

```
import pandas as pd
```

```
# Agenda
# sorting
# basic operations 2 mins
# concat
# merging
# imdb dataset
```

```
df=pd.read_csv("/Users/nikhilsanghi/Downloads/dsml-course-main-live/batches/May-Beg-Aug-Ad
```

```
df
```

	country	year	population	continent	life_exp	gdp_cap
0	Afghanistan	1952	8425333	Asia	28.801	779.445314
1	Afghanistan	1957	9240934	Asia	30.332	820.853030
2	Afghanistan	1962	10267083	Asia	31.997	853.100710
3	Afghanistan	1967	11537966	Asia	34.020	836.197138
4	Afghanistan	1972	13079460	Asia	36.088	739.981106
...
1699	Zimbabwe	1987	9216418	Africa	62.351	706.157306
1700	Zimbabwe	1992	10704340	Africa	60.377	693.420786
1701	Zimbabwe	1997	11404948	Africa	46.809	792.449960
1702	Zimbabwe	2002	11926563	Africa	39.989	672.038623
1703	Zimbabwe	2007	12311143	Africa	43.487	469.709298

1704 rows × 6 columns

```
# df.describe()
```

```
df['life_exp'].sum()
```

101344.44467999999

```
df['life_exp'].mean()
```

59.47443936619714

```
df['life_exp'].max()
```

82.603

```
df['life_exp'].min()
```

23.599

```
df['life_exp'].count()
```

1704

```
df2=df[["year","population","life_exp"]]
```

df2

	year	population	life_exp
0	1952	8425333	28.801
1	1957	9240934	30.332
2	1962	10267083	31.997
3	1967	11537966	34.020
4	1972	13079460	36.088
...
1699	1987	9216418	62.351
1700	1992	10704340	60.377
1701	1997	11404948	46.809
1702	2002	11926563	39.989
1703	2007	12311143	43.487

1704 rows × 3 columns

df

	country	year	population	continent	life_exp	gdp_cap
0	Afghanistan	1952	8425333	Asia	28.801	779.445314
1	Afghanistan	1957	9240934	Asia	30.332	820.853030
2	Afghanistan	1962	10267083	Asia	31.997	853.100710
3	Afghanistan	1967	11537966	Asia	34.020	836.197138
4	Afghanistan	1972	13079460	Asia	36.088	739.981106

```
df.sort_values(["year"])
```

	country	year	population	continent	life_exp	gdp_cap
0	Afghanistan	1952	8425333	Asia	28.801	779.445314
528	France	1952	42459667	Europe	67.410	7029.809327
540	Gabon	1952	420702	Africa	37.003	4293.476475
1656	West Bank and Gaza	1952	1030585	Asia	43.160	1515.592329
552	Gambia	1952	284320	Africa	30.000	485.230659
...
1127	Niger	2007	12894865	Africa	56.867	619.676892
1139	Nigeria	2007	135031164	Africa	46.859	2013.977305
1151	Norway	2007	4627926	Europe	80.196	49357.190170
1175	Pakistan	2007	169270617	Asia	65.483	2605.947580
1703	Zimbabwe	2007	12311143	Africa	43.487	469.709298

1704 rows × 6 columns

```
df.sort_values(["year"])
```

```
df.sort_values(["year"],ascending=False)
```

	country	year	population	continent	life_exp	gdp_cap
1703	Zimbabwe	2007	12311143	Africa	43.487	469.709298
491	Equatorial Guinea	2007	551201	Africa	51.579	12154.089750
515	Ethiopia	2007	76511887	Africa	52.947	690.805576
527	Finland	2007	5238460	Europe	79.313	33207.084400

```
df.sort_values(["year","life_exp"]).head(20)
```

	country	year	population	continent	life_exp	gdp_cap
0	Afghanistan	1952	8425333	Asia	28.801	779.445314
552	Gambia	1952	284320	Africa	30.000	485.230659
36	Angola	1952	4232095	Africa	30.015	3520.610273
1344	Sierra Leone	1952	2143249	Africa	30.331	879.787736
1032	Mozambique	1952	6446316	Africa	31.286	468.526038
192	Burkina Faso	1952	4469979	Africa	31.975	543.255241
624	Guinea-Bissau	1952	580653	Africa	32.500	299.850319
1668	Yemen, Rep.	1952	4963829	Asia	32.548	781.717576
1392	Somalia	1952	2526994	Africa	32.978	1135.749842
612	Guinea	1952	2664249	Africa	33.609	510.196492
948	Mali	1952	3838168	Africa	33.685	452.336981
504	Ethiopia	1952	20860941	Africa	34.078	362.146280
480	Equatorial Guinea	1952	216964	Africa	34.482	375.643123
420	Djibouti	1952	63149	Africa	34.812	2669.529475
252	Central African Republic	1952	1291695	Africa	35.463	1071.310713
492	Eritrea	1952	1438760	Africa	35.928	328.940557
1068	Nepal	1952	9182536	Asia	36.157	545.865723
924	Malawi	1952	2917802	Africa	36.256	369.165080
1044	Myanmar	1952	20092996	Asia	36.319	331.000000
1128	Nigeria	1952	33119096	Africa	36.324	1077.281856

```
df.sort_values(["country","life_exp"]).head(20)
```

	country	year	population	continent	life_exp	gdp_cap
0	Afghanistan	1952	8425333	Asia	28.801	779.445314
1	Afghanistan	1957	9240934	Asia	30.332	820.853030
2	Afghanistan	1962	10267083	Asia	31.997	853.100710
3	Afghanistan	1967	11537966	Asia	34.020	836.197138
4	Afghanistan	1972	13079460	Asia	36.088	739.981106
5	Afghanistan	1977	14880372	Asia	38.438	786.113360
6	Afghanistan	1982	12881816	Asia	39.854	978.011439
7	Afghanistan	1987	13867957	Asia	40.822	852.395945
8	Afghanistan	1992	16317921	Asia	41.674	649.341395
9	Afghanistan	1997	22227415	Asia	41.763	635.341351
10	Afghanistan	2002	25268405	Asia	42.129	726.734055
11	Afghanistan	2007	31889923	Asia	43.828	974.580338
12	Albania	1952	1282697	Europe	55.230	1601.056136
13	Albania	1957	1476505	Europe	59.280	1942.284244
14	Albania	1962	1728137	Europe	64.820	2312.888958
15	Albania	1967	1984060	Europe	66.220	2760.196931
16	Albania	1972	2263554	Europe	67.690	3313.422188

```
df.sort_values(["year","life_exp"],ascending=False).head(20)
```

	country	year	population	continent	life_exp	gdp_cap
803	Japan	2007	127467972	Asia	82.603	31656.06806
671	Hong Kong, China	2007	6980412	Asia	82.208	39724.97867
695	Iceland	2007	301931	Europe	81.757	36180.78919
1487	Switzerland	2007	7554661	Europe	81.701	37506.41907
71	Australia	2007	20434176	Oceania	81.235	34435.36744
1427	Spain	2007	40448191	Europe	80.941	28821.06370
1475	Sweden	2007	9031088	Europe	80.884	33859.74835

```
df2=df.sort_values(["year","life_exp"],ascending=[False,True])
df2
```

	country	year	population	continent	life_exp	gdp_cap
1463	Swaziland	2007	1133066	Africa	39.613	4513.480643
1043	Mozambique	2007	19951656	Africa	42.082	823.685621
1691	Zambia	2007	11746035	Africa	42.384	1271.211593
1355	Sierra Leone	2007	6144562	Africa	42.568	862.540756
887	Lesotho	2007	2012649	Africa	42.592	1569.331442
...
408	Denmark	1952	4334000	Europe	70.780	9692.385245
1464	Sweden	1952	7124673	Europe	71.860	8527.844662
1080	Netherlands	1952	10381988	Europe	72.130	8941.571858
684	Iceland	1952	147962	Europe	72.490	7267.688428
1140	Norway	1952	3327728	Europe	72.670	10095.421720

1704 rows × 6 columns

```
df2=df.sort_values(["year","life_exp"],ascending=[False,True])
df2
```

	country	year	population	continent	life_exp	gdp_cap
1463	Swaziland	2007	1133066	Africa	39.613	4513.480643
1043	Mozambique	2007	19951656	Africa	42.082	823.685621
1691	Zambia	2007	11746035	Africa	42.384	1271.211593
1355	Sierra Leone	2007	6144562	Africa	42.568	862.540756
887	Lesotho	2007	2012649	Africa	42.592	1569.331442

```
df2[df2["year"]==2002]
```

	country	year	population	continent	life_exp	gdp_cap
1690	Zambia	2002	10595811	Africa	39.193	1071.613938
1702	Zimbabwe	2002	11926563	Africa	39.989	672.038623
46	Angola	2002	10866106	Africa	41.003	2773.287312
1354	Sierra Leone	2002	5359092	Africa	41.012	699.489713
10	Afghanistan	2002	25268405	Asia	42.129	726.734055
...
70	Australia	2002	19546792	Oceania	80.370	30687.754730
694	Iceland	2002	288030	Europe	80.500	31163.201960
1486	Switzerland	2002	7361757	Europe	80.620	34480.957710
670	Hong Kong, China	2002	6762476	Asia	81.495	30209.015160
802	Japan	2002	127065841	Asia	82.000	28604.591900

142 rows × 6 columns

Concatenating

```
x=pd.DataFrame({"A":[10,30],"B":[20,40]})
y=pd.DataFrame({"A":[10,30,100],"C":[20,40,100],"D":[200,200,300]})
x
```

	A	B
0	10	20
1	30	40

y

	A	C	D
0	10	20	200
1	30	40	200
2	100	100	300

```
df4=pd.concat([x,y])
df4
```

	A	B	C	D
0	10	20.0	NaN	NaN
1	30	40.0	NaN	NaN
0	10	NaN	20.0	200.0
1	30	NaN	40.0	200.0
2	100	NaN	100.0	300.0


```
df4=pd.concat([x,y],ignore_index=True)
df4
```

	A	B	C	D
0	10	20.0	NaN	NaN
1	30	40.0	NaN	NaN
2	10	NaN	20.0	200.0
3	30	NaN	40.0	200.0
4	100	NaN	100.0	300.0

```
pd.concat([x,y],axis=1)
```

	A	B	A	C	D
0	10.0	20.0	10	20	200
1	30.0	40.0	30	40	200
2	NaN	NaN	100	100	300

```
pd.merge??
```

ppppdScreenshot%202022-09-02%20at%2022.30.20.png


```
users=pd.DataFrame({'userid':[1,2,3], 'name':['sarath', 'shahid', 'khushali']})
```

users

	userid	name
0	1	sarath
1	2	shahid
2	3	khushali

```
msgs=pd.DataFrame({'userid':[1,1,2,4], 'msg':['hmm', 'accha', 'theek hai', 'nice']})
```

msgs

	userid	msg
0	1	hmm
1	1	accha
2	2	theek hai
3	4	nice

```
users.merge(msgs,on="userid")
```

	userid	name	msg
0	1	sarath	hmm
1	1	sarath	accha
2	2	shahid	theek hai

pd.merge?how=

```
users.merge(msgs,on="userid",how="outer")
```

	userid	name	msg
0	1	sarath	hmm
1	1	sarath	accha
2	2	shahid	theek hai
3	3	khushali	NaN
4	4	NaN	nice

```
users.merge(msgs,on="userid",how="left")
```

	userid	name	msg
0	1	sarath	hmm
1	1	sarath	accha
2	2	shahid	theek hai
3	3	khushali	NaN

```
users.merge(msgs,on="userid",how="right")
```

	userid	name	msg
0	1	sarath	hmm
1	1	sarath	accha
2	2	shahid	theek hai
3	4	NaN	nice

```
users.merge(msgs,on="userid",how="inner")
```

	userid	name	msg
0	1	sarath	hmm
1	1	sarath	accha
2	2	shahid	theek hai

```
users.rename(columns={"userid":"id"},inplace=True)
```

msgs

	userid	msg
0	1	hmm
1	1	accha
2	2	theek hai
3	4	nice

users

id name

```
users.merge(msgs,left_on="id",right_on="userid")
```

	id	name	userid	msg
0	1	sarath	1	hmm
1	1	sarath	1	accha
2	2	shahid	2	theek hai

pd.merge?

```
users= pd.DataFrame({"id":[3,2,1],"name":["sarath","shahid","khushali"]})
```

```
msgs=pd.DataFrame({"userid":[1,1,2,4],"msg":["hmm","accha","theek hai","nice"]})
```

users

	id	name
0	3	sarath
1	2	shahid
2	1	khushali

msgs

	userid	msg
0	1	hmm
1	1	accha
2	2	theek hai
3	4	nice

```
users.merge(msgs,left_on="id",right_on="userid")
```

	id	name	userid	msg
0	2	shahid	2	theek hai
1	1	khushali	1	hmm
2	1	khushali	1	accha

#imdb dataset

https://drive.google.com/file/d/1Ws-_s1fHZ9nHfGLVUQurbHDvStePlEJm/view

```
To: /Users/nikhilsanghi/Downloads/dsml-course-main-live/batches/May-Beg-Aug-Adv/moviec
100%|██████████| 112k/112k [00:00<00:00, 551kB/s]
```



To: /Users/nikhilsanghi/Downloads/dsml-course-main-live/batches/May-Beg-Aug-Adv/directories
100% | ██████████ 65.4k/65.4k [00:00<00:00, 1.53MB/s]



	director_name	id	gender
0	James Cameron	4762	Male
1	Gore Verbinski	4763	Male
2	Sam Mendes	4764	Male
3	Christopher Nolan	4765	Male
4	Andrew Stanton	4766	Male
...
2344	Shane Carruth	7106	Male
2345	Neill Dela Llana	7107	NaN
2346	Scott Smith	7108	NaN
2347	Daniel Hsia	7109	Male
2348	Brian Herzlinger	7110	Male

	id	budget	popularity	revenue	title	vote_average	vote_count
0	43597	237000000	150	2787965087	Avatar	7.2	11800
1	43598	300000000	139	961000000	Pirates of the Caribbean: At World's End	6.9	4500
2	43599	245000000	107	880674609	Spectre	6.3	4466
3	43600	250000000	112	1084939099	The Dark Knight Rises	7.6	9106
5	43602	258000000	115	890871626	Spider- Man 3	5.9	3576
...
4736	48363	0	3	321952	The Last Waltz	7.9	64

movies

	id	budget	popularity	revenue	title	vote_average	vote_count
0	43597	237000000	150	2787965087	Avatar	7.2	11800
1	43598	300000000	139	961000000	Pirates of the Caribbean: At World's End	6.9	4500
2	43599	245000000	107	880674609	Spectre	6.3	4466
3	43600	250000000	112	1084939099	The Dark Knight Rises	7.6	9106
5	43602	258000000	115	890871626	Spider- Man 3	5.9	3576
...
4736	48363	0	3	321952	The Last Waltz	7.9	64



```
# movies.merge(directors,left_on="director_id",right_on="id",)
```

```
import numpy as np
```

```
np.all(movies["director_id"].isin(directors["id"]))
```

True

```
data=movies.merge(directors,left_on="director_id",right_on="id",how="inner")
```

data

	id_x	budget	popularity	revenue	title	vote_average	vote_count
0	43597	237000000	150	2787965087	Avatar	7.2	11800
1	43622	200000000	100	1845034188	Titanic	7.5	7562
2	43876	100000000	101	520000000	Terminator 2: Judgment Day	7.7	4185
3	43879	115000000	38	378882411	True Lies	6.8	1116
4	44184	70000000	24	90000098	The Abyss	7.1	808
...
1460	46859	0	14	25288872	Enough Said	6.6	348
1461	47023	6500000	11	13368437	Friends with Money	5.1	128
1462	47524	3000000	5	0	Please Give	6.0	57
1463	47962	0	0	0	Walking and Talking	6.6	7
1464	48229	250000	1	4186931	Lovely &	6.3	23

```
data.drop(columns=["director_id","id_y"],inplace=True)
```

data

	id_x	budget	popularity	revenue	title	vote_average	vote_count
0	43597	237000000	150	2787965087	Avatar	7.2	11800
1	43622	200000000	100	1845034188	Titanic	7.5	7562
2	43876	100000000	101	520000000	Terminator 2: Judgment Day	7.7	4185
3	43879	115000000	38	378882411	True Lies	6.8	1116

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1465 entries, 0 to 1464
Data columns (total 12 columns):
#   Column          Non-Null Count  Dtype
---  -
0   id_x            1465 non-null   int64
1   budget          1465 non-null   int64
2   popularity      1465 non-null   int64
3   revenue         1465 non-null   int64
4   title           1465 non-null   object
5   vote_average    1465 non-null   float64
6   vote_count      1465 non-null   int64
7   year            1465 non-null   int64
8   month           1465 non-null   object
9   day             1465 non-null   object
10  director_name   1465 non-null   object
11  gender          1341 non-null   object
dtypes: float64(1), int64(6), object(5)
memory usage: 148.8+ KB
```

```
data["director_name"].unique()
```

```
array(['James Cameron', 'Gore Verbinski', 'Sam Mendes',
      'Christopher Nolan', 'Sam Raimi', 'Zack Snyder', 'Bryan Singer',
      'Marc Forster', 'Andrew Adamson', 'Rob Marshall',
      'Barry Sonnenfeld', 'Peter Jackson', 'Ridley Scott', 'Chris Weitz',
      'Peter Berg', 'Tim Burton', 'Brett Ratner', 'Michael Bay',
      'Martin Campbell', 'McG', 'James Wan', 'Mike Newell',
      'Guillermo del Toro', 'Steven Spielberg', 'Justin Lin',
      'Roland Emmerich', 'Robert Zemeckis', 'Lilly Wachowski',
      'Jon Favreau', 'Martin Scorsese', 'Rob Cohen', 'David Ayer',
      'Tom Shadyac', 'Doug Liman', 'Kevin Reynolds', 'David Fincher',
      'Francis Lawrence', 'Jon Turteltaub', 'Wolfgang Petersen',
      'Michael Apted', 'Oliver Stone', 'Shawn Levy', 'George Miller',
      'Ron Howard', 'Kenneth Branagh', 'Jonathan Liebesman',
      'M. Night Shyamalan', 'Joe Wright', 'Rob Minkoff', 'Lee Tamahori',
      'Edward Zwick', 'Alex Proyas', 'Richard Donner', 'Ang Lee',
      'Jon M. Chu', 'Bill Condon', 'Louis Leterrier',
      'Alejandro González Iñárritu', 'Paul Greengrass', 'Phillip Noyce',
      'Darren Aronofsky', 'Chris Columbus', 'Robert Schwentke',
```

```
'Guy Ritchie', 'Paul Verhoeven', 'John McTiernan',
'Joel Schumacher', 'John Woo', 'Tim Story', 'James Mangold',
'Roger Donaldson', 'Steven Soderbergh', 'Raja Gosnell',
'Jan de Bont', 'Frank Coraci', 'Michael Mann', 'Peter Chelsom',
'Tony Scott', 'Paul Weitz', 'Adam McKay', 'Chuck Russell',
'Quentin Tarantino', 'Simon West', 'Peter Hyams', 'Tom Tykwer',
'Zhang Yimou', 'Frank Oz', 'Jay Roach', 'Luc Besson',
'Mark Waters', 'Renny Harlin', 'Ben Stiller', 'Dennis Dugan',
'Sydney Pollack', 'Brian De Palma', 'Paul W.S. Anderson',
'Nancy Meyers', 'Peter Segal', 'George A. Romero', 'Todd Phillips',
'Gary Winick', 'Adam Shankman', 'Les Mayfield', 'Ivan Reitman',
'Stephen Hopkins', 'Jonathan Demme', 'Terry Gilliam', 'Joe Dante',
'John Singleton', 'Mike Nichols', 'F. Gary Gray', 'Antoine Fuqua',
'Robert Luketic', 'Barry Levinson', 'Andy Tennant', 'Judd Apatow',
'Garry Marshall', 'Cameron Crowe', 'George Clooney',
'Andrzej Bartkowiak', 'Bobby Farrelly', 'Lawrence Kasdan',
'Clint Eastwood', 'Larry Charles', 'Taylor Hackford',
'Roman Polanski', 'Robert Rodriguez', 'Rob Reiner', 'Tim Hill',
'Robert Redford', 'Kenny Ortega', 'Brian Robbins', 'Brian Levant',
'David O. Russell', 'Jean-Pierre Jeunet', 'Harold Ramis',
'Donald Petrie', 'Joel Coen', 'Rod Lurie', 'David Koepp',
'Uwe Boll', 'Stephen Herek', 'John Madden', 'Wayne Wang',
'Francis Ford Coppola', 'Curtis Hanson', 'John Whitesell',
'Neil Jordan', 'Spike Lee', 'Brian Helgeland',
'Jaume Collet-Serra', 'Andy Fickman', 'Gary Fleder', 'John Landis',
'Danny Boyle', 'Andrew Niccol', 'John Carpenter', 'Wes Anderson',
'David Cronenberg', 'David Gordon Green', 'Richard LaGravenese',
'Stephen Frears', 'David Zucker', 'David R. Ellis', 'David Lynch',
'Gus Van Sant', 'John Glen', 'Catherine Hardwicke',
'Anne Fletcher', 'Wes Craven', 'Nicholas Stoller',
'Stephen Daldry', 'Malcolm D. Lee', 'Steve Miner',
'Paul Thomas Anderson', 'Kirk Jones', 'Kevin Smith', 'Scott Hicks',
'Lasse Hallström', 'Jason Reitman', 'Alexander Payne',
'Woody Allen', 'Jason Friedberg', "Gavin O'Connor",
'Miguel Arteta', 'Richard Linklater', 'Michael Winterbottom',
'Tyler Perry', 'Atom Egoyan', 'Sidney Lumet', 'Mira Nair',
'Michael Moore', 'Mike Leigh', 'James Ivory', 'Brad Anderson',
'Michael Polish', 'Paul Schrader', 'Darren Lynn Bousman',
'Nicole Holofcener'], dtype=object)
```

```
data["director_name"].nunique()
```

```
199
```

```
data.shape
```

```
(1465, 12)
```

```
data
```


	id_x	budget	popularity	revenue	title	vote_average	vote_count
0	43597	237000000	150	2787965087	Avatar	7.2	11800
1	43622	200000000	100	1845034188	Titanic	7.5	7562
2	43876	100000000	101	520000000	Terminator 2: Judgment Day	7.7	4185
3	43879	115000000	38	378882411	True Lies	6.8	1116
4	44184	70000000	24	90000098	The Abyss	7.1	808

data[data["vote_average"]>7]

	id_x	budget	popularity	revenue	title	vote_average	vote_count
0	43597	237000000	150	2787965087	Avatar	7.2	11800
1	43622	200000000	100	1845034188	Titanic	7.5	7562
2	43876	100000000	101	520000000	Terminator 2: Judgment Day	7.7	4185
4	44184	70000000	24	90000098	The Abyss	7.1	808
5	46000	18500000	67	183316455	Aliens	7.7	3220
...
1424	47488	4000000	11	35564473	Bowling for Columbine	7.3	453

[Colab paid products](#) - [Cancel contracts here](#)

