

In [1]:

```
import pandas as pd

df=pd.read_csv("Highest Hollywood Grossing Movies.csv")
df
```

Out[1]:

	Unnamed: 0	Title	Movie Info	Distributor	Release Date	Domestic Sales (in \$)	International Sales (in \$)	Worldwide Sales (in \$)
0	0	Star Wars: Episode VII - The Force Awakens (2015)	As a new threat to the galaxy rises, Rey, a de...	Walt Disney Studios Motion Pictures	December 16, 2015	936662225	1132859475	2069521
1	1	Avengers: Endgame (2019)	After the devastating events of Avengers: Infini...	Walt Disney Studios Motion Pictures	April 24, 2019	858373000	1939128328	2797501
2	2	Avatar (2009)	A paraplegic Marine dispatched to the moon Pan...	Twentieth Century Fox	December 16, 2009	760507625	2086738578	2847246
3	3	Black Panther (2018)	T'Challa, heir to the hidden but advanced king...	Walt Disney Studios Motion Pictures	NaN	700426566	647171407	1347597
4	4	Avengers: Infinity War (2018)	The Avengers and their allies must be willing ...	Walt Disney Studios Motion Pictures	NaN	678815482	1369544272	2048359
...
913	913	The Notebook (2004)	A poor yet passionate young man falls in love ...	New Line Cinema	June 25, 2004	81001787	36813370	117815
914	914	Jimmy Neutron: Boy Genius (2001)	An eight-year-old boy genius and his friends m...	Paramount Pictures	December 21, 2001	80936232	22056304	102992
915	915	Eat Pray Love (2010)	A married woman realizes how unhappy her marri...	Sony Pictures Entertainment (SPE)	August 13, 2010	80574010	124020006	204594

	Unnamed: 0	Title	Movie Info	Distributor	Release Date	Domestic Sales (in \$)	International Sales (in \$)	World Sales (in \$)
916	916	The Texas Chainsaw Massacre (2003)	After picking up a traumatized young hitchhiker...	New Line Cinema	October 17, 2003	80571655	26792250	107363
917	917	Zookeeper (2011)	A group of zoo animals decide to break their c...	Sony Pictures Entertainment (SPE)	July 6, 2011	80360843	89491916	169852

918 rows × 11 columns

◀ ▶

In [2]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 918 entries, 0 to 917
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype  
 --- 
 0   Unnamed: 0        918 non-null    int64  
 1   Title            918 non-null    object  
 2   Movie Info       918 non-null    object  
 3   Distributor      918 non-null    object  
 4   Release Date     800 non-null    object  
 5   Domestic Sales (in $)  918 non-null    int64  
 6   International Sales (in $) 918 non-null    int64  
 7   World Sales (in $)  918 non-null    int64  
 8   Genre            918 non-null    object  
 9   Movie Runtime    918 non-null    object  
 10  License          744 non-null    object  
dtypes: int64(4), object(7)
memory usage: 79.0+ KB
```

In [4]:

```
df.isna().sum()
```

```
Out[4]: Unnamed: 0      0
Title          0
Movie Info     0
Distributor    0
Release Date   118
Domestic Sales (in $)  0
International Sales (in $) 0
World Sales (in $)  0
Genre           0
Movie Runtime   0
License         174
dtype: int64
```

In [5]:

```
#Missing Values
df_cleaned=df.fillna("*")
df_cleaned
```

Out[5]:

	Unnamed: 0	Title	Movie Info	Distributor	Release Date	Domestic Sales (in \$)	International Sales (in \$)	Wc Sales (in \$)
0	0	Star Wars: Episode VII - The Force Awakens (2015)	As a new threat to the galaxy rises, Rey, a de...	Walt Disney Studios Motion Pictures	December 16, 2015	936662225	1132859475	2069521
1	1	Avengers: Endgame (2019)	After the devastating events of Avengers: Infi...	Walt Disney Studios Motion Pictures	April 24, 2019	858373000	1939128328	2797501
2	2	Avatar (2009)	A paraplegic Marine dispatched to the moon Pan...	Twentieth Century Fox	December 16, 2009	760507625	2086738578	2847246
3	3	Black Panther (2018)	T'Challa, heir to the hidden but advanced king...	Walt Disney Studios Motion Pictures	*	700426566	647171407	1347597
4	4	Avengers: Infinity War (2018)	The Avengers and their allies must be willing ...	Walt Disney Studios Motion Pictures	*	678815482	1369544272	2048359
...
913	913	The Notebook (2004)	A poor yet passionate young man falls in love	New Line Cinema	June 25, 2004	81001787	36813370	117815
914	914	Jimmy Neutron: Boy Genius (2001)	An eight-year-old boy genius and his friends m...	Paramount Pictures	December 21, 2001	80936232	22056304	102992
915	915	Eat Pray Love (2010)	A married woman realizes how unhappy her marri...	Sony Pictures Entertainment (SPE)	August 13, 2010	80574010	124020006	204594
916	916	The Texas Chainsaw Massacre (2003)	After picking up a traumatized young hitchhike...	New Line Cinema	October 17, 2003	80571655	26792250	107363

	Unnamed: 0	Title	Movie Info	Distributor	Release Date	Domestic Sales (in \$)	International Sales (in \$)	Wc Sales (in \$)
917	917	Zookeeper (2011)	A group of zoo animals decide to break their c...	Sony Pictures Entertainment (SPE)	July 6, 2011	80360843	89491916	169852

918 rows × 11 columns

In [6]:	df
---------	----

	Unnamed: 0	Title	Movie Info	Distributor	Release Date	Domestic Sales (in \$)	International Sales (in \$)	Wc Sales (in \$)
0	0	Star Wars: Episode VII - The Force Awakens (2015)	As a new threat to the galaxy rises, Rey, a de...	Walt Disney Studios Motion Pictures	December 16, 2015	936662225	1132859475	2069521
1	1	Avengers: Endgame (2019)	After the devastating events of Avengers: Infi...	Walt Disney Studios Motion Pictures	April 24, 2019	858373000	1939128328	2797501
2	2	Avatar (2009)	A paraplegic Marine dispatched to the moon Pan...	Twentieth Century Fox	December 16, 2009	760507625	2086738578	2847246
3	3	Black Panther (2018)	T'Challa, heir to the hidden but advanced king...	Walt Disney Studios Motion Pictures	NaN	700426566	647171407	1347597
4	4	Avengers: Infinity War (2018)	The Avengers and their allies must be willing ...	Walt Disney Studios Motion Pictures	NaN	678815482	1369544272	2048359
...
913	913	The Notebook (2004)	A poor yet passionate young man falls in love ...	New Line Cinema	June 25, 2004	81001787	36813370	117815
914	914	Jimmy Neutron: Boy Genius (2001)	An eight-year-old boy genius and his friends m...	Paramount Pictures	December 21, 2001	80936232	22056304	102992

Unnamed: 0		Title	Movie Info	Distributor	Release Date	Domestic Sales (in \$)	International Sales (in \$)	World Sales (in \$)
915	915	Eat Pray Love (2010)	A married woman realizes how unhappy her marri...	Sony Pictures Entertainment (SPE)	August 13, 2010	80574010	124020006	204594
916	916	The Texas Chainsaw Massacre (2003)	After picking up a traumatized young hitchhike...	New Line Cinema	October 17, 2003	80571655	26792250	107363
917	917	Zookeeper (2011)	A group of zoo animals decide to break their c...	Sony Pictures Entertainment (SPE)	July 6, 2011	80360843	89491916	169852

918 rows × 11 columns



```
In [35]: top5 = df[['Title','World Sales (in $)']].sort_values(by='World Sales (in $)', ascending=False)
top5_titles = top5.iloc[:5]
print(top5_titles)
```

	Title	World Sales (in \$)
2	Avatar (2009)	2847246203
1	Avengers: Endgame (2019)	2797501328
6	Titanic (1997)	2201647264
0	Star Wars: Episode VII - The Force Awakens (2015)	2069521700
4	Avengers: Infinity War (2018)	2048359754

```
In [79]: # libraries
import numpy as np
import matplotlib.pyplot as plt

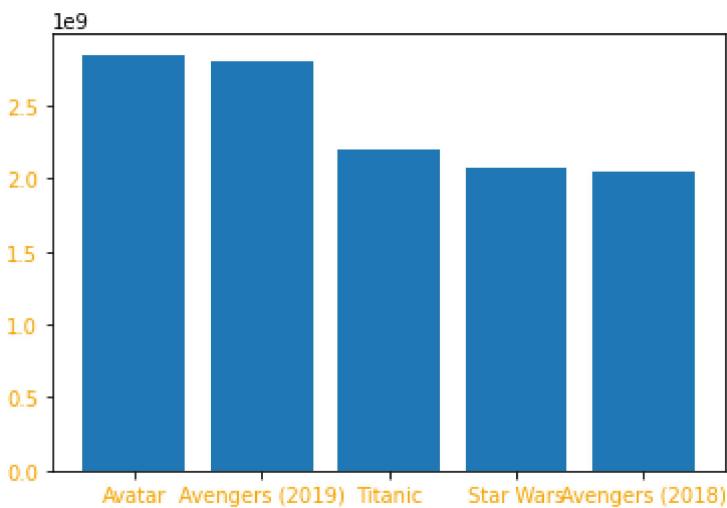
# Choose the height of the bars
height = [2847246203, 2797501328, 2201647264, 2069521700, 2048359754]

# Top 5 movies
bars = ('Avatar', 'Avengers (2019)', 'Titanic', 'Star Wars', 'Avengers (2018)')
x_pos = np.arange(len(bars))

# Create bars
plt.bar(x_pos, height)

# Create names on the x-axis
plt.xticks(x_pos, bars, color='orange')
plt.yticks(color='orange')

# Show graphic
plt.show()
```



In [125...]

```
import pandas as pd

df=pd.read_csv("Highest Hollywood Grossing Movies.csv")
df

total_df_Distributor = df.groupby(['Distributor']).count()['Title'].sort_values(ascending=True)
print(total_df_Distributor)
```

Distributor	Hits
Warner Bros.	158
Walt Disney Studios Motion Pictures	155
Universal Pictures	119
Twentieth Century Fox	117
Sony Pictures Entertainment (SPE)	101
Paramount Pictures	99
New Line Cinema	21
DreamWorks	21
Lionsgate	19
DreamWorks Distribution	17
Metro-Goldwyn-Mayer (MGM)	12
TriStar Pictures	9
Miramax	9
Revolution Studios	8
Columbia Pictures	7
Dimension Films	7
The Weinstein Company	6
Summit Entertainment	5
Fox Searchlight Pictures	4
Orion Pictures	3
STX Entertainment	3
Screen Gems	3
Focus Features	2
United Artists	2
FilmDistrict	2
USA Films	1
United Artists Releasing	1
20th Century Studios	1
Sony Pictures Classics	1
Roadside Attractions	1
Artisan Entertainment	1
Newmarket Films	1
IFC Films	1
Relativity Media	1

Name: Title, dtype: int64

In [130...]

```
#Which Distributor has most hits
```

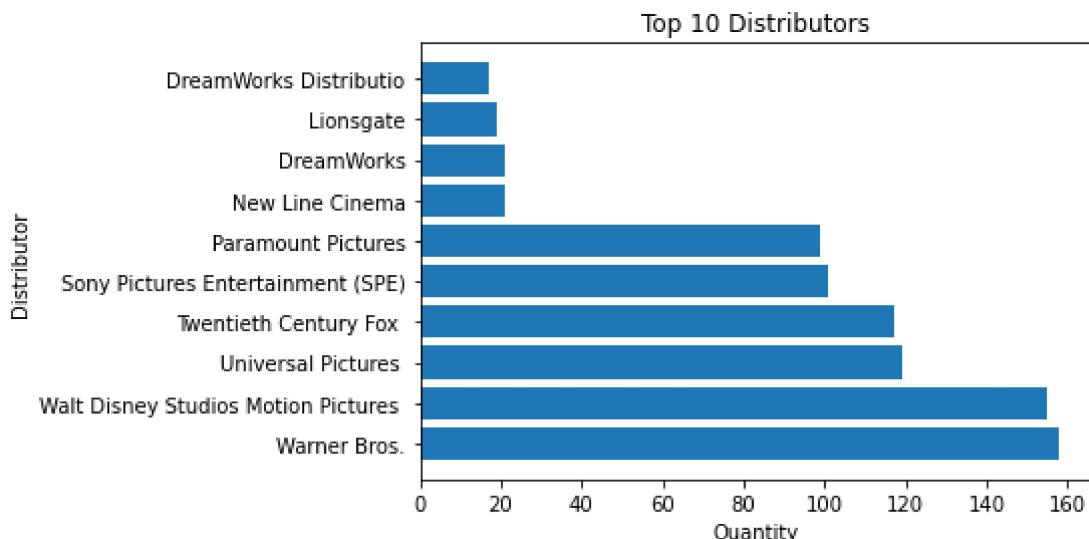
```

import matplotlib.pyplot as plt

Distributor = ['Warner Bros.', 'Walt Disney Studios Motion Pictures ', 'Universal Pict
Quantity = [158, 155, 119, 117, 101, 99, 21, 21, 19, 17]

plt.barh(Distributor, Quantity)
plt.title('Top 10 Distributors')
plt.ylabel('Distributor')
plt.xlabel('Quantity')
plt.show()

```



In [140...]

```

import pandas as pd

df=pd.read_csv("Highest Hollywood Grossing Movies.csv")
df

df['License'].nunique()

```

Out[140...]

4

In [141...]

```

print(df['License'].unique())

['PG-13' nan 'PG' 'G' 'R']

```

In [143...]

```

df_counts = df["Genre"].value_counts()
print(df_counts)

```

['Action', 'Adventure', 'Sci-Fi']	56
['Comedy']	35
['Comedy', 'Romance']	30
['Action', 'Adventure', 'Thriller']	28
['Action', 'Crime', 'Thriller']	23
..	
['Biography', 'Comedy', 'Drama', 'Romance']	1
['Drama', 'History']	1
['Drama', 'Family']	1
['Action', 'Crime', 'Mystery', 'Sci-Fi', 'Thriller']	1
['Comedy', 'Family', 'Fantasy', 'Romance']	1

Name: Genre, Length: 309, dtype: int64

In [152...]

```

df_Genre_counts = pd.DataFrame(df_counts)
df_Genre_counts = df_counts.reset_index()

```

```
df_Genre_counts.columns = ["Genre", "Number of Productions"]
df_Genre_counts
```

Out[152...]

	Genre	Number of Productions
0	['Action', 'Adventure', 'Sci-Fi']	56
1	['Comedy']	35
2	['Comedy', 'Romance']	30
3	['Action', 'Adventure', 'Thriller']	28
4	['Action', 'Crime', 'Thriller']	23
...
304	['Biography', 'Comedy', 'Drama', 'Romance']	1
305	['Drama', 'History']	1
306	['Drama', 'Family']	1
307	['Action', 'Crime', 'Mystery', 'Sci-Fi', 'Thri...']	1
308	['Comedy', 'Family', 'Fantasy', 'Romance']	1

309 rows × 2 columns

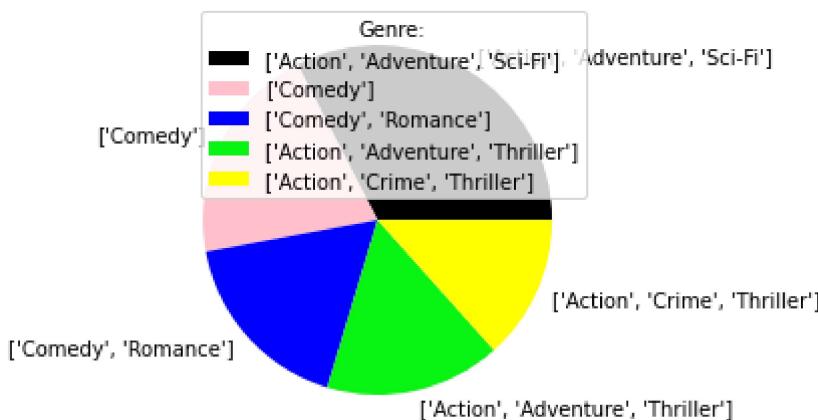
In [159...]

```
#piechart showing top 5 genre movies
```

```
import matplotlib.pyplot as plt
import numpy as np

y = np.array([56, 35, 30, 28, 23])
Genre = [['Action', 'Adventure', 'Sci-Fi'], ['Comedy'], ['Comedy', 'Romance'], ['Action', 'Adventure', 'Thriller'], ['Action', 'Crime', 'Thriller']]
mycolors = ["black", "pink", "b", "#08f312", "yellow"]

plt.pie(y, labels = Genre, colors = mycolors)
plt.legend(title = "Genre:")
plt.show()
```



In [197...]

```
import pandas as pd

df=pd.read_csv("Highest Hollywood Grossing Movies.csv")
df
# Sorting and slicing values to see top 10 movies of all time, based on the World Sales
top10 = df[['Movie Runtime', 'World Sales (in $)']].sort_values(by='World Sales (in $)', ascending=False).head(10)
```

```
top10_movies = top10.iloc[:10]
print(top10_movies)
```

	Movie	Runtime	World Sales (in \$)
2	2 hr 42 min	2847246203	
1	3 hr 1 min	2797501328	
6	3 hr 14 min	2201647264	
0	2 hr 18 min	2069521700	
4	2 hr 29 min	2048359754	
7	2 hr 4 min	1670516444	
11	1 hr 58 min	1662899439	
5	2 hr 28 min	1544455963	
8	2 hr 23 min	1518815515	
54	2 hr 17 min	1515341399	

In [231...]

```
import pandas as pd

df=pd.read_csv("Highest Hollywood Grossing Movies.csv")
df
print(df.head())
print(df.shape)
```

	Unnamed: 0	Title \
0	0	Star Wars: Episode VII - The Force Awakens (2015)
1	1	Avengers: Endgame (2019)
2	2	Avatar (2009)
3	3	Black Panther (2018)
4	4	Avengers: Infinity War (2018)

	Movie Info \
0	As a new threat to the galaxy rises, Rey, a de...
1	After the devastating events of Avengers: Infi...
2	A paraplegic Marine dispatched to the moon Pan...
3	T'Challa, heir to the hidden but advanced king...
4	The Avengers and their allies must be willing ...

	Distributor	Release Date \
0	Walt Disney Studios Motion Pictures	December 16, 2015
1	Walt Disney Studios Motion Pictures	April 24, 2019
2	Twentieth Century Fox	December 16, 2009
3	Walt Disney Studios Motion Pictures	NaN
4	Walt Disney Studios Motion Pictures	NaN

	Domestic Sales (in \$)	International Sales (in \$)	World Sales (in \$) \
0	936662225	1132859475	2069521700
1	858373000	1939128328	2797501328
2	760507625	2086738578	2847246203
3	700426566	647171407	1347597973
4	678815482	1369544272	2048359754

	Genre	Movie	Runtime	License
0	['Action', 'Adventure', 'Sci-Fi']	2 hr 18 min	PG-13	
1	['Action', 'Adventure', 'Drama', 'Sci-Fi']	3 hr 1 min	PG-13	
2	['Action', 'Adventure', 'Fantasy', 'Sci-Fi']	2 hr 42 min	PG-13	
3	['Action', 'Adventure', 'Sci-Fi']	2 hr 14 min	NaN	
4	['Action', 'Adventure', 'Sci-Fi']	2 hr 29 min	NaN	
(918, 11)				

In [234...]

```
import pandas as pd

df=pd.read_csv("Highest Hollywood Grossing Movies.csv")
df
```

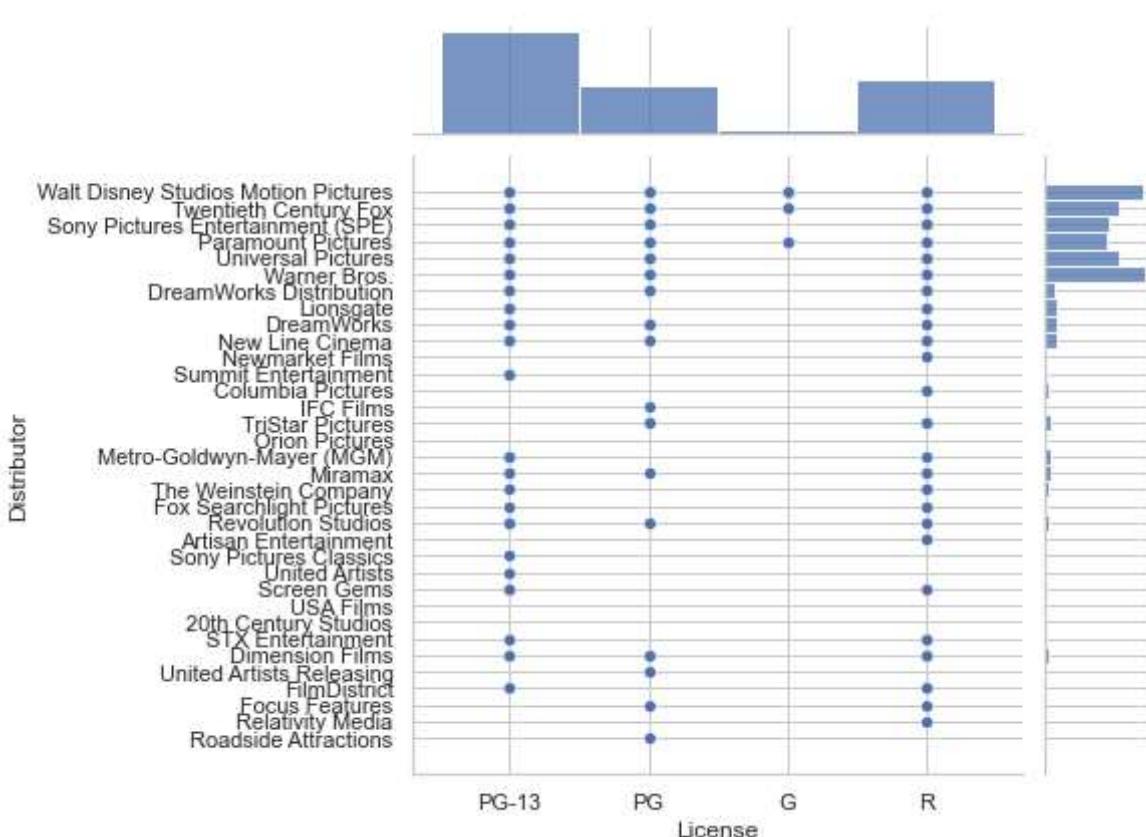
```
droprows=df.dropna()  
print(df.shape,droprows.shape)
```

```
(918, 11) (744, 11)
```

In [280...]

```
import pandas as pd  
  
df=pd.read_csv("Highest Hollywood Grossing Movies.csv")  
df  
  
sns.jointplot(data=df, x='License', y='Distributor')
```

Out[280...]



In [537...]

```
import pandas as pd  
  
df=pd.read_csv("Highest Hollywood Grossing Movies.csv")  
df  
# Sorting and slicing values to see top 10 movies of all time, based on the World Sales  
top10 = df[['Movie Runtime', 'World Sales (in $)']].sort_values(by='World Sales (in $)', ascending=False)  
top10_movies = top10.iloc[:10]  
print(top10_movies)  
  
list_a=[2847246203,2797501328,2201647264,2069521700,2048359754,1670516444,1662899439  
list_lenght=len(list_a)  
print(list_lenght)
```

	Movie Runtime	World Sales (in \$)
2	2 hr 42 min	2847246203
1	3 hr 1 min	2797501328
6	3 hr 14 min	2201647264
0	2 hr 18 min	2069521700
4	2 hr 29 min	2048359754
7	2 hr 4 min	1670516444
11	1 hr 58 min	1662899439
5	2 hr 28 min	1544455963

```
8    2 hr 23 min          1518815515
54   2 hr 17 min          1515341399
10
```

In [406...]

```
import pandas as pd

df=pd.read_csv("Highest Hollywood Grossing Movies.csv")

small_df = df[['Title','Movie Runtime']]
run_time = small_df['Movie Runtime']
print(small_df)
print("Longest Movie Runtime")
print(run_time.max())
print("Shortest Movie Runtime")
print(run_time.min())
```

	Title	Movie Runtime
0	Star Wars: Episode VII - The Force Awakens (2015)	2 hr 18 min
1	Avengers: Endgame (2019)	3 hr 1 min
2	Avatar (2009)	2 hr 42 min
3	Black Panther (2018)	2 hr 14 min
4	Avengers: Infinity War (2018)	2 hr 29 min
..
913	The Notebook (2004)	2 hr 3 min
914	Jimmy Neutron: Boy Genius (2001)	1 hr 22 min
915	Eat Pray Love (2010)	2 hr 13 min
916	The Texas Chainsaw Massacre (2003)	1 hr 38 min
917	Zookeeper (2011)	1 hr 42 min

[918 rows x 2 columns]

Longest Movie Runtime

3 hr 9 min

Shortest Movie Runtime

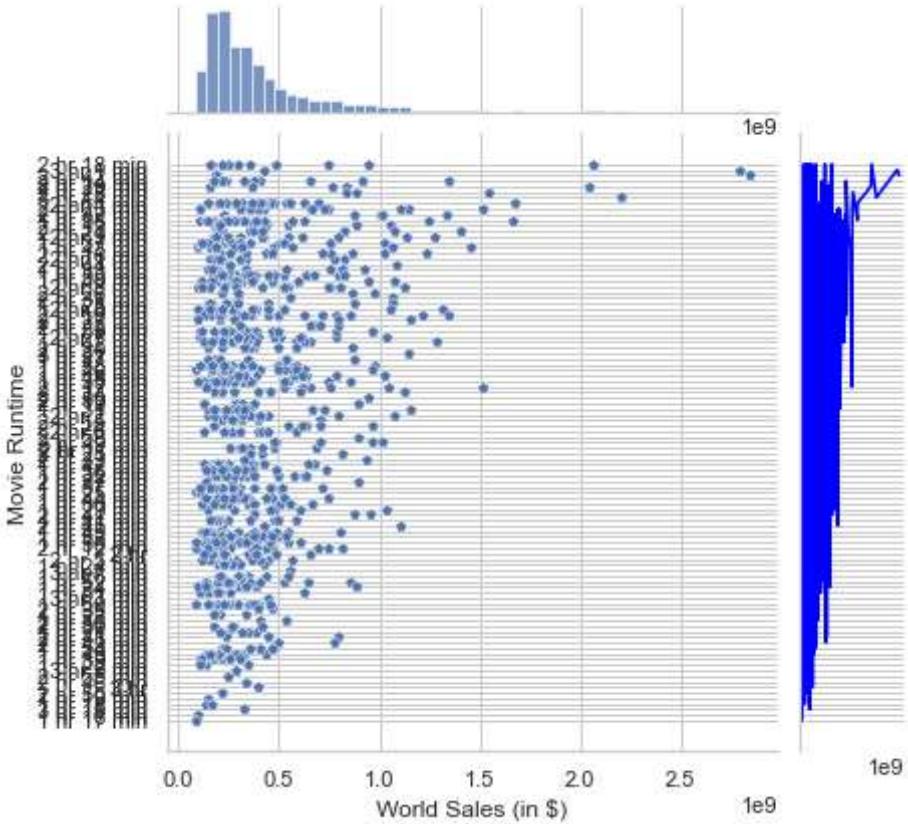
1 hr 16 min

In [405...]

```
import pandas as pd

df=pd.read_csv("Highest Hollywood Grossing Movies.csv")
df

sns.jointplot(data=df, x='World Sales (in $)', y='Movie Runtime',marker='p')
sns.lineplot(x="World Sales (in $)", y="Movie Runtime",data=df, color="blue")
plt.xlabel("World Sales (in $)")
plt.ylabel("Movie Runtime")
plt.show()
```



In [416...]

```

import pandas as pd

df=pd.read_csv("Highest Hollywood Grossing Movies.csv")
df
# Sorting and slicing values to see top 10 movies of all time, based on the World Sales
top10 = df[['Movie Runtime', 'World Sales (in $)']].sort_values(by='World Sales (in $)', ascending=False)
top10_movies = top10.iloc[:10]
print(top10_movies)

import pandas as pd

df=pd.read_csv("Highest Hollywood Grossing Movies.csv")
df

Sales = [2847246203, 2797501328, 2201647264, 2069521700, 2048359754, 1670516444, 166289943
for Sales in df:
    print(Sales)

```

	Movie Runtime	World Sales (in \$)
2	2 hr 42 min	2847246203
1	3 hr 1 min	2797501328
6	3 hr 14 min	2201647264
0	2 hr 18 min	2069521700
4	2 hr 29 min	2048359754
7	2 hr 4 min	1670516444
11	1 hr 58 min	1662899439
5	2 hr 28 min	1544455963
8	2 hr 23 min	1518815515
54	2 hr 17 min	1515341399
Unnamed: 0		
Title		
Movie Info		
Distributor		
Release Date		
Domestic Sales (in \$)		
International Sales (in \$)		

```
World Sales (in $)
Genre
Movie Runtime
License
```

In [516...]

```
#Use of dictionary - part of assignment
music = {"Artist":["Elton John", "Harry Styles", "John Lennon"], "Song":["Tiny Dancer",
for rows in music:
    print(rows)

for rows in music.items():
    print(rows)
```

```
Artist
Song
Length
Year
('Artist', ['Elton John', 'Harry Styles', 'John Lennon'])
('Song', ['Tiny Dancer', 'Golden', 'Imagine'])
('Length', [6.12, 3.28, 3.03])
('Year', [1971, 2019, 1971])
```

In [532...]

```
music = {"Artist":["Elton John", "Harry Styles", "John Lennon"], "Song":["Tiny Dancer",
for rows in music:
    print(rows)

df = pd.DataFrame(music)
for index, row in df.iterrows():
    print(row["Artist"])
```

```
Artist
Song
Length
Year
Elton John
Harry Styles
John Lennon
```

In []: