Colab Link: https://colab.research.google.com/drive/1OhbcF5GTsRA4obL06YOc7lUJAVkwrB__?usp=sharing

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

```
a = pd.DataFrame({"A":[10, 30], "B": [20, 40]})
b = pd.DataFrame({"A":[10, 30], "C": [20, 40]})
a
```

	A	B
0	10	20
1	30	40

```
b
```

	A	C
0	10	20
1	30	40

```
pd.concat([a, b], axis=1)
```

To undo cell deletion use ⌘/Ctrl+M Z or the 'Undo' option in the 'Edit' menu

0	10	20	10	20
1	30	40	30	40

```
pd.concat([a, b], axis=0)
```

	A	B	C
0	10	20.0	NaN
1	30	40.0	NaN
0	10	NaN	20.0
1	30	NaN	40.0

```
pd.concat([a, b])
```

	A	B	C
0	10	20.0	NaN
1	30	40.0	NaN
0	10	NaN	20.0
1	30	NaN	40.0

```
pd.concat([a, b]).loc[0]
```

	A	B	C
0	10	20.0	NaN
0	10	NaN	20.0

```
pd.concat([a, b], ignore_index=True)
```

	A	B	C
0	10	20.0	NaN
1	30	40.0	NaN
2	10	NaN	20.0
3	30	NaN	40.0

```
pd.concat([a, b], axis=1, keys=["x", "y"])
```

	x		y	
0	10	20	10	20
1	30	40	30	40

To undo cell deletion use ⌘/Ctrl+M Z or the 'Undo' option in the 'Edit' menu

```
pd.concat([a, b], axis=1, keys=["x", "y"]).columns
```

```
MultiIndex([('x', 'A'),
            ('x', 'B'),
            ('y', 'A'),
            ('y', 'C')],
           )
```

```
pd.concat([a, b], axis=0, keys=["x", "y"])
```

		A	B	C
x	**0**	10	20.0	NaN
	1	30	40.0	NaN
y	**0**	10	NaN	20.0

```
pd.concat([a, b], ignore_index=True)
```

	A	B	C
0	10	20.0	NaN
1	30	40.0	NaN
2	10	NaN	20.0
3	30	NaN	40.0

```
a
```

	A	B
0	10	20
1	30	40

```
b
```

	A	C
0	10	20
1	30	40

To undo cell deletion use ⌘/Ctrl+M Z or the 'Undo' option in the 'Edit' menu

	A	B	C
0	10	20.0	NaN
1	30	40.0	NaN
0	10	NaN	20.0
1	30	NaN	40.0

```
pd.concat([a, b], join="inner")
```

	A
0	10
1	30

```
pd.concat([a, b], axis=1, join="inner")
```

	A	B	A	C
0	10	20	10	20
1	30	40	30	40

```
pd.concat([a, b], axis=1, join="outer")
```

	A	B	A	C
0	10	20	10	20
1	30	40	30	40

```
a = pd.DataFrame({"A":[10, 30], "B": [20, 40]})
b = pd.DataFrame({"A":[10, 30], "C": [20, 40]})
```

```
a.index = [3, 4]
a
```

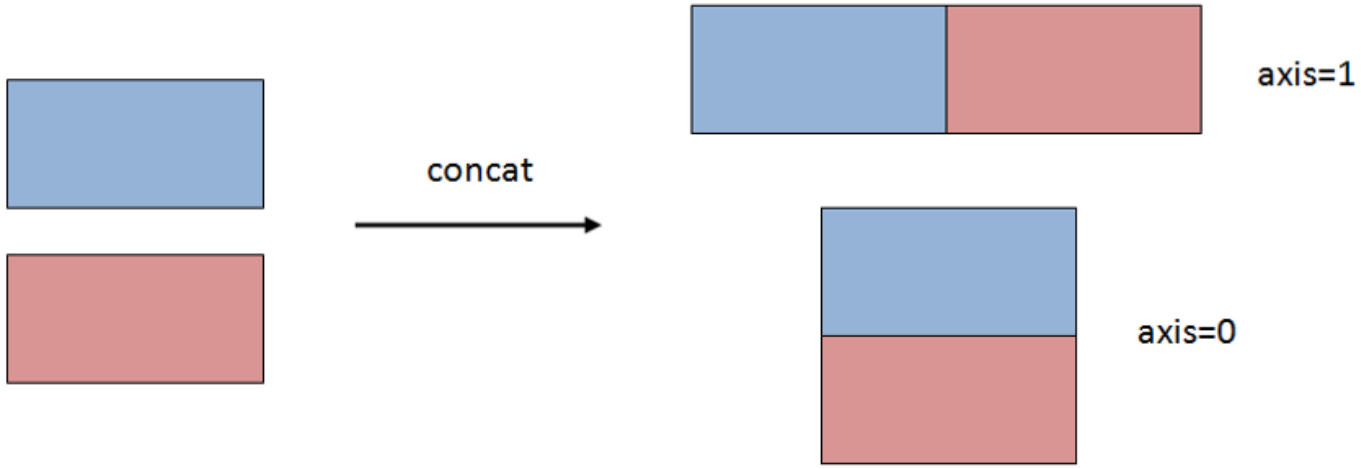
	A	B
3	10	20
4	30	40

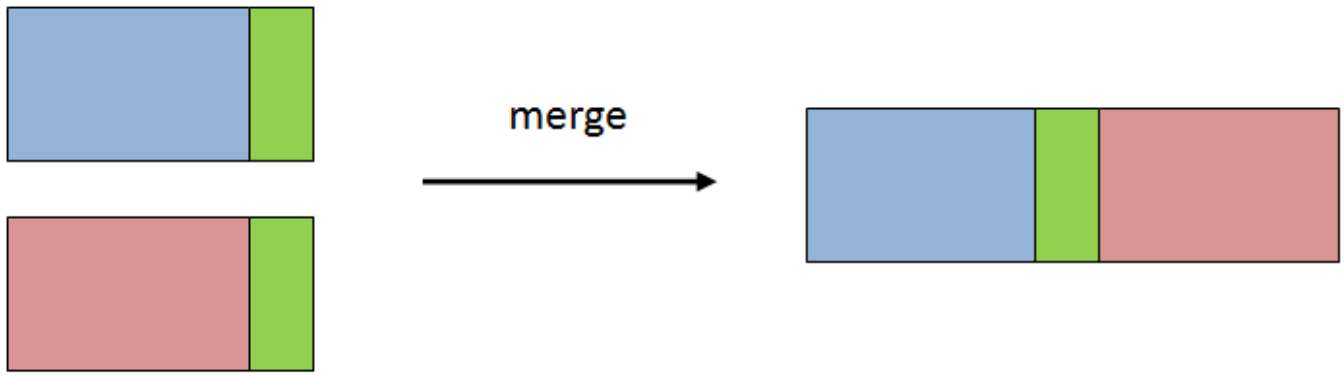
To undo cell deletion use ⌘/Ctrl+M Z or the 'Undo' option in the 'Edit' menu

	A	C
0	10	20
1	30	40

```
pd.concat([a, b], axis=1)
```

	A	B	A	C






```
users = pd.DataFrame({'userid':[1, 2, 3], 'name':['A', 'B', 'C']})
users
```

To undo cell deletion use ⌘/Ctrl+M Z or the 'Undo' option in the 'Edit' menu

1	2	B
2	3	C

```
msgs = pd.DataFrame({'userid':[1, 1, 2], 'msg':['hello', 'bye', 'hi']})
msgs
```

	userid	msg
0	1	hello
1	1	bye
2	2	hi

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

	userid	name	msg
0	1	A	hello
1	1	A	bye
2	2	B	hi

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

	userid	name	msg
0	1	A	hello
1	1	A	bye
2	2	B	hi

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

	userid	name	msg
0	1	A	hello
1	1	A	bye
2	2	B	hi
3	3	C	NaN

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

users
```

To undo cell deletion use ⌘/Ctrl+M Z or the 'Undo' option in the 'Edit' menu

0	1	A
1	2	B
2	3	C

```
msgs
```

	userid	msg
0	1	hello
1	1	bye
2	2	hi

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

	id	name	userid	msg
0	1	A	1	hello
1	1	A	1	bye
2	2	B	2	hi

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

	id	name	userid	msg
0	1	A	1	hello
1	1	A	1	bye
2	2	B	2	hi

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

	id	name	userid	msg
0	1	A	1.0	hello
1	1	A	1.0	bye
2	2	B	2.0	hi
3	3	C	NaN	NaN

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

To undo cell deletion use ⌘/Ctrl+M Z or the 'Undo' option in the 'Edit' menu

1	1	A	1	bye
2	2	B	2	hi

```
!gdown 1s2TkjSpzNc4SyxqRrQleZyDIHlc7bxnd
```

```
Downloading...
From: https://drive.google.com/uc?id=1s2TkjSpzNc4SyxqRrQleZyDIHlc7bxnd
To: /content/movies.csv
100% 112k/112k [00:00<00:00, 74.7MB/s]
```

```
!gdown 1Ws-_s1fHZ9nHfGLVUQurbHDvStePlEJm
```

```
Downloading...
```

```
From: https://drive.google.com/uc?id=1Ws-_s1fHZ9nHfGLVUQurbHDvStePlEJm
To: /content/directors.csv
100% 65.4k/65.4k [00:00<00:00, 63.6MB/s]
```

```
movies = pd.read_csv("movies.csv", index_col=0)
movies.head()
```

	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

```
directors = pd.read_csv("directors.csv", index_col=0)
directors.head()
```

	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

```
movies.shape
```

To undo cell deletion use ⌘/Ctrl+M Z or the 'Undo' option in the 'Edit' menu

```
directors.shape
```

```
(2349, 3)
```

```
movies["director_id"].nunique()
```

```
199
```

```
directors["id"].nunique()
```

```
2349
```

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

```
True
```

```
# unique_movies = movies["director_id"].unique()
# unique_directors = directors["id"].unique()
# np.all(unique_movies.isin(unique_directors))
```

```
data = movies.merge(directors, how="left", left_on="director_id", right_on="ic
data.head()
```

	id_x	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
4	43602	258000000	115	890871626	Spider-Man 3	5.9	3576

```
data["title"].nunique() == data.shape[0]
```

```
True
```

```
data.drop(["director_id", "id_x", "id_y"], axis=1, inplace=True)
```

```
data
```

To undo cell deletion use ⌘/Ctrl+M Z or the 'Undo' option in the 'Edit' menu

	budget	popularity	revenue	title	vote_average	vote_count	yea
0	237000000	150	2787965087	Avatar	7.2	11800	200
				Pirates of			

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1465 entries, 0 to 1464
Data columns (total 11 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   budget         1465 non-null   int64
 1   popularity     1465 non-null   int64
 2   revenue        1465 non-null   int64
 3   title          1465 non-null   object
 4   vote_average   1465 non-null   float64
 5   vote_count     1465 non-null   int64
 6   year           1465 non-null   int64
 7   month          1465 non-null   object
 8   day            1465 non-null   object
 9   director_name  1465 non-null   object
 10  gender         1341 non-null   object
dtypes: float64(1), int64(5), object(5)
memory usage: 137.3+ KB
```

```
data.describe()
```

	budget	popularity	revenue	vote_average	vote_count	ye
count	1.465000e+03	1465.000000	1.465000e+03	1465.000000	1465.000000	1465.0000
mean	4.802295e+07	30.855973	1.432539e+08	6.368191	1146.396587	2002.6150
std	4.935541e+07	34.845214	2.064918e+08	0.818033	1578.077438	8.6801
min	0.000000e+00	0.000000	0.000000e+00	3.000000	1.000000	1976.0000
					216.000000	1998.0000
					571.000000	2004.0000
75%	6.600000e+07	41.000000	1.792469e+08	6.900000	1387.000000	2009.0000
max	3.800000e+08	724.000000	2.787965e+09	8.300000	13752.000000	2016.0000

To undo cell deletion use ⌘/Ctrl+M Z or the 'Undo' option in the 'Edit' menu

```
data.describe(include=object)
```

```
data["revenue"] = (data["revenue"]/1000000).round(2)
```

```
data["budget"] = (data["budget"]/1000000).round(2)
```

top	Avatar	Dec	Friday	Steven Spielberg	Male

```
data
```

	budget	popularity	revenue	title	vote_average	vote_count	year	mo
0	237.00	150	2787.97	Avatar	7.2	11800	2009	
1	300.00	139	961.00	Pirates of the Caribbean: At World's End	6.9	4500	2007	
2	245.00	107	880.67	Spectre	6.3	4466	2015	
3	250.00	112	1084.94	The Dark Knight Rises	7.6	9106	2012	
4	258.00	115	890.87	Spider-Man 3	5.9	3576	2007	
...	...	...	...	...	...	...	...	
1460	0.00	3	0.32	The Last Waltz	7.9	64	1978	
1461	0.03	19	3.15	Clerks	7.4	755	1994	

```
# Quering a dataframe
```

```
data["vote_average"] > 7
```

To undo cell deletion use ⌘/Ctrl+M Z or the 'Undo' option in the 'Edit' menu

```
2       False
3        True
4       False
        ...
1460     True
1461     True
1462    False
1463    False
1464    False
Name: vote_average, Length: 1465, dtype: bool
```

```
data.loc[data["vote_average"] > 7]
```

	budget	popularity	revenue	title	vote_average	vote_count	year	mo
0	237.00	150	2787.97	Avatar	7.2	11800	2009	
3	250.00	112	1084.94	The Dark Knight Rises	7.6	9106	2012	
14	250.00	120	956.02	The Hobbit: The Battle of the Five Armies	7.1	4760	2014	
16	250.00	94	958.40	The Hobbit: The Desolation of Smaug	7.6	4524	2013	
19	200.00	100	1845.03	Titanic	7.5	7562	1997	
...	...	...	...	...	...	...	...	
1456	0.01	20	7.00	Eraserhead	7.5	485	1977	

```
data[data["vote_average"] > 7] # dont use it
```

	budget	popularity	revenue	title	vote_average	vote_count	year	mo
0	237.00	150	2787.97	Avatar	7.2	11800	2009	
3	250.00	112	1084.94	The Dark Knight Rises	7.6	9106	2012	
14	250.00	120	956.02	The Hobbit: The Battle of the Five Armies	7.1	4760	2014	
				The Hobbit:				
					6	4524	2013	
19	200.00	100	1845.03	Titanic	7.5	7562	1997	
...	...	...	...	...	...	...	...	
1456	0.01	20	7.00	Eraserhead	7.5	485	1977	
1457	0.00	5	0.00	The Mighty	7.1	51	1998	
1458	0.06	27	3.22	Pi	7.1	586	1998	

To undo cell deletion use ⌘/Ctrl+M Z or the 'Undo' option in the 'Edit' menu

```
data.loc[data["vote_average"] > 7, ["title", "vote_average"]]
```

	title	vote_average
0	Avatar	7.2
3	The Dark Knight Rises	7.6
14	The Hobbit: The Battle of the Five Armies	7.1
16	The Hobbit: The Desolation of Smaug	7.6
19	Titanic	7.5
...	...	...
1456	Eraserhead	7.5
1457	The Mighty	7.1
1458	Pi	7.1
1460	The Last Waltz	7.9

```
data.loc[(data["vote_average"] > 7) & (data["year"] >= 2015), ["title", "vote_
```

	title	vote_average
30	Furious 7	7.3
78	Mad Max: Fury Road	7.2
106	The Revenant	7.3
162	The Martian	7.6
312	The Man from U.N.C.L.E.	7.1
394	The Hateful Eight	7.6
625	The Intern	7.1
635	Bridge of Spies	7.2

To undo cell deletion use ⌘/Ctrl+M Z or the 'Undo' option in the 'Edit' menu

839	The Big Short	7.3
1344	Race	7.1

```
# Give me all the movues which are ahhabeeticalkly  after "Avengers"
```

```
"Anant" < "Mudit"
```

```
True
```

```
data.loc[data["title"] > "Avengers"]
```

	budget	popularity	revenue	title	vote_average
1	300.00	139	961.00	Pirates of the Caribbean: At World's End	6.9
2	245.00	107	880.67	Spectre	6.3
3	250.00	112	1084.94	The Dark Knight Rises	7.6
4	258.00	115	890.87	Spider-Man 3	5.9
5	250.00	155	873.26	Batman v Superman: Dawn of Justice	5.7
...	...	...	...	...	..
1460	0.00	3	0.32	The Last Waltz	7.9
1461	0.03	19	3.15	Clerks	7.4
1462	0.00	7	0.00	Rampage	6.0
1463	0.00	3	0.00	Slacker	6.4
1464	0.22	14	2.04	El Mariachi	6.6

1340 rows × 11 columns

```
# String Methods in Pandas, remaining ones after Regex

# Find the movies which has "Batman" in the title

"Batman" in "Batman and Robin"
```

```
True
```

```
data.loc[data["title"].str.contains("Batman")]
```

	budget	popularity	revenue	title	vote_average	v
5	250.0	155	873.26	Batman v Superman: Dawn of Justice	5.7	
				Begins	7.5	
128	125.0	50	238.21	Batman & Robin	4.2	
184	100.0	48	336.53	Batman Forever	5.2	
257	80.0	59	280.00	Batman Returns	6.6	
704	35.0	44	411.35	Batman	7.0	

To undo cell deletion use ⌘/Ctrl+M Z or the 'Undo' option in the 'Edit' menu

```
data.loc[data["title"].str.startswith("Batman")]
```

	budget	popularity	revenue	title	vote_average	v
5	250.0	155	873.26	Batman v Superman: Dawn of Justice	5.7	
74	150.0	115	374.22	Batman Begins	7.5	
128	125.0	50	238.21	Batman & Robin	4.2	

```
df = data
```

```
df.loc[df["director_name"] == "Christopher Nolan", "title"]
```

```
3       The Dark Knight Rises
45            The Dark Knight
58               Interstellar
59                  Inception
74              Batman Begins
565                  Insomnia
641              The Prestige
1341                  Memento
Name: title, dtype: object
```

```
# a. df.loc[(df['month']='Jan') | (df['month']='Nov')]

# b. df.loc[(df['month']=='Jan') || (df['month']=='Nov')]

# c. df.loc[df['month']=='Jan'| df['month']=='Nov']

# d. df.loc[(df['month']=='Jan') | (df['month']=='Nov')]
```

```
# a. df['year'].isin([2015, 2016, 2012])

# b. df['year'].in([2015, 2016, 2012])

# c. df['year']==([2015, 2016, 2012])
```

To undo cell deletion use ⌘/Ctrl+M Z or the 'Undo' option in the 'Edit' menu

```
"].shape[0]
```

```
8
```

```
df.loc[df["director_name"] == "Christopher Nolan", "title"].count()
```

```
8
```

```
df["director_name"].value_counts()
```

```
Steven Spielberg      26
Martin Scorsese       19
Clint Eastwood        19
Woody Allen           18
Ridley Scott          16
                      ..
```

```
Tim Hill               5
Jonathan Liebesman     5
Roman Polanski         5
Larry Charles          5
Nicole Holofcener      5
Name: director_name, Length: 199, dtype: int64
```

✓ 0s completed at 22:43

To undo cell deletion use ⌘/Ctrl+M Z or the 'Undo' option in the 'Edit' menu