In [86]:

```python
import numpy as np #linear algebra
import pandas as pd #data processing
from datetime import date, time, datetime
import re
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('display.expand_frame_repr', False)
pd.set_option('max_colwidth', None)
```

In [87]:

```python
def testingDateTime(string):
    print(f' the if her top {string.strftime("%b %d %Y")}')

def testingDateTime1(string):
    print(type(string))

    thedate = string.strftime("%b %d %Y").strip()

    print(thedate)

    if thedate == "Nov 20 1975":
        print(f' working {string}')

    print(f' the else {string}')

def stringTimeToMins(string):
    mins = 0
    #print(f' the top {string}')
    if string !="0":
        if string.split():
            #print(f'first if {string.split()}')
            splitTime = string.split()

            if len(splitTime):

                splitLen = len(splitTime)
                #print(f'second if {splitTime}')

                if splitLen == 4:
                    #print(f'third if {splitTime}')
                    #print(splitTime)
                    mins = int(splitTime[0])*60 + int(splitTime[2])

                elif splitLen == 2:
                    if ("hr" in splitTime):
                        mins = int(splitTime[0])*60
                    elif ("min" in splitTime):
                        mins = int(splitTime[0])
    return mins
```

In [88]:

```
# Importing into dataframe
df_Bom_MovieGross = pd.read_csv("./Prj_Data/DownLoadedData_FlatIron/bom.movi
df_TN_Movie_Budgets = pd.read_csv("./Prj_Data/DownLoadedData_FlatIron/tn.mov
df_TN_Movie_Budgets = pd.read_csv("./Prj_Data/DownLoadedData_FlatIron/tn.mov
df_Scrp_Financials = pd.read_excel("./Prj_Data/ImdbScrapingData/df_Financial
df_IMDB_Akas_english = pd.read_excel("./Prj_Data/DownLoadedData_Imdb/df_IMDB
df_Studio_x_ref_For_import = pd.read_excel("./Prj_Data/ImdbScrapingData/df_S
# df_Studio_cross_Ref_FI = pd.read_excel("./data/DownLoadedData_FlatIron/df_

```

In [ ]:

```

```

In [89]:

```
#PREPARE THE META DATA FROM FINANCIAL TABLES:  WE USED SOURCES,2 FROM FLATIR
#***************************************************************************
#prepraring to work with Movie Gross Df, cleaning up data and create right d
df_Bom_MovieGross.name = "df_Bom_MovieGross"
df_Bom_MovieGross['year_str_BOM'] = df_Bom_MovieGross['year'].astype(str)
df_Bom_MovieGross['year'] = df_Bom_MovieGross['year'].astype(int)
df_Bom_MovieGross['title'] = df_Bom_MovieGross['title'].str.title() #******K
df_Bom_MovieGross["titleyear"] = df_Bom_MovieGross['title'] + df_Bom_MovieGr

#Renaming to aid in consolidating between the three sources
df_Bom_MovieGross.rename(columns={"year": "year_BOM",
                                             "title":"title_BOM","studio":"s
#-------------------------------------------------------------

#prepraring to work with Movie budgets, cleaning up data and create right da
df_TN_Movie_Budgets.name = "df_TN_Movie_Budgets"

df_TN_Movie_Budgets['movie'] = df_TN_Movie_Budgets['movie'].str.title() #***
df_TN_Movie_Budgets["year"] = df_TN_Movie_Budgets['release_date'].str[-4:].a
df_TN_Movie_Budgets["year_str_TN"] = df_TN_Movie_Budgets['release_date'].str
df_TN_Movie_Budgets["titleyear"] = df_TN_Movie_Budgets['movie'] + df_TN_Movi
df_TN_Movie_Budgets = df_TN_Movie_Budgets.drop('id', axis = 1)

df_TN_Movie_Budgets.rename(columns={"year": "year_TN","movie":"title_TN","re

#needed to do this steep to collapse data given there are dups in a few movi
from pandasql import sqldf
pysqldf = lambda q: sqldf(q, globals())
q3 = """SELECT titleyear as titleyear, max(title_TN) as title_TN, max(year_T
min(rd_TN) as rd_TN FROM df_TN_Movie_Budgets GROUP BY titleyear, title_TN
        """
df_TN_Movie_Budgets = pysqldf(q3)
#-------------------------------------------------------------

# Finanally merger data from flatiron
df_fI_Financials = df_TN_Movie_Budgets.merge(df_Bom_MovieGross,
     on='titleyear', how='outer', indicator='Combing_FL_Financials', suffixe

# Add tconst Key to FLatIronTables then drop
df_FI_financials_With_tconst = df_fI_Financials.merge(df_IMDB_Akas_english[[
```

In [90]:

```
#Use all data scraped from IMDB as the default for domestic, foreign ww sale
df_FI_financials_With_tconst.title_TN.fillna("0", inplace=True)
df_FI_financials_With_tconst.year_str_TN.fillna("0", inplace=True)
df_FI_financials_With_tconst.year_TN.fillna(0, inplace=True)

df_FI_financials_With_tconst.title_BOM.fillna("0", inplace=True)
df_FI_financials_With_tconst.year_str_BOM.fillna("0", inplace=True)
df_FI_financials_With_tconst.year_BOM.fillna(0, inplace=True)

df_FI_financials_With_tconst['title_FI'] = df_FI_financials_With_tconst['tit
df_FI_financials_With_tconst['year_str_FI'] = df_FI_financials_With_tconst['
df_FI_financials_With_tconst['year_FI'] = df_FI_financials_With_tconst['year

df_FI_financials_With_tconst['title_FI'] = df_FI_financials_With_tconst.titl
df_FI_financials_With_tconst['year_str_FI'] = df_FI_financials_With_tconst.y
df_FI_financials_With_tconst['year_FI'] = df_FI_financials_With_tconst.year_

df_FI_financials_With_tconst['title_FI'] = df_FI_financials_With_tconst.appl
df_FI_financials_With_tconst['year_str_FI'] = df_FI_financials_With_tconst.a
df_FI_financials_With_tconst['year_FI'] = df_FI_financials_With_tconst.apply
```

In [91]:

```
#DROP UNNEEDED FIELDS
# df_FI_financials_With_tconst.drop(["year_BOM","year_TN", "year_str_TN","ye
```

In [92]:

```
df_FI_financials_With_tconst.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2616 entries, 0 to 2615
Data columns (total 17 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   titleyear              2616 non-null   object
 1   title_TN               2616 non-null   object
 2   year_TN                2616 non-null   float64
 3   year_str_TN            2616 non-null   object
 4   rd_TN                  2000 non-null   object
 5   title_BOM              2616 non-null   object
 6   studio_BOM             1646 non-null   object
 7   domestic_gross         1641 non-null   float64
 8   foreign_gross          1210 non-null   object
 9   year_BOM               2616 non-null   float64
 10  year_str_BOM           2616 non-null   object
 11  Combing_FL_Financials  2616 non-null   category
 12  tconst                 2616 non-null   object
 13  Adding tconst          2616 non-null   category
 14  title_FI               2616 non-null   object
 15  year_str_FI            2616 non-null   object
 16  year_FI                2616 non-null   float64
dtypes: category(2), float64(4), object(11)
memory usage: 332.3+ KB
```

In [93]:

```
#Merg flatiron financials with scraping finacials IMDB Site :https://www.box
#***************************************************************************
#preprating to work with Movie Gross Df, cleaning up data and create right d
df_MasterFinancials = df_Scrp_Financials.merge(df_FI_financials_With_tconst,
                                               on='tconst', how='outer', suf
fieldsToConvert = {'Domestic Opening': 0}
df_MasterFinancials.fillna(value=fieldsToConvert, inplace=True)
```

In [94]:

```
df_MasterFinancials_MetaData1 = df_MasterFinancials[['tconst', 'Domestic Ope
                                                     'Genres_IMDB','genres',
                                                     'studio_IMDB', 'original
                                                     'studio_BOM','title_BOM'
                                                     'titleyear_IM','titleyea
                                                     'year','year_BOM', 'year
df_MasterFinancials_MetaData = df_MasterFinancials_MetaData1.copy()

```

In [95]:

```
df_MasterFinancials_MetaData.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5543 entries, 0 to 5542
Data columns (total 21 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   tconst              5543 non-null   object 
 1   Domestic Opening    5543 non-null   float64
 2   Genres_IMDB         4920 non-null   object 
 3   genres              4920 non-null   object 
 4   MPAA                3315 non-null   object 
 5   Running Time_IMDB   4915 non-null   object 
 6   runtimeMinutes      4920 non-null   object 
 7   studio_IMDB         4754 non-null   object 
 8   originalTitle       4920 non-null   object 
 9   rd_IMDB             4920 non-null   object 
 10  rd_TN               2000 non-null   object 
 11  studio_BOM          1646 non-null   object 
 12  title_BOM           2616 non-null   object 
 13  title_IMDB          4920 non-null   object 
 14  title_TN            2616 non-null   object 
 15  titleyear_IM        4920 non-null   object 
 16  titleyear_fl        2616 non-null   object 
 17  ww_IMDB             4920 non-null   float64
 18  year                4920 non-null   float64
 19  year_BOM            2616 non-null   float64
 20  year_TN             2616 non-null   float64
dtypes: float64(5), object(16)
memory usage: 952.7+ KB
```

In [96]:

```
df_MasterFinancials_MetaData["Genres_S"] = df_MasterFinancials_MetaData.Genr
df_MasterFinancials_MetaData["Genres_S"].fillna("", inplace=True)
df_MasterFinancials_MetaData["Genres_S"] = df_MasterFinancials_MetaData["Gen
# df_MasterFinancials_MetaData["Genres_S"] = df_MasterFinancials_MetaData["G
```

In [97]:

```
#Use all data scraped from IMDB as the default for domestic, foreign ww sale

df_MasterFinancials_MetaData["year"] = df_MasterFinancials_MetaData["year"].
df_MasterFinancials_MetaData["titleyear_fin"] = df_MasterFinancials_MetaData
df_MasterFinancials_MetaData["rd_IMDB"] = df_MasterFinancials_MetaData["rd_I
df_MasterFinancials_MetaData["studio_IMDB"] = df_MasterFinancials_MetaData["
df_MasterFinancials_MetaData["studio_BOM"] = df_MasterFinancials_MetaData["s


df_MasterFinancials_MetaData["titleyear_IM"] = df_MasterFinancials_MetaData[
df_MasterFinancials_MetaData["titleyear_fl"] = df_MasterFinancials_MetaData[
df_MasterFinancials_MetaData["Running Time_IMDB"] = df_MasterFinancials_Meta
df_MasterFinancials_MetaData["rd_IMDB"] = df_MasterFinancials_MetaData["rd_I
df_MasterFinancials_MetaData["rd_TN"] = df_MasterFinancials_MetaData["rd_TN"



df_MasterFinancials_MetaData['titleyear_fin'] = df_MasterFinancials_MetaData
df_MasterFinancials_MetaData['genres_fin'] = df_MasterFinancials_MetaData['G
df_MasterFinancials_MetaData['MPAA_fin'] = df_MasterFinancials_MetaData['MPA
df_MasterFinancials_MetaData['title_fin'] = df_MasterFinancials_MetaData['ti
df_MasterFinancials_MetaData['year_fin'] = df_MasterFinancials_MetaData['yea


#fill in the blanks fromvalus from the flatIron data where missing
df_MasterFinancials_MetaData['title_fin'] = df_MasterFinancials_MetaData['ti
df_MasterFinancials_MetaData['title_fin'] = df_MasterFinancials_MetaData.app
df_MasterFinancials_MetaData['title_fin'] = df_MasterFinancials_MetaData['ti
df_MasterFinancials_MetaData['title_fin'] = df_MasterFinancials_MetaData.app

df_MasterFinancials_MetaData['genres_fin'] = df_MasterFinancials_MetaData['g
df_MasterFinancials_MetaData['genres_fin'] = df_MasterFinancials_MetaData.ap

df_MasterFinancials_MetaData["year_BOM"] = df_MasterFinancials_MetaData["yea
df_MasterFinancials_MetaData["year_TN"] = df_MasterFinancials_MetaData["year

df_MasterFinancials_MetaData['year_fin'] = df_MasterFinancials_MetaData.appl

df_MasterFinancials_MetaData['titleyear_fin'] = df_MasterFinancials_MetaData

df_MasterFinancials_MetaData['titleyear_fin'] = df_MasterFinancials_MetaData
df_MasterFinancials_MetaData['RunningTime_fin'] = df_MasterFinancials_MetaDa

```

In [98]:

```
#this code syncs the release dates

df_MasterFinancials_MetaData["rd_IMDB"] = df_MasterFinancials_MetaData["rd_I
df_MasterFinancials_MetaData["rd_TN"] = df_MasterFinancials_MetaData["rd_TN"

df_MasterFinancials_MetaData['rd_string_IMDB_step1'] = df_MasterFinancials_M

df_MasterFinancials_MetaData['rd_string_IMDB_step2'] = df_MasterFinancials_M

df_MasterFinancials_MetaData['rd_string_IMDB_step3'] = pd.to_datetime(df_Mas

df_MasterFinancials_MetaData['rd_string_TN_step3'] = pd.to_datetime(df_Maste

df_MasterFinancials_MetaData['rd_fin'] = df_MasterFinancials_MetaData['rd_st

df_MasterFinancials_MetaData['rd_fin'] = df_MasterFinancials_MetaData.apply(
```

In [99]:

```
df_MasterFinancials_MetaData.drop(["Domestic Opening","Genres_IMDB","genres"
                                  "rd_TN","title_BOM","title_IMDB","title_TN
                                  "Genres_S", "rd_string_IMDB_step1", "rd_st
                                  "rd_string_TN_step3","year","year_BOM","ye
```

In [100]:

```
df_MasterFinancials_MetaData.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5543 entries, 0 to 5542
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype         
---  ------           --------------  -----         
 0   tconst           5543 non-null   object        
 1   MPAA             3315 non-null   object        
 2   studio_IMDB      5543 non-null   object        
 3   studio_BOM       5543 non-null   object        
 4   titleyear_fin    5543 non-null   object        
 5   genres_fin       5543 non-null   object        
 6   MPAA_fin         3315 non-null   object        
 7   title_fin        5543 non-null   object        
 8   year_fin         5543 non-null   float64       
 9   RunningTime_fin  5543 non-null   int64         
 10  rd_fin           5543 non-null   datetime64[ns]
dtypes: datetime64[ns](1), float64(1), int64(1), object(8)
memory usage: 519.7+ KB
```

In [101]:

```
df_MasterFinancials_MetaData_Studios_Only = df_MasterFinancials_MetaData[["t
```

In [102]:

```
df_Studio_x_ref_For_import.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 399 entries, 0 to 398
Data columns (total 4 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   Studio_Desc  341 non-null    object
 1   studio_IMDB  341 non-null    object
 2   studio_BOM   103 non-null    object
 3   studio_BOM1  399 non-null    object
dtypes: object(4)
memory usage: 12.6+ KB
```

In [103]:

```
df_MasterFinancials_MetaData_Studios_Only.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5543 entries, 0 to 3476
Data columns (total 4 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   titleyear_fin  5543 non-null   object
 1   studio_BOM     5543 non-null   object
 2   studio_IMDB    5543 non-null   object
 3   tconst         5543 non-null   object
dtypes: object(4)
memory usage: 216.5+ KB
```

In [104]:

```
dfxrefForIMDB = df_Studio_x_ref_For_import[["Studio_Desc","studio_IMDB"]]
dfxrefForBOM = df_Studio_x_ref_For_import[["Studio_Desc","studio_BOM1"]]

dfxrefForIMDB.sort_values(by="studio_IMDB").drop_duplicates(inplace=True)
dfxrefForBOM.sort_values(by="studio_BOM1").drop_duplicates(inplace=True)
```

In [105]:

```
dfxrefForIMDB.info()
dfxrefForBOM.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 399 entries, 0 to 398
Data columns (total 2 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   Studio_Desc  341 non-null    object
 1   studio_IMDB  341 non-null    object
dtypes: object(2)
memory usage: 6.4+ KB
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 399 entries, 0 to 398
Data columns (total 2 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   Studio_Desc  341 non-null    object
 1   studio_BOM1  399 non-null    object
dtypes: object(2)
memory usage: 6.4+ KB
```

In [106]:

```
df_Bom_MovieGross_withStudio_IMDB = df_MasterFinancials_MetaData_Studios_Onl
                                          on='studio_IMDB', how='left',
                                        indicator='mergingstudio_IMDB',
```

In [107]:

```
df_Bom_MovieGross_withStudio_IMDB.drop(["mergingstudio_IMDB", "studio_IMDB"]
df_Bom_MovieGross_withStudio_IMDB.drop_duplicates(inplace = True)
```

In [108]:

```
df_Bom_MovieGross_withStudio_IMDB.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5543 entries, 0 to 5542
Data columns (total 4 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   titleyear_fin  5543 non-null   object
 1   studio_BOM     5543 non-null   object
 2   tconst         5543 non-null   object
 3   Studio_Desc    4680 non-null   object
dtypes: object(4)
memory usage: 216.5+ KB
```

In [109]:

```
df_Bom_MovieGross_withStudio_IMDB_BOM = df_Bom_MovieGross_withStudio_IMDB.me
                                          left_on='studio_BOM', right_o
                                        indicator='mergingstudio_BOM', s
```

In [110]:

```
df_Bom_MovieGross_withStudio_IMDB_BOM.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5825 entries, 0 to 5824
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype   
---  ------             --------------  -----   
 0   titleyear_fin      5825 non-null   object  
 1   studio_BOM         5825 non-null   object  
 2   tconst             5825 non-null   object  
 3   Studio_Desc_Fin    4896 non-null   object  
 4   Studio_Desc_Xref   1653 non-null   object  
 5   studio_BOM1        1729 non-null   object  
 6   mergingstudio_BOM  5825 non-null   category
dtypes: category(1), object(6)
memory usage: 324.3+ KB
```

In [111]:

```
df_Bom_MovieGross_withStudio_IMDB_BOM.drop(["mergingstudio_BOM", "studio_BOM
df_Bom_MovieGross_withStudio_IMDB_BOM.drop_duplicates(inplace = True)
```

In [112]:

```
df_Bom_MovieGross_withStudio_IMDB_BOM.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5543 entries, 0 to 5824
Data columns (total 5 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   titleyear_fin     5543 non-null   object
 1   studio_BOM        5543 non-null   object
 2   tconst            5543 non-null   object
 3   Studio_Desc_Fin   4680 non-null   object
 4   Studio_Desc_Xref  1371 non-null   object
dtypes: object(5)
memory usage: 259.8+ KB
```

In [113]:

```
df_Bom_MovieGross_withStudio_IMDB_BOM["StudioDesc"] = df_Bom_MovieGross_with
df_Bom_MovieGross_withStudio_IMDB_BOM["StudioDesc"].fillna("0")
```

Out[113]:

```
0                                   Twentieth Century Fox
1                                               IFC Films
2                                             Eleven Arts
3                                            AMC Theaters
4                                               IFC Films
5                                                       0
6                                              RADiUS-TWC
7                                          Variance Films
8                                 Millennium Entertainment
9                                               IFC Films
10                    Walt Disney Studios Motion Pictures
11                                        Drafthouse Films
12                                       Lionsgate Premiere
13                                       Magnolia Pictures
14                                                      0
15                                                      0
16                             The Samuel Goldwyn Company
17                                                      0
18                                                      0
19                                             RADiUS-TWC
```

In [114]:

```
df_Bom_MovieGross_withStudio_IMDB_BOM["StudioDesc"] = df_Bom_MovieGross_with
# df_Bom_MovieGross_withStudio_IMDB_BOM["StudioDesc"].fillna("0")
```

In [115]:

```
df_Bom_MovieGross_withStudio_IMDB_BOM.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5543 entries, 0 to 5824
Data columns (total 6 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   titleyear_fin     5543 non-null   object
 1   studio_BOM        5543 non-null   object
 2   tconst            5543 non-null   object
 3   Studio_Desc_Fin   4680 non-null   object
 4   Studio_Desc_Xref  1371 non-null   object
 5   StudioDesc        4962 non-null   object
dtypes: object(6)
memory usage: 303.1+ KB
```

In [116]:

```
df_Bom_MovieGross_withStudio_IMDB_BOM.drop(['studio_BOM', "Studio_Desc_Fin",
```

In [117]:

```
df_Bom_MovieGross_withStudio_IMDB_BOM.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5543 entries, 0 to 5824
Data columns (total 3 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   titleyear_fin  5543 non-null   object
 1   tconst         5543 non-null   object
 2   StudioDesc     4962 non-null   object
dtypes: object(3)
memory usage: 173.2+ KB
```

In [118]:

```
df_MasterFinancials_MetaData_all = df_MasterFinancials_MetaData.merge(df_Bom
```

In [119]:

```
df_MasterFinancials_MetaData_all.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5543 entries, 0 to 5542
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype         
---  ------           --------------  -----         
 0   tconst           5543 non-null   object        
 1   MPAA             3315 non-null   object        
 2   studio_IMDB      5543 non-null   object        
 3   studio_BOM       5543 non-null   object        
 4   titleyear_fin_x  5543 non-null   object        
 5   genres_fin       5543 non-null   object        
 6   MPAA_fin         3315 non-null   object        
 7   title_fin        5543 non-null   object        
 8   year_fin         5543 non-null   float64       
 9   RunningTime_fin  5543 non-null   int64         
 10  rd_fin           5543 non-null   datetime64[ns]
 11  titleyear_fin_y  5543 non-null   object        
 12  StudioDesc       4962 non-null   object        
dtypes: datetime64[ns](1), float64(1), int64(1), object(10)
memory usage: 606.3+ KB
```

In [120]:

```
df_MasterFinancials_MetaData_all = df_MasterFinancials_MetaData_all.copy()
```

In [121]:

```
df_MasterFinancials_MetaData_all.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5543 entries, 0 to 5542
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype         
---  ------           --------------  -----         
 0   tconst           5543 non-null   object        
 1   MPAA             3315 non-null   object        
 2   studio_IMDB      5543 non-null   object        
 3   studio_BOM       5543 non-null   object        
 4   titleyear_fin_x  5543 non-null   object        
 5   genres_fin       5543 non-null   object        
 6   MPAA_fin         3315 non-null   object        
 7   title_fin        5543 non-null   object        
 8   year_fin         5543 non-null   float64       
 9   RunningTime_fin  5543 non-null   int64         
 10  rd_fin           5543 non-null   datetime64[ns]
 11  titleyear_fin_y  5543 non-null   object        
 12  StudioDesc       4962 non-null   object        
dtypes: datetime64[ns](1), float64(1), int64(1), object(10)
memory usage: 606.3+ KB
```

In [122]:

```
# df_MasterFinancials_MetaData_all.drop(["studio_IMDB"],axis=1, inplace=True
```

In [123]:

```
# df_MasterFinancials_MetaData_all = df_MasterFinancials_MetaData_all[['tcon
#         'genres_fin', 'MPAA_fin', 'title_fin', 'year_fin', 'RunningTime_fin
#         'rd_fin', 'titleyear_fin_y', 'StudioDesc']]
```

In [124]:

```
df_MasterFinancials_MetaData_all.drop(["titleyear_fin_x","MPAA", "studio_IMD
# df_MasterFinancials_MetaData_all.drop(["titleyear_fin_x", "titleyear_fin_y
```

In [125]:

```
df_MasterFinancials_MetaData_all.rename(columns={"studio_BOM": "studio_BOM",
                                                  "MPAA_fin":"MPAA", "title_f
                                                "year_fin":"year", "RunningTi
                                                 ,"titleyear_fin_y":"titleyea
```

In [128]:

```
df_MasterFinancials_MetaData_all.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5543 entries, 0 to 5542
Data columns (total 10 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   tconst       5543 non-null   object
 1   studio_BOM   5543 non-null   object
 2   genres       5543 non-null   object
 3   MPAA         3315 non-null   object
 4   title        5543 non-null   object
 5   year         5543 non-null   float64
 6   RunningTime  5543 non-null   int64
 7   rd           5543 non-null   datetime64[ns]
 8   titleyear    5543 non-null   object
 9   StudioDesc   4962 non-null   object
dtypes: datetime64[ns](1), float64(1), int64(1), object(7)
memory usage: 476.4+ KB
```

In [127]:

```
df_MasterFinancials_MetaData_all.to_excel("df_MasterFinancials_MetaData.xlsx
```

In [ ]:

```
#--------------------CODE FOR CHECKING ABOVE IF REQUIRED
```

In [ ]:

```
df_MasterFinancials_MetaData[["rd_IMDB",  'rd_string_IMDB_step3', 'rd_string
```

In [ ]:

```
df_Bom_MovieGross_withStudio_IMDB_BOM.groupby(["tconst"])[["studio_BOM1"]].c
```

In [ ]:

```

```

In [ ]:

```
above_1  = df_Bom_MovieGross_withStudio_IMDB_BOM[df_Bom_MovieGross_withStudi
above_1
```