correlation w pandas

July 2, 2024

0.1 Movie industry

0.1.1 Intro

This is a data set acquired from kaggle.com and it consists of three decades of movie data (1986-2016).

The researcher/author mentions that 'there are 6820 movies in the dataset (220 movies per year, 1986-2016).

Each movie has the following attributes (kaggle.com):

- budget: the budget of a movie. Some movies don't have this, so it appears as 0
- Company
- Country
- Director
- Genre
- Gross
- name
- rating
- released
- runtime
- score (IMDb)
- votes
- star
- writer
- year

0.1.2 Research Questions:

1. Main RQ: Is the movie industry dying? Are production companies making as much money today compared to 10 years ago?
2. RQ: Which movie has made the most money in terms of gross revenue in the last three decades?
3. RQ: What correlations are there?
    1. What can we say about the relationship between budget and gross revenue?
    2. What can be said about the relationship between genre and gross revenue? Does a particular genre provide more returns?

0.1.3 Findings

RQ1: Is the movie industry dying? Are production companies making as much money today compared to 10 years ago? A particular production company was sampled to assess whether they are earnings less now compared to 10 or 20 years ago.

Findings illustrate that this particular production company was still making profit from theatre box office. This can help us to conclude that people are still willing to watch a movie at a cinema, despite recent development of Netflix and the sort

RQ2: In order to gain insight into which movie has made the most money in the last three decades, the df was sorted according to gross revenue.

Findings illustrate that up until 2016, the movie 'Avatar' stood as the highest revenue making movie at the International box office. 2024 data illustrates that figures have since changed.

RQ3: In order to gain insight into correlation between budget and gross revenue, a linear regression graph was utilised to determine the relationship between the two variables. A pearson correlation value was also determined in order to supplement the scatter plot/regression graph.

Findings through Linear regression, illustrates that there is a positive relationship between gross revenue and budget. This implies that the bigger the budget – meaning the more money invested in a movie – the more gross revenue to be expected

The pearson correlation (0.74) betweeen budget and gross revenue further solidifies that there is a strong positive relationship between the two variables.

In order to assess the relationship between genre and gross revenue, a box-and-whisker plot was used. This is because genre is a categorical value, while budget is a numeircal value therefore, linear regression is not possible for this two vairables together. A box-and-whisker plot provides an idea of the distribution of gross income for the different genres.

Findings illustrate that there is no particular genre that is making significantly more income on average.

[ ]:

[ ]:

[69]:
```
# First let's import the packages we will use in this project
# You can do this all now or as you need them
import pandas as pd
import numpy as np
import seaborn as sns
```

```
import matplotlib.pyplot as plt
import matplotlib.mlab as mlab
import matplotlib
plt.style.use('ggplot')
from matplotlib.pyplot import figure

%matplotlib inline
matplotlib.rcParams['figure.figsize'] = (12,8)

pd.options.mode.chained_assignment = None


import plotly.express as px
import plotly.graph_objects as go
from plotly.offline import offline, iplot


pd.options.display.float_format = "{:,.1f}".format


def update_layout(title_font_size = 28, hover_font_size = 16, hover_bgcolor =␣
 ↪"#45FFCA", showlegend = False):
    fig.update_layout(
        showlegend = showlegend,
        title = {
            "font" : {
                "size" :title_font_size,
                "family" :"tahoma"
            }
        },
        hoverlabel={
            "bgcolor": hover_bgcolor,
            "font_size": hover_font_size,
            "font_family": "tahoma"
        }
    )

# Now we need to read in the data
df = pd.read_csv('movies.csv')
```

0.1.4 Data cleaning

[72]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7668 entries, 0 to 7667
Data columns (total 15 columns):
```

```
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   name      7668 non-null   object
 1   rating    7591 non-null   object
 2   genre     7668 non-null   object
 3   year      7668 non-null   int64
 4   released  7666 non-null   object
 5   score     7665 non-null   float64
 6   votes     7665 non-null   float64
 7   director  7668 non-null   object
 8   writer    7665 non-null   object
 9   star      7667 non-null   object
 10  country   7665 non-null   object
 11  budget    5497 non-null   float64
 12  gross     7479 non-null   float64
 13  company   7651 non-null   object
 14  runtime   7664 non-null   float64
dtypes: float64(5), int64(1), object(9)
memory usage: 898.7+ KB
```

[74]:
```
df.describe()
```

[74]:

	year	score	votes	budget	gross	runtime
count	7,668.0	7,665.0	7,665.0	5,497.0	7,479.0	7,664.0
mean	2,000.4	6.4	88,108.5	35,589,876.2	78,500,541.0	107.3
std	11.2	1.0	163,323.8	41,457,296.6	165,725,124.3	18.6
min	1,980.0	1.9	7.0	3,000.0	309.0	55.0
25%	1,991.0	5.8	9,100.0	10,000,000.0	4,532,055.5	95.0
50%	2,000.0	6.5	33,000.0	20,500,000.0	20,205,757.0	104.0
75%	2,010.0	7.1	93,000.0	45,000,000.0	76,016,691.5	116.0
max	2,020.0	9.3	2,400,000.0	356,000,000.0	2,847,246,203.0	366.0

[76]:
```
# We need to see if we have any missing data
# Let's loop through the data and see if there is anything missing

for col in df.columns:
    pct_missing = np.mean(df[col].isnull())
    print('{} - {}%'.format(col, round(pct_missing*100)))
```

```
name - 0%
rating - 1%
genre - 0%
year - 0%
released - 0%
score - 0%
votes - 0%
director - 0%
writer - 0%
```

```
star - 0%
country - 0%
budget - 28%
gross - 2%
company - 0%
runtime - 0%
```

[78]:
```
# We need to see if we have any missing data
# Let's loop through the data and see if there is anything missing

df.isnull().sum()
```

[78]:
```
name           0
rating        77
genre          0
year           0
released       2
score          3
votes          3
director       0
writer         3
star           1
country        3
budget      2171
gross        189
company       17
runtime        4
dtype: int64
```

[80]:
```
# We need to see if we have any missing data
# Let's loop through the data and see if there is anything missing


missing_data = df.isnull()
missing_data.head(5)


for column in missing_data.columns.values.tolist():
    print(column)
    print (missing_data[column].value_counts())
    print("")
```

```
name
name
False    7668
Name: count, dtype: int64

rating
```

```
rating
False    7591
True       77
Name: count, dtype: int64

genre
genre
False    7668
Name: count, dtype: int64

year
year
False    7668
Name: count, dtype: int64

released
released
False    7666
True        2
Name: count, dtype: int64

score
score
False    7665
True        3
Name: count, dtype: int64

votes
votes
False    7665
True        3
Name: count, dtype: int64

director
director
False    7668
Name: count, dtype: int64

writer
writer
False    7665
True        3
Name: count, dtype: int64

star
star
False    7667
True        1
```

```
Name: count, dtype: int64

country
country
False    7665
True        3
Name: count, dtype: int64

budget
budget
False    5497
True     2171
Name: count, dtype: int64

gross
gross
False    7479
True      189
Name: count, dtype: int64

company
company
False    7651
True       17
Name: count, dtype: int64

runtime
runtime
False    7664
True        4
Name: count, dtype: int64
```

[82]:
```
#drop all rows with null values because we can t replace them

df = df.dropna()
```

[84]:
```
df
```

[84]:

	name	rating	genre
0	The Shining	R	Drama
1	The Blue Lagoon	R	Adventure
2	Star Wars: Episode V - The Empire Strikes Back	PG	Action
3	Airplane!	PG	Comedy
4	Caddyshack	R	Comedy
…	…	…	…
7648	Bad Boys for Life	R	Action

```
7649                              Sonic the Hedgehog         PG     Action
7650                                        Dolittle         PG  Adventure
7651                            The Call of the Wild         PG  Adventure
7652                               The Eight Hundred  Not Rated     Action

      year                             released  score       votes  \
0     1980      June 13, 1980 (United States)    8.4   927,000.0
1     1980       July 2, 1980 (United States)    5.8    65,000.0
2     1980      June 20, 1980 (United States)    8.7 1,200,000.0
3     1980       July 2, 1980 (United States)    7.7   221,000.0
4     1980      July 25, 1980 (United States)    7.3   108,000.0
…     …                                    …    …         …
7648  2020   January 17, 2020 (United States)    6.6   140,000.0
7649  2020  February 14, 2020 (United States)    6.5   102,000.0
7650  2020   January 17, 2020 (United States)    5.6    53,000.0
7651  2020  February 21, 2020 (United States)    6.8    42,000.0
7652  2020    August 28, 2020 (United States)    6.8     3,700.0

             director                  writer               star  \
0     Stanley Kubrick             Stephen King     Jack Nicholson
1      Randal Kleiser  Henry De Vere Stacpoole     Brooke Shields
2      Irvin Kershner           Leigh Brackett        Mark Hamill
3        Jim Abrahams             Jim Abrahams        Robert Hays
4        Harold Ramis       Brian Doyle-Murray        Chevy Chase
…                 …                        …                  …
7648     Adil El Arbi             Peter Craig         Will Smith
7649      Jeff Fowler               Pat Casey       Ben Schwartz
7650   Stephen Gaghan           Stephen Gaghan  Robert Downey Jr.
7651    Chris Sanders            Michael Green      Harrison Ford
7652          Hu Guan                  Hu Guan    Zhi-zhong Huang

             country        budget         gross  \
0     United Kingdom  19,000,000.0  46,998,772.0
1      United States   4,500,000.0  58,853,106.0
2      United States  18,000,000.0 538,375,067.0
3      United States   3,500,000.0  83,453,539.0
4      United States   6,000,000.0  39,846,344.0
…                ….           …            …
7648   United States  90,000,000.0 426,505,244.0
7649   United States  85,000,000.0 319,715,683.0
7650   United States 175,000,000.0 245,487,753.0
7651          Canada 135,000,000.0 111,105,497.0
7652           China  80,000,000.0 461,421,559.0

                                company  runtime
0                         Warner Bros.    146.0
1                     Columbia Pictures    104.0
```

```
2                                Lucasfilm    124.0
3                      Paramount Pictures     88.0
4                          Orion Pictures     98.0
…                                     …      …
7648                    Columbia Pictures    124.0
7649                   Paramount Pictures     99.0
7650                   Universal Pictures    101.0
7651                 20th Century Studios    100.0
7652  Beijing Diqi Yinxiang Entertainment    149.0

[5421 rows x 15 columns]
```

[86]:
```
#checking again to see. Everything should be zero
df.isnull().sum()
```

[86]:
```
name        0
rating      0
genre       0
year        0
released    0
score       0
votes       0
director    0
writer      0
star        0
country     0
budget      0
gross       0
company     0
runtime     0
dtype: int64
```

[88]:
```
# Data Types for our columns

print(df.dtypes)
```

```
name         object
rating       object
genre        object
year          int64
released     object
score       float64
votes       float64
director     object
writer       object
star         object
country      object
```

```
budget      float64
gross       float64
company      object
runtime     float64
dtype: object
```

[90]:
```
#change data type of columns

# df['budget'] = df['budget'].astype('int64') this ussually works but it didnt, it brought anerror

# df['gross'] = df['gross'].astype('int64') this ussually works but it didnt for some reason, it brought an error

# If anyone else is having issues due to IntCastingNanError, I advise to try the following:

df['budget'] = pd.to_numeric(df['budget'], errors='coerce').fillna(0).astype(int)
df['gross'] = pd.to_numeric(df['gross'], errors='coerce').fillna(0).astype(int)
```

[92]:
```
df
```

[92]:

	name	rating	genre	year	released	score	votes
0	The Shining	R	Drama	1980	June 13, 1980 (United States)	8.4	927,000.0
1	The Blue Lagoon	R	Adventure	1980	July 2, 1980 (United States)	5.8	65,000.0
2	Star Wars: Episode V - The Empire Strikes Back	PG	Action	1980	June 20, 1980 (United States)	8.7	1,200,000.0
3	Airplane!	PG	Comedy	1980	July 2, 1980 (United States)	7.7	221,000.0
4	Caddyshack	R	Comedy	1980	July 25, 1980 (United States)	7.3	108,000.0
…	…	…	…	…	…	…	…
7648	Bad Boys for Life	R	Action	2020	January 17, 2020 (United States)	6.6	140,000.0
7649	Sonic the Hedgehog	PG	Action	2020	February 14, 2020 (United States)	6.5	102,000.0
7650	Dolittle	PG	Adventure	2020	January 17, 2020 (United States)	5.6	53,000.0
7651	The Call of the Wild	PG	Adventure	2020	February 21, 2020 (United States)	6.8	42,000.0

```
7652  2020    August 28, 2020 (United States)    6.8     3,700.0

             director                   writer               star  \
0     Stanley Kubrick             Stephen King     Jack Nicholson
1      Randal Kleiser  Henry De Vere Stacpoole     Brooke Shields
2      Irvin Kershner           Leigh Brackett        Mark Hamill
3        Jim Abrahams             Jim Abrahams        Robert Hays
4        Harold Ramis       Brian Doyle-Murray        Chevy Chase
…                …                        …                  …
7648     Adil El Arbi             Peter Craig         Will Smith
7649      Jeff Fowler               Pat Casey       Ben Schwartz
7650   Stephen Gaghan           Stephen Gaghan  Robert Downey Jr.
7651    Chris Sanders            Michael Green      Harrison Ford
7652          Hu Guan                  Hu Guan    Zhi-zhong Huang

             country     budget      gross  \
0     United Kingdom   19000000   46998772
1      United States    4500000   58853106
2      United States   18000000  538375067
3      United States    3500000   83453539
4      United States    6000000   39846344
…                …        …        …
7648   United States   90000000  426505244
7649   United States   85000000  319715683
7650   United States  175000000  245487753
7651          Canada  135000000  111105497
7652           China   80000000  461421559

                                company  runtime
0                         Warner Bros.    146.0
1                      Columbia Pictures    104.0
2                              Lucasfilm    124.0
3                    Paramount Pictures     88.0
4                        Orion Pictures     98.0
…                                   …      …
7648                   Columbia Pictures    124.0
7649                 Paramount Pictures     99.0
7650                 Universal Pictures    101.0
7651              20th Century Studios     100.0
7652  Beijing Diqi Yinxiang Entertainment    149.0

[5421 rows x 15 columns]
```

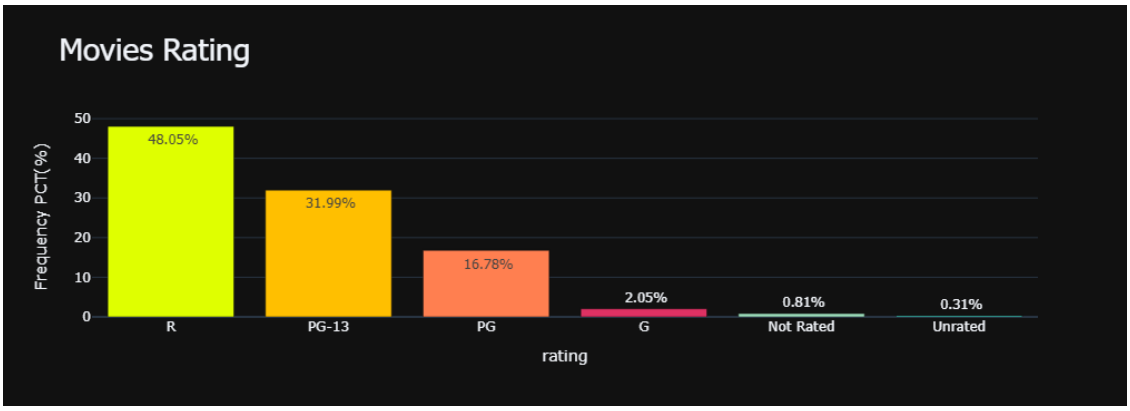
0.2 Exploratory analysis

0.2.1 Lets take a look at the movie ratings

[96]:
```
rating = df["rating"].value_counts()
(rating / df.shape[0] * 100).apply(lambda x: f"{x: 0.2f} %")
```

[96]:
```
rating
R             47.91 %
PG-13         31.89 %
PG            16.73 %
G              2.05 %
Not Rated      0.81 %
Unrated        0.31 %
NC-17          0.22 %
TV-MA          0.04 %
Approved       0.02 %
X              0.02 %
Name: count, dtype: object
```

[104]:
```
rating = rating[0:6]
fig = px.bar(data_frame= rating,
       x = rating.index,
       y = rating / sum(rating) * 100,
       color=rating.index,
       color_discrete_sequence=["#DFFF00", "#FFBF00", "#FF7F50", "#DE3163",␣
 ↪"#9FE2BF", "#40E0D0"],
       labels = {"index": "Movie Rating", "y" : "Frequency PCT(%)"},
       title = "Movies Rating",
       text = rating.apply(lambda x: f"{x / sum(rating) * 100: 0.2f}%"),
       template = "plotly_dark",
      )


update_layout(hover_bgcolor="#111")


fig.update_traces(
    textfont = {
        "family": "tahoma",
         "size": 13,
    },
    hovertemplate= "Rating: %{label}<br>Popularity: %{value:0.2f}%"
)
iplot(fig)
```



-

[ ]:

[ ]:

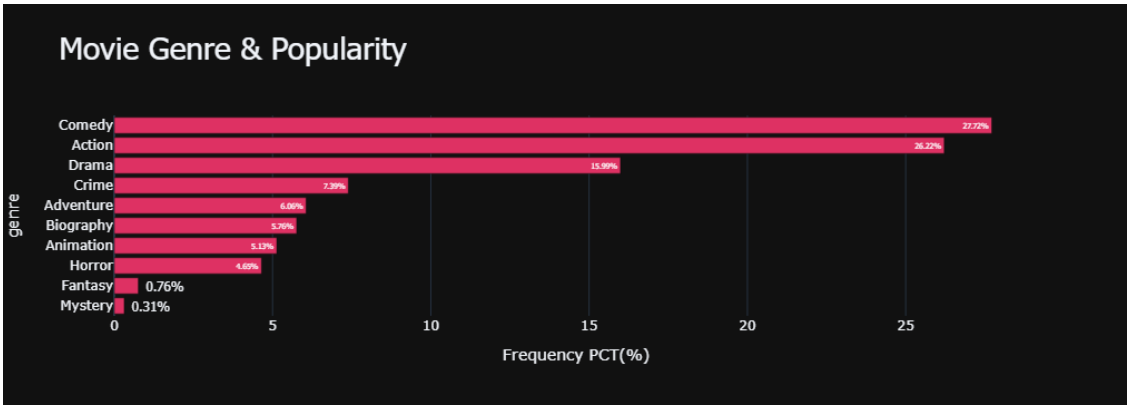
[ ]:

0.2.2 Lets take a look at which genre is most popular

[108]:
```
genre = df["genre"].value_counts()
(genre / sum(genre) * 100).apply(lambda x: f"{x:0.2f} %")
```

[108]:
```
genre
Comedy       27.60 %
Action       26.10 %
Drama        15.92 %
Crime         7.36 %
Adventure     6.03 %
Biography     5.74 %
Animation     5.11 %
Horror        4.63 %
Fantasy       0.76 %
Mystery       0.31 %
Thriller      0.13 %
Sci-Fi        0.11 %
Romance       0.09 %
Family        0.07 %
Western       0.04 %
Name: count, dtype: object
```

[110]:
```
genre = genre.nlargest(10)[::-1]
fig = px.bar(data_frame= genre,
             orientation = "h",
       x = genre / sum(genre) * 100,

       y = genre.index,
       color_discrete_sequence=["#DE3163"],
       labels = {"index": "Movie Genre", "x" : "Frequency PCT(%)"},
       title = "Movie Genre & Popularity",
       text = genre.apply(lambda x: f"{x / sum(genre) * 100: 0.2f}%"),
       template = "plotly_dark",
      )

fig.update_traces(
    textfont = {
        "family": "tahoma",
         "size": 13,
    },
    hovertemplate= "Rating: %{label}<br>Popularity: %{value:0.2f}%"
)

update_layout()

iplot(fig)
```



-

[ ]:

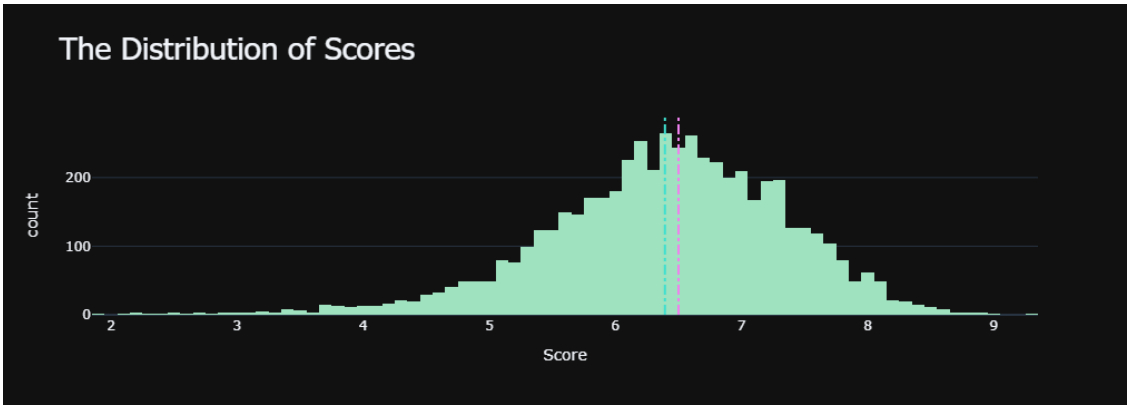
[ ]:

[ ]:

0.3 Lets take a look at each movie score

[114]:
```
fig = px.histogram(df["score"],
                   template = "plotly_dark",
                   color_discrete_sequence=["#9FE2BF"],
                   labels={"value" :"Score", "count" :"Frequency"},
                   title = "The Distribution of Scores",
                  )
##  Adding The Mean Line To The Histogram
fig.add_shape(type='line',
              x0=df["score"].mean(),
              y0=0,
              x1=df["score"].mean(),
              y1=df["score"].value_counts().max()+25,
              line = {
                  "color" :"#40E0D0",
                  "width" : 2,
                  "dash" : "dashdot"
              },
              label={
                  "text" : f"Mean: {df['score'].mean(): 0.1f}\t",
                  "textposition": "end",
                  "yanchor" :"top",
                  "xanchor" :"right",
                  "textangle" :0,
                  "font": {
                      "size": 14,
                      "color" :"#9FE2BF",
                      "family" : "tahoma"

                  },
              }
             )
##  Adding The Median Line To The Histogram
fig.add_shape(type='line',
              x0=df["score"].median(),
              y0=0,
              x1=df["score"].median(),
              y1=df["score"].value_counts().max()+25,
              line = {
                  "color" :"violet",
                  "width" : 2,
                  "dash" : "dashdot"
```

```
            },
            label={
                "text" :f"Median: {df['score'].median(): 0.1f}",
                "textposition": "end",
                "yanchor" :"top",
                "xanchor" :"left",
                "textangle" :0,
                "font": {
                    "size": 14,
                    "color" :"violet",
                    "family" : "tahoma"

                },
            }
           )
update_layout()

iplot(fig)
```



-

  0.3.1 So we can see that most movies have been given a score of between 6 and 7. Given that scores are out 10, we can conclude that the people enjoyed these movies

-

0.3.2 The distirbution seems to be negtaively skewed but we can also check below

[81]:
```
print(f"The Skew of The Score Data: {df['score'].skew(): 0.2f}")
```

```
The Skew of The Score Data: -0.63
```

[ ]:

[ ]:

[ ]:

0.3.3 RQ: Is the movie industry dying? Are production companies making as much money today compared to 10 years ago?

Let us sample a particular production company ex. Twentieth century Fox.

Here we have a list of all movies produced by Twentieth Century Fox, spanning over 3 decades

[9]:
```
df2 = df[df['company'] == 'Twentieth Century Fox']
df2
```

[9]:

	name	rating	genre	year	released	score	votes	director
28	Brubaker	R	Crime	1980	June 20, 1980 (United States)	7.2	17,000.0	Stuart Rosenberg
30	My Bodyguard	PG	Comedy	1980	September 26, 1980 (United States)	7.1	8,900.0	Tony Bill
77	Willie & Phil	R	Comedy	1980	August 15, 1980 (United States)	5.9	415.0	Paul Mazursky
131	The Final Conflict	R	Horror	1981	March 20, 1981 (United States)	5.6	19,000.0	Graham Baker
137	Eyewitness	R	Crime	1981	February 13, 1981 (United States)	6.0	4,500.0	Peter Yates
…	…	…	…	…	…	…	…	…
7245	Deadpool 2	R	Action	2018	May 18, 2018 (United States)	7.7	505,000.0	David Leitch
7246	Bohemian Rhapsody	PG-13	Biography	2018				
7259	The Predator	R	Action	2018				
7468	Alita: Battle Angel	PG-13	Action	2019				
7493	X-Men: Dark Phoenix	PG-13	Action	2019				

```
7246    November 2, 2018 (United States)    7.9 476,000.0      Bryan Singer
7259  September 14, 2018 (United States)    5.3 120,000.0       Shane Black
7468   February 14, 2019 (United States)    7.3 240,000.0  Robert Rodriguez
7493        June 7, 2019 (United States)    5.7 166,000.0     Simon Kinberg

                writer             star         country        budget  \
28         W.D. Richter   Robert Redford   United States   9,000,000.0
30          Alan Ormsby  Chris Makepeace   United States           NaN
77         Jean Gruault  Michael Ontkean   United States   5,500,000.0
131       David Seltzer        Sam Neill  United Kingdom   5,000,000.0
137        Steve Tesich     William Hurt   United States   8,500,000.0
…                 …                …               …           …
7245       Rhett Reese    Ryan Reynolds          Canada 110,000,000.0
7246  Anthony McCarten      Rami Malek  United Kingdom  52,000,000.0
7259       Fred Dekker    Boyd Holbrook   United States  88,000,000.0
7468     James Cameron     Rosa Salazar   United States 170,000,000.0
7493     Simon Kinberg     James McAvoy   United States 200,000,000.0

              gross                company  runtime
28    37,121,708.0  Twentieth Century Fox    131.0
30    22,482,952.0  Twentieth Century Fox    102.0
77     4,400,000.0  Twentieth Century Fox    115.0
131   20,471,382.0  Twentieth Century Fox    108.0
137    6,400,000.0  Twentieth Century Fox    103.0
…              …                     …      …
7245 786,470,484.0  Twentieth Century Fox    119.0
7246 911,902,649.0  Twentieth Century Fox    134.0
7259 160,542,134.0  Twentieth Century Fox    107.0
7468 404,980,543.0  Twentieth Century Fox    122.0
7493 252,442,974.0  Twentieth Century Fox    113.0

[240 rows x 15 columns]
```

[ ]:

If take a closer look, we can see that they made their highest paying movie in 2009. This was Avatar as shown below

In order to determine whether Century Fox is making less money today, we need to take a look at gross earnings of a movie they made 10 or 20 years before 2009

In the df below, If we take a look at 1989, we see that Avatar produced a movie called 'How I got into college', and this earned them just 1,642,239.0 which is significantly less compared to their earnings in 2009

This is one example that can help us conclude that movie production companies are still making profit from theatre box office, and people are still willing to watch a movie

at a cinema, despite recent development of Netflix and the sort

[13]:
```
df2.sort_values(by=['gross'], inplace=False, ascending=False)
```

[13]:
```
                        name rating      genre  year  \
5445                  Avatar  PG-13     Action  2009
3045                 Titanic  PG-13      Drama  1997
7246       Bohemian Rhapsody  PG-13  Biography  2018
2844        Independence Day  PG-13     Action  1996
7245              Deadpool 2      R     Action  2018
…                        …    …        …   …
621         The Buddy System     PG      Drama  1984
1616  How I Got Into College  PG-13     Comedy  1989
1831             Vital Signs      R      Drama  1990
1819   Come See the Paradise      R      Drama  1990
4853               Idiocracy      R  Adventure  2006

                                  released  score         votes  \
5445  December 18, 2009 (United States)    7.8 1,100,000.0
3045  December 19, 1997 (United States)    7.8 1,100,000.0
7246   November 2, 2018 (United States)    7.9   476,000.0
2844       July 3, 1996 (United States)    7.0   543,000.0
7245       May 18, 2018 (United States)    7.7   505,000.0
…                                    …    …           …
621    January 20, 1984 (United States)    5.7       794.0
1616       May 19, 1989 (United States)    5.8     2,100.0
1831     April 13, 1990 (United States)    5.4       703.0
1819       January 1991 (United States)    6.7     2,600.0
4853         January 25, 2007 (Germany)    6.6   150,000.0

                   director               writer               star  \
5445          James Cameron        James Cameron    Sam Worthington
3045          James Cameron        James Cameron  Leonardo DiCaprio
7246           Bryan Singer     Anthony McCarten         Rami Malek
2844        Roland Emmerich          Dean Devlin         Will Smith
7245           David Leitch          Rhett Reese      Ryan Reynolds
…                      …                  …                 …
621            Glenn Jordan  Mary Agnes Donoghue   Richard Dreyfuss
1616  Savage Steve Holland       Terrel Seltzer    Anthony Edwards
1831          Marisa Silver         Larry Ketron       Adrian Pasdar
1819            Alan Parker          Alan Parker       Dennis Quaid
4853             Mike Judge           Mike Judge        Luke Wilson

             country         budget            gross              company  \
5445   United States 237,000,000.0 2,847,246,203.0  Twentieth Century Fox
3045   United States 200,000,000.0 2,201,647,264.0  Twentieth Century Fox
7246  United Kingdom  52,000,000.0   911,902,649.0  Twentieth Century Fox
2844   United States  75,000,000.0   817,400,891.0  Twentieth Century Fox
```

```
7245          Canada 110,000,000.0   786,470,484.0  Twentieth Century Fox
…                …           …              …                      …
621    United States           NaN     1,820,049.0  Twentieth Century Fox
1616   United States  10,000,000.0     1,642,239.0  Twentieth Century Fox
1831   United States           NaN     1,224,605.0  Twentieth Century Fox
1819   United States  17,500,000.0       947,306.0  Twentieth Century Fox
4853   United States           NaN       495,303.0  Twentieth Century Fox

      runtime
5445    162.0
3045    194.0
7246    134.0
2844    145.0
7245    119.0
…         …
621     110.0
1616     86.0
1831    103.0
1819    138.0
4853     84.0

[240 rows x 15 columns]
```

[ ]:

[ ]:

0.3.4 RQ: Which movie has made the most money in terms of gross revenue in the last three decades?

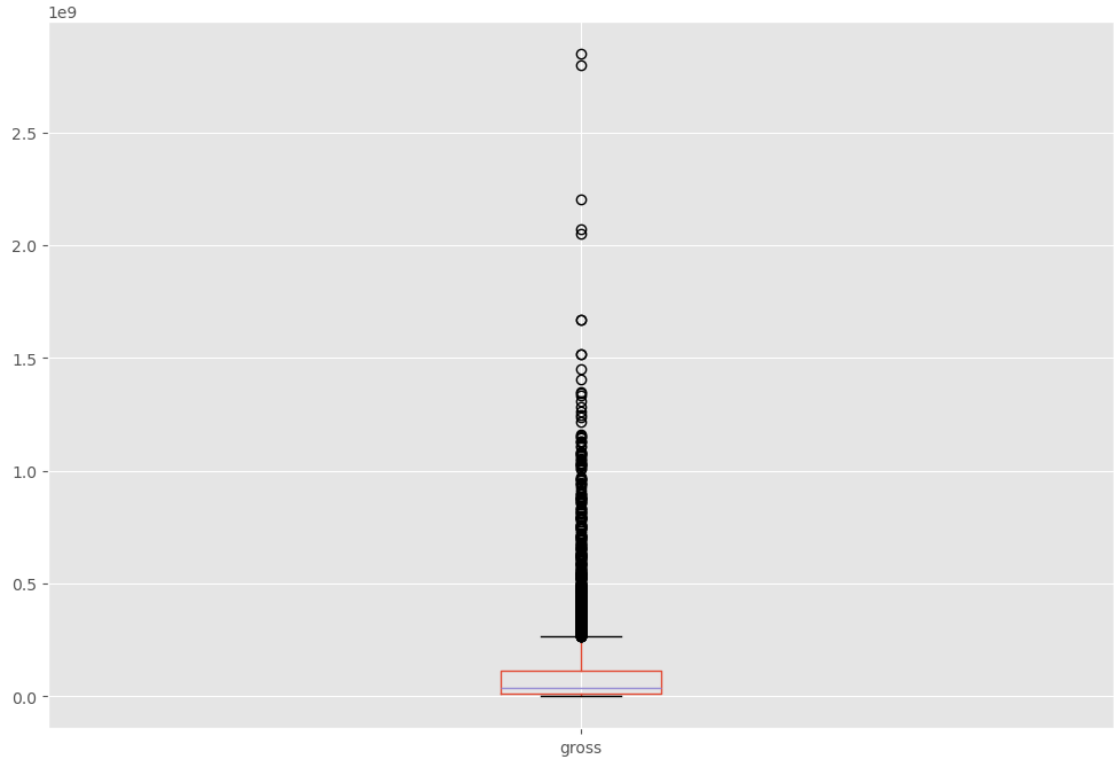
[28]:
```
# Order our Data a little bit to see
#just checking to see which movie makes the most money, that will be in terms␣
 ↪of gross
#gross is the only word in this data that means revenue
# We dont want to save the df like this so inplace=false
#if you put ascending= true youll see the movie that made the least amount of␣
 ↪money

df.sort_values(by=['gross'], inplace=False, ascending=False)
```

[28]:
```
                                          name     rating   genre  year  \
5445                                    Avatar      PG-13  Action  2009
7445                          Avengers: Endgame      PG-13  Action  2019
3045                                   Titanic      PG-13   Drama  1997
6663  Star Wars: Episode VII - The Force Awakens      PG-13  Action  2015
7244                      Avengers: Infinity War      PG-13  Action  2018
…                                           …         …      …     …
```

```
5640                                Tanner Hall          R   Drama  2009
2434                Philadelphia Experiment II      PG-13  Action  1993
3681                               Ginger Snaps  Not Rated   Drama  2000
272                                    Parasite          R  Horror  1982
3203                                Trojan War      PG-13  Comedy  1997

                             released  score      votes  \
5445  December 18, 2009 (United States)    7.8  1100000.0
7445     April 26, 2019 (United States)    8.4   903000.0
3045  December 19, 1997 (United States)    7.8  1100000.0
6663  December 18, 2015 (United States)    7.8   876000.0
7244     April 27, 2018 (United States)    8.4   897000.0
…                                   …    …        …
5640          January 15, 2015 (Sweden)    5.8     3500.0
2434         June 4, 1994 (South Korea)    4.5     1900.0
3681              May 11, 2001 (Canada)    6.8    43000.0
272      March 12, 1982 (United States)    3.9     2300.0
3203           October 1, 1997 (Brazil)    5.7     5800.0

                director                    writer               star  \
5445         James Cameron             James Cameron    Sam Worthington
7445         Anthony Russo        Christopher Markus  Robert Downey Jr.
3045         James Cameron             James Cameron  Leonardo DiCaprio
6663           J.J. Abrams           Lawrence Kasdan       Daisy Ridley
7244         Anthony Russo        Christopher Markus  Robert Downey Jr.
…                      …                         …                  …
5640  Francesca Gregorini  Tatiana von Fürstenberg        Rooney Mara
2434     Stephen Cornwell        Wallace C. Bennett       Brad Johnson
3681          John Fawcett              Karen Walton      Emily Perkins
272           Charles Band             Alan J. Adler    Robert Glaudini
3203          George Huang                 Andy Burg       Will Friedle

           country     budget       gross                    company  runtime
5445  United States  237000000  2847246203      Twentieth Century Fox    162.0
7445  United States  356000000  2797501328             Marvel Studios    181.0
3045  United States  200000000  2201647264      Twentieth Century Fox    194.0
6663  United States  245000000  2069521700                  Lucasfilm    138.0
7244  United States  321000000  2048359754             Marvel Studios    149.0
…               …          …           …                         …      …
5640  United States    3000000        5073           Two Prong Lesson     96.0
2434  United States    5000000        2970           Trimark Pictures     97.0
3681         Canada    5000000        2554  Copperheart Entertainment    108.0
272   United States     800000        2270           Embassy Pictures     85.0
3203  United States   15000000         309                   Daybreak     85.0

[5421 rows x 15 columns]
```

[30]:
```
#any outliers?
#are there any movies making a lot more money?
#yes as you can see

df.boxplot(column=['gross'])
```

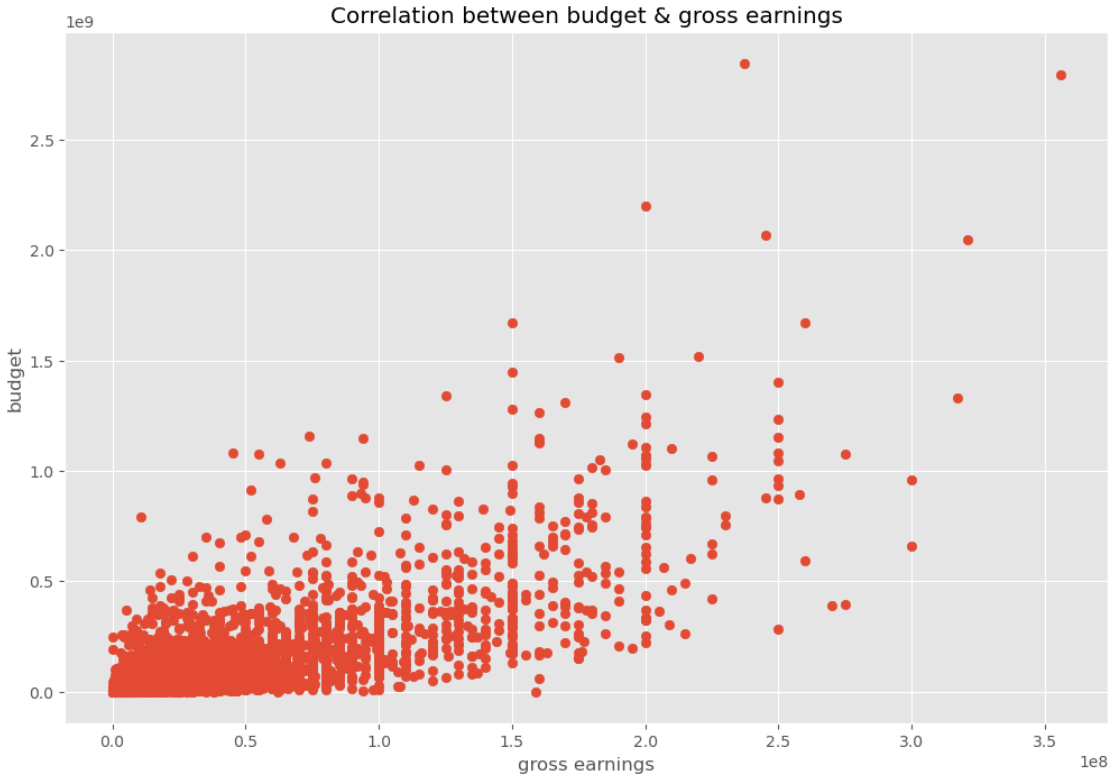
[30]: <Axes: >



0.3.5 RQ:What is the relationship between budget and gross?

[32]:
```
#what is the relationship between budget and gross?
#correlation?
#assumption is that there is a strong positive correlation such that the more␣
 ↪inside the budget, the greater the returns

plt.scatter(x=df['budget'], y=df['gross'])

plt.title("Correlation between budget & gross earnings")

plt.xlabel("gross earnings")

plt.ylabel("budget")
```

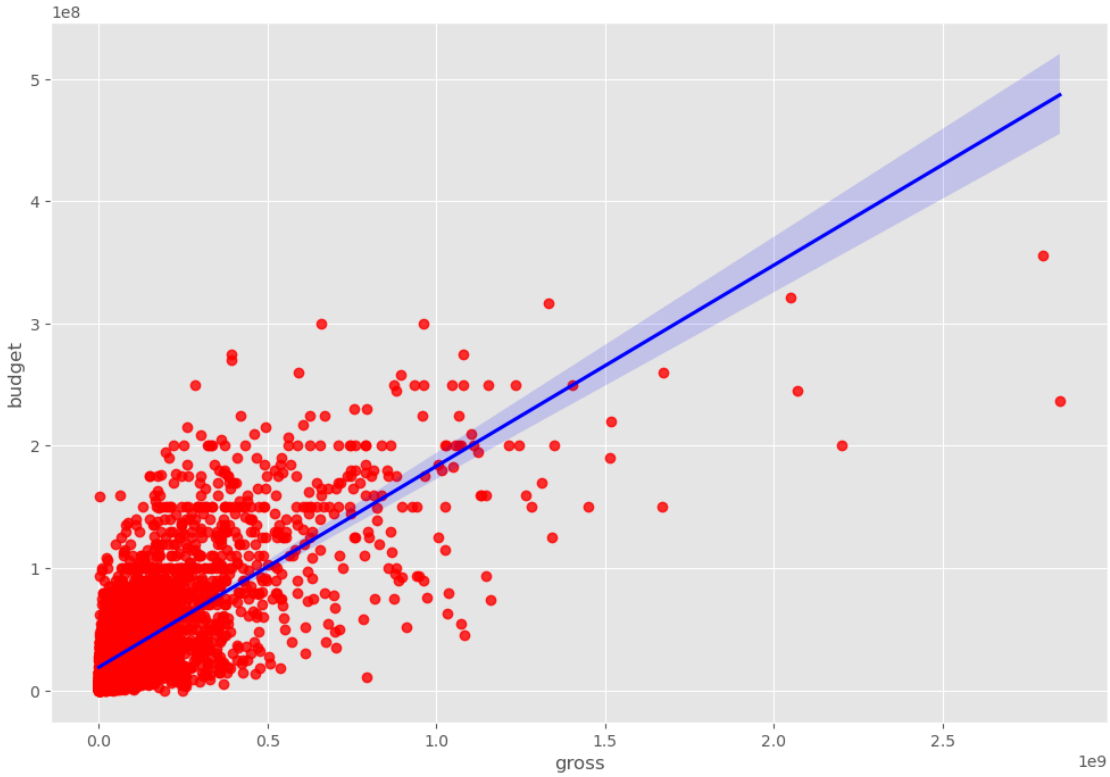
```
plt.show()
```



[ ]:

[74]:
```
#short cut of doing everything we just did above
#plot the scatter
#add the line


sns.regplot(x="gross", y="budget", data=df, scatter_kws={'color': 'red'},␣
↪line_kws={'color': 'blue'})
```

[74]:
```
<Axes: xlabel='gross', ylabel='budget'>
```



[39]:
```
#pearson, kendall, spearman #three different types of correlation
#pearson is ussually default
#as you can see, there is a strong posiitive correlation between budget and␣
 ↪gross of 0.740
#our asumption was correct

correlation_matrix = df.corr(method='pearson', numeric_only=True)
correlation_matrix




# sns.heatmap(correlation_matrix, annot=True)

# plt.show()
```

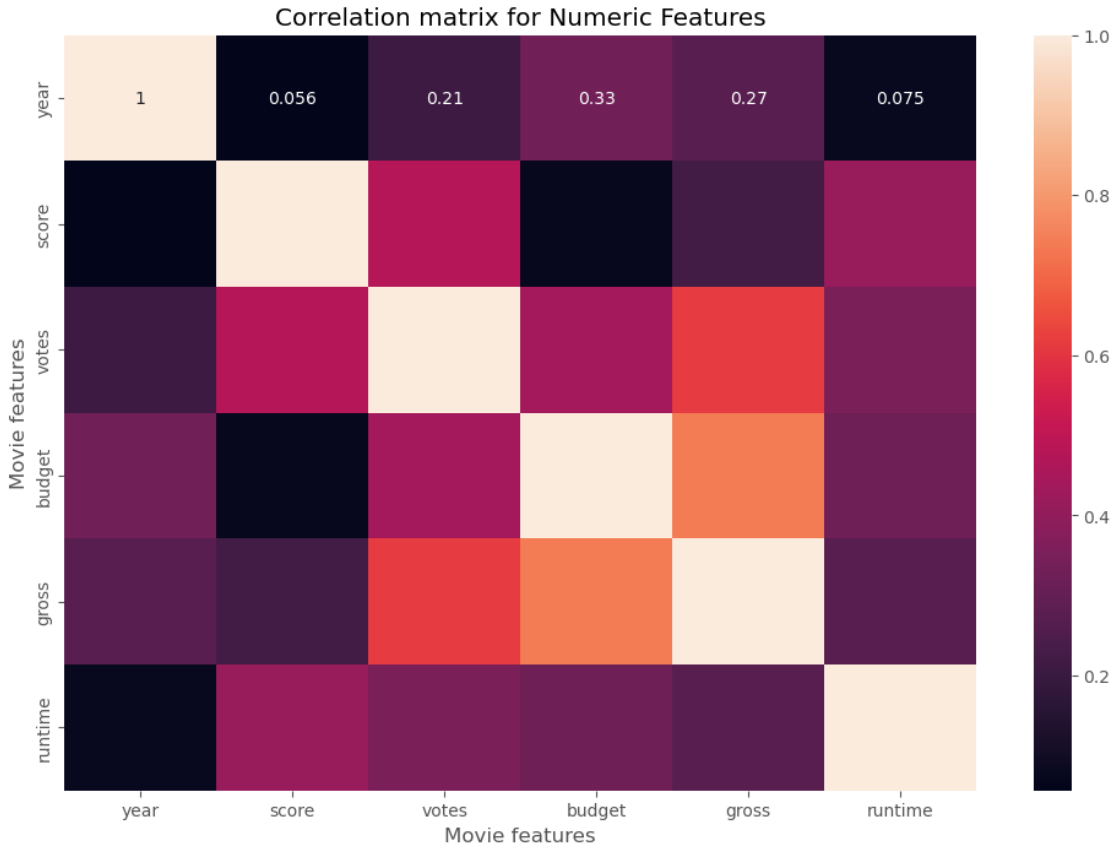
[39]:

	year	score	votes	budget	gross	runtime
year	1.000000	0.056386	0.206021	0.327722	0.274321	0.075077
score	0.056386	1.000000	0.474256	0.072001	0.222556	0.414068
votes	0.206021	0.474256	1.000000	0.439675	0.614751	0.352303
budget	0.327722	0.072001	0.439675	1.000000	0.740247	0.318695

```
gross     0.274321  0.222556  0.614751  0.740247  1.000000  0.275796
runtime   0.075077  0.414068  0.352303  0.318695  0.275796  1.000000
```

[ ]:

[46]:
```
sns.heatmap(correlation_matrix, annot=True)

plt.title("Correlation matrix for Numeric Features")

plt.xlabel("Movie features")

plt.ylabel("Movie features")

plt.show()
```



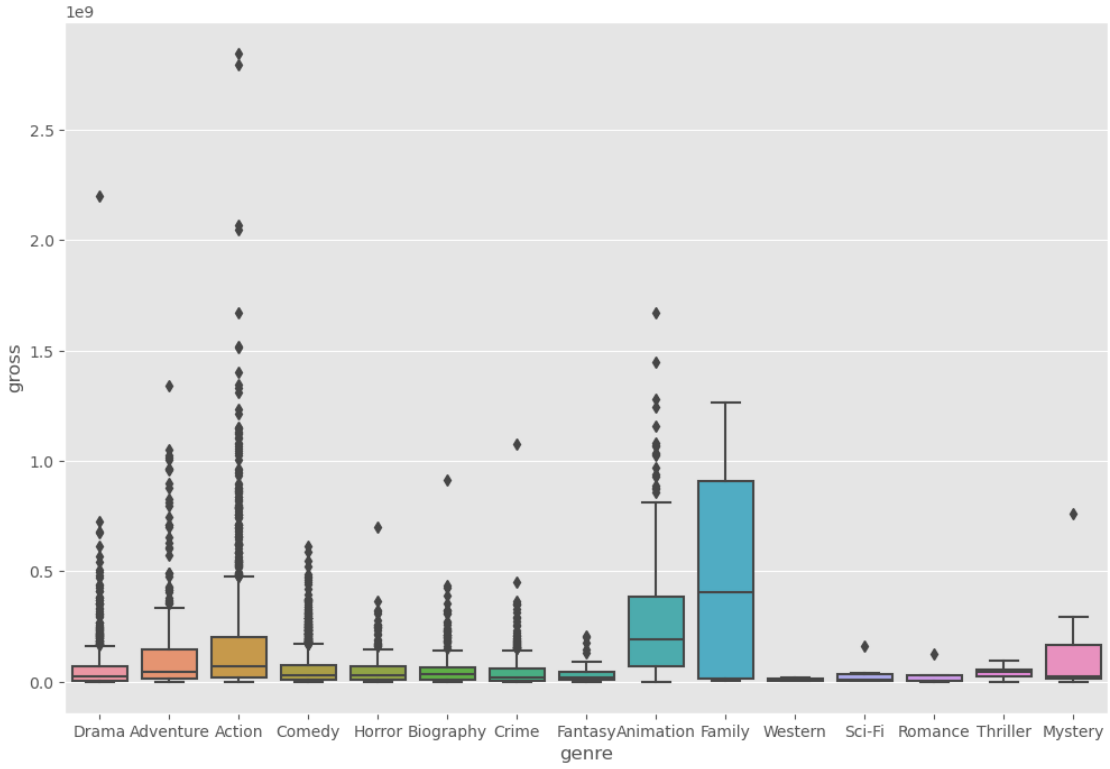
[ ]:

RQ: Does a particular genre make more gross income? The other relationship that would interesting to explore is that between genre and gross income. Does a particular genre provide more returns? To look at the relationship between these two, we will start with a box-and-whisker

plot that provides an idea of the distribution of gross income for the different genres

[48]:
```
sns.boxplot(x="genre", y="gross", data=df)
```

[48]:
```
<Axes: xlabel='genre', ylabel='gross'>
```



We can see that the distributions of gross income between the different genre categories have a significant overlap. There is no particular genre that is making significantly more income on average.

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: