pkiar / **Phase-2-Film-Production-Project**

Code Issues Pull requests Actions Projects Wiki Security

Phase-2-Film-Production-Project / index.ipynb

pkiar Update index file with conflict 9330f11 · yesterday

2916

P

Raw

Switch branches/tags

Find or create a branch...

Branches Tags

main default

EOkeyo

Kossie

Wangari

geoffrey

✓ pkiarie

View all branches

In [38]:

```
# Reading into the csv files to clean and aggregate the columns we need
import pandas as pd
bom = pd.read_csv("bom.movie_gross.csv")

bom.columns
```

Out[38]:

```
Index(['title', 'studio', 'domestic_gross', 'foreign_gross', 'year'], d
type='object')
```

In [39]:

```
bom
```

Out[39]:

	title	studio	domestic_gross	foreign_gross	year
0	Toy Story 3	BV	415000000.0	652000000	2010
1	Alice in Wonderland (2010)	BV	334200000.0	691300000	2010
2	Harry Potter and the Deathly Hallows Part 1	WB	296000000.0	664300000	2010
3	Inception	WB	292600000.0	535700000	2010
4	Shrek Forever After	P/DW	238700000.0	513900000	2010
...	...	...	...	...	...
3382	The Quake	Magn.	6200.0	NaN	2018
3383	Edward II (2018 re-release)	FM	4800.0	NaN	2018
3384	El Pacto	Sony	2500.0	NaN	2018
3385	The Swan	Synergetic	2400.0	NaN	2018
3386	An Actor Prepares	Grav.	1700.0	NaN	2018

3387 rows × 5 columns

In [40]:

```
bom.describe()
```

Out[40]:

	domestic_gross	year
count	3.359000e+03	3387.000000
mean	2.874585e+07	2013.958075
std	6.698250e+07	2.478141
min	1.000000e+02	2010.000000
25%	1.200000e+05	2012.000000
50%	1.400000e+06	2014.000000
75%	2.790000e+07	2016.000000
max	9.367000e+08	2018.000000

In [41]:

```
bom.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3387 entries, 0 to 3386
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   title           3387 non-null   object 
 1   studio          3382 non-null   object 
 2   domestic_gross  3359 non-null   float64
 3   foreign_gross   2037 non-null   object 
 4   year            3387 non-null   int64  
dtypes: float64(1), int64(1), object(3)
memory usage: 132.4+ KB
```

In [42]:

```
bom['foreign_gross'] = pd.to_numeric(bom['foreign_gross'], errors='coer
```

In [43]:

```
bom['foreign_gross'].astype(float)
```

Out[43]:

```
0       652000000.0
1       691300000.0
2       664300000.0
3       535700000.0
4       513900000.0
           ...     
3382            NaN
3383            NaN
3384            NaN
3385            NaN
3386            NaN
Name: foreign_gross, Length: 3387, dtype: float64
```

In [44]:

```
bom.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3387 entries, 0 to 3386
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   title           3387 non-null   object 
 1   studio          3382 non-null   object 
 2   domestic_gross  3359 non-null   float64
 3   foreign_gross   2032 non-null   float64
 4   year            3387 non-null   int64  
dtypes: float64(2), int64(1), object(2)
memory usage: 132.4+ KB
```

In [45]:

```
#Check missing rows in the bomdf
bom.isna().sum()
```

Out[45]:

```
title                0
studio               5
domestic_gross      28
foreign_gross     1355
```

```
year                     0
dtype: int64
```

In [46]:

```
bom['foreign_gross'] = bom['foreign_gross'].fillna(bom['foreign_gross']
bom['domestic_gross'] = bom['domestic_gross'].fillna(bom['domestic_gros
```

In [47]:

```
bom.isna().sum()
```

Out[47]:

```
title             0
studio            5
domestic_gross    0
foreign_gross     0
year              0
dtype: int64
```

In [48]:

```
bom=bom.dropna()
```

In [49]:

```
bom['studio'].isna().sum()
```

Out[49]:

```
0
```

In [50]:

```
bom.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 3382 entries, 0 to 3386
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   title           3382 non-null   object 
 1   studio          3382 non-null   object 
 2   domestic_gross  3382 non-null   float64
 3   foreign_gross   3382 non-null   float64
 4   year            3382 non-null   int64  
dtypes: float64(2), int64(1), object(2)
memory usage: 158.5+ KB
```

This dataset will answer question to do with revenue ?

1. What movie will make most of the money ? correlation between year and domestic gross or foreign gross vs year or studio

In [51]:

```
import sqlite3

conn = sqlite3.connect("im.db")

df2 = """
         SELECT name
         FROM SQLITE_MASTER
      """
pd.read_sql_query(df2,conn)
```

Out[51]:

name

In [52]:

```
col_0 = pd.read_sql("PRAGMA table_info(movie_basics);", conn)
col_1 = pd.read_sql("PRAGMA table_info(directors);", conn)
col_2 = pd.read_sql("PRAGMA table_info(known_for);", conn)
col_3 = pd.read_sql("PRAGMA table_info(movie_akas);", conn)
col_4 = pd.read_sql("PRAGMA table_info(movie_ratings);", conn)
col_5 = pd.read_sql("PRAGMA table_info(persons);", conn)
col_6 = pd.read_sql("PRAGMA table_info(principals);", conn)
col_7 = pd.read_sql("PRAGMA table_info(writers);", conn)

col_0,col_1,col_2,col_3,col_4,col_5,col_6,col_7
```

Out[52]:

```
(Empty DataFrame
 Columns: [cid, name, type, notnull, dflt_value, pk]
 Index: [],
 Empty DataFrame
 Columns: [cid, name, type, notnull, dflt_value, pk]
 Index: [],
 Empty DataFrame
 Columns: [cid, name, type, notnull, dflt_value, pk]
 Index: [],
 Empty DataFrame
 Columns: [cid, name, type, notnull, dflt_value, pk]
 Index: [],
 Empty DataFrame
 Columns: [cid, name, type, notnull, dflt_value, pk]
 Index: [],
 Empty DataFrame
 Columns: [cid, name, type, notnull, dflt_value, pk]
 Index: [],
 Empty DataFrame
 Columns: [cid, name, type, notnull, dflt_value, pk]
 Index: [],
 Empty DataFrame
 Columns: [cid, name, type, notnull, dflt_value, pk]
 Index: [])
```

1. Does genre make movie to have a higher rating ?

   genre vs ratings

2. Does movie the language affect movie ratings >?

   language vs ratings

In [53]:

```
df3 = pd.read_table("rt.movie_info.tsv")
df3.head()
```

Out[53]:

	id	synopsis	rating	genre	director	writer	th
0	1	This gritty, fast-paced, and innovative police...	R	Action and Adventure|Classics|Drama	William Friedkin	Ernest Tidyman	(

1	3	New York City, not-too-distant-future: Eric Pa...	R	Drama|Science Fiction and Fantasy	David Cronenberg	David Cronenberg|Don DeLillo	Au
2	5	Illeana Douglas delivers a superb performance ...	R	Drama|Musical and Performing Arts	Allison Anders	Allison Anders	Se
3	6	Michael Douglas runs afoul of a treacherous su...	R	Drama|Mystery and Suspense	Barry Levinson	Paul Attanasio|Michael Crichton	[
4	7	NaN	NR	Drama|Romance	Rodney Bennett	Giles Cooper	

In [54]:

```
df3.columns
```

Out[54]:

```
Index(['id', 'synopsis', 'rating', 'genre', 'director', 'writer',
       'theater_date', 'dvd_date', 'currency', 'box_office', 'runtime',
       'studio'],
      dtype='object')
```

1. Does genre make movie to have a higher rating ?

   genre vs ratings

2. Does director have any impact on ratings?

   ratings vs director

3. how do form watched affect movie ratings

   Movies watches in theater vs movie watched in dvd.

In [55]:

```
df4 = pd.read_table("rt.reviews.tsv" , encoding="latin1")
df4.head()
```

Out[55]:

	id	review	rating	fresh	critic	top_critic	publisher	date
0	3	A distinctly gallows take on contemporary fina...	3/5	fresh	PJ Nabarro	0	Patrick Nabarro	November 10, 2018
1	3	It's an allegory in search of a meaning that n...	NaN	rotten	Annalee Newitz	0	io9.com	May 23, 2018
2	3	... life lived in a bubble in financial dealin...	NaN	fresh	Sean Axmaker	0	Stream on Demand	January 4, 2018

3	3	Continuing along a line introduced in last yea...	NaN	fresh	Daniel Kasman	0	MUBI	November 16, 2017
4	3	... a perverse twist on neorealism...	NaN	fresh	NaN	0	Cinema Scope	October 12, 2017

In [56]:

```
df5 = pd.read_csv("tmdb.movies.csv")
df5.columns
```

Out[56]:

```
Index(['Unnamed: 0', 'genre_ids', 'id', 'original_language', 'original_
title',
       'popularity', 'release_date', 'title', 'vote_average', 'vote_cou
nt'],
      dtype='object')
```

In [57]:

```
df5.head()
```

Out[57]:

	Unnamed: 0	genre_ids	id	original_language	original_title	popularity	release_c
0	0	[12, 14, 10751]	12444	en	Harry Potter and the Deathly Hallows: Part 1	33.533	2010-1
1	1	[14, 12, 16, 10751]	10191	en	How to Train Your Dragon	28.734	2010-0
2	2	[12, 28, 878]	10138	en	Iron Man 2	28.515	2010-0
3	3	[16, 35, 10751]	862	en	Toy Story	28.005	1995-1
4	4	[28, 878, 12]	27205	en	Inception	27.920	2010-0

In [58]:

```
df5["genre_ids"].value_counts().sort_index(ascending=False)
```

Out[58]:

```
genre_ids
[]                                 2479
[99]                               3700
[99, 99]                              2
[99, 99, 99]                          1
[99, 9648]                            4
                                   ...
[10402, 10751, 14, 10770, 35]         1
[10402, 10749]                        3
[10402, 10749, 35]                    2
[10402, 10749, 35, 18]                3
```

```
[10402, 10749, 35, 18]                3
[10402, 10749, 18]                    2
Name: count, Length: 2477, dtype: int64
```

1. Does language affect popularity ?

   original language vs populaity and vote count

2. Does release date increase popularity ?

   Release date vs popularity

In [59]:

```
df6 = pd.read_csv("tn.movie_budgets.csv")
df6.columns
```

Out[59]:

```
Index(['id', 'release_date', 'movie', 'production_budget', 'domestic_gr
oss',
       'worldwide_gross'],
      dtype='object')
```

In [60]:

```
df6.head()
```

Out[60]:

	id	release_date	movie	production_budget	domestic_gross	worldwide_gross
0	1	Dec 18, 2009	Avatar	$425,000,000	$760,507,625	$2,776,345,279
1	2	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	$410,600,000	$241,063,875	$1,045,663,875
2	3	Jun 7, 2019	Dark Phoenix	$350,000,000	$42,762,350	$149,762,350
3	4	May 1, 2015	Avengers: Age of Ultron	$330,600,000	$459,005,868	$1,403,013,963