

```
In [1]: import pandas as pd
import numpy as np
```

## Data Frame (Revisited)

```
In [2]: dict = {'Mango':[4, 5, 6, 3, 1], 'Apple':[5, 4, 3, 0, 2], 'Banana':[2, 3, 5, 2, 7]}
dict
```

```
Out[2]: {'Mango': [4, 5, 6, 3, 1], 'Apple': [5, 4, 3, 0, 2], 'Banana': [2, 3, 5, 2, 7]}
```

```
In [3]: array = np.array([[4, 5, 2], [5, 4, 3], [6, 3, 5], [3, 0, 2], [1, 2, 7]])
array
```

```
Out[3]: array([[4, 5, 2],
               [5, 4, 3],
               [6, 3, 5],
               [3, 0, 2],
               [1, 2, 7]])
```

```
In [4]: fruits = pd.DataFrame(dict)
fruits
```

```
Out[4]:
```

	Mango	Apple	Banana
0	4	5	2
1	5	4	3
2	6	3	5
3	3	0	2
4	1	2	7

```
In [5]: fruits = pd.DataFrame(array)
fruits
```

```
Out[5]:
```

	0	1	2
0	4	5	2
1	5	4	3
2	6	3	5
3	3	0	2
4	1	2	7

```
In [6]: fruits = pd.DataFrame(array, columns = ['Mango', 'Apple', 'Banana'])
fruits
```

```
Out[6]:
```

	Mango	Apple	Banana
0	4	5	2
1	5	4	3
2	6	3	5
3	3	0	2
4	1	2	7

```
In [7]: fruits = pd.DataFrame({'Mango':[4, 5, 6, 3, 1], 'Apple':[5, 4, 3, 0, 2], 'Banana':[2, 3
```

```
In [8]: fruits
```

```
Out[8]:
```

	Mango	Apple	Banana
0	4	5	2
1	5	4	3
2	6	3	5
3	3	0	2
4	1	2	7

```
In [9]: fruits = pd.DataFrame(np.array([[4, 5, 2], [5, 4, 3], [6, 3, 5], [3, 0, 2], [1, 2, 7]]))
```

```
In [10]: fruits
```

```
Out[10]:
```

	Mango	Apple	Banana
0	4	5	2
1	5	4	3
2	6	3	5
3	3	0	2
4	1	2	7

## Data Frame Manipulation (Continued)

### (1) Columns

#### 1) Selecting Multiple Columns

```
In [11]: movie = pd.read_csv('movie.csv')
movie_actor_director = movie[['actor_1_name', 'actor_2_name', 'actor_3_name', 'director_name']]
movie_actor_director
```

```
Out[11]:
```

	actor_1_name	actor_2_name	actor_3_name	director_name
0	CCH Pounder	Joel David Moore	Wes Studi	James Cameron
1	Johnny Depp	Orlando Bloom	Jack Davenport	Gore Verbinski
2	Christoph Waltz	Rory Kinnear	Stephanie Sigman	Sam Mendes
3	Tom Hardy	Christian Bale	Joseph Gordon-Levitt	Christopher Nolan
4	Doug Walker	Rob Walker	NaN	Doug Walker
...	...	...	...	...
4911	Eric Mabius	Daphne Zuniga	Crystal Lowe	Scott Smith
4912	Natalie Zea	Valorie Curry	Sam Underwood	NaN
4913	Eva Boehnke	Maxwell Moody	David Chandler	Benjamin Roberds
4914	Alan Ruck	Daniel Henney	Eliza Coupe	Daniel Hsia
4915	John August	Brian Herzlinger	Jon Gunn	Jon Gunn

4916 rows × 4 columns

```
In [12]: movie[['director_name']]
```

```
Out[12]:
```

	director_name
0	James Cameron
1	Gore Verbinski
2	Sam Mendes
3	Christopher Nolan
4	Doug Walker
...	...
4911	Scott Smith
4912	NaN
4913	Benjamin Roberds
4914	Daniel Hsia
4915	Jon Gunn

4916 rows × 1 columns

```
In [13]: movie['director_name']
```

```
Out[13]: 0      James Cameron
1      Gore Verbinski
2      Sam Mendes
3      Christopher Nolan
4      Doug Walker
...
4911    Scott Smith
4912      NaN
4913    Benjamin Roberds
4914    Daniel Hsia
4915      Jon Gunn
Name: director_name, Length: 4916, dtype: object
```

```
In [14]: movie['director_name'].to_frame()
```

```
Out[14]:
```

	director_name
0	James Cameron
1	Gore Verbinski
2	Sam Mendes
3	Christopher Nolan
4	Doug Walker
...	...
4911	Scott Smith
4912	NaN
4913	Benjamin Roberds
4914	Daniel Hsia
4915	Jon Gunn

4916 rows × 1 columns

```
In [15]: movie = pd.read_csv('movie.csv')
```

```
In [16]: movie.columns
```

```
Out[16]: Index(['color', 'director_name', 'num_critic_for_reviews', 'duration',
'director_facebook_likes', 'actor_3_facebook_likes', 'actor_2_name',
'actor_1_facebook_likes', 'gross', 'genres', 'actor_1_name',
'movie_title', 'num_voted_users', 'cast_total_facebook_likes',
'actor_3_name', 'facenumber_in_poster', 'plot_keywords',
'movie_imdb_link', 'num_user_for_reviews', 'language', 'country',
'content_rating', 'budget', 'title_year', 'actor_2_facebook_likes',
'imdb_score', 'aspect_ratio', 'movie_facebook_likes'],
dtype='object')
```

```
In [17]: movie_actor_director = movie.loc[:, ['actor_1_name', 'actor_2_name', 'actor_3_name', 'd
movie_actor_director
```

```
Out[17]:
```

	actor_1_name	actor_2_name	actor_3_name	director_name
0	CCH Pounder	Joel David Moore	Wes Studi	James Cameron
1	Johnny Depp	Orlando Bloom	Jack Davenport	Gore Verbinski
2	Christoph Waltz	Rory Kinnear	Stephanie Sigman	Sam Mendes
3	Tom Hardy	Christian Bale	Joseph Gordon-Levitt	Christopher Nolan
4	Doug Walker	Rob Walker	NaN	Doug Walker
...	...	...	...	...
4911	Eric Mabius	Daphne Zuniga	Crystal Lowe	Scott Smith
4912	Natalie Zea	Valorie Curry	Sam Underwood	NaN
4913	Eva Boehnke	Maxwell Moody	David Chandler	Benjamin Roberds
4914	Alan Ruck	Daniel Henney	Eliza Coupe	Daniel Hsia
4915	John August	Brian Herzlinger	Jon Gunn	Jon Gunn

4916 rows × 4 columns

```
In [18]: movie.loc[:, 'color':'duration']
```

```
Out[18]:
```

	color	director_name	num_critic_for_reviews	duration
0	Color	James Cameron	723.0	178.0
1	Color	Gore Verbinski	302.0	169.0
2	Color	Sam Mendes	602.0	148.0
3	Color	Christopher Nolan	813.0	164.0
4	NaN	Doug Walker	NaN	NaN
...	...	...	...	...
4911	Color	Scott Smith	1.0	87.0
4912	Color	NaN	43.0	43.0
4913	Color	Benjamin Roberds	13.0	76.0
4914	Color	Daniel Hsia	14.0	100.0
4915	Color	Jon Gunn	43.0	90.0

4916 rows × 4 columns

```
In [19]: movie.loc[:, ['director_name']]
```

```
Out[19]:
```

	director_name
0	James Cameron
1	Gore Verbinski

	director_name
2	Sam Mendes
3	Christopher Nolan
4	Doug Walker
...	...
4911	Scott Smith
4912	NaN
4913	Benjamin Roberds
4914	Daniel Hsia
4915	Jon Gunn

4916 rows × 1 columns

```
In [20]: movie.loc[:, 'director_name']
```

```
Out[20]: 0      James Cameron
1      Gore Verbinski
2      Sam Mendes
3      Christopher Nolan
4      Doug Walker
...
4911    Scott Smith
4912           NaN
4913    Benjamin Roberds
4914    Daniel Hsia
4915    Jon Gunn
Name: director_name, Length: 4916, dtype: object
```

```
In [21]: movie.loc[:, 'director_name'].to_frame()
```

	director_name
0	James Cameron
1	Gore Verbinski
2	Sam Mendes
3	Christopher Nolan
4	Doug Walker
...	...
4911	Scott Smith
4912	NaN
4913	Benjamin Roberds
4914	Daniel Hsia
4915	Jon Gunn

4916 rows × 1 columns

## 2) Deleting Multiple Columns

```
In [22]: movie = pd.read_csv('movie.csv')
```

```
In [23]: movie.columns
```

```
Out[23]: Index(['color', 'director_name', 'num_critic_for_reviews', 'duration',  
               'director_facebook_likes', 'actor_3_facebook_likes', 'actor_2_name',  
               'actor_1_facebook_likes', 'gross', 'genres', 'actor_1_name',  
               'movie_title', 'num_voted_users', 'cast_total_facebook_likes',  
               'actor_3_name', 'facenumber_in_poster', 'plot_keywords',  
               'movie_imdb_link', 'num_user_for_reviews', 'language', 'country',  
               'content_rating', 'budget', 'title_year', 'actor_2_facebook_likes',  
               'imdb_score', 'aspect_ratio', 'movie_facebook_likes'],  
              dtype='object')
```

```
In [24]: movie = movie.drop(['gross', 'genres'], axis = 'columns')
```

```
In [25]: movie.columns
```

```
Out[25]: Index(['color', 'director_name', 'num_critic_for_reviews', 'duration',  
               'director_facebook_likes', 'actor_3_facebook_likes', 'actor_2_name',  
               'actor_1_facebook_likes', 'actor_1_name', 'movie_title',  
               'num_voted_users', 'cast_total_facebook_likes', 'actor_3_name',  
               'facenumber_in_poster', 'plot_keywords', 'movie_imdb_link',  
               'num_user_for_reviews', 'language', 'country', 'content_rating',  
               'budget', 'title_year', 'actor_2_facebook_likes', 'imdb_score',  
               'aspect_ratio', 'movie_facebook_likes'],  
              dtype='object')
```

```
In [26]: movie = movie.drop('color', axis = 'columns')
```

```
In [27]: movie.columns
```

```
Out[27]: Index(['director_name', 'num_critic_for_reviews', 'duration',  
               'director_facebook_likes', 'actor_3_facebook_likes', 'actor_2_name',  
               'actor_1_facebook_likes', 'actor_1_name', 'movie_title',  
               'num_voted_users', 'cast_total_facebook_likes', 'actor_3_name',  
               'facenumber_in_poster', 'plot_keywords', 'movie_imdb_link',  
               'num_user_for_reviews', 'language', 'country', 'content_rating',  
               'budget', 'title_year', 'actor_2_facebook_likes', 'imdb_score',  
               'aspect_ratio', 'movie_facebook_likes'],  
              dtype='object')
```

```
In [28]: movie = movie.drop(['director_name'], axis = 'columns')
```

```
In [29]: movie.columns
```

```
Out[29]: Index(['num_critic_for_reviews', 'duration', 'director_facebook_likes',
               'actor_3_facebook_likes', 'actor_2_name', 'actor_1_facebook_likes',
               'actor_1_name', 'movie_title', 'num_voted_users',
               'cast_total_facebook_likes', 'actor_3_name', 'facenumber_in_poster',
               'plot_keywords', 'movie_imdb_link', 'num_user_for_reviews', 'language',
               'country', 'content_rating', 'budget', 'title_year',
               'actor_2_facebook_likes', 'imdb_score', 'aspect_ratio',
               'movie_facebook_likes'],
              dtype='object')
```

### 3) Ordering Column Names In Alphabetical Order

```
In [30]: movie = pd.read_csv('movie.csv')
```

```
In [31]: movie.columns
```

```
Out[31]: Index(['color', 'director_name', 'num_critic_for_reviews', 'duration',
               'director_facebook_likes', 'actor_3_facebook_likes', 'actor_2_name',
               'actor_1_facebook_likes', 'gross', 'genres', 'actor_1_name',
               'movie_title', 'num_voted_users', 'cast_total_facebook_likes',
               'actor_3_name', 'facenumber_in_poster', 'plot_keywords',
               'movie_imdb_link', 'num_user_for_reviews', 'language', 'country',
               'content_rating', 'budget', 'title_year', 'actor_2_facebook_likes',
               'imdb_score', 'aspect_ratio', 'movie_facebook_likes'],
              dtype='object')
```

```
In [32]: movie = movie.sort_index(axis = 'columns')
```

```
In [33]: movie
```

```
Out[33]:
```

	actor_1_facebook_likes	actor_1_name	actor_2_facebook_likes	actor_2_name	actor_3_facebook_likes
<b>0</b>	1000.0	CCH Pounder	936.0	Joel David Moore	855.0
<b>1</b>	40000.0	Johnny Depp	5000.0	Orlando Bloom	1000.0
<b>2</b>	11000.0	Christoph Waltz	393.0	Rory Kinnear	161.0
<b>3</b>	27000.0	Tom Hardy	23000.0	Christian Bale	23000.0
<b>4</b>	131.0	Doug Walker	12.0	Rob Walker	NaN
...	...	...	...	...	...
<b>4911</b>	637.0	Eric Mabius	470.0	Daphne Zuniga	318.0



	actor_1_facebook_likes	actor_1_name	actor_2_facebook_likes	actor_2_name	actor_3_facebook_likes
<b>4912</b>	841.0	Natalie Zea	593.0	Valorie Curry	319.0
<b>4913</b>	0.0	Eva Boehnke	0.0	Maxwell Moody	0.0
<b>4914</b>	946.0	Alan Ruck	719.0	Daniel Henney	489.0
<b>4915</b>	86.0	John August	23.0	Brian Herzlinger	16.0

4916 rows × 28 columns

## 4) Ordering Column Names Manually

```
In [34]: movie = pd.read_csv('movie.csv')
```

```
In [35]: movie.columns
```

```
Out[35]: Index(['color', 'director_name', 'num_critic_for_reviews', 'duration',
               'director_facebook_likes', 'actor_3_facebook_likes', 'actor_2_name',
               'actor_1_facebook_likes', 'gross', 'genres', 'actor_1_name',
               'movie_title', 'num_voted_users', 'cast_total_facebook_likes',
               'actor_3_name', 'facenumber_in_poster', 'plot_keywords',
               'movie_imdb_link', 'num_user_for_reviews', 'language', 'country',
               'content_rating', 'budget', 'title_year', 'actor_2_facebook_likes',
               'imdb_score', 'aspect_ratio', 'movie_facebook_likes'],
              dtype='object')
```

```
In [36]: disc_core = ['movie_title', 'title_year', 'content_rating', 'genres']
disc_people = ['director_name', 'actor_1_name', 'actor_2_name', 'actor_3_name']
disc_other = ['color', 'country', 'language', 'plot_keywords', 'movie_imdb_link']
cont_fb = ['director_facebook_likes', 'actor_1_facebook_likes', 'actor_2_facebook_likes',
           'actor_3_facebook_likes', 'cast_total_facebook_likes', 'movie_facebook_likes']
cont_finance = ['budget', 'gross']
cont_num_reviews = ['num_voted_users', 'num_user_for_reviews', 'num_critic_for_reviews']
cont_other = ['imdb_score', 'duration', 'aspect_ratio', 'facenumber_in_poster']
```

```
In [37]: new_col_order = disc_core + disc_people + disc_other + cont_fb + cont_finance + cont_num_reviews + cont_other
```

```
In [38]: movie2 = movie[new_col_order]
movie2.columns
```

```
Out[38]: Index(['movie_title', 'title_year', 'content_rating', 'genres',
               'director_name', 'actor_1_name', 'actor_2_name', 'actor_3_name',
               'color', 'country', 'language', 'plot_keywords', 'movie_imdb_link',
               'director_facebook_likes', 'actor_1_facebook_likes',
               'actor_2_facebook_likes', 'actor_3_facebook_likes',
```

```
'cast_total_facebook_likes', 'movie_facebook_likes', 'budget', 'gross',
'num_voted_users', 'num_user_for_reviews', 'num_critic_for_reviews',
'imdb_score', 'duration', 'aspect_ratio', 'facenumber_in_poster'],
dtype='object')
```

## 5) Selecting Columns With Methods

In [39]: `movie.filter(like = 'facebook')`

Out[39]:

	director_facebook_likes	actor_3_facebook_likes	actor_1_facebook_likes	cast_total_facebook_likes	ac
0	0.0	855.0	1000.0	4834	
1	563.0	1000.0	40000.0	48350	
2	0.0	161.0	11000.0	11700	
3	22000.0	23000.0	27000.0	106759	
4	131.0	NaN	131.0	143	
...	...	...	...	...	...
4911	2.0	318.0	637.0	2283	
4912	NaN	319.0	841.0	1753	
4913	0.0	0.0	0.0	0	
4914	0.0	489.0	946.0	2386	
4915	16.0	16.0	86.0	163	

4916 rows × 6 columns



## (2) Rows

### 1) Selecting Multiple Rows

In [40]: `movie_new = pd.read_csv('movie.csv', index_col = 'movie_title')`

In [41]: `movie_new`

Out[41]:

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes	movie_title
Avatar	Color	James Cameron	723.0	178.0	0.0		
Pirates of the Caribbean: The Curse of the Black Pearl	Color	Gore Verbinski	302.0	169.0	563.0		

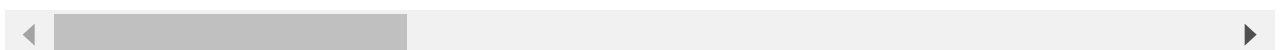
	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes
movie_title						
<b>At World's End</b>						
<b>Spectre</b>	Color	Sam Mendes	602.0	148.0	0.0	
<b>The Dark Knight Rises</b>	Color	Christopher Nolan	813.0	164.0	22000.0	
<b>Star Wars: Episode VII - The Force Awakens</b>	NaN	Doug Walker	NaN	NaN	131.0	
...	...	...	...	...	...	
<b>Signed Sealed Delivered</b>	Color	Scott Smith	1.0	87.0	2.0	
<b>The Following</b>	Color	NaN	43.0	43.0	NaN	
<b>A Plague So Pleasant</b>	Color	Benjamin Roberds	13.0	76.0	0.0	
<b>Shanghai Calling</b>	Color	Daniel Hsia	14.0	100.0	0.0	
<b>My Date with Drew</b>	Color	Jon Gunn	43.0	90.0	16.0	

4916 rows × 7 columns

```
In [42]: movie_favorites = movie_new.loc[['Avatar', 'Spectre', 'The Dark Knight Rises']]
movie_favorites
```

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes
movie_title						
<b>Avatar</b>	Color	James Cameron	723.0	178.0	0.0	
<b>Spectre</b>	Color	Sam Mendes	602.0	148.0	0.0	
<b>The Dark Knight Rises</b>	Color	Christopher Nolan	813.0	164.0	22000.0	

3 rows × 7 columns

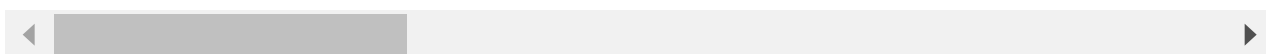


```
In [43]: movie_favorites = movie_new.loc['Avatar':'The Dark Knight Rises']
movie_favorites
```

```
Out[43]:
```

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes
<b>Avatar</b>	Color	James Cameron	723.0	178.0	0.0	
<b>Pirates of the Caribbean: At World's End</b>	Color	Gore Verbinski	302.0	169.0	563.0	
<b>Spectre</b>	Color	Sam Mendes	602.0	148.0	0.0	
<b>The Dark Knight Rises</b>	Color	Christopher Nolan	813.0	164.0	22000.0	

4 rows × 27 columns



```
In [44]: movie_favorites = movie_new.loc[['Avatar', 'Spectre', 'The Dark Knight Rises'], ['director_name', 'duration']]
movie_favorites
```

```
Out[44]:
```

	director_name	duration
<b>Avatar</b>	James Cameron	178.0
<b>Spectre</b>	Sam Mendes	148.0
<b>The Dark Knight Rises</b>	Christopher Nolan	164.0

```
In [45]: movie_favorites = movie_new.loc['Avatar':'The Dark Knight Rises', 'director_name':'duration']
movie_favorites
```

```
Out[45]:
```

	director_name	num_critic_for_reviews	duration
<b>Avatar</b>	James Cameron	723.0	178.0
<b>Pirates of the Caribbean: At World's End</b>	Gore Verbinski	302.0	169.0
<b>Spectre</b>	Sam Mendes	602.0	148.0
<b>The Dark Knight Rises</b>	Christopher Nolan	813.0	164.0

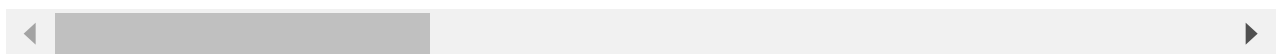
```
In [46]: movie_new_2 = movie_new.reset_index()
movie_new_2
```

Out[46]:

	movie_title	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3
--	-------------	-------	---------------	------------------------	----------	-------------------------	---------

0	Avatar	Color	James Cameron	723.0	178.0	0.0	
1	Pirates of the Caribbean: At World's End	Color	Gore Verbinski	302.0	169.0	563.0	
2	Spectre	Color	Sam Mendes	602.0	148.0	0.0	
3	The Dark Knight Rises	Color	Christopher Nolan	813.0	164.0	22000.0	
4	Star Wars: Episode VII - The Force Awakens	NaN	Doug Walker	NaN	NaN	131.0	
...	...	...	...	...	...	...	
4911	Signed Sealed Delivered	Color	Scott Smith	1.0	87.0	2.0	
4912	The Following	Color	NaN	43.0	43.0	NaN	
4913	A Plague So Pleasant	Color	Benjamin Roberds	13.0	76.0	0.0	
4914	Shanghai Calling	Color	Daniel Hsia	14.0	100.0	0.0	
4915	My Date with Drew	Color	Jon Gunn	43.0	90.0	16.0	

4916 rows × 28 columns



In [47]:

```
movie_favorites = movie_new_2.loc[0:3]
movie_favorites
```

Out[47]:

	movie_title	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_fa
--	-------------	-------	---------------	------------------------	----------	-------------------------	------------

0	Avatar	Color	James Cameron	723.0	178.0	0.0	
1	Pirates of the Caribbean: At World's End	Color	Gore Verbinski	302.0	169.0	563.0	
2	Spectre	Color	Sam Mendes	602.0	148.0	0.0	
3	The Dark Knight	Color	Christopher Nolan	813.0	164.0	22000.0	



```

actor_1_name          CCH Pounder
num_voted_users       886204
cast_total_facebook_likes  4834
actor_3_name          Wes Studi
facenumber_in_poster   0.0
plot_keywords         avatar|future|marine|native|paraplegic
movie_imdb_link        http://www.imdb.com/title/tt0499549/?ref_=fn_t...
num_user_for_reviews   3054.0
language              English
country               USA
content_rating         PG-13
budget                237000000.0
title_year            2009.0
actor_2_facebook_likes  936.0
imdb_score             7.9
aspect_ratio           1.78
movie_facebook_likes   33000
Name: Avatar, dtype: object

```

```
In [51]: type(movie_favorites_1)
```

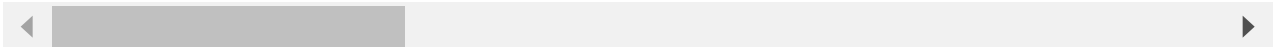
```
Out[51]: pandas.core.series.Series
```

```
In [52]: movie_favorites_2 = movie_new.loc[['Avatar']]
movie_favorites_2
```

```
Out[52]:
```

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes
movie_title						
Avatar	Color	James Cameron	723.0	178.0	0.0	

1 rows × 7 columns



```
In [53]: type(movie_favorites_2)
```

```
Out[53]: pandas.core.frame.DataFrame
```

```
In [54]: movie_favorites_1.to_frame()
```

```
Out[54]:
```

	Avatar
color	Color
director_name	James Cameron
num_critic_for_reviews	723.0
duration	178.0
director_facebook_likes	0.0
actor_3_facebook_likes	855.0
actor_2_name	Joel David Moore

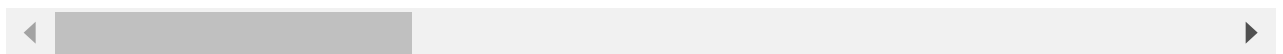
	Avatar
actor_1_facebook_likes	1000.0
gross	760505847.0
genres	Action Adventure Fantasy Sci-Fi
actor_1_name	CCH Pounder
num_voted_users	886204
cast_total_facebook_likes	4834
actor_3_name	Wes Studi
facenumber_in_poster	0.0
plot_keywords	avatar future marine native paraplegic
movie_imdb_link	http://www.imdb.com/title/tt0499549/?ref_=fn_t...
num_user_for_reviews	3054.0
language	English
country	USA
content_rating	PG-13
budget	237000000.0
title_year	2009.0
actor_2_facebook_likes	936.0
imdb_score	7.9
aspect_ratio	1.78
movie_facebook_likes	33000

```
In [55]: movie_favorites_1.to_frame().transpose()
```

```
Out[55]:
```

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes
<b>Avatar</b>	Color	James Cameron	723.0	178.0	0.0	8

1 rows × 7 columns



```
In [56]: movie = pd.read_csv('movie.csv')
movie.columns
```

```
Out[56]: Index(['color', 'director_name', 'num_critic_for_reviews', 'duration',
'director_facebook_likes', 'actor_3_facebook_likes', 'actor_2_name',
'actor_1_facebook_likes', 'gross', 'genres', 'actor_1_name',
'movie_title', 'num_voted_users', 'cast_total_facebook_likes',
'actor_3_name', 'facenumber_in_poster', 'plot_keywords',
'movie_imdb_link', 'num_user_for_reviews', 'language', 'country',
```



```
'content_rating', 'budget', 'title_year', 'actor_2_facebook_likes',  
'imdb_score', 'aspect_ratio', 'movie_facebook_likes'],  
dtype='object')
```

```
In [57]: movie[['color', 'duration']][:3]
```

```
Out[57]:
```

	color	duration
0	Color	178.0
1	Color	169.0
2	Color	148.0

```
In [59]: movie.loc[:3, ['color', 'duration']]
```

```
Out[59]:
```

	color	duration
0	Color	178.0
1	Color	169.0
2	Color	148.0
3	Color	164.0

```
In [60]: movie.loc[:3, 'color':'duration']
```

```
Out[60]:
```

	color	director_name	num_critic_for_reviews	duration
0	Color	James Cameron	723.0	178.0
1	Color	Gore Verbinski	302.0	169.0
2	Color	Sam Mendes	602.0	148.0
3	Color	Christopher Nolan	813.0	164.0

```
In [61]: movie['color']
```

```
Out[61]:
```

0	Color
1	Color
2	Color
3	Color
4	NaN
	...
4911	Color
4912	Color
4913	Color
4914	Color
4915	Color

Name: color, Length: 4916, dtype: object

```
In [62]: movie.loc[0]
```

```
Out[62]: color                                Color
director_name                                James Cameron
num_critic_for_reviews                        723.0
duration                                      178.0
director_facebook_likes                       0.0
actor_3_facebook_likes                        855.0
actor_2_name                                  Joel David Moore
actor_1_facebook_likes                        1000.0
gross                                          760505847.0
genres                                         Action|Adventure|Fantasy|Sci-Fi
actor_1_name                                  CCH Pounder
movie_title                                    Avatar
num_voted_users                                886204
cast_total_facebook_likes                     4834
actor_3_name                                  Wes Studi
facenumber_in_poster                          0.0
plot_keywords                                avatar|future|marine|native|paraplegic
movie_imdb_link                               http://www.imdb.com/title/tt0499549/?ref_=fn_t...
num_user_for_reviews                          3054.0
language                                       English
country                                       USA
content_rating                                PG-13
budget                                         237000000.0
title_year                                    2009.0
actor_2_facebook_likes                        936.0
imdb_score                                    7.9
aspect_ratio                                  1.78
movie_facebook_likes                          33000
Name: 0, dtype: object
```

## 2) Deleting Multiple Rows

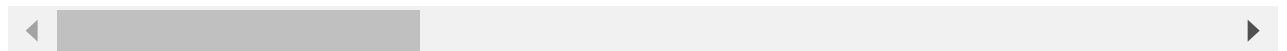
```
In [63]: movie = pd.read_csv('movie.csv')
```

```
In [64]: movie.head()
```

```
Out[64]:
```

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes
0	Color	James Cameron	723.0	178.0	0.0	855.0
1	Color	Gore Verbinski	302.0	169.0	563.0	1000.0
2	Color	Sam Mendes	602.0	148.0	0.0	161.0
3	Color	Christopher Nolan	813.0	164.0	22000.0	23000.0
4	NaN	Doug Walker	NaN	NaN	131.0	NaN

5 rows × 28 columns

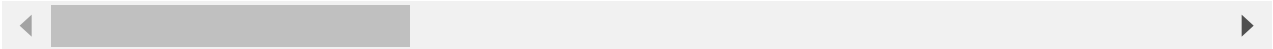


```
In [65]: movie = movie.drop([0])
movie
```

Out[65]:

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_li
1	Color	Gore Verbinski	302.0	169.0	563.0	100
2	Color	Sam Mendes	602.0	148.0	0.0	16
3	Color	Christopher Nolan	813.0	164.0	22000.0	2300
4	NaN	Doug Walker	NaN	NaN	131.0	N
5	Color	Andrew Stanton	462.0	132.0	475.0	53
...	...	...	...	...	...	...
4911	Color	Scott Smith	1.0	87.0	2.0	31
4912	Color	NaN	43.0	43.0	NaN	31
4913	Color	Benjamin Roberds	13.0	76.0	0.0	
4914	Color	Daniel Hsia	14.0	100.0	0.0	48
4915	Color	Jon Gunn	43.0	90.0	16.0	1

4915 rows × 28 columns



In [166...

```
movie_new = movie.set_index('movie_title')
movie_new
```

Out[166...

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebo
movie_title						
Avatar	Color	James Cameron	723.0	178.0	0.0	
Pirates of the Caribbean: At World's End	Color	Gore Verbinski	302.0	169.0	563.0	
Spectre	Color	Sam Mendes	602.0	148.0	0.0	
The Dark Knight Rises	Color	Christopher Nolan	813.0	164.0	22000.0	
Star Wars: Episode VII - The Force Awakens	NaN	Doug Walker	NaN	NaN	131.0	

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes
movie_title						
...	...	...	...	...	...	...
<b>Signed Sealed Delivered</b>	Color	Scott Smith	1.0	87.0	2.0	
<b>The Following</b>	Color	NaN	43.0	43.0	NaN	
<b>A Plague So Pleasant</b>	Color	Benjamin Roberds	13.0	76.0	0.0	
<b>Shanghai Calling</b>	Color	Daniel Hsia	14.0	100.0	0.0	
<b>My Date with Drew</b>	Color	Jon Gunn	43.0	90.0	16.0	

4916 rows × 7 columns

```
In [68]: movie_new = movie_new.drop(['John Carter', 'Spider-Man 3'])
movie_new
```

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes
movie_title						
<b>Pirates of the Caribbean: At World's End</b>	Color	Gore Verbinski	302.0	169.0	563.0	
<b>Spectre</b>	Color	Sam Mendes	602.0	148.0	0.0	
<b>The Dark Knight Rises</b>	Color	Christopher Nolan	813.0	164.0	22000.0	
<b>Star Wars: Episode VII - The Force Awakens</b>	NaN	Doug Walker	NaN	NaN	131.0	
<b>Tangled</b>	Color	Nathan Greno	324.0	100.0	15.0	
...	...	...	...	...	...	...
<b>Signed Sealed Delivered</b>	Color	Scott Smith	1.0	87.0	2.0	
<b>The</b>	Color	NaN	43.0	43.0	NaN	

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes
movie_title						
<b>Following</b>						
<b>A Plague So Pleasant</b>	Color	Benjamin Roberds	13.0	76.0	0.0	
<b>Shanghai Calling</b>	Color	Daniel Hsia	14.0	100.0	0.0	
<b>My Date with Drew</b>	Color	Jon Gunn	43.0	90.0	16.0	

4913 rows × 7 columns

### 3) Ordering Rows In Alphabetical Order of Index Names

```
In [72]: movie = pd.read_csv('movie.csv', index_col = 'movie_title')
```

```
In [73]: movie
```

```
Out[73]:
```

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes
movie_title						
<b>Avatar</b>	Color	James Cameron	723.0	178.0	0.0	
<b>Pirates of the Caribbean: At World's End</b>	Color	Gore Verbinski	302.0	169.0	563.0	
<b>Spectre</b>	Color	Sam Mendes	602.0	148.0	0.0	
<b>The Dark Knight Rises</b>	Color	Christopher Nolan	813.0	164.0	22000.0	
<b>Star Wars: Episode VII - The Force Awakens</b>	NaN	Doug Walker	NaN	NaN	131.0	
...	...	...	...	...	...	
<b>Signed Sealed Delivered</b>	Color	Scott Smith	1.0	87.0	2.0	
<b>The Following</b>	Color	NaN	43.0	43.0	NaN	

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes
movie_title						
<b>A Plague So Pleasant</b>	Color	Benjamin Roberds	13.0	76.0	0.0	
<b>Shanghai Calling</b>	Color	Daniel Hsia	14.0	100.0	0.0	
<b>My Date with Drew</b>	Color	Jon Gunn	43.0	90.0	16.0	

4916 rows × 27 columns

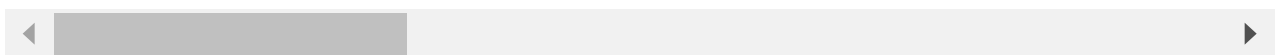
In [168...

```
movie = movie.sort_values(['director_name'])
# movie.sort_index(axis = 0)
movie
```

Out[168...

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes
<b>4498</b>	Color	A. Raven Cruz	3.0	97.0	0.0	9
<b>4221</b>	Color	Aaron Hann	29.0	87.0	0.0	9
<b>3430</b>	Color	Aaron Schneider	160.0	100.0	11.0	97
<b>2156</b>	Color	Aaron Seltzer	99.0	85.0	64.0	72
<b>2862</b>	Color	Abel Ferrara	48.0	99.0	220.0	59
...	...	...	...	...	...	
<b>4683</b>	Color	NaN	75.0	60.0	NaN	83
<b>4688</b>	Color	NaN	11.0	22.0	NaN	
<b>4704</b>	Color	NaN	23.0	43.0	NaN	57
<b>4752</b>	Color	NaN	11.0	58.0	NaN	25
<b>4912</b>	Color	NaN	43.0	43.0	NaN	31

4916 rows × 28 columns



## 4) Adding a New Row Into a Specific Place

In [78]:

```
movie = pd.read_csv('movie.csv')
```

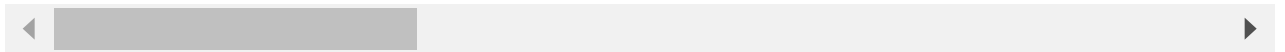
In [79]:

```
movie.head()
```

Out[79]:

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes
<b>0</b>	Color	James Cameron	723.0	178.0	0.0	855.0
<b>1</b>	Color	Gore Verbinski	302.0	169.0	563.0	1000.0
<b>2</b>	Color	Sam Mendes	602.0	148.0	0.0	161.0
<b>3</b>	Color	Christopher Nolan	813.0	164.0	22000.0	23000.0
<b>4</b>	NaN	Doug Walker	NaN	NaN	131.0	NaN

5 rows × 28 columns



In [80]:

```
line = pd.DataFrame({'color':'Color', 'director_name':'Blake Pappas'}, index = [3.5])
line
```

Out[80]:

	color	director_name
<b>3.5</b>	Color	Blake Pappas

In [81]:

```
movie = movie.append(line)
movie
```

Out[81]:

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes
<b>0.0</b>	Color	James Cameron	723.0	178.0	0.0	{
<b>1.0</b>	Color	Gore Verbinski	302.0	169.0	563.0	10
<b>2.0</b>	Color	Sam Mendes	602.0	148.0	0.0	'
<b>3.0</b>	Color	Christopher Nolan	813.0	164.0	22000.0	230
<b>4.0</b>	NaN	Doug Walker	NaN	NaN	131.0	
<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	
<b>4912.0</b>	Color	NaN	43.0	43.0	NaN	:
<b>4913.0</b>	Color	Benjamin Roberds	13.0	76.0	0.0	
<b>4914.0</b>	Color	Daniel Hsia	14.0	100.0	0.0	4
<b>4915.0</b>	Color	Jon Gunn	43.0	90.0	16.0	

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_
3.5	Color	Blake Pappas	NaN	NaN	NaN	

4917 rows × 28 columns

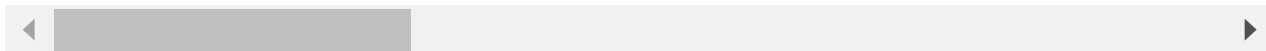
```
In [82]: movie = movie.sort_index()
```

```
In [83]: movie
```

```
Out[83]:
```

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_
0.0	Color	James Cameron	723.0	178.0	0.0	{
1.0	Color	Gore Verbinski	302.0	169.0	563.0	10
2.0	Color	Sam Mendes	602.0	148.0	0.0	'
3.0	Color	Christopher Nolan	813.0	164.0	22000.0	230
3.5	Color	Blake Pappas	NaN	NaN	NaN	
...	...	...	...	...	...	
4911.0	Color	Scott Smith	1.0	87.0	2.0	:
4912.0	Color	NaN	43.0	43.0	NaN	:
4913.0	Color	Benjamin Roberds	13.0	76.0	0.0	
4914.0	Color	Daniel Hsia	14.0	100.0	0.0	4
4915.0	Color	Jon Gunn	43.0	90.0	16.0	

4917 rows × 28 columns



```
In [84]: movie = pd.read_csv('movie.csv')
```

```
In [85]: line = pd.DataFrame({'color':'Color', 'director_name':'Blake Pappas'}, index = [3.5])
line
```

```
Out[85]:
```

	color	director_name
3.5	Color	Blake Pappas



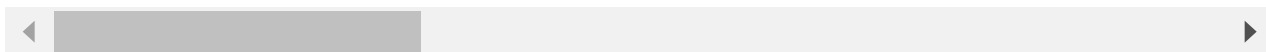
```
In [86]: movie = movie.append(line)
```

```
In [87]: movie.tail()
```

```
Out[87]:
```

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_
<b>4912.0</b>	Color	NaN	43.0	43.0	NaN	:
<b>4913.0</b>	Color	Benjamin Roberds	13.0	76.0	0.0	
<b>4914.0</b>	Color	Daniel Hsia	14.0	100.0	0.0	4
<b>4915.0</b>	Color	Jon Gunn	43.0	90.0	16.0	
<b>3.5</b>	Color	Blake Pappas	NaN	NaN	NaN	

5 rows × 28 columns

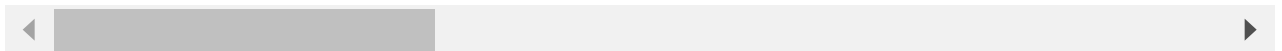


```
In [88]: movie.sort_index().reset_index()
```

Out[88]:

	index	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_face
<b>0</b>	0.0	Color	James Cameron	723.0	178.0	0.0	
<b>1</b>	1.0	Color	Gore Verbinski	302.0	169.0	563.0	
<b>2</b>	2.0	Color	Sam Mendes	602.0	148.0	0.0	
<b>3</b>	3.0	Color	Christopher Nolan	813.0	164.0	22000.0	
<b>4</b>	3.5	Color	Blake Pappas	NaN	NaN	NaN	
...	...	...	...	...	...	...	
<b>4912</b>	4911.0	Color	Scott Smith	1.0	87.0	2.0	
<b>4913</b>	4912.0	Color	NaN	43.0	43.0	NaN	
<b>4914</b>	4913.0	Color	Benjamin Roberds	13.0	76.0	0.0	
<b>4915</b>	4914.0	Color	Daniel Hsia	14.0	100.0	0.0	
<b>4916</b>	4915.0	Color	Jon Gunn	43.0	90.0	16.0	

4917 rows × 29 columns



In [89]: `movie.sort_index().reset_index(drop = True)`

Out[89]:

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes
<b>0</b>	Color	James Cameron	723.0	178.0	0.0	85
<b>1</b>	Color	Gore Verbinski	302.0	169.0	563.0	100
<b>2</b>	Color	Sam Mendes	602.0	148.0	0.0	16
<b>3</b>	Color	Christopher Nolan	813.0	164.0	22000.0	2300
<b>4</b>	Color	Blake Pappas	NaN	NaN	NaN	N
...	...	...	...	...	...	
<b>4912</b>	Color	Scott Smith	1.0	87.0	2.0	31
<b>4913</b>	Color	NaN	43.0	43.0	NaN	31
<b>4914</b>	Color	Benjamin Roberds	13.0	76.0	0.0	
<b>4915</b>	Color	Daniel Hsia	14.0	100.0	0.0	48

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_lil
<b>4916</b>	Color	Jon Gunn	43.0	90.0	16.0	1

4917 rows × 28 columns

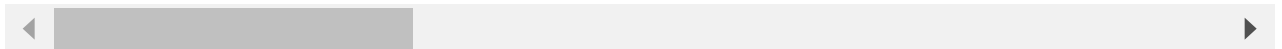
```
In [90]: movie = movie.sort_index().reset_index(drop = True)
```

```
In [91]: movie
```

```
Out[91]:
```

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_lil
<b>0</b>	Color	James Cameron	723.0	178.0	0.0	85
<b>1</b>	Color	Gore Verbinski	302.0	169.0	563.0	100
<b>2</b>	Color	Sam Mendes	602.0	148.0	0.0	16
<b>3</b>	Color	Christopher Nolan	813.0	164.0	22000.0	2300
<b>4</b>	Color	Blake Pappas	NaN	NaN	NaN	N
...	...	...	...	...	...	
<b>4912</b>	Color	Scott Smith	1.0	87.0	2.0	31
<b>4913</b>	Color	NaN	43.0	43.0	NaN	31
<b>4914</b>	Color	Benjamin Roberds	13.0	76.0	0.0	
<b>4915</b>	Color	Daniel Hsia	14.0	100.0	0.0	48
<b>4916</b>	Color	Jon Gunn	43.0	90.0	16.0	1

4917 rows × 28 columns



## 5) Selecting Rows With Methods

```
In [92]: movie = pd.read_csv('movie.csv', index_col = 'movie_title')
```

```
In [93]: movie.head()
```

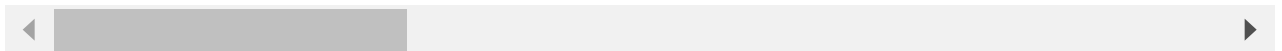
Out[93]:

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes
--	-------	---------------	------------------------	----------	-------------------------	------------------------

movie\_title

<b>Avatar</b>	Color	James Cameron	723.0	178.0	0.0	
<b>Pirates of the Caribbean: At World's End</b>	Color	Gore Verbinski	302.0	169.0	563.0	
<b>Spectre</b>	Color	Sam Mendes	602.0	148.0	0.0	
<b>The Dark Knight Rises</b>	Color	Christopher Nolan	813.0	164.0	22000.0	
<b>Star Wars: Episode VII - The Force Awakens</b>	NaN	Doug Walker	NaN	NaN	131.0	

5 rows × 7 columns



In [94]:

```
movie.filter(like = 'Star Wars', axis = 'index')
```

Out[94]:

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes
--	-------	---------------	------------------------	----------	-------------------------	------------------------

movie\_title

<b>Star Wars: Episode VII - The Force Awakens</b>	NaN	Doug Walker	NaN	NaN	131.0	
<b>Star Wars: Episode III - Revenge of the Sith</b>	Color	George Lucas	359.0	140.0	0.0	
<b>Star Wars: Episode II - Attack of the Clones</b>	Color	George Lucas	284.0	142.0	0.0	
<b>Star Wars: Episode I - The Phantom Menace</b>	Color	George Lucas	320.0	136.0	0.0	
<b>Star Wars: Episode VI - Return of the Jedi</b>	Color	Richard Marquand	197.0	134.0	37.0	

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes
movie_title						
<b>Star Wars: Episode V - The Empire Strikes Back</b>	Color	Irvin Kershner	223.0	127.0	883.0	
<b>Star Wars: Episode IV - A New Hope</b>	Color	George Lucas	282.0	125.0	0.0	
<b>Star Wars: The Clone Wars</b>	Color	NaN	47.0	23.0	NaN	

8 rows × 7 columns

## Summary: Key Methods

(1) loc

(2) drop

(3) append

(4) filter

(5) sort\_index

(5) reset\_index

(6) transpose

## Data Frame Operations

### (1) Working With Operators on a Series

```
In [95]: movie = pd.read_csv('movie.csv', index_col = 'movie_title')
movie
```

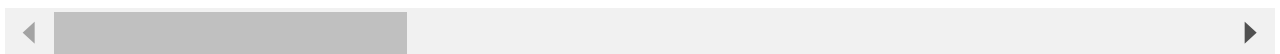
Out[95]:

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes
--	-------	---------------	------------------------	----------	-------------------------	------------------------

movie_title
-------------

Avatar	Color	James Cameron	723.0	178.0	0.0
Pirates of the Caribbean: At World's End	Color	Gore Verbinski	302.0	169.0	563.0
Spectre	Color	Sam Mendes	602.0	148.0	0.0
The Dark Knight Rises	Color	Christopher Nolan	813.0	164.0	22000.0
Star Wars: Episode VII - The Force Awakens	NaN	Doug Walker	NaN	NaN	131.0
...	...	...	...	...	...
Signed Sealed Delivered	Color	Scott Smith	1.0	87.0	2.0
The Following	Color	NaN	43.0	43.0	NaN
A Plague So Pleasant	Color	Benjamin Roberds	13.0	76.0	0.0
Shanghai Calling	Color	Daniel Hsia	14.0	100.0	0.0
My Date with Drew	Color	Jon Gunn	43.0	90.0	16.0

4916 rows × 27 columns



In [96]:

```
imdb_score = movie['imdb_score']
imdb_score
```

Out[96]:

movie_title	
Avatar	7.9
Pirates of the Caribbean: At World's End	7.1
Spectre	6.8
The Dark Knight Rises	8.5
Star Wars: Episode VII - The Force Awakens	7.1
...	...
Signed Sealed Delivered	7.7
The Following	7.5
A Plague So Pleasant	6.3
Shanghai Calling	6.3

```
My Date with Drew                                6.6
Name: imdb_score, Length: 4916, dtype: float64
```

```
In [97]:  imdb_score * 2.5
```

```
Out[97]: movie_title
Avatar                                19.75
Pirates of the Caribbean: At World's End  17.75
Spectre                               17.00
The Dark Knight Rises                  21.25
Star Wars: Episode VII - The Force Awakens  17.75
...
Signed Sealed Delivered                19.25
The Following                          18.75
A Plague So Pleasant                   15.75
Shanghai Calling                       15.75
My Date with Drew                      16.50
Name: imdb_score, Length: 4916, dtype: float64
```

```
In [98]:  imdb_score // 7
```

```
Out[98]: movie_title
Avatar                                1.0
Pirates of the Caribbean: At World's End  1.0
Spectre                               0.0
The Dark Knight Rises                  1.0
Star Wars: Episode VII - The Force Awakens  1.0
...
Signed Sealed Delivered                1.0
The Following                          1.0
A Plague So Pleasant                   0.0
Shanghai Calling                       0.0
My Date with Drew                      0.0
Name: imdb_score, Length: 4916, dtype: float64
```

```
In [99]:  imdb_score > 7
```

```
Out[99]: movie_title
Avatar                                True
Pirates of the Caribbean: At World's End  True
Spectre                               False
The Dark Knight Rises                  True
Star Wars: Episode VII - The Force Awakens  True
...
Signed Sealed Delivered                True
The Following                          True
A Plague So Pleasant                   False
Shanghai Calling                       False
My Date with Drew                      False
Name: imdb_score, Length: 4916, dtype: bool
```

```
In [100...  imdb_score
```

```
Out[100... movie_title
Avatar                                7.9
Pirates of the Caribbean: At World's End  7.1
Spectre                               6.8
The Dark Knight Rises                  8.5
Star Wars: Episode VII - The Force Awakens  7.1
...
```

Signed Sealed Delivered 7.7  
The Following 7.5  
A Plague So Pleasant 6.3  
Shanghai Calling 6.3  
My Date with Drew 6.6  
Name: imdb\_score, Length: 4916, dtype: float64

## (2) Working With Operators On a Data Frame

In [101...

```
college = pd.read_csv('college.csv')  
college
```

Out[101...

	INSTNM	CITY	STABBR	HBCU	MENONLY	WOMENONLY	RELAFFIL	SATVRMID	SATM
0	Alabama A & M University	Normal	AL	1.0	0.0	0.0	0	424.0	
1	University of Alabama at Birmingham	Birmingham	AL	0.0	0.0	0.0	0	570.0	
2	Amridge University	Montgomery	AL	0.0	0.0	0.0	1	NaN	
3	University of Alabama in Huntsville	Huntsville	AL	0.0	0.0	0.0	0	595.0	
4	Alabama State University	Montgomery	AL	1.0	0.0	0.0	0	425.0	
...	...	...	...	...	...	...	...	...	
7530	SAE Institute of Technology San Francisco	Emeryville	CA	NaN	NaN	NaN	1	NaN	
7531	Rasmussen College - Overland Park	Overland Park	KS	NaN	NaN	NaN	1	NaN	
7532	National Personal Training Institute of Cleveland	Highland Heights	OH	NaN	NaN	NaN	1	NaN	
7533	Bay Area Medical Academy - San Jose Satellite ...	San Jose	CA	NaN	NaN	NaN	1	NaN	



	INSTNM	CITY	STABBR	HBCU	MENONLY	WOMENONLY	RELAFFIL	SATVRMID	SATM
	Excel Learning <b>7534</b> Center-San Antonio South	San Antonio	TX	NaN	NaN	NaN	1	NaN	

7535 rows × 27 columns

In [102...

```
college.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7535 entries, 0 to 7534
Data columns (total 27 columns):
#   Column                      Non-Null Count  Dtype
---  -
0   INSTNM                      7535 non-null   object
1   CITY                        7535 non-null   object
2   STABBR                      7535 non-null   object
3   HBCU                        7164 non-null   float64
4   MENONLY                    7164 non-null   float64
5   WOMENONLY                  7164 non-null   float64
6   RELAFFIL                   7535 non-null   int64
7   SATVRMID                   1185 non-null   float64
8   SATMTMID                   1196 non-null   float64
9   DISTANCEONLY               7164 non-null   float64
10  UGDS                       6874 non-null   float64
11  UGDS_WHITE                 6874 non-null   float64
12  UGDS_BLACK                 6874 non-null   float64
13  UGDS_HISP                  6874 non-null   float64
14  UGDS_ASIAN                 6874 non-null   float64
15  UGDS_AIAN                  6874 non-null   float64
16  UGDS_NHPI                  6874 non-null   float64
17  UGDS_2MOR                  6874 non-null   float64
18  UGDS_NRA                   6874 non-null   float64
19  UGDS_UNKN                  6874 non-null   float64
20  PPTUG_EF                   6853 non-null   float64
21  CURROPER                   7535 non-null   int64
22  PCTPELL                    6849 non-null   float64
23  PCTFLOAN                   6849 non-null   float64
24  UG25ABV                    6718 non-null   float64
25  MD_EARN_WNE_P10            6413 non-null   object
26  GRAD_DEBT_MDN_SUPP         7503 non-null   object
dtypes: float64(20), int64(2), object(5)
memory usage: 1.6+ MB
```

In [103...

```
college.columns
```

Out[103...

```
Index(['INSTNM', 'CITY', 'STABBR', 'HBCU', 'MENONLY', 'WOMENONLY', 'RELAFFIL',
      'SATVRMID', 'SATMTMID', 'DISTANCEONLY', 'UGDS', 'UGDS_WHITE',
      'UGDS_BLACK', 'UGDS_HISP', 'UGDS_ASIAN', 'UGDS_AIAN', 'UGDS_NHPI',
      'UGDS_2MOR', 'UGDS_NRA', 'UGDS_UNKN', 'PPTUG_EF', 'CURROPER', 'PCTPELL',
      'PCTFLOAN', 'UG25ABV', 'MD_EARN_WNE_P10', 'GRAD_DEBT_MDN_SUPP'],
      dtype='object')
```

In [105...

```
college = pd.read_csv('college.csv', index_col = 'INSTNM')
college_ugds_ = college.filter(like = 'UGDS_')
```

college\_ugds\_

Out[105...

	UGDS_WHITE	UGDS_BLACK	UGDS_HISP	UGDS_ASIAN	UGDS_AIAN	UGDS_NHPI	UGDS_2M
INSTNM							
Alabama A & M University	0.0333	0.9353	0.0055	0.0019	0.0024	0.0019	0.0
University of Alabama at Birmingham	0.5922	0.2600	0.0283	0.0518	0.0022	0.0007	0.0
Amridge University	0.2990	0.4192	0.0069	0.0034	0.0000	0.0000	0.0
University of Alabama in Huntsville	0.6988	0.1255	0.0382	0.0376	0.0143	0.0002	0.0
Alabama State University	0.0158	0.9208	0.0121	0.0019	0.0010	0.0006	0.0
...	...	...	...	...	...	...	...
SAE Institute of Technology San Francisco	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Rasmussen College - Overland Park	NaN	NaN	NaN	NaN	NaN	NaN	NaN
National Personal Training Institute of Cleveland	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Bay Area Medical Academy - San Jose Satellite Location	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Excel Learning Center-San Antonio South	NaN	NaN	NaN	NaN	NaN	NaN	NaN

7535 rows × 9 columns



In [106...

```
college_ugds_round_per = college_ugds_ * 100
college_ugds_round_per
```

Out[106...

	UGDS_WHITE	UGDS_BLACK	UGDS_HISP	UGDS_ASIAN	UGDS_AIAN	UGDS_NHPI	UGDS_21
INSTNM							
Alabama A & M University	3.33	93.53	0.55	0.19	0.24	0.19	
University of Alabama at Birmingham	59.22	26.00	2.83	5.18	0.22	0.07	
Amridge University	29.90	41.92	0.69	0.34	0.00	0.00	
University of Alabama in Huntsville	69.88	12.55	3.82	3.76	1.43	0.02	
Alabama State University	1.58	92.08	1.21	0.19	0.10	0.06	
...	...	...	...	...	...	...	
SAE Institute of Technology San Francisco	NaN	NaN	NaN	NaN	NaN	NaN	
Rasmussen College - Overland Park	NaN	NaN	NaN	NaN	NaN	NaN	
National Personal Training Institute of Cleveland	NaN	NaN	NaN	NaN	NaN	NaN	
Bay Area Medical Academy - San Jose Satellite Location	NaN	NaN	NaN	NaN	NaN	NaN	
Excel Learning Center-San Antonio South	NaN	NaN	NaN	NaN	NaN	NaN	

7535 rows × 9 columns

In [107...

```
college_ugds_round_per.sort_values('UGDS_ASIAN', ascending = False)
```

Out[107...

	UGDS_WHITE	UGDS_BLACK	UGDS_HISP	UGDS_ASIAN	UGDS_AIAN	UGDS_NHPI	UGDS_;
INSTNM							
<b>Cosmopolitan Beauty and Tech School</b>	0.91	0.00	1.82	97.27	0.0	0.0	
<b>United Beauty College</b>	0.00	0.00	3.30	96.70	0.0	0.0	
<b>Diamond Beauty College</b>	0.00	0.00	2.74	96.58	0.0	0.0	
<b>Asian American International Beauty College</b>	1.73	0.58	1.73	95.95	0.0	0.0	
<b>Rosemead Beauty School</b>	0.00	0.00	4.76	95.24	0.0	0.0	
...	...	...	...	...	...	...	
<b>SAE Institute of Technology San Francisco</b>	NaN	NaN	NaN	NaN	NaN	NaN	
<b>Rasmussen College - Overland Park</b>	NaN	NaN	NaN	NaN	NaN	NaN	
<b>National Personal Training Institute of Cleveland</b>	NaN	NaN	NaN	NaN	NaN	NaN	
<b>Bay Area Medical Academy - San Jose Satellite Location</b>	NaN	NaN	NaN	NaN	NaN	NaN	
<b>Excel Learning Center-San Antonio South</b>	NaN	NaN	NaN	NaN	NaN	NaN	

7535 rows × 9 columns

### (3) Count a Number of Values Using `value_counts()`

```
In [111... movie = pd.read_csv('movie.csv')
```

```
In [112... director = movie['director_name']  
actor_1_fb_likes = movie['actor_1_facebook_likes']
```

```
In [113... director
```

```
Out[113... 0      James Cameron  
1      Gore Verbinski  
2      Sam Mendes  
3      Christopher Nolan  
4      Doug Walker  
...  
4911    Scott Smith  
4912      NaN  
4913    Benjamin Roberds  
4914    Daniel Hsia  
4915      Jon Gunn  
Name: director_name, Length: 4916, dtype: object
```

```
In [114... director.value_counts()
```

```
Out[114... Steven Spielberg    26  
Woody Allen            22  
Martin Scorsese        20  
Clint Eastwood         20  
Ridley Scott           16  
..  
Eric Schaeffer         1  
Michael Meredith       1  
Gia Coppola            1  
Dito Montiel           1  
Brandon Camp           1  
Name: director_name, Length: 2397, dtype: int64
```

```
In [115... director.value_counts(normalize = True)
```

```
Out[115... Steven Spielberg    0.005401  
Woody Allen            0.004570  
Martin Scorsese        0.004155  
Clint Eastwood         0.004155  
Ridley Scott           0.003324  
...  
Eric Schaeffer         0.000208  
Michael Meredith       0.000208  
Gia Coppola            0.000208
```

```
Dito Montiel          0.000208
Brandon Camp          0.000208
Name: director_name, Length: 2397, dtype: float64
```

```
In [116...] director.value_counts(normalize = True)['Spike Lee']
```

```
Out[116...] 0.0033236393851267137
```

```
In [117...] actor_1_fb_likes
```

```
Out[117...] 0          1000.0
1          40000.0
2          11000.0
3          27000.0
4           131.0
...
4911        637.0
4912        841.0
4913         0.0
4914        946.0
4915         86.0
Name: actor_1_facebook_likes, Length: 4916, dtype: float64
```

```
In [118...] actor_1_fb_likes.value_counts()
```

```
Out[118...] 1000.0      436
11000.0     206
2000.0      189
3000.0      150
12000.0     131
...
564.0        1
46000.0       1
49.0          1
447.0         1
161.0         1
Name: actor_1_facebook_likes, Length: 877, dtype: int64
```

```
In [119...] actor_1_fb_likes.value_counts(normalize = True)
```

```
Out[119...] 1000.0      0.088816
11000.0     0.041964
2000.0      0.038501
3000.0      0.030556
12000.0     0.026686
...
564.0      0.000204
46000.0    0.000204
49.0       0.000204
447.0      0.000204
161.0      0.000204
Name: actor_1_facebook_likes, Length: 877, dtype: float64
```

## (4) Descriptive Statistics

```
In [120... movie = pd.read_csv('movie.csv')
```

```
In [121... director = movie['director_name']  
director.describe()
```

```
Out[121... count          4814  
unique          2397  
top      Steven Spielberg  
freq              26  
Name: director_name, dtype: object
```

```
In [122... actor_1_fb_likes = movie['actor_1_facebook_likes']  
actor_1_fb_likes.describe()
```

```
Out[122... count      4909.000000  
mean       6494.488491  
std        15106.986884  
min          0.000000  
25%         607.000000  
50%         982.000000  
75%        11000.000000  
max        640000.000000  
Name: actor_1_facebook_likes, dtype: float64
```

```
In [123... actor_1_fb_likes.min()
```

```
Out[123... 0.0
```

```
In [124... actor_1_fb_likes.max()
```

```
Out[124... 640000.0
```

```
In [125... actor_1_fb_likes.mean()
```

```
Out[125... 6494.488490527602
```

```
In [126... actor_1_fb_likes.std()
```

```
Out[126... 15106.986883848309
```

```
In [127... actor_1_fb_likes.median()
```

```
Out[127... 982.0
```

```
In [128... actor_1_fb_likes.sum()
```

```
Out[128... 31881444.0
```

```
In [129... actor_1_fb_likes.quantile()
```

```
Out[129... 982.0
```

```
In [130... actor_1_fb_likes.quantile(0.5)
```

```
Out[130... 982.0
```

```
In [131... actor_1_fb_likes.quantile(0.2)
```

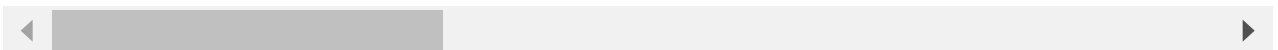
```
Out[131... 510.0
```

```
In [132... actor_1_fb_likes.quantile([0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9])
```

```
Out[132... 0.1      240.0
0.2      510.0
0.3      694.0
0.4      854.0
0.5      982.0
0.6     1000.0
0.7     8000.0
0.8     13000.0
0.9     18000.0
Name: actor_1_facebook_likes, dtype: float64
```

```
In [133... movie.describe()
```

```
Out[133...      num_critic_for_reviews  duration  director_facebook_likes  actor_3_facebook_likes  actor_1_facebook_likes
count      4867.000000    4901.000000      4814.000000      4893.000000      4901.000000
mean         137.988905     107.090798        691.014541        631.276313        649.000000
std          120.239379      25.286015      2832.954125      1625.874802      1510.000000
min           1.000000       7.000000         0.000000         0.000000         0.000000
25%           49.000000      93.000000         7.000000       132.000000        60.000000
50%          108.000000     103.000000        48.000000       366.000000        98.000000
75%          191.000000     118.000000       189.750000       633.000000       1100.000000
max          813.000000     511.000000     23000.000000     23000.000000     64000.000000
```



```
In [134... movie.min()
```

```
Out[134... num_critic_for_reviews      1.0
duration                    7.0
director_facebook_likes     0.0
actor_3_facebook_likes      0.0
actor_1_facebook_likes      0.0
gross                      162.0
```



genres	Action
movie_title	#Horror
num_voted_users	5
cast_total_facebook_likes	0
facenumber_in_poster	0.0
movie_imdb_link	<a href="http://www.imdb.com/title/tt0006864/?ref_=fn_t...">http://www.imdb.com/title/tt0006864/?ref_=fn_t...</a>
num_user_for_reviews	1.0
budget	218.0
title_year	1916.0
actor_2_facebook_likes	0.0
imdb_score	1.6
aspect_ratio	1.18
movie_facebook_likes	0
dtype:	object

## (5) Handling Null Values

```
In [135... # movie = pd.read_csv('movie.csv')
director = movie['director_name']
```

```
In [136... director
```

```
Out[136... 0      James Cameron
1      Gore Verbinski
2      Sam Mendes
3      Christopher Nolan
4      Doug Walker
...
4911    Scott Smith
4912      NaN
4913    Benjamin Roberds
4914    Daniel Hsia
4915      Jon Gunn
Name: director_name, Length: 4916, dtype: object
```

```
In [137... director.isnull()
# director.notnull()
```

```
Out[137... 0      False
1      False
2      False
3      False
4      False
...
4911    False
4912     True
4913    False
4914    False
4915    False
Name: director_name, Length: 4916, dtype: bool
```

```
In [138... director.isnull().any()
```

```
Out[138... True
```

```
In [139... director.isnull().sum()
```

Out[139...] 102

```
In [140...] actor_1_fb_likes.isnull()
```

```
Out[140...] 0      False
            1      False
            2      False
            3      False
            4      False
            ...
            4911   False
            4912   False
            4913   False
            4914   False
            4915   False
            Name: actor_1_facebook_likes, Length: 4916, dtype: bool
```

```
In [141...] actor_1_fb_likes.isnull().sum()
```

Out[141...] 7

```
In [142...] actor_1_fb_likes_filled = actor_1_fb_likes.fillna(0)
            actor_1_fb_likes_filled.isnull().any()
```

Out[142...] False

```
In [143...] actor_1_fb_likes = movie['actor_1_facebook_likes']
            len(actor_1_fb_likes)
```

Out[143...] 4916

```
In [144...] actor_1_fb_likes_dropped = actor_1_fb_likes.dropna()
```

```
In [145...] len(actor_1_fb_likes_dropped)
```

Out[145...] 4909

```
In [146...] actor_1_fb_likes_dropped = actor_1_fb_likes.dropna()
            print(len(actor_1_fb_likes), len(actor_1_fb_likes_dropped))
            # actor_1_fb_likes_dropped.size
            # actor_1_fb_likes_dropped.count()
```

4916 4909

## (6) Transposing the Direction of a Data Frame Operation

In [147...

```
college = pd.read_csv('college.csv', index_col = 'INSTNM')
college_ugds_ = college.filter(like = 'UGDS_')
college_ugds_
```

Out[147...

	UGDS_WHITE	UGDS_BLACK	UGDS_HISP	UGDS_ASIAN	UGDS_AIAN	UGDS_NHPI	UGDS_2M
INSTNM							
Alabama A & M University	0.0333	0.9353	0.0055	0.0019	0.0024	0.0019	0.0
University of Alabama at Birmingham	0.5922	0.2600	0.0283	0.0518	0.0022	0.0007	0.0
Amridge University	0.2990	0.4192	0.0069	0.0034	0.0000	0.0000	0.0
University of Alabama in Huntsville	0.6988	0.1255	0.0382	0.0376	0.0143	0.0002	0.0
Alabama State University	0.0158	0.9208	0.0121	0.0019	0.0010	0.0006	0.0
...	...	...	...	...	...	...	...
SAE Institute of Technology San Francisco	NaN	NaN	NaN	NaN	NaN	NaN	
Rasmussen College - Overland Park	NaN	NaN	NaN	NaN	NaN	NaN	
National Personal Training Institute of Cleveland	NaN	NaN	NaN	NaN	NaN	NaN	
Bay Area Medical Academy - San Jose Satellite Location	NaN	NaN	NaN	NaN	NaN	NaN	
Excel Learning Center-San Antonio South	NaN	NaN	NaN	NaN	NaN	NaN	

7535 rows × 9 columns

In [148...

```
college_ugds_.count()
```

Out[148...

```
UGDS_WHITE    6874
UGDS_BLACK    6874
UGDS_HISP     6874
UGDS_ASIAN    6874
UGDS_AIAN     6874
UGDS_NHPI     6874
UGDS_2MOR     6874
UGDS_NRA      6874
UGDS_UNKN     6874
dtype: int64
```

In [149...

```
college_ugds_.count(axis = 'index')
```

Out[149...

```
UGDS_WHITE    6874
UGDS_BLACK    6874
UGDS_HISP     6874
UGDS_ASIAN    6874
UGDS_AIAN     6874
UGDS_NHPI     6874
UGDS_2MOR     6874
UGDS_NRA      6874
UGDS_UNKN     6874
dtype: int64
```

In [150...

```
college_ugds_.count(axis = 'columns')
```

Out[150...

```
INSTNM
Alabama A & M University          9
University of Alabama at Birmingham 9
Amridge University               9
University of Alabama in Huntsville 9
Alabama State University         9
..
SAE Institute of Technology San Francisco 0
Rasmussen College - Overland Park    0
National Personal Training Institute of Cleveland 0
Bay Area Medical Academy - San Jose Satellite Location 0
Excel Learning Center-San Antonio South 0
Length: 7535, dtype: int64
```

In [152...

```
college_ugds_.count(axis = 'columns').sort_values()
```

Out[152...

```
INSTNM
Excel Learning Center-San Antonio South 0
Albany Law School                      0
Albany Medical College                 0
Institute for the Psychological Sciences 0
Forest Institute of Professional Psychology 0
..
Farmingdale State College             9
SUNY College of Agriculture and Technology at Cobleskill 9
SUNY College of Technology at Delhi    9
SUNY College of Technology at Alfred    9
```

```
In [153... college_ugds_ = college_ugds_.dropna(how = 'all')
```

```
In [154... college_ugds_.count(axis = 'columns').sort_values()
```

```
Out[154... INSTNM
Alabama A & M University          9
Pike County Joint Vocational School District  9
South Texas College              9
Professional Technical Institution Inc        9
Franklin Technology-MSSU          9
..
CUNY Graduate School and University Center    9
CUNY City College                      9
College of Staten Island CUNY             9
CUNY Bronx Community College             9
Coastal Pines Technical College          9
Length: 6874, dtype: int64
```

```
In [155... college_ugds_.count()
```

```
Out[155... UGDS_WHITE    6874
UGDS_BLACK    6874
UGDS_HISP     6874
UGDS_ASIAN    6874
UGDS_AIAN     6874
UGDS_NHPI     6874
UGDS_2MOR     6874
UGDS_NRA      6874
UGDS_UNKN     6874
dtype: int64
```

```
In [156... college_ugds_
```

	UGDS_WHITE	UGDS_BLACK	UGDS_HISP	UGDS_ASIAN	UGDS_AIAN	UGDS_NHPI	UGDS_2M
INSTNM							
Alabama A & M University	0.0333	0.9353	0.0055	0.0019	0.0024	0.0019	0.0
University of Alabama at Birmingham	0.5922	0.2600	0.0283	0.0518	0.0022	0.0007	0.0
Amridge University	0.2990	0.4192	0.0069	0.0034	0.0000	0.0000	0.0
University of Alabama in Huntsville	0.6988	0.1255	0.0382	0.0376	0.0143	0.0002	0.0
Alabama State	0.0158	0.9208	0.0121	0.0019	0.0010	0.0006	0.0

	UGDS_WHITE	UGDS_BLACK	UGDS_HISP	UGDS_ASIAN	UGDS_AIAN	UGDS_NHPI	UGDS_2M
INSTNM							
University							
...	...	...	...	...	...	...	...
Hollywood Institute of Beauty Careers- West Palm Beach	0.2182	0.4182	0.2364	0.0182	0.0000	0.0000	0.0
Hollywood Institute of Beauty Careers- Casselberry	0.1200	0.3333	0.4400	0.0000	0.0000	0.0000	0.0
Coachella Valley Beauty College- Beaumont	0.3284	0.1045	0.4925	0.0149	0.0299	0.0149	0.0
Dewey University- Mayaguez	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0
Coastal Pines Technical College	0.6762	0.2508	0.0359	0.0045	0.0034	0.0017	0.0

6874 rows × 9 columns

## Summary: Handling Null Values

In [157...

```
college = pd.read_csv('college.csv', index_col = 'INSTNM')
college_ugds_ = college.filter(like = 'UGDS_')
college_ugds_
```

Out[157...

	UGDS_WHITE	UGDS_BLACK	UGDS_HISP	UGDS_ASIAN	UGDS_AIAN	UGDS_NHPI	UGDS_2M
INSTNM							
Alabama A & M University	0.0333	0.9353	0.0055	0.0019	0.0024	0.0019	0.0
University of Alabama at Birmingham	0.5922	0.2600	0.0283	0.0518	0.0022	0.0007	0.0
Amridge University	0.2990	0.4192	0.0069	0.0034	0.0000	0.0000	0.0

	UGDS_WHITE	UGDS_BLACK	UGDS_HISP	UGDS_ASIAN	UGDS_AIAN	UGDS_NHPI	UGDS_2I
INSTNM							
University of Alabama in Huntsville	0.6988	0.1255	0.0382	0.0376	0.0143	0.0002	0.0
Alabama State University	0.0158	0.9208	0.0121	0.0019	0.0010	0.0006	0.0
...	...	...	...	...	...	...	...
SAE Institute of Technology San Francisco	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Rasmussen College - Overland Park	NaN	NaN	NaN	NaN	NaN	NaN	NaN
National Personal Training Institute of Cleveland	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Bay Area Medical Academy - San Jose Satellite Location	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Excel Learning Center-San Antonio South	NaN	NaN	NaN	NaN	NaN	NaN	NaN

7535 rows × 9 columns

In [158...

```
college_ugds_.isnull().sum(axis = 'columns').sort_values(ascending = False)
```

Out[158...

```
INSTNM
Excel Learning Center-San Antonio South    9
Western State College of Law at Argosy University    9
Albany Law School    9
Albany Medical College    9
A T Still University of Health Sciences    9
..
SUNY at Binghamton    0
SUNY at Albany    0
Morrisville State College    0
Farmingdale State College    0
```

The University of Texas at Austin 0  
Length: 7535, dtype: int64

```
In [159... college_ugds_ = college_ugds_.dropna(how = 'all')
```

```
In [160... college_ugds_.isnull().sum(axis = 'columns').sort_values(ascending = False)
```

Out[160... INSTNM  
Alabama A & M University 0  
California State University-Monterey Bay 0  
Lorain County Joint Vocational School District 0  
Pike County Joint Vocational School District 0  
South Texas College 0  
..  
CUNY Hunter College 0  
CUNY Hostos Community College 0  
CUNY Graduate School and University Center 0  
CUNY City College 0  
Coastal Pines Technical College 0  
Length: 6874, dtype: int64

```
In [161... college_ugds_.isnull().sum()
```

Out[161... UGDS\_WHITE 0  
UGDS\_BLACK 0  
UGDS\_HISP 0  
UGDS\_ASIAN 0  
UGDS\_AIAN 0  
UGDS\_NHPI 0  
UGDS\_2MOR 0  
UGDS\_NRA 0  
UGDS\_UNKN 0  
dtype: int64

```
In [162... college_ugds_
```

Out[162...

	UGDS_WHITE	UGDS_BLACK	UGDS_HISP	UGDS_ASIAN	UGDS_AIAN	UGDS_NHPI	UGDS_2M
INSTNM							
Alabama A & M University	0.0333	0.9353	0.0055	0.0019	0.0024	0.0019	0.0
University of Alabama at Birmingham	0.5922	0.2600	0.0283	0.0518	0.0022	0.0007	0.0
Amridge University	0.2990	0.4192	0.0069	0.0034	0.0000	0.0000	0.0
University of Alabama in Huntsville	0.6988	0.1255	0.0382	0.0376	0.0143	0.0002	0.0
Alabama State	0.0158	0.9208	0.0121	0.0019	0.0010	0.0006	0.0



	UGDS_WHITE	UGDS_BLACK	UGDS_HISP	UGDS_ASIAN	UGDS_AIAN	UGDS_NHPI	UGDS_2M
INSTNM							
University							
...	...	...	...	...	...	...	...
Hollywood Institute of Beauty Careers- West Palm Beach	0.2182	0.4182	0.2364	0.0182	0.0000	0.0000	0.0
Hollywood Institute of Beauty Careers- Casselberry	0.1200	0.3333	0.4400	0.0000	0.0000	0.0000	0.0
Coachella Valley Beauty College- Beaumont	0.3284	0.1045	0.4925	0.0149	0.0299	0.0149	0.0
Dewey University- Mayaguez	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0
Coastal Pines Technical College	0.6762	0.2508	0.0359	0.0045	0.0034	0.0017	0.0

6874 rows × 9 columns