Ressources

- Page GitHub du projet Data for Good

https://github.com/dataforgoodfr/bechdelai

- Documentation de l'API

https://bechdeltest.com/api/v1/doc#getMovieByImdbId

- Licence à citer dans README

Étape 1 : Préparation et Importation des Données

In [1]:
```
# 1. Importer les bibliothèques nécessaires

import requests
import json
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
# 2. Récupérer les données à partir de l'API

link = 'http://bechdeltest.com/api/v1/getAllMovies'


# Effectuer la requête GET

r = requests.get(link)


# Vérifier que la requête a réussi (code HTTP 200)
if r.status_code == 200:
    # Récupérer la réponse JSON
    data = r.json()

    print(f"Statut HTTP: {r.status_code}")
    print("Contenu de la réponse :", r.text)

# Enregistrer la réponse JSON dans un fichier
    file = '..\gitignore\dataforgood_bechdelai.json'
    with open(file, 'w', encoding='utf-8') as f:
        json.dump(data, f, ensure_ascii=False, indent=4)
```

```
    print("Données enregistrées dans 'gitignore\dataforgood_bechdelai.json'")
else:
    print(f"Erreur lors de la requête : {response.status_code}")
```

```
<>:20: SyntaxWarning: invalid escape sequence '\g'
<>:24: SyntaxWarning: invalid escape sequence '\d'
<>:20: SyntaxWarning: invalid escape sequence '\g'
<>:24: SyntaxWarning: invalid escape sequence '\d'
C:\Users\jpvt\AppData\Local\Temp\ipykernel_8840\868025735.py:20: SyntaxWarning: i
nvalid escape sequence '\g'
  file = '..\gitignore\dataforgood_bechdelai.json'
C:\Users\jpvt\AppData\Local\Temp\ipykernel_8840\868025735.py:24: SyntaxWarning: i
nvalid escape sequence '\d'
  print("Données enregistrées dans 'gitignore\dataforgood_bechdelai.json'")
```

```
[{'rating': 0,
  'id': 9602,
  'imdbid': '3155794',
  'title': 'Passage de Venus',
  'year': 1874},
 {'rating': 0,
  'imdbid': '14495706',
  'id': 9804,
  'title': 'La Rosace Magique',
  'year': 1877},
 {'rating': 0,
  'id': 9603,
  'imdbid': '2221420',
  'title': 'Sallie Gardner at a Gallop',
  'year': 1878},
 {'year': 1878,
  'title': 'Le singe musicien',
  'imdbid': '12592084',
  'id': 9806,
  'rating': 0},
 {'year': 1881,
  'title': 'Athlete Swinging a Pick',
  'id': 9816,
  'imdbid': '7816420',
  'rating': 0},
 {'rating': 0,
  'id': 9831,
  'imdbid': '5459794',
  'title': 'Buffalo Running',
  'year': 1883},
 {'rating': 0,
  'imdbid': '8588366',
  'id': 9832,
  'title': 'L&#39;homme machine',
  'year': 1885},
 {'title': 'Man Walking Around the Corner',
  'year': 1887,
  'rating': 0,
  'imdbid': '2075247',
  'id': 9614},
 {'title': 'Cockatoo Flying',
  'year': 1887,
  'rating': 0,
  'imdbid': '8133192',
  'id': 9836},
 {'rating': 0,
  'id': 9837,
  'imdbid': '7411790',
  'title': 'Child Carrying Flowers to Woman',
  'year': 1887},
 {'title': 'Jumping Over a Man&#39;s Back-Leapfrog',
  'year': 1887,
  'rating': 0,
  'id': 9838,
  'imdbid': '7541160'},
 {'title': 'Man Riding Jumping Horse',
  'year': 1887,
  'rating': 0,
  'imdbid': '7754902',
  'id': 9841},
```

	rating	id	imdbid	title	year
0	0	9602	3155794	Passage de Venus	1874
1	0	9804	14495706	La Rosace Magique	1877
2	0	9603	2221420	Sallie Gardner at a Gallop	1878
3	0	9806	12592084	Le singe musicien	1878
4	0	9816	7816420	Athlete Swinging a Pick	1881
...	...	...	...	...	...
10442	3	11507	27410895	Let go	2024
10443	1	11508	9218128	Gladiator II	2024
10444	3	11509	1262426	Wicked: Part 1	2024
10445	2	11510	31807233	Her story	2024
10446	1	11513	24176060	Queer	2024

10447 rows × 5 columns

In [4]:
```
# 3. Aperçu initial des données :

print(data.head())  # Affiche les 5 premières lignes
print(data.info())  # Donne des informations sur les types de colonnes et les va
print(data.describe())  # Statistiques descriptives basiques (numériques uniquem
```

```
   rating    id    imdbid                       title  year
0       0  9602   3155794            Passage de Venus  1874
1       0  9804  14495706           La Rosace Magique  1877
2       0  9603   2221420  Sallie Gardner at a Gallop  1878
3       0  9806  12592084           Le singe musicien  1878
4       0  9816   7816420     Athlete Swinging a Pick  1881
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10447 entries, 0 to 10446
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   rating  10447 non-null  int64 
 1   id      10447 non-null  int64 
 2   imdbid  10447 non-null  object
 3   title   10447 non-null  object
 4   year    10447 non-null  int64 
dtypes: int64(3), object(2)
memory usage: 408.2+ KB
None
             rating            id          year
count  10447.000000  10447.000000  10447.000000
mean       2.135733   5586.786254   1997.041256
std        1.098775   3276.820216     25.077329
min        0.000000      1.000000   1874.000000
25%        1.000000   2722.500000   1989.000000
50%        3.000000   5539.000000   2006.000000
75%        3.000000   8430.500000   2014.000000
max        3.000000  11513.000000   2024.000000
```

Étape 2 : Inspection des Données

In [5]:
```python
# 1. Vérification des dimensions :

print(data.shape)  # Nombre de lignes et de colonnes

# 2. Recherche des valeurs manquantes :

print(f"Valeurs manquantes : {data.isnull().sum()}")  # Total des valeurs manqua
print(f"Valeurs manquantes : {data.isna().sum()}")  # Total des valeurs manquant
```

```
(10447, 5)
Valeurs manquantes : rating    0
id        0
imdbid    0
title     0
year      0
dtype: int64
Valeurs manquantes : rating    0
id        0
imdbid    0
title     0
year      0
dtype: int64
```

In [6]:
```python
# 3. Identification des doublons :

# Doublons de lignes entières
print(data.duplicated().sum())  # Nombre de lignes dupliquées
```

```
0
```

In [7]:
```python
# Doublons sur identifiants :
print(f"Valeurs uniques : {data["imdbid"].unique()}")
print(f"Nombre de ligne par valeur : {data["imdbid"].value_counts()}")

# Récupérer les valeurs pour lesquelles value_counts == 2
imdbid_counts = data["imdbid"].value_counts()
imdbid_with_two_occurrences = imdbid_counts[imdbid_counts >= 2].index.tolist()

# Afficher la liste des valeurs correspondantes
print(f"Liste des valeurs apparaissant 2 fois ou plus : {imdbid_with_two_occurre
```

```
Valeurs uniques : ['3155794' '14495706' '2221420' ... '1262426' '31807233' '24176
060']
Nombre de ligne par valeur : imdbid
2043900     2
0117056     2
            2
0035279     2
2457282     2
           ..
0120461     1
0118842     1
0119859     1
0120493     1
24176060    1
Name: count, Length: 10441, dtype: int64
Liste des valeurs apparaissant 2 fois ou plus : ['2043900', '0117056', '', '00352
79', '2457282', '0086425']
```

In [8]:
```python
# Focus sur les lignes avec id identique

focus = data[data["imdbid"]=="2457282"]
print(focus)
print(focus["title"][7511])
print(focus["title"][7523])
```

```
      rating    id   imdbid  \
7517       3  4980  2457282
7529       3  5026  2457282

                                                 title  year
7517  Puella Magi Madoka Magica the Movie Part III: ...  2013
7529                    Madoka Magica: Rebellion Story  2013
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
File d:\BigPapaProject\Anaconda\Lib\site-packages\pandas\core\indexes\base.py:3805, in Index.get_loc(self, key)
   3804 try:
-> 3805     return self._engine.get_loc(casted_key)
   3806 except KeyError as err:

File index.pyx:167, in pandas._libs.index.IndexEngine.get_loc()

File index.pyx:196, in pandas._libs.index.IndexEngine.get_loc()

File pandas\\_libs\\hashtable_class_helper.pxi:2606, in pandas._libs.hashtable.Int64HashTable.get_item()

File pandas\\_libs\\hashtable_class_helper.pxi:2630, in pandas._libs.hashtable.Int64HashTable.get_item()

KeyError: 7511

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Cell In[8], line 5
      3 focus = data[data["imdbid"]=="2457282"]
      4 print(focus)
----> 5 print(focus["title"][7511])
      6 print(focus["title"][7523])

File d:\BigPapaProject\Anaconda\Lib\site-packages\pandas\core\series.py:1121, in Series.__getitem__(self, key)
   1118     return self._values[key]
   1120 elif key_is_scalar:
-> 1121     return self._get_value(key)
   1123 # Convert generator to list before going through hashable part
   1124 # (We will iterate through the generator there to check for slices)
   1125 if is_iterator(key):

File d:\BigPapaProject\Anaconda\Lib\site-packages\pandas\core\series.py:1237, in Series._get_value(self, label, takeable)
   1234     return self._values[label]
   1236 # Similar to Index.get_value, but we do not fall back to positional
-> 1237 loc = self.index.get_loc(label)
   1239 if is_integer(loc):
   1240     return self._values[loc]

File d:\BigPapaProject\Anaconda\Lib\site-packages\pandas\core\indexes\base.py:3812, in Index.get_loc(self, key)
   3807     if isinstance(casted_key, slice) or (
   3808         isinstance(casted_key, abc.Iterable)
   3809         and any(isinstance(x, slice) for x in casted_key)
   3810     ):
   3811         raise InvalidIndexError(key)
-> 3812     raise KeyError(key) from err
   3813 except TypeError:
   3814     # If we have a listlike key, _check_indexing_error will raise
   3815     #  InvalidIndexError. Otherwise we fall through and re-raise
   3816     #  the TypeError.
   3817     self._check_indexing_error(key)
```

```
KeyError: 7511
```

In [ ]:
```
# Focus sur les lignes avec id ''

focus = data[data["imdbid"]==""]
print(focus)
```

```
       year                  title     id imdbid  rating
5846   2008                 Machan  11315              1
10383  2024  A Little Family Drama  11379              3
```

Actions à mettre en oeuvre

- supprimer les doublons d'imdbid
- supprimer les lignes qui ont un imdbid = ''
- changer le format du 'imdbid' pour avoir tt000000

In [ ]:
```
# Supprimer les lignes avec doublons sur la colonne 'imdbid' (garder la première

data_bechdel_clean = data.drop_duplicates(subset="imdbid", keep='first')

# Supprimer les lignes avec valeurs '' sur la colonne 'imdbid'

data_bechdel_clean = data_bechdel_clean[data_bechdel_clean["imdbid"] != '']

# Changer le format du 'imdbid' pour avoir tt000000

data_bechdel_clean["imdbid"] = 'tt' + data_bechdel_clean["imdbid"]

# Suppression des colonnes 'year', 'title', 'id'

data_bechdel_clean = data_bechdel_clean.drop(['year', 'title', 'id'],axis=1)
```

In [ ]:
```
# Voir le résultat :

display(data_bechdel_clean)
```

	imdbid	rating
0	tt3155794	0
1	tt14495706	0
2	tt2221420	0
3	tt12592084	0
4	tt7816420	0
...	...	...
10403	tt15574270	3
10404	tt33343397	3
10405	tt21187072	3
10406	tt17526714	1
10407	tt27489557	3

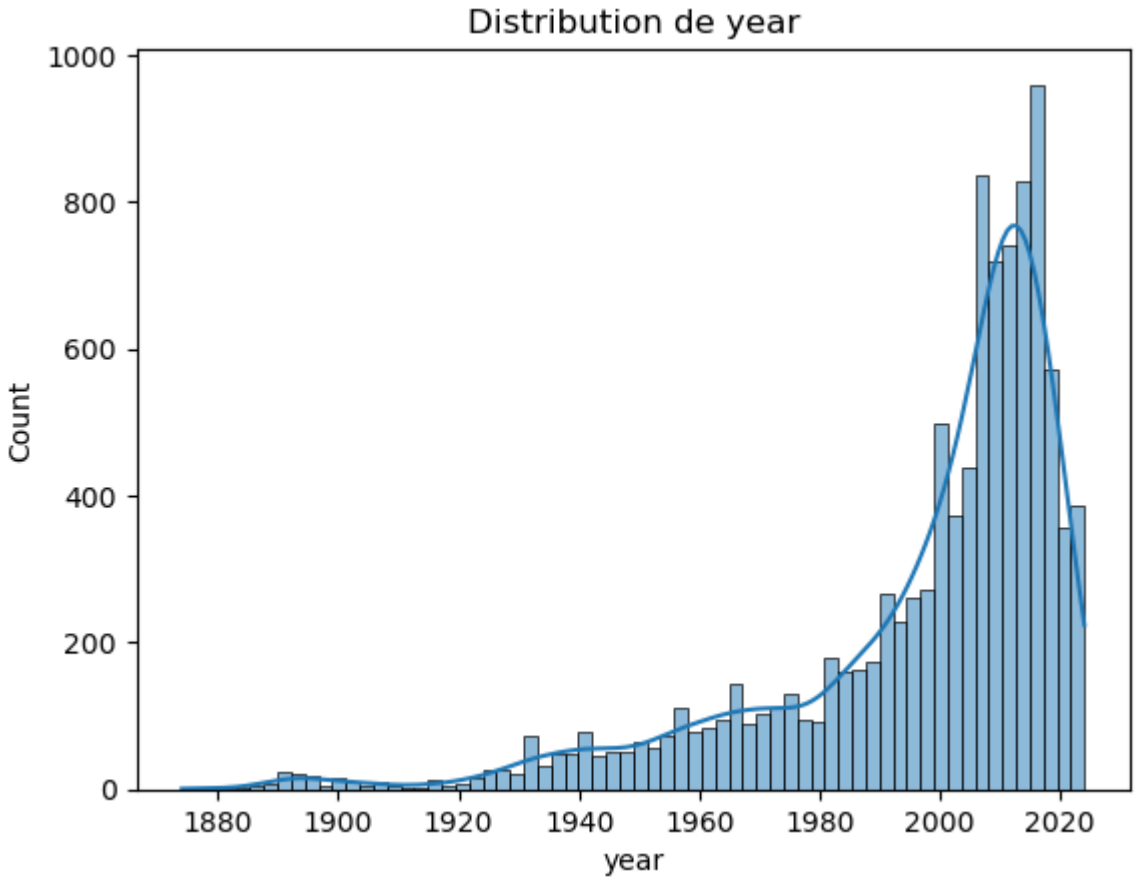
10401 rows × 2 columns

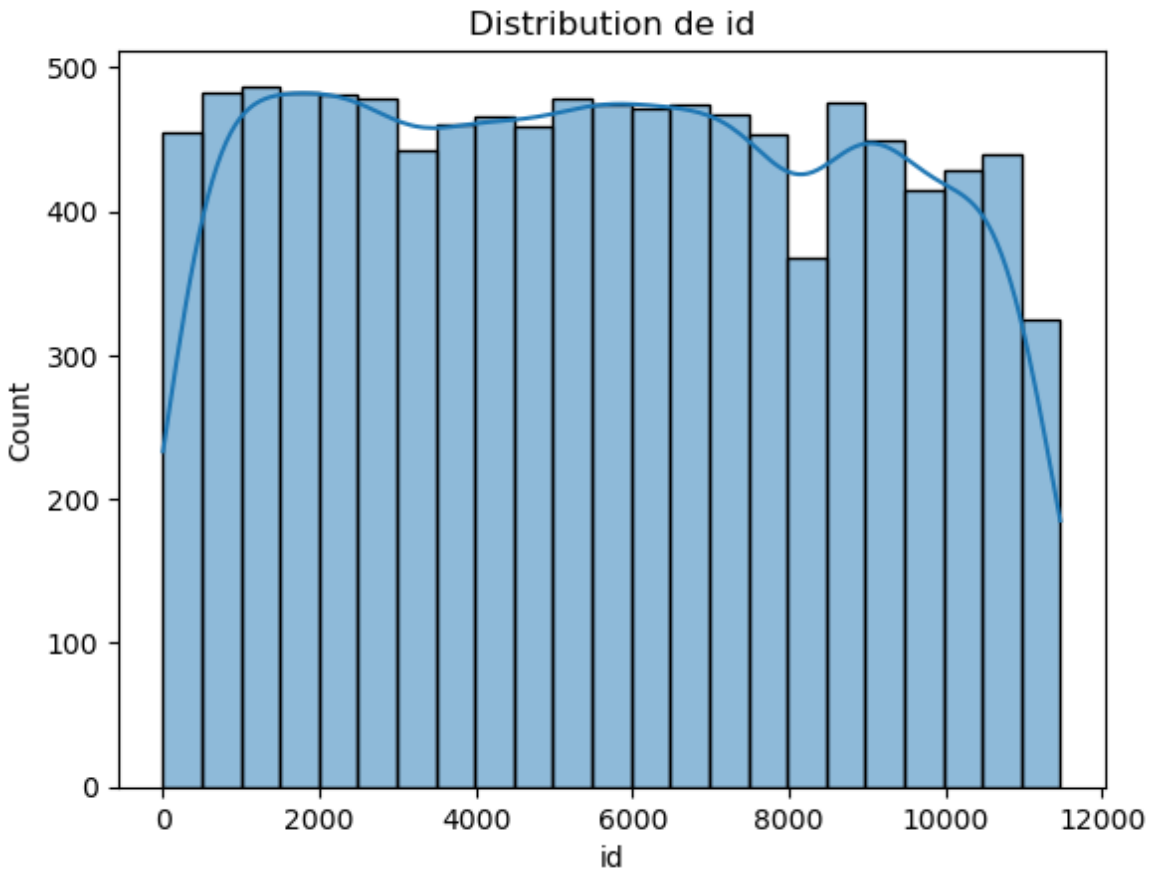
Étape 3 : Analyse des Variables

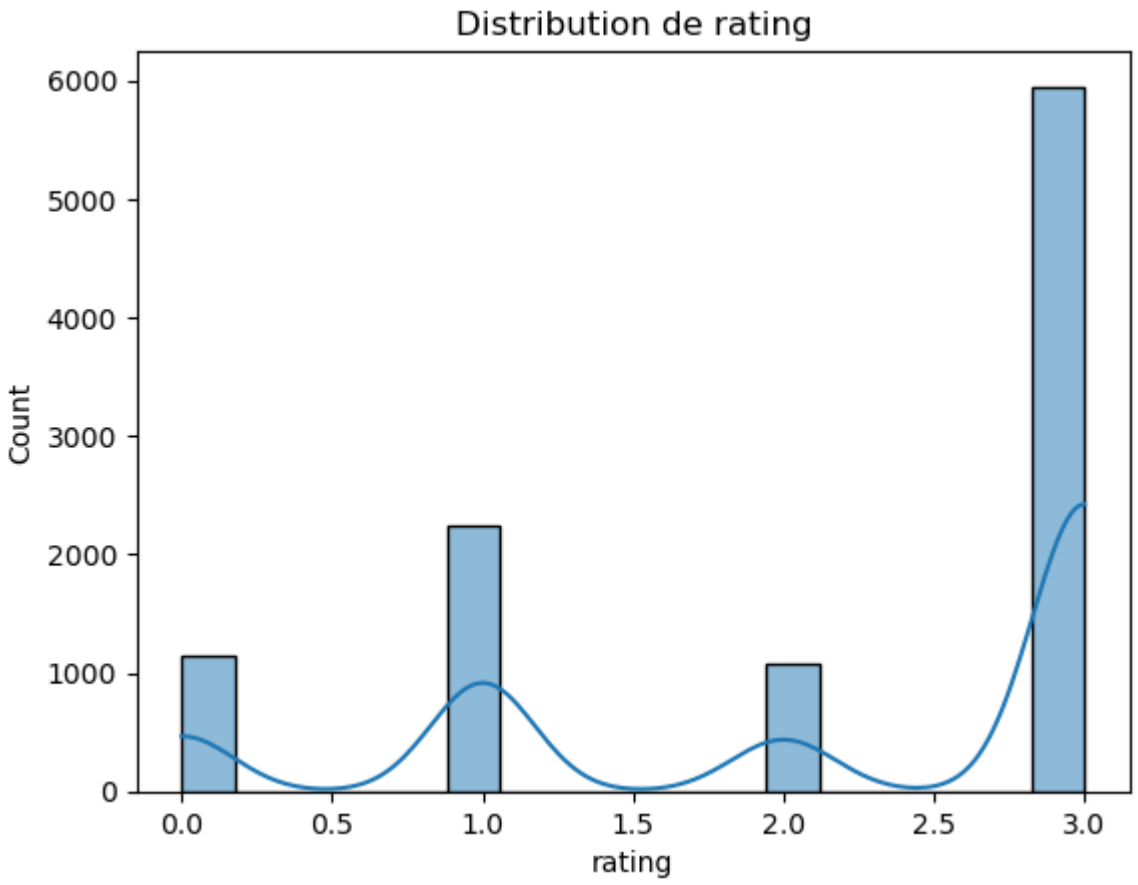
In [ ]:
```
# 2. Variables numériques :

# Statistiques de base et distribution :

for col in data.select_dtypes(include=['int64', 'float64']):
    sns.histplot(data[col], kde=True)
    plt.title(f"Distribution de {col}")
    plt.show()
```


Distribution de year
Count
year
0
200
400
600
800
1000
1880
1900
1920
1940
1960
1980
2000
2020


Distribution de id
Count
id
0
100
200
300
400
500
0
2000
4000
6000
8000
10000
12000



In [ ]:
```
# Exporter en CSV

data_bechdel_clean.to_csv('../gitignore/data_bechdel.csv', index=False, encoding
```