmann', 'RA Jadeja', 'M Rawat', 'D Salunkhe', 'SK Warne', 'SK Trivedi',
        'MM Patel']]",
        'Aleem Dar', 'GA Pratapkumar'],
       [335983, 'Chandigarh', '2008-04-19', ...,
        "['PA Patel', 'ML Hayden', 'MEK Hussey', 'MS Dhoni', 'SK Raina', 'JDP Oram', 'S Badrinath', 'Joginder Sharma', 'P Amarnath',
        'MS Gony', 'M Muralitharan']]",
        'MR Benson', 'SL Shastri'],
       [335982, 'Bangalore', '2008-04-18', ...,
        "['SC Ganguly', 'BB McCullum', 'RT Ponting', 'DJ Hussey', 'Mohammad Hafeez', 'LR Shukla', 'WP Saha', 'AB Agarkar', 'AB Dinda',
        'M Kartik', 'I Sharma']]",
        'Asad Rauf', 'RE Koertzen']], dtype=object)

# head and tail
movies.head(2)
```

	title_x	imdb_id	poster_path	
	Uri: The Surgical Strike	tt8291224	https://upload.wikimedia.org/wikipedia/en/thumb/https://en.wikipedia.org/wiki/U	
1	Battalion 609	tt9472208	NaN	https://en.wikipedia.org/

```
ipl.tail(2)
```

	ID	City	Date	Season	MatchNumber	Team1	Team2	Venue	Tc
948	335983	Chandigarh	2008-04-19	2007/08	2	Kings XI Punjab	Chennai Super Kings	Punjab Cricket Association Stadium, Mohali	Si
949	335982	Bangalore	2008-04-18	2007/08	1	Royal Challengers Bangalore	Kolkata Knight Riders	M Chinnaswamy Stadium	C

```
# sample
ipl.sample(5)
```

	ID	City	Date	Season	MatchNumber	Team1	Team2	Venue
336	1082628	Mumbai	2017-05-01	2017	38	Mumbai Indians	Royal Challengers Bangalore	Wankhede Stadium
98	1254107	Sharjah	2021-09-25	2021	37	Punjab Kings	Sunrisers Hyderabad	Sharjah Cricket Stadium
890	392182	Cape Town	2009-04-18	2009	2	Royal Challengers Bangalore	Rajasthan Royals	Newlands
157	1216533	Abu Dhabi	2020-10-19	2020/21	37	Chennai Super Kings	Rajasthan Royals	Sheikh Zayed Stadium
386	980991	Chandigarh	2016-05-15	2016	46	Kings XI Punjab	Sunrisers Hyderabad	Punjab Cricket Association IS Bindra Stadium, ...

```
# info
movies.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1629 entries, 0 to 1628
Data columns (total 18 columns):
#   Column                Non-Null Count  Dtype
---  -
0   title_x                1629 non-null   object
1   imdb_id                1629 non-null   object
2   poster_path            1526 non-null   object
3   wiki_link              1629 non-null   object
4   title_y                1629 non-null   object
5   original_title         1629 non-null   object
6   is_adult               1629 non-null   int64
7   year_of_release        1629 non-null   int64
8   runtime                1629 non-null   object
9   genres                 1629 non-null   object
10  imdb_rating            1629 non-null   float64
11  imdb_votes             1629 non-null   int64
12  story                  1609 non-null   object
13  summary                1629 non-null   object
14  tagline                557 non-null    object
15  actors                 1624 non-null   object
```

```
16 wins_nominations  707 non-null    object
17 release_date      1522 non-null   object
dtypes: float64(1), int64(3), object(14)
memory usage: 229.2+ KB

ipl.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 950 entries, 0 to 949
Data columns (total 20 columns):
#   Column                Non-Null Count  Dtype
---  -
0   ID                    950 non-null   int64
1   City                 899 non-null   object
2   Date                 950 non-null   object
3   Season              950 non-null   object
4   MatchNumber         950 non-null   object
5   Team1               950 non-null   object
6   Team2               950 non-null   object
7   Venue               950 non-null   object
8   TossWinner          950 non-null   object
9   TossDecision        950 non-null   object
10  SuperOver           946 non-null   object
11  WinningTeam         946 non-null   object
12  WonBy               950 non-null   object
13  Margin              932 non-null   float64
14  method              19 non-null    object
15  Player_of_Match     946 non-null   object
16  Team1Players        950 non-null   object
17  Team2Players        950 non-null   object
18  Umpire1             950 non-null   object
19  Umpire2             950 non-null   object
dtypes: float64(1), int64(1), object(18)
memory usage: 148.6+ KB

# describe
movies.describe()
```

	is_adult	year_of_release	imdb_rating	imdb_votes
count	1629.0	1629.000000	1629.000000	1629.000000
mean	0.0	2010.263966	5.557459	5384.263352
std	0.0	5.381542	1.567609	14552.103231
min	0.0	2001.000000	0.000000	0.000000
25%	0.0	2005.000000	4.400000	233.000000
50%	0.0	2011.000000	5.600000	1000.000000
75%	0.0	2015.000000	6.800000	4287.000000
max	0.0	2019.000000	9.400000	310481.000000

```
ipl.describe()
```

	ID	Margin
count	9.500000e+02	932.000000
mean	8.304852e+05	17.056867
std	3.375678e+05	21.633109
min	3.359820e+05	1.000000
25%	5.012612e+05	6.000000
50%	8.297380e+05	8.000000
75%	1.175372e+06	19.000000
max	1.312200e+06	146.000000

```
# isnull
movies.isnull().sum()

title_x      0
imdb_id      0
poster_path  103
```

```
wiki_link      0
title_y       0
original_title 0
is_adult       0
year_of_release 0
runtime       0
genres        0
imdb_rating   0
imdb_votes    0
story        20
summary       0
tagline      1072
actors        5
wins_nominations 922
release_date  107
dtype: int64
```

```
# duplicated
movies.duplicated().sum()
```

```
0
```

```
students.duplicated().sum()
```

```
1
```

```
# rename
students
```

	iq	percent	lpa
0	100	80	10
1	90	70	7
2	120	100	14
3	80	50	2
4	0	0	0
5	0	0	0

```
students.rename(columns={'marks':'percent','package':'lpa'},inplace=True)
```

Math Methods

```
# sum -> axis argument
students.sum(axis=0)
```

```
iq      390
percent 300
lpa      33
dtype: int64
```

```
students.mean(axis=1)
```

```
0    63.333333
1    55.666667
2    78.000000
3    44.000000
4     0.000000
5     0.000000
dtype: float64
```

```
students.var()
```

```
iq      2710.0
percent 1760.0
lpa      33.5
dtype: float64
```

Start coding or [generate](#) with AI.

✓ Selecting cols from a DataFrame

```
# single cols
movies['title_x']
```

```
0          Uri: The Surgical Strike
1          Battalion 609
2    The Accidental Prime Minister (film)
3          Why Cheat India
4          Evening Shadows
...
1624        Tera Mera Saath Rahen
1625        Yeh Zindagi Ka Safar
1626        Sabse Bada Sukh
1627          Daaka
1628        Humsafar
Name: title_x, Length: 1629, dtype: object
```

```
ipl['Venue']
```

```
0          Narendra Modi Stadium, Ahmedabad
1          Narendra Modi Stadium, Ahmedabad
2          Eden Gardens, Kolkata
3          Eden Gardens, Kolkata
4          Wankhede Stadium, Mumbai
...
945          Eden Gardens
946          Wankhede Stadium
947          Feroz Shah Kotla
948    Punjab Cricket Association Stadium, Mohali
949          M Chinnaswamy Stadium
Name: Venue, Length: 950, dtype: object
```

```
# multiple cols
movies[['year_of_release', 'actors', 'title_x']]
```

	year_of_release	actors	title_x
0	2019	Vicky Kaushal Paresh Rawal Mohit Raina Yami Ga...	Uri: The Surgical Strike
1	2019	Vicky Ahuja Shoaib Ibrahim Shrikant Kamat Elen...	Battalion 609
2	2019	Anupam Kher Akshaye Khanna Aahana Kumra Atul S...	The Accidental Prime Minister (film)
3	2019	Emraan Hashmi Shreya Dhanwanthary Snigdha Deep ...	Why Cheat India
4	2018	Mona Ambegaonkar Ananth Narayan Mahadevan Deva...	Evening Shadows
...
1624	2001	Ajay Devgn Sonali Bendre Namrata Shirodkar Pre...	Tera Mera Saath Rahen
1625	2001	Ameesha Patel Jimmy Sheirgill Nafisa Ali Gulsh...	Yeh Zindagi Ka Safar

```
ipl[['Team1', 'Team2', 'WinningTeam']]
```


	Team1	Team2	WinningTeam
0	Rajasthan Royals	Gujarat Titans	Gujarat Titans
1	Royal Challengers Bangalore	Rajasthan Royals	Rajasthan Royals
2	Royal Challengers Bangalore	Lucknow Super Giants	Royal Challengers Bangalore
3	Rajasthan Royals	Gujarat Titans	Gujarat Titans
4	Sunrisers Hyderabad	Punjab Kings	Punjab Kings
...
945	Kolkata Knight Riders	Deccan Chargers	Kolkata Knight Riders
946	Mumbai Indians	Royal Challengers Bangalore	Royal Challengers Bangalore
947	Delhi Daredevils	Rajasthan Royals	Delhi Daredevils
948	Kings XI Punjab	Chennai Super Kings	Chennai Super Kings
949	Royal Challengers Bangalore	Kolkata Knight Riders	Kolkata Knight Riders

950 rows × 3 columns

✓ Selecting rows from a DataFrame

- **iloc** - searches using index positions
- **loc** - searches using index labels

single row

movies.iloc[5]

```

title_x                               Soni (film)
imdb_id                             tt6078866
poster_path  https://upload.wikimedia.org/wikipedia/en/thum...
wiki_link    https://en.wikipedia.org/wiki/Soni_(film)
title_y                               Soni
original_title                        Soni
is_adult                               0
year_of_release                       2018
runtime                               97
genres                                Drama
imdb_rating                           7.2
imdb_votes                           1595
story      Soni a young policewoman in Delhi and her su...
summary    While fighting crimes against women in Delhi ...
tagline                                          NaN
actors      Geetika Vidya Ohlyan|Saloni Batra|Vikas Shukla...
wins_nominations  3 wins & 5 nominations
release_date      18 January 2019 (USA)
Name: 5, dtype: object

```

multiple row

movies.iloc[:5]

	title_x	imdb_id	poster_path	
0	Uri: The Surgical Strike	tt8291224	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wiki/
1	Battalion 609	tt9472208	NaN	https://en.wikipedia.or
2	The Accidental Prime Minister (film)	tt6986710	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wiki/
3	Why Cheat India	tt8108208	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wil
4	Evening Shadows	tt6028796	NaN	https://en.wikipedia.org/wik

```
# fancy indexing
movies.iloc[[0,4,5]]
```

	title_x	imdb_id	poster_path	
0	Uri: The Surgical Strike	tt8291224	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wiki/L
4	Evening Shadows	tt6028796	NaN	https://en.wikipedia.org/wiki/l
5	Soni (film)	tt6078866	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.c

```
# loc
students
```

	iq	marks	package
name			
nitish	100	80	10
ankit	90	70	7
rupesh	120	100	14
rishabh	80	50	2
amit	0	0	0
ankita	0	0	0

```
students.loc['nitish']
```

```
iq      100
marks   80
package 10
Name: nitish, dtype: int64
```

```
students.loc['nitish':'rishabh':2]
```

	iq	marks	package
name			
nitish	100	80	10
rupesh	120	100	14

```
students.loc[['nitish','ankita','rupesh']]
```

	iq	marks	package
name			
nitish	100	80	10
ankita	0	0	0
rupesh	120	100	14

```
students.iloc[[0,3,4]]
```

	iq	marks	package
name			
nitish	100	80	10
rishabh	80	50	2
amit	0	0	0

▼ Selecting both rows and cols

```
movies.iloc[0:3,0:3]
```

		title_x	imdb_id	poster_path
0	Uri: The Surgical Strike	tt8291224	https://upload.wikimedia.org/wikipedia/en/thum...	
1	Battalion 609	tt9472208		NaN
2	The Accidental Prime Minister (film)	tt6986710	https://upload.wikimedia.org/wikipedia/en/thum...	

```
movies.loc[0:2,'title_x':'poster_path']
```

	title_x	imdb_id	poster_path
0	Uri: The Surgical Strike	tt8291224	https://upload.wikimedia.org/wikipedia/en/thum...
1	Battalion 609	tt9472208	NaN
2	The Accidental Prime Minister (film)	tt6986710	https://upload.wikimedia.org/wikipedia/en/thum...

▼ Filtering a DataFrame

```
ipl.head(2)
```

	ID	City	Date	Season	MatchNumber	Team1	Team2	Venue	Tos
0	1312200	Ahmedabad	2022-05-29	2022	Final	Rajasthan Royals	Gujarat Titans	Narendra Modi Stadium, Ahmedabad	R
1	1312199	Ahmedabad	2022-05-27	2022	Qualifier 2	Royal Challengers Bangalore	Rajasthan Royals	Narendra Modi Stadium, Ahmedabad	R

```
# find all the final winners
mask = ipl['MatchNumber'] == 'Final'
new_df = ipl[mask]
new_df[['Season', 'WinningTeam']]
```

```
ipl[ipl['MatchNumber'] == 'Final'][['Season', 'WinningTeam']]
```

	Season	WinningTeam
0	2022	Gujarat Titans
74	2021	Chennai Super Kings
134	2020/21	Mumbai Indians
194	2019	Mumbai Indians
254	2018	Chennai Super Kings
314	2017	Mumbai Indians
373	2016	Sunrisers Hyderabad
433	2015	Mumbai Indians
492	2014	Kolkata Knight Riders
552	2013	Mumbai Indians
628	2012	Kolkata Knight Riders
702	2011	Chennai Super Kings
775	2009/10	Chennai Super Kings
835	2009	Deccan Chargers
892	2007/08	Rajasthan Royals

```
# how many super over finishes have occurred
ipl[ipl['SuperOver'] == 'Y'].shape[0]
```

```
14
```

```
# how many matches has csk won in kolkata
ipl[(ipl['City'] == 'Kolkata') & (ipl['WinningTeam'] == 'Chennai Super Kings')].shape[0]
```

```
5
```

```
# toss winner is match winner in percentage
(ipl[ipl['TossWinner'] == ipl['WinningTeam']].shape[0]/ipl.shape[0])*100

51.473684210526315

# movies with rating higher than 8 and votes>10000
movies[(movies['imdb_rating'] > 8.5) & (movies['imdb_votes'] > 10000)].shape[0]

0

# Action movies with rating higher than 7.5
# mask1 = movies['genres'].str.split('|').apply(lambda x:'Action' in x)
mask1 = movies['genres'].str.contains('Action')
mask2 = movies['imdb_rating'] > 7.5

movies[mask1 & mask2]
```

	title_x	imdb_id	poster_path	wiki_link	title_y	origi
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: Tl
41	Family of Thakurganj	tt8897986	https://upload.wikimedia.org/wikipedia/en/9/99...	https://en.wikipedia.org/wiki/Family_of_Thakur...	Family of Thakurganj	.
84	Mukkabaaz	tt7180544	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wiki/Mukkabaaz	The Brawler	I
106	Raazi	tt7098658	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wiki/Raazi	Raazi	
110	Parmanu: The Story of Pokhran	tt6826438	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wiki/Parmanu:_The_Sto...	Parmanu: The Story of Pokhran	Par Story
112	Bhavesh Joshi Superhero	tt6129302	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wiki/Bhavesh_Joshi_Su...	Bhavesh Joshi Superhero	Bha
169	The Ghazi Attack	tt6299040	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wiki/The_Ghazi_Attack...	The Ghazi Attack	The G
219	Raag Desh (film)	tt6080746	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wiki/Raagdes	Raag Desh	I
258	Irudhi Suttru	tt5310090	https://upload.wikimedia.org/wikipedia/en/f/fe...	https://en.wikipedia.org/wiki/Saala_Khadoos	Saala Khadoos	Saal
280	Laal Rang	tt5600714	NaN	https://en.wikipedia.org/wiki/Laal_Rang	Laal Rang	

```
# write a function that can return the track record of 2 teams against each other
```

▼ Adding new cols

```
# completely new
movies['Country'] = 'India'
movies.head()
```

	title_x	imdb_id	poster_path	
0	Uri: The Surgical Strike	tt8291224	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wiki/
1	Battalion 609	tt9472208	NaN	https://en.wikipedia.or
2	The Accidental Prime Minister (film)	tt6986710	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wiki/
3	Why Cheat India	tt8108208	https://upload.wikimedia.org/wikipedia/en/thum...	https://en.wikipedia.org/wil
4	Evening Shadows	tt6028796	NaN	https://en.wikipedia.org/wik

```
# from existing ones
movies.dropna(inplace=True)
```

```
movies['lead actor'] = movies['actors'].str.split('|').apply(lambda x:x[0])
movies.head()
```

```

title_x    imdb_id    poster_path

11    Gully
    Bov    tt2395469    https://upload.wikimedia.org/wikipedia/en/thum...    https://en.wikipedia.c

movies.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 298 entries, 11 to 1623
Data columns (total 19 columns):
#   Column                Non-Null Count  Dtype
---  -
0   title_x                298 non-null   object
1   imdb_id                298 non-null   object
2   poster_path            298 non-null   object
3   wiki_link              298 non-null   object
4   title_y                298 non-null   object
5   original_title         298 non-null   object
6   is_adult               298 non-null   int64
7   year_of_release        298 non-null   int64
8   runtime                298 non-null   object
9   genres                 298 non-null   object
10  imdb_rating             298 non-null   float64
11  imdb_votes              298 non-null   int64
12  story                   298 non-null   object
13  summary                 298 non-null   object
14  tagline                 298 non-null   object
15  actors                  298 non-null   object
16  wins_nominations        298 non-null   object
17  release_date            298 non-null   object
18  Country                 298 non-null   object
dtypes: float64(1), int64(3), object(15)
memory usage: 46.6+ KB
```

▼ Important DataFrame Functions

```

# astype
ipl.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 950 entries, 0 to 949
Data columns (total 20 columns):
#   Column                Non-Null Count  Dtype
---  -
0   ID                    950 non-null   int64
1   City                  899 non-null   object
2   Date                  950 non-null   object
3   Season                950 non-null   object
4   MatchNumber           950 non-null   object
5   Team1                 950 non-null   object
6   Team2                 950 non-null   object
7   Venue                 950 non-null   object
8   TossWinner            950 non-null   object
9   TossDecision          950 non-null   object
10  SuperOver             946 non-null   object
11  WinningTeam           946 non-null   object
12  WonBy                 950 non-null   object
13  Margin                932 non-null   float64
14  method                19 non-null    object
15  Player_of_Match       946 non-null   object
16  Team1Players           950 non-null   object
17  Team2Players           950 non-null   object
18  Umpire1               950 non-null   object
19  Umpire2               950 non-null   object
dtypes: float64(1), int64(1), object(18)
memory usage: 148.6+ KB
```

```
ipl['ID'] = ipl['ID'].astype('int32')
```

```
ipl.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 950 entries, 0 to 949
Data columns (total 20 columns):
```