Agenda

- **Concatenation**
  - pd.concat()
  - axis for concat
- **Merge**
  - Concat v/s Merge
  - `left_on` and `right_on`
  - Joins
- **Intoduction to IMDB dataset**
  - Reading two datasets
- **Merging the dataframes**
  - `unique()` and `nunique()`
  - `isin()`
  - Using Left Join for `merge()`
- **Feature Exploration**
  - Create new features
- **Fetching data using pandas**
  - Quering from dataframe - Masking, Filtering, `&` and `|`
- **Apply**

In [ ]:
```
#Concatenation
```

In [2]:
```
import pandas as pd
```

In [3]:
```
users = pd.DataFrame({"userid":[1,2,3],"name":["Ichigo","Santosh","Avinas
users
```

Out[3]:

	userid	name
0	1	Ichigo
1	2	Santosh
2	3	Avinash

In [4]:
```
msgs = pd.DataFrame({"userid":[1,1,2,4],"msg":["hmm","acha","theek hain",
msgs
```

Out[4]:

	userid	msg
0	1	hmm
1	1	acha
2	2	theek hain
3	4	nice

In [5]:
```
1 pd.concat([users,msgs])
```

Out[5]:

	userid	name	msg
0	1	Ichigo	NaN
1	2	Santosh	NaN
2	3	Avinash	NaN
0	1	NaN	hmm
1	1	NaN	acha
2	2	NaN	theek hain
3	4	NaN	nice

In [6]:
```
1 pd.concat([users,msgs],ignore_index=True)
```

Out[6]:

	userid	name	msg
0	1	Ichigo	NaN
1	2	Santosh	NaN
2	3	Avinash	NaN
3	1	NaN	hmm
4	1	NaN	acha
5	2	NaN	theek hain
6	4	NaN	nice

In [7]:
```
1 pd.concat([users,msgs],axis=1)
```

Out[7]:

	userid	name	userid	msg
0	1.0	Ichigo	1	hmm
1	2.0	Santosh	1	acha
2	3.0	Avinash	2	theek hain
3	NaN	NaN	4	nice

In [8]:
```
1 users
```

Out[8]:

	userid	name
0	1	Ichigo
1	2	Santosh
2	3	Avinash

In [9]:
```
1 msgs
```

Out[9]:

	userid	msg
0	1	hmm
1	1	acha
2	2	theek hain
3	4	nice

In [10]:
```
1 pd.merge(users,msgs,on="userid")
```

Out[10]:

	userid	name	msg
0	1	Ichigo	hmm
1	1	Ichigo	acha
2	2	Santosh	theek hain

In [11]:
```
1 #more intuitive method - Second approach
2
3 users.merge(msgs,on="userid")
```

Out[11]:

	userid	name	msg
0	1	Ichigo	hmm
1	1	Ichigo	acha
2	2	Santosh	theek hain

In [12]:
```
1 users.merge(msgs,on="userid",how="left")
```

Out[12]:

	userid	name	msg
0	1	Ichigo	hmm
1	1	Ichigo	acha
2	2	Santosh	theek hain
3	3	Avinash	NaN

In [13]:
```
1 users.merge(msgs,on="userid",how="right")
```

Out[13]:

	userid	name	msg
0	1	Ichigo	hmm
1	1	Ichigo	acha
2	2	Santosh	theek hain
3	4	NaN	nice

In [14]:
```
1 users.merge(msgs,on="userid",how="outer")
```

Out[14]:

	userid	name	msg
0	1	Ichigo	hmm
1	1	Ichigo	acha
2	2	Santosh	theek hain
3	3	Avinash	NaN
4	4	NaN	nice

In [15]:
```
1 users
```

Out[15]:

	userid	name
0	1	Ichigo
1	2	Santosh
2	3	Avinash

In [16]:
```
1 users.rename(columns={"userid":"id"},inplace=True)
```

In [17]:
```
1 users
```

Out[17]:

	id	name
0	1	Ichigo
1	2	Santosh
2	3	Avinash

In [18]:
```
1 msgs
```

Out[18]:

	userid	msg
0	1	hmm
1	1	acha
2	2	theek hain
3	4	nice

In [19]:
```
merged = users.merge(msgs,left_on="id",right_on="userid")
merged
```

Out[19]:

	id	name	userid	msg
0	1	Ichigo	1	hmm
1	1	Ichigo	1	acha
2	2	Santosh	2	theek hain

In [20]:
```
merged.drop(columns="userid")
```

Out[20]:

	id	name	msg
0	1	Ichigo	hmm
1	1	Ichigo	acha
2	2	Santosh	theek hain

In [ ]:
```

```

In [ ]:
```
#IMDB dataset
```

In [21]:
```
!gdown 1s2TkjSpzNc4SyxqRrQleZyDIHlc7bxnd
```

```
Downloading...
From: https://drive.google.com/uc?id=1s2TkjSpzNc4SyxqRrQleZyDIHlc7bxnd (https://drive.google.com/uc?id=1s2TkjSpzNc4SyxqRrQleZyDIHlc7bxnd)
To: /Users/ritnil/Scaler-cohorts/movies.csv
100%|████████████████████████████████████████| 112k/112k [00:00<00:00, 704kB/s]
```

In [22]:
```
!gdown 1Ws-_s1fHZ9nHfGLVUQurbHDvStePlEJm
```

```
Downloading...
From: https://drive.google.com/uc?id=1Ws-_s1fHZ9nHfGLVUQurbHDvStePlEJm (https://drive.google.com/uc?id=1Ws-_s1fHZ9nHfGLVUQurbHDvStePlEJm)
To: /Users/ritnil/Scaler-cohorts/directors.csv
100%|████████████████████████████████████████| 65.4k/65.4k [00:00<00:00, 894kB/s]
```

In [23]:
```
movies=pd.read_csv("/Users/ritnil/Scaler-cohorts/movies.csv")
movies
```

Out[23]:

	Unnamed: 0	id	budget	popularity	revenue	title	vote_average	vote_cou
0	0	43597	237000000	150	2787965087	Avatar	7.2	118
1	1	43598	300000000	139	961000000	Pirates of the Caribbean: At World's End	6.9	45
2	2	43599	245000000	107	880674609	Spectre	6.3	44
3	3	43600	250000000	112	1084939099	The Dark Knight Rises	7.6	91
4	5	43602	258000000	115	890871626	Spider-Man 3	5.9	35
...	...	...	...	...	...	...	...	
1460	4736	48363	0	3	321952	The Last Waltz	7.9	
1461	4743	48370	27000	19	3151130	Clerks	7.4	7
1462	4748	48375	0	7	0	Rampage	6.0	1
1463	4749	48376	0	3	0	Slacker	6.4	
1464	4768	48395	220000	14	2040920	El Mariachi	6.6	2

1465 rows × 12 columns

In [24]:
```
movies.drop(columns=["Unnamed: 0"],inplace=True)
```

In [25]:
```
1 movies
```

Out[25]:

	id	budget	popularity	revenue	title	vote_average	vote_count	director_
0	43597	237000000	150	2787965087	Avatar	7.2	11800	47
1	43598	300000000	139	961000000	Pirates of the Caribbean: At World's End	6.9	4500	47
2	43599	245000000	107	880674609	Spectre	6.3	4466	47
3	43600	250000000	112	1084939099	The Dark Knight Rises	7.6	9106	47
4	43602	258000000	115	890871626	Spider-Man 3	5.9	3576	47
...	...	...	...	...	...	...	...	
1460	48363	0	3	321952	The Last Waltz	7.9	64	48
1461	48370	27000	19	3151130	Clerks	7.4	755	53
1462	48375	0	7	0	Rampage	6.0	131	51
1463	48376	0	3	0	Slacker	6.4	77	55
1464	48395	220000	14	2040920	El Mariachi	6.6	238	50

1465 rows × 11 columns

In [26]:
```
directors=pd.read_csv("/Users/ritnil/Scaler-cohorts/directors.csv")
directors
```

Out[26]:

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

2349 rows × 4 columns

In [27]:
```
directors.drop(columns="Unnamed: 0",inplace=True)
```

In [28]:
```
directors
```

Out[28]:

	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

2349 rows × 3 columns

In [29]:
```
1 movies.shape
```

Out[29]: (1465, 11)

In [30]:
```
1 directors.shape
```

Out[30]: (2349, 3)

In [31]:
```
1 movies["director_id"].nunique()
```

Out[31]: 199

In [32]:
```
1 directors["id"].nunique()
```

Out[32]: 2349

In [ ]:
```
1 #isin function
```

In [33]:
```
1 movies["director_id"].isin(directors["id"])
```

Out[33]:
```
0       True
1       True
2       True
3       True
4       True
        ...
1460    True
1461    True
1462    True
1463    True
1464    True
Name: director_id, Length: 1465, dtype: bool
```

In [34]:
```
1 pd.Series([2000,3000,4000]).isin(pd.Series([2000,3000]))
```

Out[34]:
```
0     True
1     True
2    False
dtype: bool
```

In [35]:
```
1 import numpy as np
2
3 np.all(pd.Series([2000,3000,4000]).isin(pd.Series([2000,3000])))
```

Out[35]: False

In [36]:
```
np.all(movies["director_id"].isin(directors["id"]))
```

Out[36]: True

In [52]:
```
data=pd.merge(movies,directors,left_on="director_id",right_on="id",how="i

data
```

Out[52]:

	id_x	budget	popularity	revenue	title	vote_average	vote_count	director_
0	43597	237000000	150	2787965087	Avatar	7.2	11800	47
1	43622	200000000	100	1845034188	Titanic	7.5	7562	47
2	43876	100000000	101	520000000	Terminator 2: Judgment Day	7.7	4185	47
3	43879	115000000	38	378882411	True Lies	6.8	1116	47
4	44184	70000000	24	90000098	The Abyss	7.1	808	47
...	...	...	...	...	...	...	...	
1460	46859	0	14	25288872	Enough Said	6.6	348	62
1461	47023	6500000	11	13368437	Friends with Money	5.1	128	62
1462	47524	3000000	5	0	Please Give	6.0	57	62
1463	47962	0	0	0	Walking and Talking	6.6	7	62
1464	48229	250000	1	4186931	Lovely & Amazing	6.3	23	62

1465 rows × 14 columns

In [53]:
```
1 movies
```

Out[53]:

	id	budget	popularity	revenue	title	vote_average	vote_count	director_
0	43597	237000000	150	2787965087	Avatar	7.2	11800	47
1	43598	300000000	139	961000000	Pirates of the Caribbean: At World's End	6.9	4500	47
2	43599	245000000	107	880674609	Spectre	6.3	4466	47
3	43600	250000000	112	1084939099	The Dark Knight Rises	7.6	9106	47
4	43602	258000000	115	890871626	Spider-Man 3	5.9	3576	47
...	...	...	...	...	...	...	...	
1460	48363	0	3	321952	The Last Waltz	7.9	64	48
1461	48370	27000	19	3151130	Clerks	7.4	755	53
1462	48375	0	7	0	Rampage	6.0	131	51
1463	48376	0	3	0	Slacker	6.4	77	55
1464	48395	220000	14	2040920	El Mariachi	6.6	238	50

1465 rows × 11 columns

In [54]:
```
1 directors
```

Out[54]:

	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

2349 rows × 3 columns

In [55]:
```
1 data.drop(columns=["director_id","id_y"],inplace=True)
```

In [56]:
```
1 data
```

Out[56]:

	id_x	budget	popularity	revenue	title	vote_average	vote_count	year	m
0	43597	237000000	150	2787965087	Avatar	7.2	11800	2009	
1	43622	200000000	100	1845034188	Titanic	7.5	7562	1997	
2	43876	100000000	101	520000000	Terminator 2: Judgment Day	7.7	4185	1991	
3	43879	115000000	38	378882411	True Lies	6.8	1116	1994	
4	44184	70000000	24	90000098	The Abyss	7.1	808	1989	
...	...	...	...	...	...	...	...	...	
1460	46859	0	14	25288872	Enough Said	6.6	348	2013	
1461	47023	6500000	11	13368437	Friends with Money	5.1	128	2006	
1462	47524	3000000	5	0	Please Give	6.0	57	2010	
1463	47962	0	0	0	Walking and Talking	6.6	7	1996	
1464	48229	250000	1	4186931	Lovely & Amazing	6.3	23	2001	

1465 rows × 12 columns

In [57]:
```
1 data
```

Out[57]:

	id_x	budget	popularity	revenue	title	vote_average	vote_count	year	m
0	43597	237000000	150	2787965087	Avatar	7.2	11800	2009	
1	43622	200000000	100	1845034188	Titanic	7.5	7562	1997	
2	43876	100000000	101	520000000	Terminator 2: Judgment Day	7.7	4185	1991	
3	43879	115000000	38	378882411	True Lies	6.8	1116	1994	
4	44184	70000000	24	90000098	The Abyss	7.1	808	1989	
...	...	...	...	...	...	...	...	...	...
1460	46859	0	14	25288872	Enough Said	6.6	348	2013	
1461	47023	6500000	11	13368437	Friends with Money	5.1	128	2006	
1462	47524	3000000	5	0	Please Give	6.0	57	2010	
1463	47962	0	0	0	Walking and Talking	6.6	7	1996	
1464	48229	250000	1	4186931	Lovely & Amazing	6.3	23	2001	

1465 rows × 12 columns

In [58]:
```
1 data.shape
```

Out[58]: (1465, 12)

In [59]:
```
1 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
```

In [60]:
```
1 data.describe()
```

Out[60]:

	id_x	budget	popularity	revenue	vote_average	vote_count	
count	1465.000000	1.465000e+03	1465.000000	1.465000e+03	1465.000000	1465.000000	14
mean	45225.191126	4.802295e+07	30.855973	1.432539e+08	6.368191	1146.396587	20
std	1189.096396	4.935541e+07	34.845214	2.064918e+08	0.818033	1578.077438	
min	43597.000000	0.000000e+00	0.000000	0.000000e+00	3.000000	1.000000	19
25%	44236.000000	1.400000e+07	11.000000	1.738013e+07	5.900000	216.000000	19
50%	45022.000000	3.300000e+07	23.000000	7.578164e+07	6.400000	571.000000	20
75%	45990.000000	6.600000e+07	41.000000	1.792469e+08	6.900000	1387.000000	20
max	48395.000000	3.800000e+08	724.000000	2.787965e+09	8.300000	13752.000000	20

In [61]:
```
1 data.describe(include="all")
```

Out[61]:

	id_x	budget	popularity	revenue	title	vote_average	vote_c
count	1465.000000	1.465000e+03	1465.000000	1.465000e+03	1465	1465.000000	1465.00
unique	NaN	NaN	NaN	NaN	1465	NaN	
top	NaN	NaN	NaN	NaN	Avatar	NaN	
freq	NaN	NaN	NaN	NaN	1	NaN	
mean	45225.191126	4.802295e+07	30.855973	1.432539e+08	NaN	6.368191	1146.39
std	1189.096396	4.935541e+07	34.845214	2.064918e+08	NaN	0.818033	1578.07
min	43597.000000	0.000000e+00	0.000000	0.000000e+00	NaN	3.000000	1.00
25%	44236.000000	1.400000e+07	11.000000	1.738013e+07	NaN	5.900000	216.00
50%	45022.000000	3.300000e+07	23.000000	7.578164e+07	NaN	6.400000	571.00
75%	45990.000000	6.600000e+07	41.000000	1.792469e+08	NaN	6.900000	1387.00
max	48395.000000	3.800000e+08	724.000000	2.787965e+09	NaN	8.300000	13752.00

In [62]:
```
data["revenue"]= (data["revenue"]/1000000)
data
```

Out[62]:

	id_x	budget	popularity	revenue	title	vote_average	vote_count	year	n
0	43597	237000000	150	2787.965087	Avatar	7.2	11800	2009	
1	43622	200000000	100	1845.034188	Titanic	7.5	7562	1997	
2	43876	100000000	101	520.000000	Terminator 2: Judgment Day	7.7	4185	1991	
3	43879	115000000	38	378.882411	True Lies	6.8	1116	1994	
4	44184	70000000	24	90.000098	The Abyss	7.1	808	1989	
...	...	...	...	...	...	...	...	...	...
1460	46859	0	14	25.288872	Enough Said	6.6	348	2013	
1461	47023	6500000	11	13.368437	Friends with Money	5.1	128	2006	
1462	47524	3000000	5	0.000000	Please Give	6.0	57	2010	
1463	47962	0	0	0.000000	Walking and Talking	6.6	7	1996	
1464	48229	250000	1	4.186931	Lovely & Amazing	6.3	23	2001	

1465 rows × 12 columns

In [64]:
```
data["revenue"]= (data["revenue"]).round(2)
```

In [65]:
```
1 data
```

Out[65]:

	id_x	budget	popularity	revenue	title	vote_average	vote_count	year	mont
0	43597	237000000	150	2787.97	Avatar	7.2	11800	2009	De
1	43622	200000000	100	1845.03	Titanic	7.5	7562	1997	No
2	43876	100000000	101	520.00	Terminator 2: Judgment Day	7.7	4185	1991	Ju
3	43879	115000000	38	378.88	True Lies	6.8	1116	1994	Ju
4	44184	70000000	24	90.00	The Abyss	7.1	808	1989	Au
...	...	...	...	...	...	...	...	...	.
1460	46859	0	14	25.29	Enough Said	6.6	348	2013	Se
1461	47023	6500000	11	13.37	Friends with Money	5.1	128	2006	Se
1462	47524	3000000	5	0.00	Please Give	6.0	57	2010	Ja
1463	47962	0	0	0.00	Walking and Talking	6.6	7	1996	Ju
1464	48229	250000	1	4.19	Lovely & Amazing	6.3	23	2001	Au

1465 rows × 12 columns

In [66]:
```
1 data["budget"]= (data["budget"]/1000000).round(2)  #for learners, rename
```

In [67]:
```
1 data
```

Out[67]:

	id_x	budget	popularity	revenue	title	vote_average	vote_count	year	month
0	43597	237.00	150	2787.97	Avatar	7.2	11800	2009	Dec
1	43622	200.00	100	1845.03	Titanic	7.5	7562	1997	Nov
2	43876	100.00	101	520.00	Terminator 2: Judgment Day	7.7	4185	1991	Jul
3	43879	115.00	38	378.88	True Lies	6.8	1116	1994	Jul
4	44184	70.00	24	90.00	The Abyss	7.1	808	1989	Aug
...	...	...	...	...	...	...	...	...	...
1460	46859	0.00	14	25.29	Enough Said	6.6	348	2013	Sep
1461	47023	6.50	11	13.37	Friends with Money	5.1	128	2006	Sep
1462	47524	3.00	5	0.00	Please Give	6.0	57	2010	Jan
1463	47962	0.00	0	0.00	Walking and Talking	6.6	7	1996	Jul
1464	48229	0.25	1	4.19	Lovely & Amazing	6.3	23	2001	Aug

1465 rows × 12 columns

In [68]:
```
1 data["vote_average"]>7 #this is a mask
```

Out[68]:
```
0        True
1        True
2        True
3       False
4        True
        ...
1460    False
1461    False
1462    False
1463    False
1464    False
Name: vote_average, Length: 1465, dtype: bool
```

In [69]:
```
1 data[data["vote_average"]>7]
```

Out[69]:

	id_x	budget	popularity	revenue	title	vote_average	vote_count	year	month
0	43597	237.00	150	2787.97	Avatar	7.2	11800	2009	Dec
1	43622	200.00	100	1845.03	Titanic	7.5	7562	1997	Nov
2	43876	100.00	101	520.00	Terminator 2: Judgment Day	7.7	4185	1991	Jul
4	44184	70.00	24	90.00	The Abyss	7.1	808	1989	Aug
5	46000	18.50	67	183.32	Aliens	7.7	3220	1986	Jul
...	...	...	...	...	...	...	...	...	...
1424	47488	4.00	11	35.56	Bowling for Columbine	7.3	453	2002	Oct
1426	48310	0.16	3	6.71	Roger & Me	7.4	90	1989	Sep
1433	46168	11.50	15	23.24	The Remains of the Day	7.5	202	1993	Nov
1438	47597	0.00	11	0.00	Maurice	7.1	61	1987	Sep
1441	47232	5.00	35	8.20	The Machinist	7.3	1247	2004	Feb

301 rows × 12 columns

In [70]:
```
1 data.loc[data["vote_average"]>7,data.loc[(data["vote_average"]>7) & (data
```

Out[70]:

	title	director_name
0	Avatar	James Cameron
1	Titanic	James Cameron
2	Terminator 2: Judgment Day	James Cameron
4	The Abyss	James Cameron
5	Aliens	James Cameron
...	...	...
1424	Bowling for Columbine	Michael Moore
1426	Roger & Me	Michael Moore
1433	The Remains of the Day	James Ivory
1438	Maurice	James Ivory
1441	The Machinist	Brad Anderson

301 rows × 2 columns

In [71]:
```
1 data.loc[(data["day"]=="Friday") | (data["day"]=="Saturday")]
```

Out[71]:

	id_x	budget	popularity	revenue	title	vote_average	vote_count	year	month
5	46000	18.50	67	183.32	Aliens	7.7	3220	1986	Jul
6	47036	6.40	74	78.37	The Terminator	7.3	4128	1984	Oct
7	43598	300.00	139	961.00	Pirates of the Caribbean: At World's End	6.9	4500	2007	May
16	44041	80.00	49	181.00	Road to Perdition	7.3	1077	2002	Jul
17	44154	72.00	32	96.89	Jarhead	6.6	765	2005	Nov
...	...	...	...	...	...	...	...	...	...
1455	46574	10.00	33	163.88	Saw III	6.1	1071	2006	Oct
1457	46827	0.00	5	0.19	Repo! The Genetic Opera	6.7	100	2008	Jul
1458	47152	4.00	42	152.93	Saw II	6.3	1251	2005	Oct
1462	47524	3.00	5	0.00	Please Give	6.0	57	2010	Jan
1464	48229	0.25	1	4.19	Lovely & Amazing	6.3	23	2001	Aug

701 rows × 12 columns

In [72]:
```
1 data.sort_values(by=["popularity"],ascending=False).head(5)
```

Out[72]:

	id_x	budget	popularity	revenue	title	vote_average	vote_count	year	month	
23	43692	165.0	724	675.12	Interstellar	8.1	10867	2014	Nov	W
361	43724	150.0	434	378.86	Mad Max: Fury Road	7.2	9427	2015	May	W
11	43796	140.0	271	655.01	Pirates of the Caribbean: The Curse of the Bla...	7.5	6985	2003	Jul	W
315	43797	125.0	206	752.10	The Hunger Games: Mockingjay - Part 1	6.6	5584	2014	Nov	
22	43662	185.0	187	1004.56	The Dark Knight	8.2	12002	2008	Jul	W

In [73]:
```
1 data.loc[data["director_name"]=="Christopher Nolan"]
```

Out[73]:

	id_x	budget	popularity	revenue	title	vote_average	vote_count	year	month	
21	43600	250.0	112	1084.94	The Dark Knight Rises	7.6	9106	2012	Jul	
22	43662	185.0	187	1004.56	The Dark Knight	8.2	12002	2008	Jul	We
23	43692	165.0	724	675.12	Interstellar	8.1	10867	2014	Nov	We
24	43693	160.0	167	825.53	Inception	8.1	13752	2010	Jul	We
25	43716	150.0	115	374.22	Batman Begins	7.5	7359	2005	Jun	
26	44630	46.0	41	113.71	Insomnia	6.8	1148	2002	May	
27	44793	40.0	74	109.68	The Prestige	8.0	4391	2006	Oct	-
28	47170	9.0	60	39.72	Memento	8.1	4028	2000	Oct	We

In [74]:
```
1 data
```

Out[74]:

	id_x	budget	popularity	revenue	title	vote_average	vote_count	year	month
0	43597	237.00	150	2787.97	Avatar	7.2	11800	2009	Dec
1	43622	200.00	100	1845.03	Titanic	7.5	7562	1997	Nov
2	43876	100.00	101	520.00	Terminator 2: Judgment Day	7.7	4185	1991	Jul
3	43879	115.00	38	378.88	True Lies	6.8	1116	1994	Jul
4	44184	70.00	24	90.00	The Abyss	7.1	808	1989	Aug
...	...	...	...	...	...	...	...	...	...
1460	46859	0.00	14	25.29	Enough Said	6.6	348	2013	Sep
1461	47023	6.50	11	13.37	Friends with Money	5.1	128	2006	Sep
1462	47524	3.00	5	0.00	Please Give	6.0	57	2010	Jan
1463	47962	0.00	0	0.00	Walking and Talking	6.6	7	1996	Jul
1464	48229	0.25	1	4.19	Lovely & Amazing	6.3	23	2001	Aug

1465 rows × 12 columns

In [75]:
```
1 def myfunc(data):
2     if data =="Male":
3         return 1
4     elif data=="Female":
5         return 0
6     else:
7         return np.nan
```

In [76]:
```
data["gender"]=data["gender"].apply(myfunc)
data
```

Out[76]:

	id_x	budget	popularity	revenue	title	vote_average	vote_count	year	month
0	43597	237.00	150	2787.97	Avatar	7.2	11800	2009	Dec
1	43622	200.00	100	1845.03	Titanic	7.5	7562	1997	Nov
2	43876	100.00	101	520.00	Terminator 2: Judgment Day	7.7	4185	1991	Jul
3	43879	115.00	38	378.88	True Lies	6.8	1116	1994	Jul
4	44184	70.00	24	90.00	The Abyss	7.1	808	1989	Aug
...	...	...	...	...	...	...	...	...	...
1460	46859	0.00	14	25.29	Enough Said	6.6	348	2013	Sep
1461	47023	6.50	11	13.37	Friends with Money	5.1	128	2006	Sep
1462	47524	3.00	5	0.00	Please Give	6.0	57	2010	Jan
1463	47962	0.00	0	0.00	Walking and Talking	6.6	7	1996	Jul
1464	48229	0.25	1	4.19	Lovely & Amazing	6.3	23	2001	Aug

1465 rows × 12 columns

In [79]:
```
#simple method to create new column :
data["profit_simple_method"]=data['revenue']-data["budget"]
```

In [78]:
```
1 data
```

Out[78]:

	id_x	budget	popularity	revenue	title	vote_average	vote_count	year	month
0	43597	237.00	150	2787.97	Avatar	7.2	11800	2009	Dec
1	43622	200.00	100	1845.03	Titanic	7.5	7562	1997	Nov
2	43876	100.00	101	520.00	Terminator 2: Judgment Day	7.7	4185	1991	Jul
3	43879	115.00	38	378.88	True Lies	6.8	1116	1994	Jul
4	44184	70.00	24	90.00	The Abyss	7.1	808	1989	Aug
...	...	...	...	...	...	...	...	...	...
1460	46859	0.00	14	25.29	Enough Said	6.6	348	2013	Sep
1461	47023	6.50	11	13.37	Friends with Money	5.1	128	2006	Sep
1462	47524	3.00	5	0.00	Please Give	6.0	57	2010	Jan
1463	47962	0.00	0	0.00	Walking and Talking	6.6	7	1996	Jul
1464	48229	0.25	1	4.19	Lovely & Amazing	6.3	23	2001	Aug

1465 rows × 13 columns

In [80]:
```
1 #Not so simple :
2 def profit_func(x):
3     return x["revenue"]-x["budget"]
```

In [81]:
```
1 data["profit_apply_method"]=data[["revenue","budget"]].apply(profit_func,
```

In [82]:
```
1 data
```

Out[82]:

	id_x	budget	popularity	revenue	title	vote_average	vote_count	year	month
0	43597	237.00	150	2787.97	Avatar	7.2	11800	2009	Dec
1	43622	200.00	100	1845.03	Titanic	7.5	7562	1997	Nov
2	43876	100.00	101	520.00	Terminator 2: Judgment Day	7.7	4185	1991	Jul
3	43879	115.00	38	378.88	True Lies	6.8	1116	1994	Jul
4	44184	70.00	24	90.00	The Abyss	7.1	808	1989	Aug
...	...	...	...	...	...	...	...	...	...
1460	46859	0.00	14	25.29	Enough Said	6.6	348	2013	Sep
1461	47023	6.50	11	13.37	Friends with Money	5.1	128	2006	Sep
1462	47524	3.00	5	0.00	Please Give	6.0	57	2010	Jan
1463	47962	0.00	0	0.00	Walking and Talking	6.6	7	1996	Jul
1464	48229	0.25	1	4.19	Lovely & Amazing	6.3	23	2001	Aug

1465 rows × 14 columns

In [83]:
```
1 data[["revenue","budget"]].apply(np.sum,axis=1)
```

Out[83]:
```
0       3024.97
1       2045.03
2        620.00
3        493.88
4        160.00
         ...
1460      25.29
1461      19.87
1462       3.00
1463       0.00
1464       4.44
Length: 1465, dtype: float64
```

In [84]:
```
1 data[["revenue","budget"]].apply(np.sum,axis=0) #change axis
```

Out[84]:
```
revenue    209867.04
budget      70353.62
dtype: float64
```

In [85]:
```
1 data[["revenue","budget"]]
```

Out[85]:

	revenue	budget
0	2787.97	237.00
1	1845.03	200.00
2	520.00	100.00
3	378.88	115.00
4	90.00	70.00
...	...	...
1460	25.29	0.00
1461	13.37	6.50
1462	0.00	3.00
1463	0.00	0.00
1464	4.19	0.25

1465 rows × 2 columns

In [86]:
```
1 #--The end--- :-)
```

In [ ]:
```
1
```

In [ ]:
```
1
```