In [84]:

```python
import pandas as pd
import numpy as np
```

In [85]:

```python
df = pd.read_excel('hollywood.xlsx')
df['movie_board_rating_display_name'].value_counts()
```

Out[85]:

```
R            448
PG-13        441
PG           182
Not Rated     83
G             39
NC-17          3
Name: movie_board_rating_display_name, dtype: int64
```

In [86]:

```python
df
```

Out[86]:

	id	name	display_name	production_year	movie_sequel	creative_type
0	7950115	Avatar	Avatar	2009	0	Science Fiction
1	50950115	Harry Potter and the Deathly Hallows: Part II	Harry Potter and the Deathly Hallows: Part II	2011	1	Fantasy
2	119870115	Transformers 3	Transformers: Dark of the Moon	2011	1	Science Fiction
3	119590115	Toy Story 3	Toy Story 3	2010	1	Kids Fiction
4	91700115	Pirates of the Caribbean 4	Pirates of the Caribbean: On Stranger Tides	2011	1	Fantasy
...	...	...	...	...	...	...
1191	144410115	Red State	Red State	2011	0	Contemporary Fiction
1192	133000115	Mission, La	La Mission	2009	0	Contemporary Fiction
1193	630115	2010 Oscar Shorts	2010 Oscar Shorts	2009	1	Multiple Creative Types
1194	133360115	sept jours du talion, Les	Les sept jours du talion	2010	0	Contemporary Fiction
1195	146890115	Damsels in Distress	Damsels in Distress	2010	0	Contemporary Fiction

1196 rows × 15 columns

In [87]:

```
## To check if there is any null value in the dataframe
df.isnull().any()
```

Out[87]:

```
id                                    False
name                                  False
display_name                          False
production_year                       False
movie_sequel                          False
creative_type                         False
source                                False
production_method                     False
genre                                 False
language                              False
board_rating_reason                   False
movie_board_rating_display_name       False
movie_release_pattern_display_name    False
total                                 False
Category                              False
dtype: bool
```

1) What are the top three genre of movies with the highest average earnings?

In [88]:

```
movies_grouped_by_genre = df[['total','genre']].groupby('genre',as_index=Fals
```

In [89]:

```
## This line of code finds total earning by each genre
total_earnings_by_genre = movies_grouped_by_genre.sum()
```

In [90]:

```
## This line of code finds total number of movies by each genre
total_number_of_movies_by_genre = movies_grouped_by_genre.count()

total_earnings_by_genre['avg_earnings'] = total_earnings_by_genre['total']/to
total_earnings_by_genre
```

Out[90]:

	genre	total	avg_earnings
0	Action	25180	203.064516
1	Adventure	32369	302.514019
2	Black Comedy	648	54.000000
3	Comedy	23460	90.230769
4	Concert/Performance	270	45.000000
5	Documentary	914	16.925926
6	Drama	16493	51.380062
7	Horror	5822	78.675676
8	Multiple Genres	31	3.444444
9	Musical	1718	143.166667
10	Romantic Comedy	5900	73.750000
11	Thriller/Suspense	11935	91.106870
12	Western	485	80.833333

In [91]:

```
total_earnings_by_genre = total_earnings_by_genre.sort_values(['avg_earnings'
```

Final Answer

In [92]:

```
total_earnings_by_genre['genre'].head(3)
```

Out[92]:

```
1    Adventure
0       Action
9      Musical
Name: genre, dtype: object
```

2) Do movies with sequels tend to have higher gross earnings than the movies without sequels? You can pick the last movie to come out for a prequel.

In [93]:

```
movie_with_sequel = df[df['movie_sequel']==1]
movie_with_sequel['total'].mean()
```

Out[93]:

```
315.55555555555554
```

In [94]:

```
movies_without_sequal = df[df['movie_sequel']==0]
movies_without_sequal['total'].mean()
```

Out[94]:

```
81.83966635773865
```

3) Find the proportion (percentage) of movies by ratings for English language and others (all other languages). Are proportions significantly different from each other?

In [95]:

```
movies_by_language = df[['name','language']].groupby('language')
movies_by_language.count()
english_movies = df[df['language']=='English']
other_language_movies = df[df['language']!='English']
```

In [100…

```
proportions_by_rating = pd.DataFrame(english_movies['movie_board_rating_displ
rating_labels = proportions_by_rating.index
proportions_by_rating['rating'] = rating_labels
proportions_by_rating.index = [i for i in range(1,len(proportions_by_rating)+
total = sum(english_movies['movie_board_rating_display_name'].value_counts())
proportions_by_rating['total_movies'] = proportions_by_rating['movie_board_ra
proportions_by_rating['proportion(in %)'] = proportions_by_rating['movie_boar
proportions_by_rating.drop('movie_board_rating_display_name',axis=1,inplace =
```

In [102…

```
proportions_by_rating['proportion(in %)'].round(decimals=2)
proportions_by_rating
```

Out[102…

	rating	total_movies	proportion(in %)
1	PG-13	435	38.024476

	rating	total_movies	proportion(in %)
2	R	431	37.674825
3	PG	178	15.559441
4	Not Rated	59	5.157343
5	G	38	3.321678
6	NC-17	3	0.262238

In [103…

```
proportions_by_rating = pd.DataFrame(other_language_movies['movie_board_ratin
rating_labels = proportions_by_rating.index
proportions_by_rating['rating'] = rating_labels
proportions_by_rating.index = [i for i in range(1,len(proportions_by_rating)+
total = sum(other_language_movies['movie_board_rating_display_name'].value_co
proportions_by_rating['total_movies'] = proportions_by_rating['movie_board_ra
proportions_by_rating['proportion(in %)'] = proportions_by_rating['movie_boar
proportions_by_rating.drop('movie_board_rating_display_name',axis=1,inplace =
```

In [104…

```
proportions_by_rating['proportion(in %)'].round(decimals=2)
proportions_by_rating
```

Out[104…

	rating	total_movies	proportion(in %)
1	Not Rated	24	46.153846
2	R	17	32.692308
3	PG-13	6	11.538462
4	PG	4	7.692308
5	G	1	1.923077

In [ ]: