

IMDb

<https://bit.ly/2TXtBhc>

เว็บ IMDb รวบรวมข้อมูลเกี่ยวกับภาพยนตร์ต่างๆ และตัดข้อมูลบางส่วนมาไว้ใช้ทำการทดลองต่างๆ ได้ <https://www.imdb.com/interfaces/>

ไจอห์นนี่คัดกรอง [ข้อมูลบางส่วนใน IMDb](#) มา โดยโดยมีไฟล์นี้อยู่

filteredCast.tsv

- tconst (string) - alphanumeric unique identifier of the title
- ordering (integer) – a number to uniquely identify rows for a given titleid
- nconst (string) - alphanumeric unique identifier of the name/person
- category (string) - the category of job that person was in
- job (string) - the specific job title if applicable, else '\N'
- characters (string) - the name of the character played if applicable, else '\N'

filteredTitle.tsv

- tconst (string) - alphanumeric unique identifier of the title
- titleType (string) – the type/format of the title (e.g. movie, short, tvseries, tvepisode, video, etc)
- primaryTitle (string) – the more popular title / the title used by the filmmakers on promotional materials at the point of release
- originalTitle (string) - original title, in the original language
- isAdult (boolean) - 0: non-adult title; 1: adult title
- startYear (YYYY) – represents the release year of a title. In the case of TV Series, it is the series start year
- endYear (YYYY) – TV Series end year. '\N' for all other title types
- runtimeMinutes – primary runtime of the title, in minutes
- genres (string array) – includes up to three genres associated with the title

filteredStar.tsv


- nconst (string) - alphanumeric unique identifier of the name/person
- primaryName (string)– name by which the person is most often credited
- birthYear – in YYYY format
- deathYear – in YYYY format if applicable, else '\N'
- primaryProfession (array of strings)– the top-3 professions of the person
- knownForTitles (array of tconsts) – titles the person is known for

ไจอห์นนี่


1. มีภาพยนตร์กี่ประเภท (genres) แต่ละประเภทมีอย่างละกี่เรื่อง พร้อมพล็อตกราฟ
2. มีนักแสดงชายและหญิงอย่างละกี่คน(นับจากไฟล์ filteredStar)
3. นักแสดงที่ยังมีชีวิตมีอายุเท่าไรวบ้าง แต่ละช่วงมีกี่คน
4. มีนักแสดงกี่คนที่เคยแสดงในภาพยนตร์ Action
5. มีนักแสดงกี่คนที่แสดงในภาพยนตร์มากกว่า 1 ประเภท
6. มีภาพยนตร์ทั้งหมดกี่เรื่องที่เข้าฉายในปีอธิกสุรทิน
7. มีผู้กำกับกี่คนที่เป็นนักแสดงด้วย
8. หาจำนวนภาพยนตร์แนวโรแมนติกในแต่ละปี พร้อมพล็อตกราฟ(นับเฉพาะปีที่มีตั้งแต่ 1 เรื่องขึ้นไป)

▼ 1. มีภาพยนตร์กี่ประเภท (genres) แต่ละประเภทมีอย่างละกี่เรื่อง พร้อมพล็อตกราฟ


```
1 #solution
2 from google.colab import files
3 files.upload()
4 files.upload()
5 files.upload()
```

 Choose Files | filteredCast-v2.tsv

- **filteredCast-v2.tsv**(text/tab-separated-values) - 523963 bytes, last modified: 1/7/2020 - 100% done
Saving filteredCast-v2.tsv to filteredCast-v2.tsv

 Choose Files | filteredStar-v2.tsv


- **filteredStar-v2.tsv**(text/tab-separated-values) - 717387 bytes, last modified: 1/7/2020 - 100% done
Saving filteredStar-v2.tsv to filteredStar-v2.tsv

 Choose Files | filteredTitle-v2.tsv

- **filteredTitle-v2.tsv**(text/tab-separated-values) - 491899 bytes, last modified: 1/7/2020 - 100% done
Saving filteredTitle-v2.tsv to filteredTitle-v2.tsv

```
{'filteredTitle-v2.tsv': b'tconst\ttitleType\tprimaryTitle\toriginalTitle\tisAdult\tstartYear\tendYear\truntimeMinutes\tgenres\r\r\ntt0192789\tmovie\tWhile Supplie
```

```
1 pip install seaborn
```

 Requirement already satisfied: seaborn in /usr/local/lib/python3.6/dist-packages (0.10.1)

Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib/python3.6/dist-packages (from seaborn) (1.18.5)

Requirement already satisfied: matplotlib>=2.1.2 in /usr/local/lib/python3.6/dist-packages (from seaborn) (3.2.2)

Requirement already satisfied: pandas>=0.22.0 in /usr/local/lib/python3.6/dist-packages (from seaborn) (1.0.5)

Requirement already satisfied: scipy>=1.0.1 in /usr/local/lib/python3.6/dist-packages (from seaborn) (1.4.1)

Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!>=2.1.6,>=2.0.1 in /usr/local/lib/python3.6/dist-packages (from matplotlib>=2.1.2->seaborn) (2.4.7)

Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.6/dist-packages (from matplotlib>=2.1.2->seaborn) (1.2.0)

Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.6/dist-packages (from matplotlib>=2.1.2->seaborn) (0.10.0)

Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.6/dist-packages (from matplotlib>=2.1.2->seaborn) (2.8.1)

Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.6/dist-packages (from pandas>=0.22.0->seaborn) (2018.9)

Requirement already satisfied: six in /usr/local/lib/python3.6/dist-packages (from cycler>=0.10->matplotlib>=2.1.2->seaborn) (1.12.0)

```
1 import pandas as pd
2 import seaborn as sns
3 import matplotlib.pyplot as plt
```

```
4
5 casts = pd.read_csv("filteredCast-v2.tsv", delimiter='\t')
6 titles = pd.read_csv('filteredTitle-v2.tsv', delimiter='\t')
7 stars = pd.read_csv('filteredStar-v2.tsv', delimiter='\t')
```

```
1 titles.head(5)
```

	tconst	titleType	primaryTitle	originalTitle	isAdult	startYear	endYear	runtimeMinutes	genres
0	tt0192789	movie	While Supplies Last	While Supplies Last	0	2002	∅	120	Comedy,Musical
1	tt4914592	movie	Electric Heart	Electric Heart	0	2017	∅	75	Adventure,Drama,Music
2	tt4999994	movie	Rain Doll	Rain Doll	0	2016	∅	115	Drama
3	tt2690572	movie	The Blessed Ones	Polaris	0	2017	∅	79	Drama

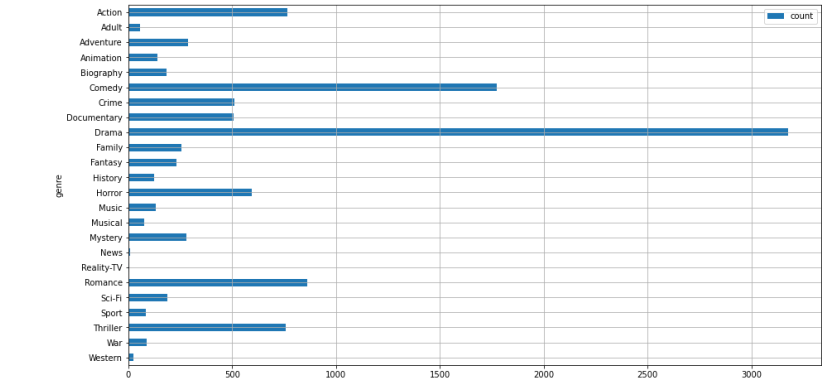
```
1 titles_genres = titles['genres'].replace('\n', None)
```

```
1 check_genre = titles_genres.str.split(',', expand=True)
```

```
1 from collections import Counter
2 count_genres = check_genre[0].append([check_genre[1], check_genre[2]])
3
4 x = pd.DataFrame.from_dict(Counter(count_genres), orient='index').reset_index()
5 x = x.rename(columns={'index' : 'genre', 0 : 'count'})
6 x = x.dropna(axis=0)
7
8 count_genres = x.groupby(x['genre']).agg('sum')
9
10 print('ภาพยนตร์ทั้งหมดมี ', len(count_genres), ' ประเภท\n')
11 print('โดยแบ่งได้ดังนี้')
12 for row, column in count_genres.iterrows():
13     print(row, ' : ', column['count'], ' เรื่อง')
14 print('\n')
15
16 count_genres.plot(kind='barh', figsize=(15, 8), legend=True, grid=True).invert_yaxis()
17
18 #sns.barplot(count_genres.index, 'count', data=count_genres)
19 #plt.show()
20
```

ภาพยนตร์ทั้งหมดมี 24 ประเภท

โดยแบ่งได้ดังนี้
Action : 764 เรื่อง
Adult : 56 เรื่อง
Adventure : 289 เรื่อง
Animation : 139 เรื่อง
Biography : 185 เรื่อง
Comedy : 1773 เรื่อง
Crime : 509 เรื่อง
Documentary : 506 เรื่อง
Drama : 3177 เรื่อง
Family : 257 เรื่อง
Fantasy : 232 เรื่อง
History : 124 เรื่อง
Horror : 596 เรื่อง
Music : 133 เรื่อง
Musical : 74 เรื่อง
Mystery : 281 เรื่อง
News : 7 เรื่อง
Reality-TV : 3 เรื่อง
Romance : 863 เรื่อง
Sci-Fi : 186 เรื่อง
Sport : 83 เรื่อง
Thriller : 759 เรื่อง
War : 88 เรื่อง
Western : 23 เรื่อง



▼ 2. มีนักแสดงชายและหญิงอย่างละกี่คน(นับจากไฟล์ filteredStar)

```
1 #solution
2 profession = stars['primaryProfession'].str.split(',', expand=True)
3 count_profession = profession[0].append([profession[1], profession[2]])
4 count_profession = pd.DataFrame.from_dict(Counter(count_profession), orient='index').reset_index()
5 count_profession.rename(columns={'index' : 'profession', 0 : 'count'}, inplace=True)
6
7 for row, column in count_profession.iterrows():
8     if column['profession'] == 'actor' :
9         print('นักแสดงชายมีทั้งหมด : ', column['count'] , ' คน')
10    elif column['profession'] == 'actress' :
11        print('นักแสดงหญิงมีทั้งหมด : ', column['count'] , ' คน')
```

```
➤  นักแสดงหญิงมีทั้งหมด :   3537   คน
   นักแสดงชายมีทั้งหมด :   4300   คน
```

▼ 3. นักแสดงที่ยังมีชีวิตมีอายุเท่าไรบ้าง แต่ละช่วงมีกี่คน

```
1 #solution
2 age = stars.drop(axis=1, columns=['primaryProfession', 'knownForTitles'])
3 age['thisYear'] = 2020
4 age = age.replace({'\\N' : 0})
5 age['deathYear'] = pd.to_numeric(age['deathYear'], downcast='integer')
6
7 for row, column in age.iterrows():
8     if age.loc[row, 'deathYear'] == 0:
9         age.loc[row, 'Age'] = age.loc[row, 'thisYear'] - age.loc[row, 'birthYear']
10    else:
11        age.loc[row, 'Age'] = 'Death'
```

```
1 count_age = Counter(age['Age'])
2 count_age = pd.DataFrame.from_dict(count_age, orient='index')
3 count_age = count_age.drop(axis=0, index=['Death']).reset_index()
4 count_age.rename(columns={'index' : 'Age', 0 : 'count'}, inplace=True)
5 count_age['Age'] = pd.to_numeric(count_age['Age'], downcast='integer')
6 count_age.sort_values(by='Age', inplace=True)
7
8 print('ช่วงอายุที่นักแสดงยังมีชีวิตอยู่ :')
9 print(count_age[['Age']], '\n')
10
11 print('นักแสดงที่ยังมีชีวิตอยู่ ดังนี้')
12 for row, column in count_age.iterrows():
13     print('ช่วงอายุ', column['Age'], ' ปี มีจำนวนทั้งหมด', column['count'], ' คน')
```

```
➤  ช่วงอายุที่นักแสดงยังมีชีวิตอยู่ :
   Age
18   31
16   32
14   33
13   34
15   35
17   36
12   37
10   38
 6   39
 7   40
 3   41
11   42
 9   43
 8   44
 1   45
 0   46
 2   47
 4   48
 5   49

นักแสดงที่ยังมีชีวิตอยู่ ดังนี้
ช่วงอายุ 31 ปี มีจำนวนทั้งหมด 213 คน
ช่วงอายุ 32 ปี มีจำนวนทั้งหมด 233 คน
ช่วงอายุ 33 ปี มีจำนวนทั้งหมด 288 คน
ช่วงอายุ 34 ปี มีจำนวนทั้งหมด 328 คน
ช่วงอายุ 35 ปี มีจำนวนทั้งหมด 317 คน
ช่วงอายุ 36 ปี มีจำนวนทั้งหมด 365 คน
ช่วงอายุ 37 ปี มีจำนวนทั้งหมด 389 คน
ช่วงอายุ 38 ปี มีจำนวนทั้งหมด 438 คน
ช่วงอายุ 39 ปี มีจำนวนทั้งหมด 430 คน
ช่วงอายุ 40 ปี มีจำนวนทั้งหมด 448 คน
ช่วงอายุ 41 ปี มีจำนวนทั้งหมด 495 คน
ช่วงอายุ 42 ปี มีจำนวนทั้งหมด 460 คน
ช่วงอายุ 43 ปี มีจำนวนทั้งหมด 442 คน
ช่วงอายุ 44 ปี มีจำนวนทั้งหมด 463 คน
ช่วงอายุ 45 ปี มีจำนวนทั้งหมด 472 คน
ช่วงอายุ 46 ปี มีจำนวนทั้งหมด 447 คน
ช่วงอายุ 47 ปี มีจำนวนทั้งหมด 505 คน
ช่วงอายุ 48 ปี มีจำนวนทั้งหมด 511 คน
ช่วงอายุ 49 ปี มีจำนวนทั้งหมด 498 คน
```

▼ 4. มีนักแสดงกี่คนที่เคยแสดงในภาพยนตร์ Action

```
1 #solution
2 action_movie = titles.drop(axis=1, columns=['isAdult', 'startYear', 'endYear', 'runtimeMinutes'])
3 action_movie['genre1'] = check_genre[0]
4 action_movie['genre2'] = check_genre[1]
5 action_movie['genre3'] = check_genre[2]
6 action_movie = action_movie.loc[(action_movie['genre1'] == 'Action') | (action_movie['genre2'] == 'Action') | (action_movie['genre3'] == 'Action')]
7 action_movie_list = action_movie['tconst']
```

```
1 action_cast = casts[casts['tconst'].isin(action_movie_list)]
2 action_cast_list = action_cast['nconst']
```

```
1 action_stars = stars[stars['nconst'].isin(action_cast_list)]
2
3 print('จำนวนภาพยนตร์ Action : ', len(action_movie), ' เรื่อง')
4 print('จำนวนนักแสดงที่เคยแสดงในภาพยนตร์ Action : ', len(action_stars), ' คน')
```

จำนวนภาพยนตร์ Action : 764 เรื่อง
จำนวนนักแสดงที่เคยแสดงในภาพยนตร์ Action : 1137 คน

5. มีนักแสดงกี่คนที่แสดงในภาพยนตร์มากกว่า 1 ประเภท

```
1 titles_genre = titles.drop(axis=1, columns=['titleType', 'primaryTitle', 'originalTitle', 'genres', 'isAdult', 'startYear', 'endYear', 'runtimeMinutes'])
2 titles_genre['genre1'] = check_genre[0]
3 titles_genre['genre2'] = check_genre[1]
4 titles_genre['genre3'] = check_genre[2]
5 titles_genre = check1_genre.replace({None : 'unknown'})
6 titles_genre.sample(5)
```

	tconst	genre1	genre2	genre3
2311	tt2094195	Crime	Drama	Mystery
308	tt0120002	Comedy	unknown	unknown
1347	tt4199898	Adventure	Animation	Comedy
5581	tt4209744	Romance	unknown	unknown
2190	tt2134058	Comedy	Romance	unknown

```
1 movies = pd.merge(casts, titles_genre, how='left', left_on=casts['tconst'], right_on=titles_genre['tconst'], suffixes=('_casts', '_titles'))
2 movies_morethan1 = movies.loc[x['genre2'] != 'unknown']
3
4 movies_morethan1.sample(5)
```

	key_0	tconst_casts	ordering	nconst	category	job	characters	tconst_titles	genre1	genre2	genre3
4804	tt1561433	tt1561433	5	nm1446064	director	W	W	tt1561433	Comedy	Musical	unknown
6929	tt2757228	tt2757228	3	nm4888185	actress	W	["Nancy"]	tt2757228	Action	Drama	Mystery
8053	tt4124122	tt4124122	1	nm1932988	actor	W	["Uji"]	tt4124122	Crime	Drama	unknown
603	tt0205843	tt0205843	3	nm0423307	actor	W	["Beldar"]	tt0205843	Adventure	Drama	Fantasy
13	tt0104382	tt0104382	2	nm0674338	actress	W	["Hanna"]	tt0104382	Comedy	Drama	Romance

```
1 print('จำนวนนักแสดงที่แสดงในภาพยนตร์มากกว่า 1 ประเภท : ', len(movies_morethan1['nconst'].unique()), 'คน')
```

จำนวนนักแสดงที่แสดงในภาพยนตร์มากกว่า 1 ประเภท : 4548 คน

6. มีภาพยนตร์ทั้งหมดกี่เรื่องที่เข้าฉายในปีอธิกสุรทิน

```
1 #solution
2 leap_year = titles.drop(axis=1, columns=['isAdult', 'endYear', 'runtimeMinutes', 'genres'])
3
4 for row, column in leap_year.iterrows() :
5     if (leap_year.loc[row, 'startYear'] % 4 == 0 & ((leap_year.loc[row, 'startYear'] % 100 != 0) | (leap_year.loc[row, 'startYear'] % 400 == 0))):
6         leap_year.loc[row, 'isLeapYear'] = 1
7     else :
8         leap_year.loc[row, 'isLeapYear'] = 0
```

```
1 print('จำนวนภาพยนตร์ที่เข้าฉายในปีอธิกสุรทิน', Counter(leap_year['isLeapYear'])[1], ' เรื่อง')
```

จำนวนภาพยนตร์ที่เข้าฉายในปีอธิกสุรทิน 1504 เรื่อง

7. มีผู้กำกับกี่คนที่เป็นนักแสดงด้วย

```
1 #solution
2 director = stars.drop(axis=1, columns=['birthYear', 'deathYear', 'knownForTitles'])
3 director['role1'] = profession[0]
4 director['role2'] = profession[1]
5 director['role3'] = profession[2]
6 director_list = director.loc[(director['role1'] == 'director') | (director['role2'] == 'director') | (director['role3'] == 'director')]
```

```
1 print('จำนวนผู้กำกับที่เป็นนักแสดง ', len(director_list), 'คน')
```

จำนวนผู้กำกับที่เป็นนักแสดง 1284 คน

8. หาจำนวนภาพยนตร์แนวโรแมนติกในแต่ละปี พร้อมพล็อตกราฟ(นับเฉพาะปีที่มีตั้งแต่ 1 เรื่องขึ้นไป)

```
1 #solution
2 romcom_movie = titles.drop(axis=1, columns=['isAdult', 'endYear', 'runtimeMinutes', 'tconst', 'titleType', 'primaryTitle', 'originalTitle'])
```

```
3 romcom_check = pd.Series(romcom_movie['genres']).str.contains('Romance,Comedy|Comedy,Romance',regex=True, case=True)
4 romcom_movie['checkRomCom'] = romcom_check
5 romcom_plot = romcom_movie.groupby(romcom_movie['startYear']).agg('sum')
6
7 print('ปีที่มีจำนวนภาพยนตร์แนวโรแมนติกตั้งแต่ 1 เรื่องขึ้นไป')
8 print(romcom_plot[romcom_plot['checkRomCom'] >= 1])
```

ปีที่มีจำนวนภาพยนตร์แนวโรแมนติกตั้งแต่ 1 เรื่องขึ้นไป

checkRomCom	
startYear	
1991	1.0
1992	1.0
1993	1.0
1994	1.0
1995	2.0
1996	5.0
1997	2.0
1998	7.0
1999	3.0
2000	2.0
2001	3.0
2002	5.0
2003	5.0
2004	3.0
2005	6.0
2006	3.0
2007	8.0
2008	3.0
2009	4.0
2010	16.0
2011	14.0
2012	6.0
2013	5.0
2014	12.0
2015	7.0
2016	11.0
2017	5.0
2018	5.0
2019	7.0
2020	4.0

```
1 print('กราฟแท่งแสดงปีที่มีจำนวนภาพยนตร์แนวโรแมนติกตั้งแต่ 1 เรื่องขึ้นไป\n')
2 romcom_plot[romcom_plot['checkRomCom'] >= 1].plot(kind='barh', figsize=(15, 10), legend=True, grid=True).invert_yaxis()
```

กราฟแท่งแสดงปีที่มีจำนวนภาพยนตร์แนวโรแมนติกตั้งแต่ 1 เรื่องขึ้นไป

