Pandas 資料分析 (1)

張家瑋 副教授
國立臺中科技大學資訊工程系

DataFrame 物件 (1/2)

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

pd.set_option('max_columns', 4, 'max_rows', 10, 'max_colwidth', 12)
```

In [2]:
```
movies = pd.read_csv('data/movie.csv')
movies
```

Out[2]:

	color	director_name	...	aspect_ratio	movie_facebook_likes
0	Color	James Ca...	...	1.78	33000
1	Color	Gore Ver...	...	2.35	0
2	Color	Sam Mendes	...	2.35	85000
3	Color	Christop...	...	2.35	164000
4	NaN	Doug Walker	...	NaN	0
...	...	...	...	...	...
4911	Color	Scott Smith	...	NaN	84
4912	Color	NaN	...	16.00	32000
4913	Color	Benjamin...	...	NaN	16
4914	Color	Daniel Hsia	...	2.35	660
4915	Color	Jon Gunn	...	1.85	456

4916 rows × 28 columns

In [4]:
```
movies.head()
```

Out[4]:

	color	director_name	...	aspect_ratio	movie_facebook_likes
0	Color	James Ca...	...	1.78	33000
1	Color	Gore Ver...	...	2.35	0
2	Color	Sam Mendes	...	2.35	85000
3	Color	Christop...	...	2.35	164000
4	NaN	Doug Walker	...	NaN	0

5 rows × 28 columns

In [5]:
```
movies.head(3)
```

Out[5]:

	color	director_name	...	aspect_ratio	movie_facebook_likes
0	Color	James Ca...	...	1.78	33000
1	Color	Gore Ver...	...	2.35	0
2	Color	Sam Mendes	...	2.35	85000

3 rows × 28 columns

DataFrame 的屬性 (2/2)

```
In [6]: movies = pd.read_csv('data/movie.csv')
        columns = movies.columns
        index = movies.index
        data = movies.values
```

```
In [7]: columns
```

```
Out[7]: 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 [8]: index
```

```
Out[8]: RangeIndex(start=0, stop=4916, step=1)
```

```
In [9]: data
```

```
Out[9]: array([['Color', 'James Cameron', 723.0, ..., 7.9, 1.78, 33000],
               ['Color', 'Gore Verbinski', 302.0, ..., 7.1, 2.35, 0],
               ['Color', 'Sam Mendes', 602.0, ..., 6.8, 2.35, 85000],
               ...,
               ['Color', 'Benjamin Roberds', 13.0, ..., 6.3, nan, 16],
               ['Color', 'Daniel Hsia', 14.0, ..., 6.3, 2.35, 660],
               ['Color', 'Jon Gunn', 43.0, ..., 6.6, 1.85, 456]], dtype=object)
```

```
In [10]: type(index)
```

```
Out[10]: pandas.core.indexes.range.RangeIndex
```

```
In [11]: type(columns)
```

```
Out[11]: pandas.core.indexes.base.Index
```

```
In [12]: type(data)
```

```
Out[12]: numpy.ndarray
```

```
In [13]: index.to_numpy()
```

```
Out[13]: array([   0,    1,    2, ..., 4913, 4914, 4915], dtype=int64)
```

```
In [14]: columns.to_numpy()
```

```
Out[14]: array(['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)
```

Series 物件

```
In [15]: fruit = pd.Series(['apple', 'banana', 'grape', 'pineapple'], index=['a', 'b', 'c', 'd'])
         fruit

Out[15]: a        apple
         b       banana
         c        grape
         d    pineapple
         dtype: object

In [16]: fruit = pd.Series(['apple', 'banana', 'grape', 'pineapple'])
         fruit

Out[16]: 0        apple
         1       banana
         2        grape
         3    pineapple
         dtype: object

In [17]: fruit.dtypes

Out[17]: dtype('O')

In [18]: fruit.size

Out[18]: 4
```

Pandas 中的資料型別

```
In [19]: movies = pd.read_csv('data/movie.csv')
         movies.dtypes
```

```
Out[19]: color                      object
         director_name              object
         num_critic_for_reviews    float64
         duration                  float64
         director_facebook_likes   float64
                                    ...
         title_year                float64
         actor_2_facebook_likes    float64
         imdb_score                float64
         aspect_ratio              float64
         movie_facebook_likes        int64
         Length: 28, dtype: object
```

```
In [20]: movies.dtypes.value_counts()
```

```
Out[20]: float64    13
         object     12
         int64       3
         dtype: int64
```

```
In [21]: movies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4916 entries, 0 to 4915
Data columns (total 28 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   color                      4897 non-null   object
 1   director_name              4814 non-null   object
 2   num_critic_for_reviews     4867 non-null   float64
 3   duration                   4901 non-null   float64
 4   director_facebook_likes    4814 non-null   float64
 5   actor_3_facebook_likes     4893 non-null   float64
 6   actor_2_name               4903 non-null   object
 7   actor_1_facebook_likes     4909 non-null   float64
 8   gross                      4054 non-null   float64
 9   genres                     4916 non-null   object
 10  actor_1_name               4909 non-null   object
 11  movie_title                4916 non-null   object
 12  num_voted_users            4916 non-null   int64
 13  cast_total_facebook_likes  4916 non-null   int64
 14  actor_3_name               4893 non-null   object
 15  facenumber_in_poster       4903 non-null   float64
 16  plot_keywords              4764 non-null   object
 17  movie_imdb_link            4916 non-null   object
 18  num_user_for_reviews       4895 non-null   float64
 19  language                   4904 non-null   object
 20  country                    4911 non-null   object
 21  content_rating             4616 non-null   object
 22  budget                     4432 non-null   float64
 23  title_year                 4810 non-null   float64
 24  actor_2_facebook_likes     4903 non-null   float64
 25  imdb_score                 4916 non-null   float64
 26  aspect_ratio               4590 non-null   float64
 27  movie_facebook_likes       4916 non-null   int64
dtypes: float64(13), int64(3), object(12)
memory usage: 1.1+ MB
```

選取 DataFrame 的欄位

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

        pd.set_option('max_columns', 4, 'max_rows', 10, 'max_colwidth', 12)
```

```
In [2]: movies = pd.read_csv('data/movie.csv')
        movies['director_name']
```

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

```
In [3]: movies.director_name
```

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

```
In [4]: movies.loc[:, 'director_name']
```

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

```
In [5]: movies.iloc[:, 1]
```

```
Out[5]: 0       James Ca...
        1       Gore Ver...
        2        Sam Mendes
        3       Christop...
        4       Doug Walker
                   ...
        4911    Scott Smith
        4912            NaN
        4913    Benjamin...
        4914    Daniel Hsia
        4915       Jon Gunn
        Name: director_name, Length: 4916, dtype: object
```

Series 的方法-資料的敘述統計

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

        movies = pd.read_csv('data/movie.csv')
        director = movies['director_name']
        fb_likes = movies['actor_1_facebook_likes']
        director.dtype
```

```
Out[1]: dtype('O')
```

```
In [2]: director.sample(n=5, random_state=42)
```

```
Out[2]: 2347      Brian Percival
        4687         Lucio Fulci
        691         Phillip Noyce
        3911        Sam Peckinpah
        2488    Rowdy Herrington
        Name: director_name, dtype: object
```

```
In [3]: director.value_counts()
```

```
Out[3]: Steven Spielberg    26
        Woody Allen         22
        Martin Scorsese     20
        Clint Eastwood      20
        Ridley Scott        16
                            ..
        John Putch           1
        Luca Guadagnino      1
        Sam Fell             1
        Dan Fogelman         1
        Daniel Hsia          1
        Name: director_name, Length: 2397, dtype: int64
```

```
In [4]: director.unique()
```

```
Out[4]: array(['James Cameron', 'Gore Verbinski', 'Sam Mendes', ...,
               'Scott Smith', 'Benjamin Roberds', 'Daniel Hsia'], dtype=object)
```

```
In [5]: fb_likes.min()
```

```
Out[5]: 0.0
```

```
In [6]: fb_likes.max()
```

```
Out[6]: 640000.0
```

```
In [7]: fb_likes.mean()
```

```
Out[7]: 6494.488490527602
```

```
In [8]: fb_likes.median()
```

```
Out[8]: 982.0
```

```
In [9]: fb_likes.std()
```

```
Out[9]: 15106.986883848185
```

```
In [10]: fb_likes.describe()
```

```
Out[10]: 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 [40]: director.value_counts(normalize=True)
```

```
Out[40]: Steven Spielberg    0.005401
         Woody Allen         0.004570
         Martin Scorsese     0.004155
         Clint Eastwood      0.004155
         Spike Lee           0.003324
                               ...
         Jon Stewart         0.000208
         Jeff Schaffer       0.000208
         Ari Folman          0.000208
         Emilio Estevez      0.000208
         David Nutter        0.000208
         Name: director_name, Length: 2397, dtype: float64
```

```
In [35]: fb_likes.quantile(.2)
```

```
Out[35]: 510.0
```

```
In [36]: fb_likes.quantile([.1, .2, .3, .4, .5, .6, .7, .8, .9])
```

```
Out[36]: 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
```

Series 的方法-資料的遺失值處理

```
In [37]: director.isna()
```

```
Out[37]: 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 [41]: director.hasnans
```

```
Out[41]: True
```

```
In [42]: director.notna()
```

```
Out[42]: 0        True
         1        True
         2        True
         3        True
         4        True
                  ...
         4911     True
         4912    False
         4913     True
         4914     True
         4915     True
         Name: director_name, Length: 4916, dtype: bool
```

```
In [38]: fb_likes_filled = fb_likes.fillna(0)
         fb_likes_filled.count()
```

```
Out[38]: 4916
```

```
In [39]: fb_likes_dropped = fb_likes.dropna()
         fb_likes_dropped.size
```

```
Out[39]: 4909
```

Series 的方法-資料欄位的計算 (1/2)

```
In [44]: movies = pd.read_csv('data/movie.csv')
         imdb_score = movies['imdb_score']
         imdb_score

Out[44]: 0       7.9
         1       7.1
         2       6.8
         3       8.5
         4       7.1
                ...
         4911    7.7
         4912    7.5
         4913    6.3
         4914    6.3
         4915    6.6
         Name: imdb_score, Length: 4916, dtype: float64
```

```
In [45]: imdb_score + 1

Out[45]: 0       8.9
         1       8.1
         2       7.8
         3       9.5
         4       8.1
                ...
         4911    8.7
         4912    8.5
         4913    7.3
         4914    7.3
         4915    7.6
         Name: imdb_score, Length: 4916, dtype: float64
```

```
In [46]: imdb_score * 2.5

Out[46]: 0       19.75
         1       17.75
         2       17.00
         3       21.25
         4       17.75
                 ...
         4911    19.25
         4912    18.75
         4913    15.75
         4914    15.75
         4915    16.50
         Name: imdb_score, Length: 4916, dtype: float64
```

```
In [47]: imdb_score // 7

Out[47]: 0       1.0
         1       1.0
         2       0.0
         3       1.0
         4       1.0
                ...
         4911    1.0
         4912    1.0
         4913    0.0
         4914    0.0
         4915    0.0
         Name: imdb_score, Length: 4916, dtype: float64
```

```
In [48]: imdb_score > 7

Out[48]: 0        True
         1        True
         2       False
         3        True
         4        True
                 ...
         4911     True
         4912     True
         4913    False
         4914    False
         4915    False
         Name: imdb_score, Length: 4916, dtype: bool
```

Series 的方法-資料欄位的計算 (2/2)

```
In [44]: movies = pd.read_csv('data/movie.csv')
         imdb_score = movies['imdb_score']
         imdb_score

Out[44]: 0       7.9
         1       7.1
         2       6.8
         3       8.5
         4       7.1
                ...
         4911    7.7
         4912    7.5
         4913    6.3
         4914    6.3
         4915    6.6
         Name: imdb_score, Length: 4916, dtype: float64
```

```
In [45]: imdb_score + 1

Out[45]: 0       8.9
         1       8.1
         2       7.8
         3       9.5
         4       8.1
                ...
         4911    8.7
         4912    8.5
         4913    7.3
         4914    7.3
         4915    7.6
         Name: imdb_score, Length: 4916, dtype: float64
```

```
In [46]: imdb_score * 2.5

Out[46]: 0       19.75
         1       17.75
         2       17.00
         3       21.25
         4       17.75
```

```
In [47]: imdb_score // 7

Out[47]: 0       1.0
         1       1.0
         2       0.0
         3       1.0
         4       1.0
                ...
         4911    1.0
         4912    1.0
         4913    0.0
         4914    0.0
         4915    0.0
         Name: imdb_score, Length: 4916, dtype: float64
```

```
In [48]: imdb_score > 7

Out[48]: 0        True
         1        True
         2       False
         3        True
         4        True
                 ...
         4911     True
         4912     True
         4913    False
         4914    False
         4915    False
         Name: imdb_score, Length: 4916, dtype: bool
```

	算符	Series 方法
數值運算	+, −, *, /, //, %, **	.add, .sub, .mul, .div, .floordiv, .mod, .pow
比較運算	<, >, <=, >=, ==, !=	.lt, .gt, .le, .ge, .eq, .ne

Series 方法的串聯使用 (1/3)

```
In [54]: movies = pd.read_csv('data/movie.csv')
         fb_likes = movies['actor_1_facebook_likes']
         director = movies['director_name']

In [55]: director.value_counts().head(3)

Out[55]: Steven Spielberg    26
         Woody Allen         22
         Martin Scorsese     20
         Name: director_name, dtype: int64
```

```
In [57]: fb_likes.dtype

Out[57]: dtype('float64')

In [58]: (fb_likes.fillna(0)
                  .astype(int)
                  .head()
         )

Out[58]: 0     1000
         1    40000
         2    11000
         3    27000
         4      131
         Name: actor_1_facebook_likes, dtype: int32
```

Series 方法的串聯使用 (2/3)

```
In [57]: fb_likes.dtype

Out[57]: dtype('float64')
```

```
In [58]: (fb_likes.fillna(0)
                  .astype(int)
                  .head()
          )

Out[58]: 0     1000
         1    40000
         2    11000
         3    27000
         4      131
         Name: actor_1_facebook_likes, dtype: int32
```

```
In [65]: fb_likes.fillna(0)\
                 .astype(int)\
                 .head()

Out[65]: 0     1000
         1    40000
         2    11000
         3    27000
         4      131
         Name: actor_1_facebook_likes, dtype: int32
```

Series 方法的串聯使用 (3/3)

In [61]:
```
def debug_df(df):
    print("BEFORE")
    print(df)
    print("AFTER")
    return df
```

In [62]:
```
(fb_likes.fillna(0)
         .pipe(debug_df)
         .astype(int)
         .head()
)
```

```
BEFORE
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
AFTER
```

Out[62]:
```
0     1000
1    40000
2    11000
3    27000
4      131
Name: actor_1_facebook_likes, dtype: int32
```

In [63]:
```
intermediate = None
def get_intermediate(df):
    global intermediate
    intermediate = df
    return df
```

In [64]:
```
res = (fb_likes.fillna(0)
         .pipe(get_intermediate)
         .astype(int)
         .head()
)

intermediate
```

Out[64]:
```
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
```

更改欄位名稱的方法 (1/4)

```
In [66]: movies = pd.read_csv('data/movie.csv')

In [67]: col_map = {'director_name':'Director Name'}

In [68]: movies.rename(columns=col_map).head()
```

Out[68]:

	color	Director Name	...	aspect_ratio	movie_facebook_likes
0	Color	James Ca...	...	1.78	33000
1	Color	Gore Ver...	...	2.35	0
2	Color	Sam Mendes	...	2.35	85000
3	Color	Christop...	...	2.35	164000
4	NaN	Doug Walker	...	NaN	0

5 rows × 28 columns

更改欄位名稱的方法 (2/4)

In [69]:
```
idx_map = {'Avatar':'Ratava',
           'Spectre': 'Ertceps',
           "Pirates of the Caribbean: At World's End": 'POC'}
col_map = {'aspect_ratio': 'aspect',
           'movie_facebook_likes': 'fblikes'}
(movies
   .set_index('movie_title')
   .rename(index=idx_map, columns=col_map)
   .head(3)
)
```

Out[69]:

movie_title	color	director_name	...	aspect	fblikes
Ratava	Color	James Ca...	...	1.78	33000
POC	Color	Gore Ver...	...	2.35	0
Ertceps	Color	Sam Mendes	...	2.35	85000

3 rows × 27 columns

更改欄位名稱的方法 (3/4)

```
movies = pd.read_csv('data/movie.csv', index_col='movie_title')
movies.head(3)
```

movie_title	color	director_name	...	aspect_ratio	movie_facebook_likes
Avatar	Color	James Ca...	...	1.78	33000
Pirates of the Caribbean: At World's End	Color	Gore Ver...	...	2.35	0
Spectre	Color	Sam Mendes	...	2.35	85000

3 rows × 27 columns

```
movies = pd.read_csv('data/movie.csv', index_col='movie_title')
ids = movies.index.tolist()
columns = movies.columns.tolist()
ids[0] = 'Ratava'
ids[1] = 'POC'
ids[2] = 'Ertceps'
columns[1] = 'director'
columns[-2] = 'aspect'
columns[-1] = 'fblikes'
movies.index = ids
movies.columns = columns
```

更改欄位名稱的方法 (4/4)

In [72]:
```
def to_clean(val):
    return val.strip().replace('_', '.')
movies.rename(columns=to_clean).head(3)
```

Out[72]:

movie_title	color	director.name	...	aspect.ratio	movie.facebook.likes
Avatar	Color	James Ca...	...	1.78	33000
Pirates of the Caribbean: At World's End	Color	Gore Ver...	...	2.35	0
Spectre	Color	Sam Mendes	...	2.35	85000

3 rows × 27 columns

In [73]:
```
cols = [col.strip().replace('_', '.')
        for col in movies.columns]
movies.columns = cols
movies.head(3)
```

Out[73]:

movie_title	color	director.name	...	aspect.ratio	movie.facebook.likes
Avatar	Color	James Ca...	...	1.78	33000
Pirates of the Caribbean: At World's End	Color	Gore Ver...	...	2.35	0
Spectre	Color	Sam Mendes	...	2.35	85000

3 rows × 27 columns

新增欄位 (1/2)

```
movies = pd.read_csv('data/movie.csv')
movies['has_seen'] = 0
movies.head(3)
```

	color	director_name	...	movie_facebook_likes	has_seen
0	Color	James Ca...	...	33000	0
1	Color	Gore Ver...	...	0	0
2	Color	Sam Mendes	...	85000	0

3 rows × 29 columns

```
col_map = {'aspect_ratio': 'aspect',
           'movie_facebook_likes': 'fblikes'}
(movies
   .rename(columns=col_map)
   .assign(has_seen=0)
)
movies.head(3)
```

	color	director_name	...	movie_facebook_likes	has_seen
0	Color	James Ca...	...	33000	0
1	Color	Gore Ver...	...	0	0
2	Color	Sam Mendes	...	85000	0

3 rows × 29 columns

新增欄位 (2/2)

```
cols = ['actor_1_facebook_likes','actor_2_facebook_likes',
        'actor_3_facebook_likes','director_facebook_likes']
sum_col = movies[cols].sum(axis='columns')
sum_col.head(5)
```

```
0     2791.0
1    46563.0
2    11554.0
3    95000.0
4      274.0
dtype: float64
```

```
movies.assign(total_likes=sum_col).head(5)
```

	color	director_name	...	has_seen	total_likes
0	Color	James Ca...	...	0	2791.0
1	Color	Gore Ver...	...	0	46563.0
2	Color	Sam Mendes	...	0	11554.0
3	Color	Christop...	...	0	95000.0
4	NaN	Doug Walker	...	0	274.0

5 rows × 30 columns

```
def sum_likes(df):
    return df[[c for c in df.columns
                if 'like' in c]].sum(axis=1)
movies.assign(total_likes=sum_likes).head(5)
```

	color	director_name	...	has_seen	total_likes
0	Color	James Ca...	...	0	40625.0
1	Color	Gore Ver...	...	0	94913.0
2	Color	Sam Mendes	...	0	108254.0
3	Color	Christop...	...	0	365759.0
4	NaN	Doug Walker	...	0	417.0

5 rows × 30 columns

刪除欄位

1. df = df.drop(columns= ‘column_name’)
2. del df[‘ column_name’]

練習 (1/3)

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

movies = pd.read_csv('data/movie.csv')
movies['has_seen'] = 0
movies.head(3)
```

Out[1]:

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes	actor_2_name	actor_1_facebook_likes	gross
0	Color	James Cameron	723.0	178.0	0.0	855.0	Joel David Moore	1000.0	760505847.0
1	Color	Gore Verbinski	302.0	169.0	563.0	1000.0	Orlando Bloom	40000.0	309404152.0
2	Color	Sam Mendes	602.0	148.0	0.0	161.0	Rory Kinnear	11000.0	200074175.0

3 rows × 29 columns

In [2]:
```
total = (movies['actor_1_facebook_likes'] +
         movies['actor_2_facebook_likes'] +
         movies['actor_3_facebook_likes'] +
         movies['director_facebook_likes'])
total.head(5)
```

Out[2]:
```
0     2791.0
1    46563.0
2    11554.0
3    95000.0
4        NaN
dtype: float64
```

練習 (2/3)

In [3]:
```
def cast_like_gt_actor_director(df):
    return df['cast_total_facebook_likes'] >= df['total_likes']

df2 = (movies.assign(total_likes=total,
                     is_cast_likes_more = cast_like_gt_actor_director)
      )
```

In [4]:
```
df2['is_cast_likes_more'].all()
```

Out[4]:
```
False
```

In [5]:
```
df2 = df2.drop(columns='total_likes')
```

In [6]:
```
actor_sum = (movies[[c for c in movies.columns if 'actor_' in c and '_likes' in c]]
             .sum(axis='columns'))

actor_sum.head(5)
```

Out[6]:
```
0     2791.0
1    46000.0
2    11554.0
3    73000.0
4      143.0
dtype: float64
```

練習 (3/3)

```
In [7]: movies['cast_total_facebook_likes'] >= actor_sum
```

```
Out[7]: 0       True
        1       True
        2       True
        3       True
        4       True
                ...
        4911    True
        4912    True
        4913    True
        4914    True
        4915    True
        Length: 4916, dtype: bool
```

```
In [8]: movies['cast_total_facebook_likes'].ge(actor_sum)
```

```
Out[8]: 0       True
        1       True
        2       True
        3       True
        4       True
                ...
        4911    True
        4912    True
        4913    True
        4914    True
        4915    True
        Length: 4916, dtype: bool
```

```
In [9]: movies['cast_total_facebook_likes'].ge(actor_sum).all()
```

```
Out[9]: True
```