In [1]:
```
import numpy as np #linear algebra
import pandas as pd #data processing
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 [2]:
```
df_IMDB_Akas = pd.read_csv("./Prj_Data/DownLoadedData_Imdb/title.akas (1).ts
IM_title_Basics = pd.read_csv("./Prj_Data/DownLoadedData_Imdb/title.basics (
df_MetaDataFromFinanceTables = pd.read_excel("./Prj_Data/ImdbScrapingData/df
```

...

In [3]:
```
#This table is used as a filter, filter on english and production type
df_IMDB_Akas = df_IMDB_Akas.loc[(df_IMDB_Akas['language']=='en')]
df_IMDB_Akas = df_IMDB_Akas.loc[(df_IMDB_Akas['types']=='imdbDisplay')]
```

In [4]:
```
df_IMDB_Akas = df_IMDB_Akas.drop_duplicates(subset='titleId', keep='first').

IM_title_Basics = IM_title_Basics.loc[(IM_title_Basics['isAdult']==0)]
IM_title_Basics.drop(['isAdult', 'endYear', 'titleType', 'originalTitle'], a
IM_title_Basics["runtimeMinutes"] = IM_title_Basics.runtimeMinutes.replace(r
IM_title_Basics['runtimeMinutes'] = IM_title_Basics['runtimeMinutes'].astype
IM_title_Basics = IM_title_Basics.loc[(IM_title_Basics['runtimeMinutes']>60)
IM_title_Basics["startYear"] = IM_title_Basics.startYear.replace(r'\N',0, re
IM_title_Basics['startYear'] = IM_title_Basics['startYear'].astype(int)
IM_title_Basics = IM_title_Basics.loc[(IM_title_Basics['startYear']>=2005) &
IM_title_Basics['startYear_str'] = IM_title_Basics['startYear'].astype(str)
IM_title_Basics['primaryTitle'] = IM_title_Basics['primaryTitle'].str.title(
IM_title_Basics['titleyear'] = IM_title_Basics['primaryTitle'] + IM_title_Ba
```

In [5]:
```
df_IMDB_Akas_english = IM_title_Basics.merge(df_IMDB_Akas, left_on="tconst",
```

In [6]:
```
df_IMDB_Akas_english.info()
```

...

In [7]:
```
df_IMDB_Akas_english.drop(["ind_Link"], axis=1, inplace=True)
```

In [8]:
```
df_IMDB_Akas_english.head()
```

...

In [9]:
```
df_MetaDataFromFinanceTables.info()
```

...

In [10]:
```
df_MetaDataFromFinanceTables.drop(["year","RunningTime", "genres", "title","
```

In [11]:
```
df_IMDB_Eng_with_metadata = df_IMDB_Akas_english.merge(df_MetaDataFromFinanc
```

In [12]:
```
1 df_IMDB_Eng_with_metadata.info()
```

...

In [13]:
```
1 df_IMDB_Eng_with_metadata.drop(["titleId","Unnamed: 0","Merg_MetaData" ], ax
```

In [14]:
```
1 df_IMDB_Eng_with_metadata.info()
```

...

In [15]:
```
1 df_IMDB_Eng_with_metadata.to_excel("df_IMDB_MovieCatalog.xlsx")
```

Data belwo is for quick cutting and copying for testing :)

In [ ]:
```
1 # df_IMDB_Akas_english.drop(["ind_Link","Unnamed: 0","titleyear_fin", "title
2 #                            "title_fin","Merg_MetaData", "year_fin", "Runnin
3
4 # df_IMDB_Akas_english.drop(["ind_Link","Unnamed: 0","titleyear_fin", "title
5 #                            "title_fin","Merg_MetaData", "year_fin", "Runnin
```

In [ ]:
```
1 testingLink.loc[testingLink['tconst'] == "tt0499549"][["primaryTitle_x", "tc
2 # .loc[df_IMDB_Akas['tconst'] == "tt0499549"][["primaryTitle", "tconst"]]
3 # df_IMDB_Akas_english[df_IMDB_Akas_english.primaryTitle.str.contains('Happy
```