

code

May 28, 2023

Final Project Submission

Please fill out:

Student name: Paul Gitonga Njoki

Student pace: full time:

Scheduled project review date/time:

Instructor name:

Blog post URL:

BEST PERFORMING MOVIE ANALYSIS

Authored by Paul Gitonga Njoki

Overview

Microsoft as a company wants to start on creating original video content but do not have enough knowledge about movie creation to move forward with their plan. Using data obtained from the Rotten Tomatoes and Box Office Mojo datasets for analysis, it helped in discovering patterns and relationships in the data in order to make better decisions and recommendations that Microsoft will use in order for them to venture into movie crteation.

DATA UNDERSTANDING

Data that is used for this task was obtained from movie websites. I chose to work with two data sets that is the Rotten Tomatoes and Box Office Mojo datasets. After importing the necessary libraries to be used, we then read the data and understand its strucrtute,data contained and cleaning it before we go ahead to analyzing them to give us efficient information about movies before making conclusions.

```
[ ]: #loading Libraries
import pandas as pd
import numpy as np

import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

```
from scipy import stats
from scipy.stats import norm
```

```
[ ]: #reading ROTTEN TOMATOES data from the tsv file
rtmovie_df = pd.read_csv("learn-co-curriculum dsc-phase-1-project master_
↳zippedData/rt.movie_info.tsv.gz", sep='\t', header=0)
rtmovie_df.head()
```

```
[ ]:      id                                     synopsis rating
0    1  This gritty, fast-paced, and innovative police...      R \
1    3  New York City, not-too-distant-future: Eric Pa...      R
2    5  Illeana Douglas delivers a superb performance ...      R
3    6  Michael Douglas runs afoul of a treacherous su...      R
4    7                                     NaN      NR

      genre                                     director
0  Action and Adventure|Classics|Drama  William Friedkin \
1    Drama|Science Fiction and Fantasy  David Cronenberg
2    Drama|Musical and Performing Arts    Allison Anders
3          Drama|Mystery and Suspense    Barry Levinson
4          Drama|Romance      Rodney Bennett

      writer  theater_date  dvd_date  currency
0    Ernest Tidyman   Oct 9, 1971  Sep 25, 2001    NaN \
1  David Cronenberg|Don DeLillo  Aug 17, 2012   Jan 1, 2013     $
2          Allison Anders  Sep 13, 1996  Apr 18, 2000    NaN
3  Paul Attanasio|Michael Crichton  Dec 9, 1994  Aug 27, 1997    NaN
4          Giles Cooper           NaN           NaN    NaN

      box_office  runtime  studio
0          NaN  104 minutes    NaN
1    600,000  108 minutes  Entertainment One
2          NaN  116 minutes    NaN
3          NaN  128 minutes    NaN
4          NaN  200 minutes    NaN
```

Rotten Tomatoes Data

```
[ ]: #dropping unwanted columns
rtmovie_df = rtmovie_df.drop(rtmovie_df.columns[0], axis='columns')
```

```
[ ]: rtmovie_df.shape
```

```
[ ]: (1560, 11)
```

```
[ ]: #checking the columns in the dataframe

rtmovie_df.columns
```

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

```
[ ]: #getting the summary of rotten tomatoes dataframe
```

```
rtmovie_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1560 entries, 0 to 1559
Data columns (total 11 columns):
#   Column          Non-Null Count  Dtype
---  -
0   synopsis        1498 non-null   object
1   rating          1557 non-null   object
2   genre           1552 non-null   object
3   director        1361 non-null   object
4   writer          1111 non-null   object
5   theater_date    1201 non-null   object
6   dvd_date        1201 non-null   object
7   currency        340 non-null    object
8   box_office      340 non-null    object
9   runtime         1530 non-null   object
10  studio          494 non-null    object
dtypes: object(11)
memory usage: 134.2+ KB
```

```
[ ]: # Checking for null values
```

```
rtmovie_df.isnull().sum()
```

```
[ ]: synopsis        62
     rating          3
     genre           8
     director       199
     writer        449
     theater_date   359
     dvd_date       359
     currency     1220
     box_office    1220
     runtime        30
     studio       1066
     dtype: int64
```

```
[ ]: # obtaining counts for each value in genre column#
```

```
rtmovie_df['genre'].value_counts()
```

```
[ ]: genre
Drama
151
Comedy
110
Comedy|Drama
80
Drama|Mystery and Suspense
67
Art House and International|Drama
62
...
Art House and International|Drama|Sports and Fitness
1
Comedy|Documentary|Musical and Performing Arts|Special Interest
1
Comedy|Cult Movies|Mystery and Suspense|Science Fiction and Fantasy
1
Action and Adventure|Art House and International|Mystery and Suspense|Special
Interest      1
Comedy|Drama|Kids and Family|Sports and Fitness
1
Name: count, Length: 299, dtype: int64
```

```
[ ]: #obtaining descriptive statistics for the genre column to determine the top genre
rtmovie_df['genre'].describe()
```

```
[ ]: count      1552
unique        299
top           Drama
freq          151
Name: genre, dtype: object
```

Data for Box Office Mojo

```
[ ]: #calling the box office mojo data from the csv file
#checking the first 5 elements of the dataframe

mojo_df = pd.read_csv("learn-co-curriculum dsc-phase-1-project master_
↳zippedData/bom.movie_gross.csv.gz")
mojo_df.head()
```

```
[ ]:
           title studio  domestic_gross \
0           Toy Story 3      BV    415000000.0
1  Alice in Wonderland (2010)    BV    334200000.0
2  Harry Potter and the Deathly Hallows Part 1    WB    296000000.0
3           Inception      WB    292600000.0
4  Shrek Forever After  P/DW    238700000.0
```

```

    foreign_gross  year
0      652000000  2010
1      691300000  2010
2      664300000  2010
3      535700000  2010
4      513900000  2010

```

```
[ ]: #obtaining columns in the dataframe
```

```
mojo_df.columns
```

```
[ ]: Index(['title', 'studio', 'domestic_gross', 'foreign_gross', 'year'],
          dtype='object')
```

```
[ ]: #obtaining number of columns and rows
```

```
mojo_df.shape
```

```
[ ]: (3387, 5)
```

```
[ ]: #getting data types per column
```

```
mojo_df.dtypes
```

```
[ ]: title           object
     studio          object
     domestic_gross  float64
     foreign_gross   object
     year            int64
     dtype: object
```

```
[ ]: # getting total number of NaN values in the dataset
```

```
mojo_df.isna().sum()
```

```
[ ]: title           0
     studio           5
     domestic_gross   28
     foreign_gross   1350
     year              0
     dtype: int64
```

```
[ ]: #checking the summary of the mojo dataframe
```

```
mojo_df.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

```
[ ]: # top studios
```

```
top10 = mojo_df['studio'].value_counts().head()
top10
```

```
[ ]: studio
```

```
IFC      166
Uni.     147
WB       140
Fox      136
Magn.    136
```

```
Name: count, dtype: int64
```

```
[ ]: type(top10)
```

```
[ ]: pandas.core.series.Series
```

```
[ ]: mojo_df.groupby(['studio']).sum()
```

```
[ ]:
      title  domestic_gross
studio
3D      Sea Rex 3D: Journey to a Prehistoric World  6100000.0 \
A23      Revenge of the Electric CarRed Obsession  164200.0
A24  Spring BreakersThe Bling RingThe Spectacular N...  324194200.0
ADC      A Royal Night OutAbsolutely Anything  248200.0
AF  BarbaraSister (2012)Caesar Must DieOmarBethleh...  2142900.0
...
XL      Storm Surfers 3DJimi: All Is By My Side  458000.0
YFG      Papa: Hemingway in Cuba  1100000.0
Yash  Band Baaja BaaraatBadmaash CompanyMere Brother...  31631400.0
Zee      Rustom  1100000.0
Zeit.  Mid-August LunchVisionLast Train HomeThe TreeB...  5663500.0

      foreign_gross  year
studio
3D      9900000  2010
A23      0  4024
```

```

A24      1760000013300000630000028000002400000210000011... 98754
ADC                                           0      4032
AF              3100000400000      12080
...
XL                                           0      4027
YFG                                           0      2016
Yash      6070064400800000008110000025900000818000003900000 28194
Zee                                           571000      2016
Zeit.              87000004600000210000012000003700000 32206

```

[257 rows x 4 columns]

```

[ ]: #descriptive statistics for each column

mojo_df['domestic_gross'].describe()

```

```

[ ]: count      3.359000e+03
      mean      2.874585e+07
      std       6.698250e+07
      min       1.000000e+02
      25%       1.200000e+05
      50%       1.400000e+06
      75%       2.790000e+07
      max       9.367000e+08
      Name: domestic_gross, dtype: float64

```

```

[ ]: #checking the years we'll be working with

mojo_df.year.unique()

```

```

[ ]: array([2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018], dtype=int64)

```

DATA PREPARATION

I prepared the data for analysis by performing data preparation, which included data cleaning, after deciding the preferred data sets to employ.

We will perform the following after data cleaning:

1. Inspect for and remove any unnecessary columns.
2. Standardization, column renaming, and data type conversion were required. Upper case values.
3. Verify any null values and remove them.
4. Search for missing values, then take appropriate action.
5. Check for duplicate values and remove them if necessary.

Regarding the BOM Data;

```

[ ]: #dropping columns in the dataframe that won't be needed during analysis

mojo_df.drop(['title'], axis=1, inplace=True)

```

```
[ ]: mojo_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3387 entries, 0 to 3386
Data columns (total 4 columns):
#   Column          Non-Null Count  Dtype
---  -
0   studio           3382 non-null   object
1   domestic_gross   3359 non-null   float64
2   foreign_gross    2037 non-null   object
3   year             3387 non-null   int64
dtypes: float64(1), int64(1), object(2)
memory usage: 106.0+ KB
```

```
[ ]: # checking if there are any duplication
mojo_df.duplicated().value_counts()
```

```
[ ]: False      3376
      True       11
      Name: count, dtype: int64
```

```
[ ]: # missing values inspection
row_count = mojo_df.shape[0]
missing_count = row_count - mojo_df.count()
missing_count
```

```
[ ]: studio           5
      domestic_gross   28
      foreign_gross    1350
      year             0
      dtype: int64
```

```
[ ]: # duplicates check
duplicateRows = mojo_df[mojo_df.duplicated()]
duplicateRows.count()
```

```
[ ]: studio           11
      domestic_gross   11
      foreign_gross     0
      year             11
      dtype: int64
```

```
[ ]: # checking null values

mojo_df.isnull().any()
```

```
[ ]: studio           True
      domestic_gross   True
```



```
foreign_gross    True
year            False
dtype: bool
```

```
[ ]: # duplicates check

mojo_df.duplicated().sum()
```

```
[ ]: 11
```

For Rotten Tomatoes Data:

I started by dropping columns that i will not need in the analysis

```
[ ]: rtmovie_df.columns
```

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

```
[ ]: rtmovie_df
```

```
[ ]:
                                synopsis rating
0      This gritty, fast-paced, and innovative police...      R \
1      New York City, not-too-distant-future: Eric Pa...      R
2      Illeana Douglas delivers a superb performance ...      R
3      Michael Douglas runs afoul of a treacherous su...      R
4                                     NaN      NR
...
1555  Forget terrorists or hijackers -- there's a ha...      R
1556  The popular Saturday Night Live sketch was exp...     PG
1557  Based on a novel by Richard Powell, when the l...      G
1558  The Sandlot is a coming-of-age story about a g...     PG
1559  Suspended from the force, Paris cop Hubert is ...      R

                                genre      director \
0      Action and Adventure|Classics|Drama  William Friedkin \
1      Drama|Science Fiction and Fantasy    David Cronenberg
2      Drama|Musical and Performing Arts     Allison Anders
3      Drama|Mystery and Suspense           Barry Levinson
4      Drama|Romance                        Rodney Bennett
...
1555  Action and Adventure|Horror|Mystery and Suspense      NaN
1556  Comedy|Science Fiction and Fantasy                    Steve Barron
1557  Classics|Comedy|Drama|Musical and Performing Arts      Gordon Douglas
1558  Comedy|Drama|Kids and Family|Sports and Fitness      David Mickey Evans
1559  Action and Adventure|Art House and Internation...      NaN
```

		writer	theater_date	
0		Ernest Tidyman	Oct 9, 1971	\
1		David Cronenberg Don DeLillo	Aug 17, 2012	
2		Allison Anders	Sep 13, 1996	
3		Paul Attanasio Michael Crichton	Dec 9, 1994	
4		Giles Cooper	NaN	
...		
1555		NaN	Aug 18, 2006	
1556	Terry Turner Tom Davis Dan Aykroyd Bonnie Turner		Jul 23, 1993	
1557		NaN	Jan 1, 1962	
1558	David Mickey Evans Robert Gunter		Apr 1, 1993	
1559		Luc Besson	Sep 27, 2001	

	dvd_date	currency	box_office	runtime	studio
0	Sep 25, 2001	NaN	NaN	104 minutes	NaN
1	Jan 1, 2013	\$	600,000	108 minutes	Entertainment One
2	Apr 18, 2000	NaN	NaN	116 minutes	NaN
3	Aug 27, 1997	NaN	NaN	128 minutes	NaN
4	NaN	NaN	NaN	200 minutes	NaN
...
1555	Jan 2, 2007	\$	33,886,034	106 minutes	New Line Cinema
1556	Apr 17, 2001	NaN	NaN	88 minutes	Paramount Vantage
1557	May 11, 2004	NaN	NaN	111 minutes	NaN
1558	Jan 29, 2002	NaN	NaN	101 minutes	NaN
1559	Feb 11, 2003	NaN	NaN	94 minutes	Columbia Pictures

[1560 rows x 11 columns]

```
[ ]: #total NaN values in the data set
rtmovie_df.isna().sum()
```

```
[ ]: synopsis      62
rating            3
genre             8
director          199
writer            449
theater_date      359
dvd_date          359
currency          1220
box_office        1220
runtime           30
studio           1066
dtype: int64
```

```
[ ]: #missing values check
row_count = rtmovie_df.shape[0]
missing_count = row_count - rtmovie_df.count()
```

```
missing_count
```

```
[ ]: synopsis      62
     rating        3
     genre         8
     director     199
     writer       449
     theater_date  359
     dvd_date     359
     currency     1220
     box_office    1220
     runtime       30
     studio       1066
     dtype: int64
```

Data Analysis

mojo_df Analysis for top 10 studios against their domestic gross

```
[ ]: #sorting the data for top to studios
```

```
mojo_dfagg = mojo_df.groupby(['studio']).agg('sum')
```

```
[ ]: mojo_dfagg = mojo_dfagg.sort_values('domestic_gross', ascending=False).head(10)
     mojo_dfagg
```

```
[ ]:      domestic_gross      foreign_gross
     studio
     BV      1.841903e+10  6520000006913000003910000002280000002456000001... \
     Uni.      1.290239e+10  2916000002164000001622000007780000059800000642...
     WB      1.216805e+10  6643000005357000003300000001112000001013000006...
     Fox      1.094950e+10  3113000001855000001946000001377000001000000008...
     Sony      8.459683e+09  1825000001752000002107000001094000001280000001...
     Par.      7.685871e+09  3115000001879000001668000008100000092800000545...
     LGF      4.118963e+09  1714000009040000051100000481000002670000046300...
     WB (NL)    3.995700e+09  1930000001060000005260000024200000921000001153...
     LG/S      2.078200e+09  5374000001613000001054000002960000027600000870...
     P/DW      1.682900e+09  5139000002773000001735000001380000077140000050...

     year
     studio
     BV      213451
     Uni.     296082
     WB      281941
     Fox      273882
     Sony     221575
     Par.     203417
```

```
LGF      207437
WB (NL)   90644
LG/S      82599
P/DW      20109
```

```
[ ]: mojo_dfagg.index
```

```
[ ]: Index(['BV', 'Uni.', 'WB', 'Fox', 'Sony', 'Par.', 'LGF', 'WB (NL)', 'LG/S',
          'P/DW'],
          dtype='object', name='studio')
```

```
[ ]: #bar graph plot for top 10 studios domestic gross

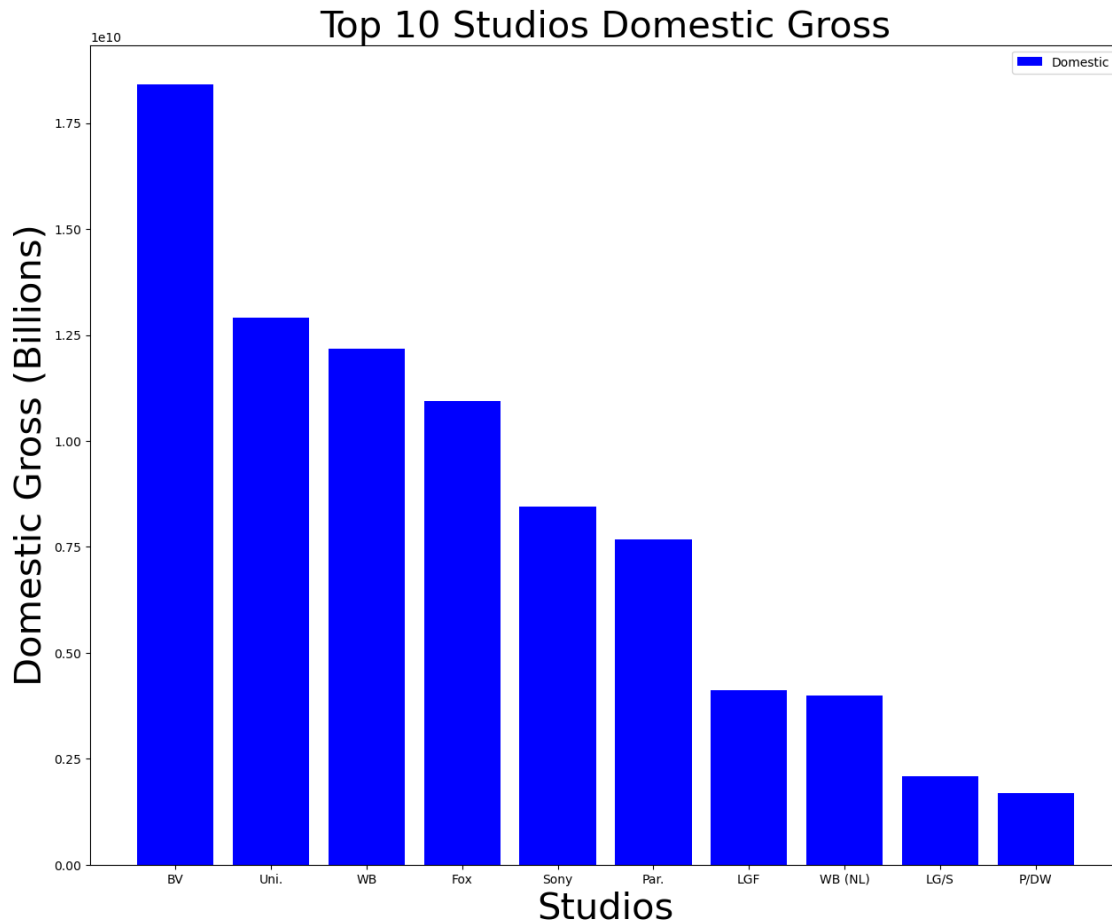
plt.figure(figsize=(15,12))

studios = mojo_dfagg.index
dom_gross = mojo_dfagg.domestic_gross

plt.bar(range(len(studios)), dom_gross, color='blue')

plt.title('Top 10 Studios Domestic Gross', fontsize=30)
plt.xlabel('Studios', fontsize=30)
plt.ylabel('Domestic Gross (Billions)', fontsize=30)
plt.xticks(range(len(studios)), studios)

plt.legend(['Domestic'])
plt.show();
```



which genre of movie is produced more?

```
[ ]: #as par to rtmovie_df
rtmovie_df.genre.value_counts()
```

```
[ ]: genre
Drama
151
Comedy
110
Comedy|Drama
80
Drama|Mystery and Suspense
67
Art House and International|Drama
62
...
```

```

Art House and International|Drama|Sports and Fitness
1
Comedy|Documentary|Musical and Performing Arts|Special Interest
1
Comedy|Cult Movies|Mystery and Suspense|Science Fiction and Fantasy
1
Action and Adventure|Art House and International|Mystery and Suspense|Special
Interest      1
Comedy|Drama|Kids and Family|Sports and Fitness
1
Name: count, Length: 299, dtype: int64

```

```

[ ]: #Frequency of movie genres

rtmovie_df['first_genre'] = rtmovie_df['genre'].str.split(',').str[0]

a = plt.cm.cool

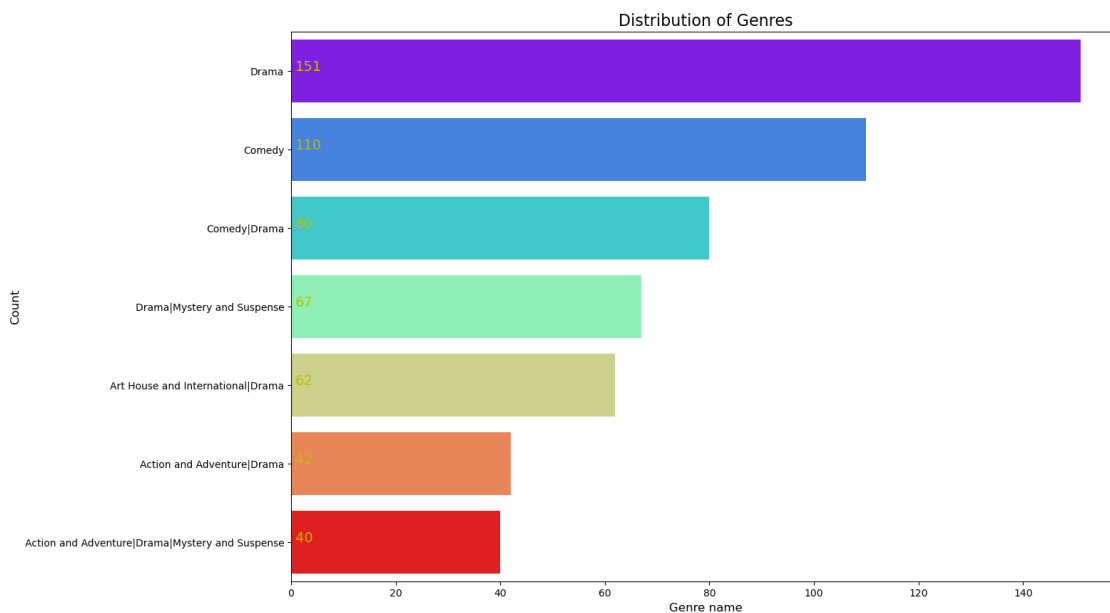
plt.figure(figsize=(15,10))
count = rtmovie_df['first_genre'].value_counts()[:7]
sns.barplot(x=count.values,y=count.index, palette=plt.cm.rainbow(np.linspace(0,1,7)))
for i, v in enumerate(count.values):
    plt.text(0.8,i,v,color='y',fontsize=14)
plt.xlabel('Genre name', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.title("Distribution of Genres", fontsize=16)

```

```

[ ]: Text(0.5, 1.0, 'Distribution of Genres')

```



By looking at the overall amount of films created in each genre, we can observe that drama films are produced more frequently than comedies, with a combination of art house and international/classics/mystery and Suspense being the least frequently produced.

```
[ ]: #viewing all the unique ratings in the dataframe
```

```
rtmovie_df['rating'].unique()
```

```
[ ]: array(['R', 'NR', 'PG', 'PG-13', nan, 'G', 'NC17'], dtype=object)
```

```
[ ]: #checking on the total number of counts per genre:
```

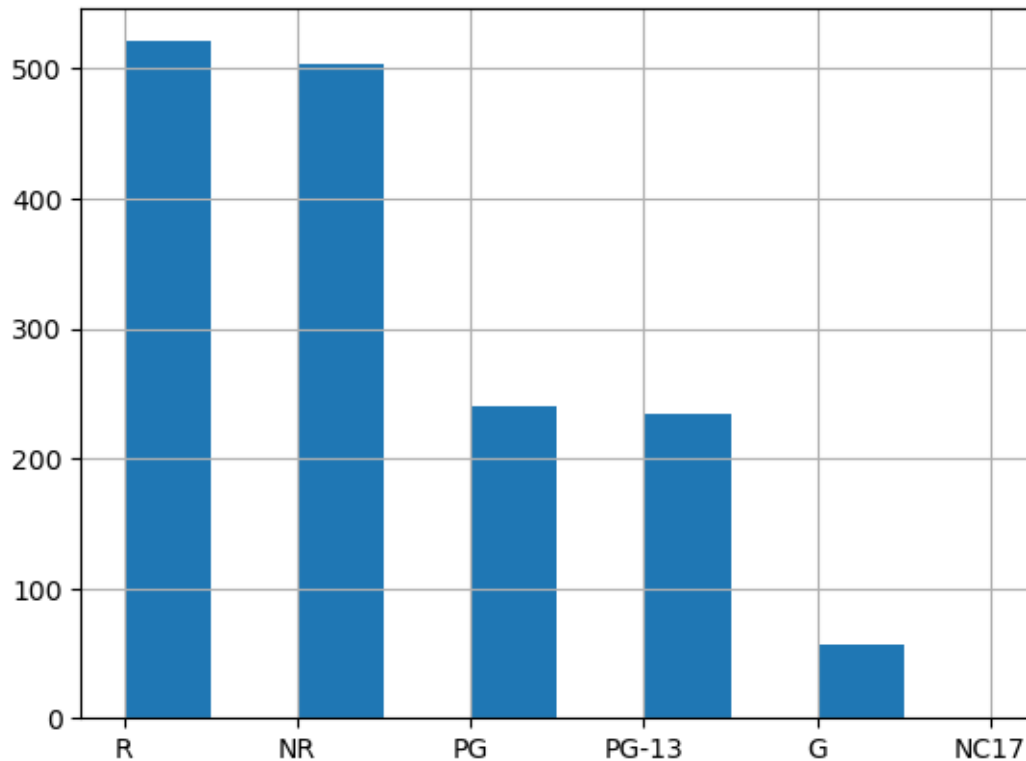
```
rtmovie_df['rating'].value_counts()
```

```
[ ]: rating
     R      521
     NR     503
     PG     240
     PG-13   235
     G       57
     NC17      1
     Name: count, dtype: int64
```

```
[ ]: #visualizing this on a histogram, we'll have
```

```
rtmovie_df['rating'].hist(bins=10)
```

```
[ ]: <Axes: >
```



By evaluating the total number of films under each category, we deduce that films with a R rating are being produced more frequently than those with an NC17 rating.

```
[ ]: # convert release date column to datetime values
rtmovie_df['dvd_date'] = pd.to_datetime(rtmovie_df['dvd_date'])
# create release month column
rtmovie_df['release_month'] = rtmovie_df['dvd_date'].dt.strftime('%B')
```

```
[ ]: # checking for successful column creation
rtmovie_df['release_month'].value_counts()
```

```
[ ]: release_month
March      128
May        117
October    110
September  107
February   107
November   102
June       99
August     98
April      93
January    81
```

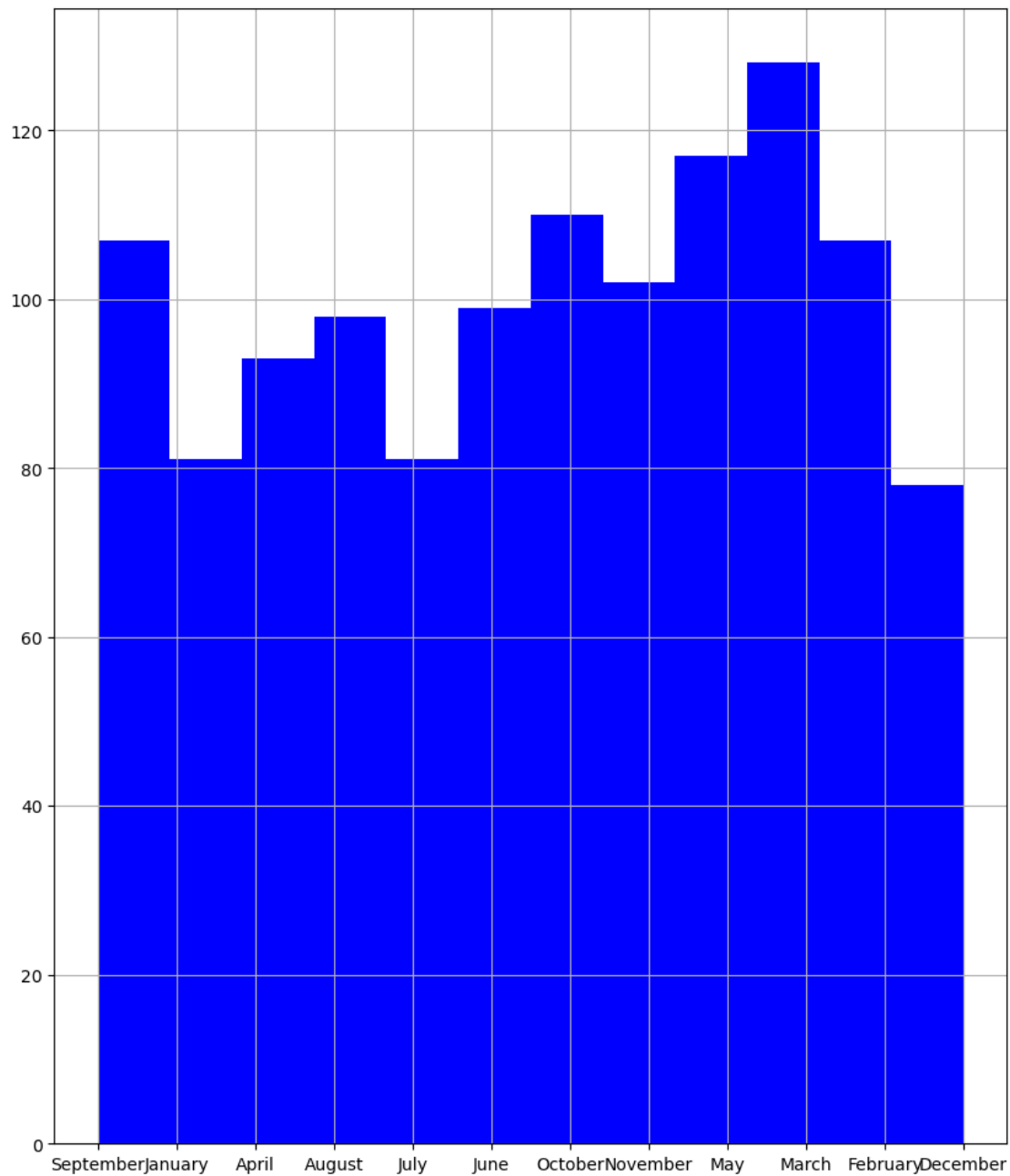


```
July          81
December      78
Name: count, dtype: int64
```

```
[ ]: #visualizing this on a histogram, we'll have
```

```
rtmovie_df['release_month'].hist(bins=12, figsize=(10,12), color=('b'))
```

```
[ ]: <Axes: >
```



```
[ ]: ##Assesing df
rtmovie_df
```

```
[ ]:
                                synopsis rating
0      This gritty, fast-paced, and innovative police...      R \
1      New York City, not-too-distant-future: Eric Pa...      R
2      Illeana Douglas delivers a superb performance ...      R
3      Michael Douglas runs afoul of a treacherous su...      R
4                                     NaN      NR
...
1555  Forget terrorists or hijackers -- there's a ha...      R
1556  The popular Saturday Night Live sketch was exp...      PG
1557  Based on a novel by Richard Powell, when the l...      G
1558  The Sandlot is a coming-of-age story about a g...      PG
1559  Suspended from the force, Paris cop Hubert is ...      R

                                genre      director
0      Action and Adventure|Classics|Drama      William Friedkin \
1      Drama|Science Fiction and Fantasy      David Cronenberg
2      Drama|Musical and Performing Arts      Allison Anders
3      Drama|Mystery and Suspense      Barry Levinson
4      Drama|Romance      Rodney Bennett
...
1555  Action and Adventure|Horror|Mystery and Suspense      NaN
1556  Comedy|Science Fiction and Fantasy      Steve Barron
1557  Classics|Comedy|Drama|Musical and Performing Arts      Gordon Douglas
1558  Comedy|Drama|Kids and Family|Sports and Fitness      David Mickey Evans
1559  Action and Adventure|Art House and Internation...      NaN

                                writer      theater_date
0      Ernest Tidyman      Oct 9, 1971 \
1      David Cronenberg|Don DeLillo      Aug 17, 2012
2      Allison Anders      Sep 13, 1996
3      Paul Attanasio|Michael Crichton      Dec 9, 1994
4      Giles Cooper      NaN
...
1555      NaN      Aug 18, 2006
1556  Terry Turner|Tom Davis|Dan Aykroyd|Bonnie Turner      Jul 23, 1993
1557      NaN      Jan 1, 1962
1558      David Mickey Evans|Robert Gunter      Apr 1, 1993
1559      Luc Besson      Sep 27, 2001

                                dvd_date      currency      box_office      runtime      studio
0      2001-09-25      NaN      NaN      104 minutes      NaN \
1      2013-01-01      $      600,000      108 minutes      Entertainment One
```

2	2000-04-18	NaN	NaN	116 minutes	NaN
3	1997-08-27	NaN	NaN	128 minutes	NaN
4	NaT	NaN	NaN	200 minutes	NaN
...
1555	2007-01-02	\$	33,886,034	106 minutes	New Line Cinema
1556	2001-04-17	NaN	NaN	88 minutes	Paramount Vantage
1557	2004-05-11	NaN	NaN	111 minutes	NaN
1558	2002-01-29	NaN	NaN	101 minutes	NaN
1559	2003-02-11	NaN	NaN	94 minutes	Columbia Pictures

	first_genre	release_month
0	Action and Adventure Classics Drama	September
1	Drama Science Fiction and Fantasy	January
2	Drama Musical and Performing Arts	April
3	Drama Mystery and Suspense	August
4	Drama Romance	NaN
...
1555	Action and Adventure Horror Mystery and Suspense	January
1556	Comedy Science Fiction and Fantasy	April
1557	Classics Comedy Drama Musical and Performing Arts	May
1558	Comedy Drama Kids and Family Sports and Fitness	January
1559	Action and Adventure Art House and Internation...	February

[1560 rows x 13 columns]

merging BOM data and Rotten Tomatoes dataframes

```
[ ]: # merging the Dfs
merged_df = pd.merge(rtmovie_df, mojo_df, how='outer')
# previewing the new DataFrame
merged_df.shape
```

```
[ ]: (9213, 16)
```

```
[ ]: merged_df.head()
```

	synopsis	rating
0	This gritty, fast-paced, and innovative police...	R \
1	This gritty, fast-paced, and innovative police...	R
2	This gritty, fast-paced, and innovative police...	R
3	This gritty, fast-paced, and innovative police...	R
4	This gritty, fast-paced, and innovative police...	R

	genre	director	writer
0	Action and Adventure Classics Drama	William Friedkin	Ernest Tidyman \
1	Action and Adventure Classics Drama	William Friedkin	Ernest Tidyman
2	Action and Adventure Classics Drama	William Friedkin	Ernest Tidyman
3	Action and Adventure Classics Drama	William Friedkin	Ernest Tidyman

```
4 Action and Adventure|Classics|Drama William Friedkin Ernest Tidyman
```

	theater_date	dvd_date	currency	box_office	runtime	studio
0	Oct 9, 1971	2001-09-25	NaN	NaN	104 minutes	NaN \
1	Oct 9, 1971	2001-09-25	NaN	NaN	104 minutes	NaN
2	Oct 9, 1971	2001-09-25	NaN	NaN	104 minutes	NaN
3	Oct 9, 1971	2001-09-25	NaN	NaN	104 minutes	NaN
4	Oct 9, 1971	2001-09-25	NaN	NaN	104 minutes	NaN

	first_genre	release_month	domestic_gross
0	Action and Adventure Classics Drama	September	96900.0 \
1	Action and Adventure Classics Drama	September	70600.0
2	Action and Adventure Classics Drama	September	NaN
3	Action and Adventure Classics Drama	September	7100.0
4	Action and Adventure Classics Drama	September	NaN

	foreign_gross	year
0	3300000	2010.0
1	3300000	2011.0
2	4000000	2012.0
3	NaN	2014.0
4	122000000	2017.0

```
[ ]: # show number of rows and columns
merged_df.shape
```

```
[ ]: (9213, 16)
```

```
[ ]: # show all column names
merged_df.columns
```

```
[ ]: Index(['synopsis', 'rating', 'genre', 'director', 'writer', 'theater_date',
          'dvd_date', 'currency', 'box_office', 'runtime', 'studio',
          'first_genre', 'release_month', 'domestic_gross', 'foreign_gross',
          'year'],
          dtype='object')
```

```
[ ]: #counts per genre of the merged dataframe

count = merged_df['genre'].value_counts()
count
```

```
[ ]: genre
Drama|Mystery and Suspense
531
Drama
489
```

```

Comedy|Drama
421
Comedy
338
Art House and International|Drama
274
...
Action and Adventure|Animation|Science Fiction and Fantasy
1
Action and Adventure|Animation|Drama|Science Fiction and Fantasy|Special
Interest
1
Kids and Family|Musical and Performing Arts
1
Action and Adventure|Animation|Art House and International|Drama|Science Fiction
and Fantasy
1
Art House and International|Documentary|Faith and Spirituality
1
Name: count, Length: 299, dtype: int64

```

```
[ ]: pop_genres = count.iloc[:20]
pop_genres
```

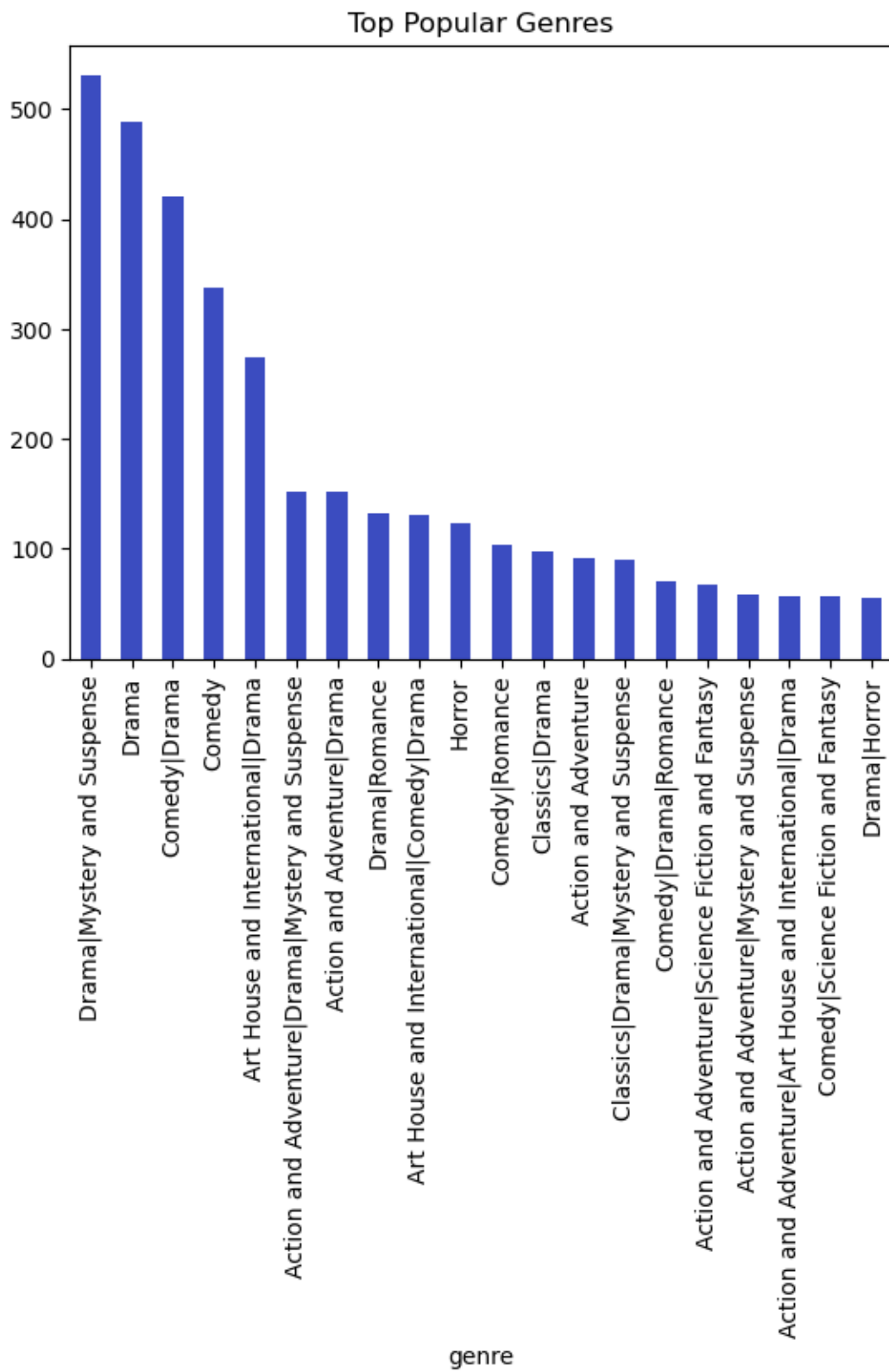
```
[ ]: genre
Drama|Mystery and Suspense      531
Drama                          489
Comedy|Drama                    421
Comedy                         338
Art House and International|Drama 274
Action and Adventure|Drama|Mystery and Suspense 152
Action and Adventure|Drama      152
Drama|Romance                   133
Art House and International|Comedy|Drama 131
Horror                         123
Comedy|Romance                 104
Classics|Drama                 97
Action and Adventure           91
Classics|Drama|Mystery and Suspense 90
Comedy|Drama|Romance           71
Action and Adventure|Science Fiction and Fantasy 68
Action and Adventure|Mystery and Suspense 58
Action and Adventure|Art House and International|Drama 57
Comedy|Science Fiction and Fantasy 57
Drama|Horror                   55
Name: count, dtype: int64

```

```
[ ]: #top 20 popular movies that is with the most value counts represented in a
↳graph:
```

```
pop_genres.plot.bar(x = 'genres', title = 'Top Popular Genres',  
                    colormap='coolwarm')
```

```
[ ]: <Axes: title={'center': 'Top Popular Genres'}, xlabel='genre'>
```



```
[ ]: # getting mean and median world domestic amounts by genre

genre_stats = merged_df.groupby('genre')['domestic_gross'].agg(['median',
↳ 'mean'])
genre_stats.sort_values(by='mean', ascending=False)
```

```
[ ]:
                                     median      mean
genre
Drama|Mystery and Suspense          30350000.0  5.553751e+07
Art House and International|Comedy|Drama|Musica... 17048450.0  3.336243e+07
Drama|Horror|Mystery and Suspense          70600.0  8.384258e+06
Action and Adventure|Mystery and Suspense          70600.0  6.719047e+06
Drama|Horror          1500000.0  6.237862e+06
...
Horror|Kids and Family|Mystery and Suspense|Sci...      NaN      NaN
Horror|Musical and Performing Arts|Science Fict...      NaN      NaN
Horror|Mystery and Suspense|Science Fiction and...      NaN      NaN
Kids and Family|Musical and Performing Arts           NaN      NaN
Mystery and Suspense|Science Fiction and Fantas...      NaN      NaN

[299 rows x 2 columns]
```

```
[ ]: #filtering the dataframe based on Drama|Mystery and Suspense which is the top
↳ genre

DramaMS=merged_df.loc[merged_df['genre'] == "Drama|Mystery and Suspense"]
DramaMS
```

```
[ ]:
                                     synopsis rating
10   Michael Douglas runs afoul of a treacherous su...      R \
11   Michael Douglas runs afoul of a treacherous su...      R
12   Michael Douglas runs afoul of a treacherous su...      R
13   Michael Douglas runs afoul of a treacherous su...      R
14   Michael Douglas runs afoul of a treacherous su...      R
...
5968 Directed by Clint Eastwood, the mysterious dra...      R
6002 Abel Ferrara's cult crime drama Bad Lieutenant...      R
6212 Filmed in the California desert on Super 16mm,...      NR
6285 Frankie is a Los Angeles drug dealer. He comes...      R
6303 Texas brothers--Toby (Chris Pine), and Tanner ...      R

                                     genre      director
10   Drama|Mystery and Suspense      Barry Levinson \
11   Drama|Mystery and Suspense      Barry Levinson
12   Drama|Mystery and Suspense      Barry Levinson
13   Drama|Mystery and Suspense      Barry Levinson
14   Drama|Mystery and Suspense      Barry Levinson
```


...
5968	Drama Mystery and Suspense	Clint Eastwood
6002	Drama Mystery and Suspense	Werner Herzog
6212	Drama Mystery and Suspense	Oren Shai
6285	Drama Mystery and Suspense	Nick Cassavetes
6303	Drama Mystery and Suspense	David Mackenzie

	writer	theater_date	dvd_date	currency	
10	Paul Attanasio Michael Crichton	Dec 9, 1994	1997-08-27	NaN	\
11	Paul Attanasio Michael Crichton	Dec 9, 1994	1997-08-27	NaN	
12	Paul Attanasio Michael Crichton	Dec 9, 1994	1997-08-27	NaN	
13	Paul Attanasio Michael Crichton	Dec 9, 1994	1997-08-27	NaN	
14	Paul Attanasio Michael Crichton	Dec 9, 1994	1997-08-27	NaN	

...
5968	Brian Helgeland	Oct 8, 2003	2004-06-08	\$	
6002	NaN	Nov 20, 2009	2010-04-06	\$	
6212	Oren Shai Webb Wilcoxon	Oct 28, 2016	2016-12-06	NaN	
6285	Nick Cassavetes	Jan 12, 2007	2007-05-01	\$	
6303	Taylor Sheridan	Aug 12, 2016	2016-11-22	\$	

	box_office	runtime	studio	
10	NaN	128 minutes	NaN	\
11	NaN	128 minutes	NaN	
12	NaN	128 minutes	NaN	
13	NaN	128 minutes	NaN	
14	NaN	128 minutes	NaN	

...
5968	88,800,000	137 minutes	WB	
6002	1,616,556	121 minutes	First Look Pictures	
6212	NaN	88 minutes	Rocking Films	
6285	15,133,185	118 minutes	Universal Studios	
6303	26,973,524	102 minutes	Film 44	

	first_genre	release_month	domestic_gross	foreign_gross	
10	Drama Mystery and Suspense	August	96900.0	3300000	\
11	Drama Mystery and Suspense	August	70600.0	3300000	
12	Drama Mystery and Suspense	August	NaN	4000000	
13	Drama Mystery and Suspense	August	7100.0	NaN	
14	Drama Mystery and Suspense	August	NaN	122000000	

...
5968	Drama Mystery and Suspense	June	3200000.0	NaN	
6002	Drama Mystery and Suspense	April	NaN	NaN	
6212	Drama Mystery and Suspense	December	NaN	NaN	
6285	Drama Mystery and Suspense	May	NaN	NaN	
6303	Drama Mystery and Suspense	November	NaN	NaN	

year

```

10    2010.0
11    2011.0
12    2012.0
13    2014.0
14    2017.0
...    ...
5968  2018.0
6002    NaN
6212    NaN
6285    NaN
6303    NaN

```

[531 rows x 16 columns]

```
[ ]: #filtering out the most common director in the Drama/Mystery and Suspense genre

DramaMS['director'].value_counts()
```

```
[ ]: director
Clint Eastwood          141
Gary Wheeler            136
Joseph Ruben            10
Gary Fleder              6
Mike Figgis              6
Barry Levinson           5
Lewis Gilbert            5
John Badham              5
Curtis Hanson            5
Shawn Christensen        5
William Beaudine         5
Andy Wolk                5
Andrew Chapman           5
Bob Rafelson|George Bud Davis  5
Robert Foster            5
Michael Fields           5
Andrew Birkin            5
Spike Lee                5
Peter Hyams              5
James Cox                5
Steven Hilliard Stern    5
Gordon Willis            5
Irving Lerner            5
Fritz Lang              5
Paul Wendkos             5
Uli Edel                 5
Nathan Hope              5
Mike Barker              5

```

Sidney Gilliat	5
John Sturges	5
David Mamet	5
Gordon Hessler	5
Sam Peckinpah	5
Boaz Yakin	5
Yves Simoneau	5
Neil Jordan	5
Bruce Robinson	5
Larry Elikann	5
Nicolas Roeg	5
Steven Spielberg	5
John Farrow	5
Michael Apted	5
Bryan Singer	3
David Koepp	1
Nick Cassavetes	1
Oren Shai	1
Werner Herzog	1
Anton Corbijn	1
Will Canon	1
Nicholas Racz	1
Perry Moore Hunter Hill	1
Craig Bolotin	1
Morten Tyldum	1
David Fincher	1
Simon West	1
Paul Thomas Anderson	1
David Mackenzie	1

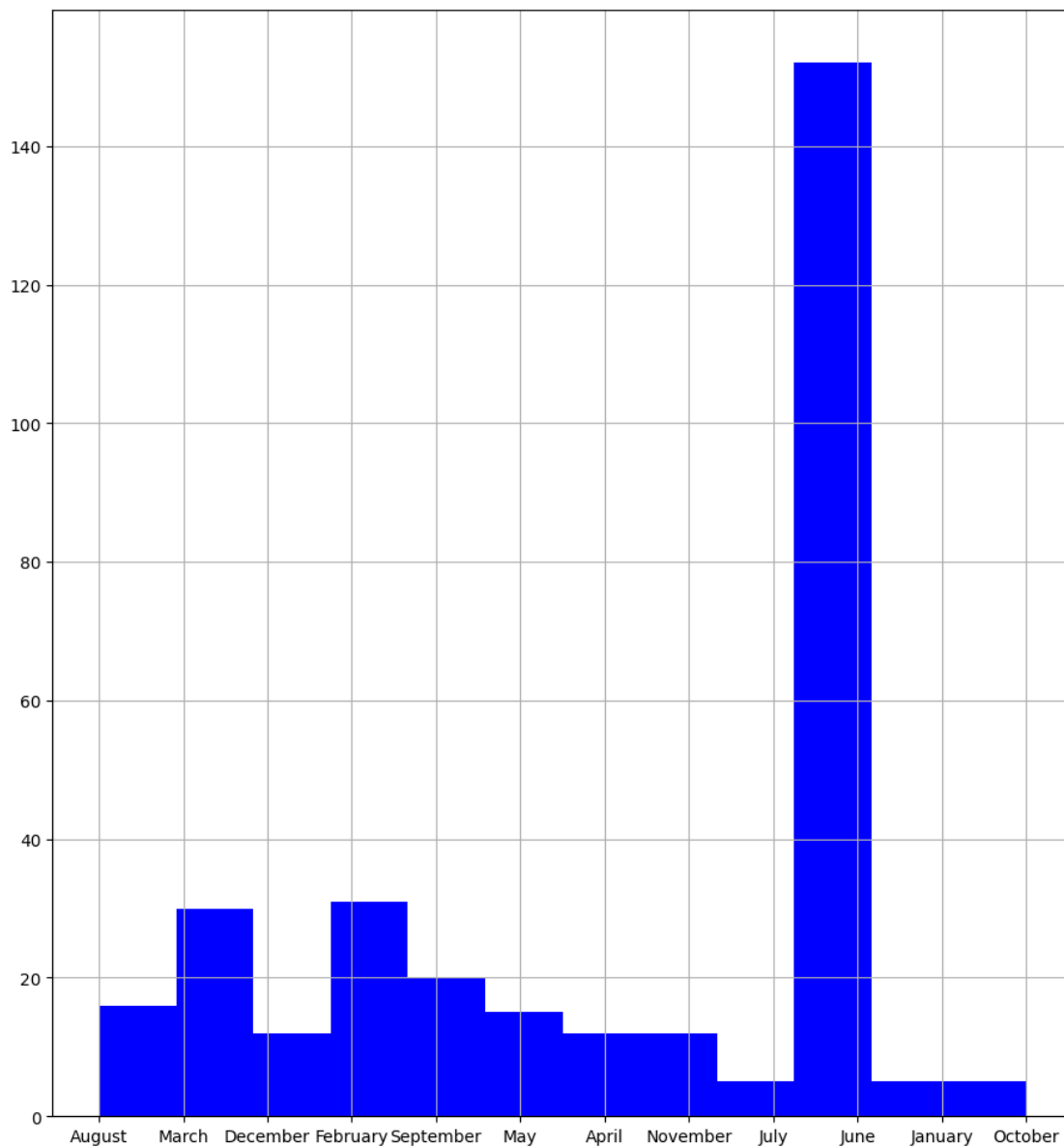
Name: count, dtype: int64

```
[ ]: DramaMS['release_month'].value_counts()
```

```
[ ]: release_month
June      152
February   31
March      30
September  20
August     16
May        15
December   12
April      12
November   12
July        5
January     5
October     5
Name: count, dtype: int64
```

```
[ ]: # Visualizing MONTH RELEASED using histogram
DramaMS['release_month'].hist(bins=12, figsize=(11,12), color=('blue'))
```

```
[ ]: <Axes: >
```



Most Drama|Mystery and Suspense movies were released in the month of June

```
[ ]: DramaMS['rating'].value_counts()
```

```
[ ]: rating
R      293
```

```
PG-13    152
NR        76
G         5
PG        5
Name: count, dtype: int64
```

```
[ ]:
```