レシピ43 DataFrameの行のマスキング（P113）

- `mask` メソッドは、whereメソッドと正反対の演算をする
- `mask` はデフォルトでは「True」を欠損値等で置き換える（whereはFalseを置き換える）

- このレシピでは、映画データセットで2010年より後の映画にマスキングをして、欠損値のある行をすべてフィルタリングする

（1）データセットを読み込み、映画の題名をインデックスにし、基準を作る

In [9]:
```
movie = pd.read_csv('movie.csv', index_col='movie_title')
c1 = movie['title_year'] >= 2010
c2 = movie['title_year'].isnull()
criteria = c1 | c2
```

（2） `mask` メソッドを使い、2010年より後の映画の行の値を欠損値にする。title_yearの元の値が欠損の映画もマスキングされる

In [13]:
```
# わかりづらいので、title_yearを先頭にもってくる
first_col = movie.pop('title_year')
movie.insert(0, 'title_year', first_col)
movie.head()
```

Out[13]:

movie_title	title_year	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_faceb
Avatar	2009.0	Color	James Cameron	723.0	178.0	0.0	
Pirates of the Caribbean: At World's End	2007.0	Color	Gore Verbinski	302.0	169.0	563.0	
Spectre	2015.0	Color	Sam Mendes	602.0	148.0	0.0	
The Dark Knight Rises	2012.0	Color	Christopher Nolan	813.0	164.0	22000.0	
Star Wars: Episode VII - The Force Awakens	NaN	NaN	Doug Walker	NaN	NaN	131.0	

5 rows × 27 columns

In [22]:
```
# cf.criteria（比較演算子は |(or)。2010年以降の映画、または年が空欄の映画を指す）
(movie['title_year'] >= 2010) | (movie['title_year'].isnull())
```

Out[22]:
```
movie_title
Avatar                                         False
Pirates of the Caribbean: At World's End       False
Spectre                                         True
The Dark Knight Rises                           True
Star Wars: Episode VII - The Force Awakens      True
                                               ...
Signed Sealed Delivered                         True
```

The Following True
A Plague So Pleasant True
Shanghai Calling True
My Date with Drew False
Name: title_year, Length: 4916, dtype: bool

mask

- **条件に対し「True」になったものに処理（欠損値にする、等）を行う**
- デフォルトでは、Trueデータを欠損値でマスクする（以下の画面では上でTrueになった行がNaNに変わった）

In [30]:
```
movie.mask(criteria).head()
```

Out[30]:

movie_title	title_year	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_faceb
Avatar	2009.0	Color	James Cameron	723.0	178.0	0.0	
Pirates of the Caribbean: At World's End	2007.0	Color	Gore Verbinski	302.0	169.0	563.0	
Spectre	NaN	NaN	NaN	NaN	NaN	NaN	
The Dark Knight Rises	NaN	NaN	NaN	NaN	NaN	NaN	
Star Wars: Episode VII - The Force Awakens	NaN	NaN	NaN	NaN	NaN	NaN	

5 rows × 27 columns

maskによってNaN行が多数できている

（3）それらの欠損行を削除すれば、取出したいものだけが残る

- すべての値がNaNである行を削除 `dropna(how='all')`

In [26]:
```
movie_mask = movie.mask(criteria).dropna(how='all')
movie_mask.head()
```

Out[26]:

movie_title	title_year	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_faceb
Avatar	2009.0	Color	James Cameron	723.0	178.0	0.0	
Pirates of the Caribbean: At World's End	2007.0	Color	Gore Verbinski	302.0	169.0	563.0	
Spider-Man 3	2007.0	Color	Sam Raimi	392.0	156.0	0.0	
Harry Potter and the Half-Blood Prince	2009.0	Color	David Yates	375.0	153.0	282.0	

movie_title	title_year	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_faceb
Superman Returns	2006.0	Color	Bryan Singer	434.0	169.0	0.0	

5 rows × 27 columns

（4）

- （3）のやり方とは別にBooleanインデックスでも同じことをやってみる。
- `equals` で結果が同じであればよい -> なぜかFalse

In [39]:
```
movie_boolean = movie[movie['title_year'] < 2010] # isNullの行は除外される?(5)の結果からは対象外になっている模様
movie_mask.equals(movie_boolean)
```

Out[39]: False

（5）Falseの理由は何？　同じshapeかを確認してみる　-> True

In [29]:
```
movie_mask.shape == movie_boolean.shape
```

Out[29]: True

（6）

- maskメソッドが多くの欠損値を作った
- 欠損値はfloat型なので、NaNが含まれる整数カラム全体もfloat型になってしまっている
- **`equals` メソッドは、カラムの値が同じでもデータ型が違うとFalseを返す**
- データ型をチェックしてみる

In [32]:
```
movie_mask.dtypes == movie_boolean.dtypes
```

Out[32]:
```
title_year                 True
color                      True
director_name              True
num_critic_for_reviews     True
duration                   True
director_facebook_likes    True
actor_3_facebook_likes     True
actor_2_name               True
actor_1_facebook_likes     True
gross                      True
genres                     True
actor_1_name               True
num_voted_users           False
cast_total_facebook_likes False
actor_3_name               True
facenumber_in_poster       True
plot_keywords              True
movie_imdb_link            True
num_user_for_reviews       True
language                   True
country                    True
content_rating             True
budget                     True
actor_2_facebook_likes     True
imdb_score                 True
```

```
aspect_ratio              True
movie_facebook_likes     False
dtype: bool
```

---

(7)

- 3つのカラムのデータ型が違っていた
- 以下よくわからない
- テストモジュール（ユーザーは開発者が主）に `assert_frame_equal` 関数があり、SeriesとDataFrameの等価性をデータ型は無視してチェックする
- **等価ならば、Noneを返す（要は見た目反応がない）、等しくない場合はエラーを返す**

In [35]:
```
from pandas.testing import assert_frame_equal
assert_frame_equal(movie_boolean, movie_mask, check_dtype=False)
```

（補足）

- マスキングを使った場合と、Booleanインデックス法ではBooleanの方が1桁速い

In [40]:
```
%timeit movie.mask(criteria).dropna(how='all')
```

```
21.8 ms ± 2.66 ms per loop (mean ± std. dev. of 7 runs, 10 loops each)
```

In [42]:
```
%timeit movie[movie['title_year'] < 2010]
```

```
1.58 ms ± 76.6 µs per loop (mean ± std. dev. of 7 runs, 1000 loops each)
```

以上

In [ ]:
```

```