THIS FILE WAS USED TO INSERT NCONST AND TCONST INTO THE DATA SCRAPED FROM THE NUMBERS, REQUIRED SO I COULD CONNECT BACK THE THE IMDB DATA ON TCONST AND NCONST WHERE NECESSARY

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)

# Importing into dataframe
# df_Title_Basics = pd.read_csv("./Prj_Data/DownLoadedData_Imdb/title.basics
# df_Name_Basics = pd.read_csv("./Prj_Data/DownLoadedData_Imdb/name.basics (

# df_Generes_With_tconst = pd.read_excel("df_Generes_With_tconst.xlsx")
df_IMDB_MovieCatalog = pd.read_excel("df_IMDB_MovieCatalog.xlsx")
df_starsplayers_knownForMovie = pd.read_excel ('./Prj_Data/ImdbScrapingData/
df_starsplayers = pd.read_excel ('./Prj_Data/ImdbScrapingData/TheNumbers/inp
# df_Principles = pd.read_csv("./Prj_Data/DownLoadedData_Imdb/title.principa

```

...

In [ ]:

```
#_________________________________Inserting nconst and tconst into Numbers M
```

In [ ]:

```
#1. Start by Narrowing the Basics and Principles tables down by only getting
#main movies tconts/ tabls
df_IMDB_MovieCatalog_Short = df_IMDB_MovieCatalog[["tconst","titleyear"]]
```

In [ ]:

```
ActorswithMoviesWeCareAbout = df_IMDB_MovieCatalog_Short.merge(df_Principles

```

In [ ]:

```
#2. Now that you have reduced the number of possible linking rows for our St
#  Remember this first requires you to make sure the names in IMDB (Robert D
# via updating the People metadata table manually cross referencing via eye

```

In [ ]:

```
#get star actors connected to nconst
df_starsplayers_WithMixedNconst = df_starsplayers.merge(ActorswithMoviesWeCa
df_starsplayers_WithMixedNconst = df_starsplayers_WithMixedNconst.drop_dupli

```

In [ ]:

```
#------SAVE----Used to create cross reference for names in IMDB to names in
# df_starsplayers_MixedNconst = df_starsplayers.merge(ActorswithMoviesWeCare
# df_starsplayers_MixedNconst = df_starsplayers_MixedNconst.drop_duplicates(
# #get star actors connected to nconst
```

In [ ]:

```
# Make sure not missing any important people when inserting nconst
# df_starsplayers_MixedNconst_Null_nconst = df_starsplayers_MixedNconst[df_s
df_starsplayers_MixedNconst[df_starsplayers_MixedNconst['nconst'].isnull()].
```

In [ ]:

```
#DF 1 NCONST ONLY
df_StarActors_With_nconst = df_starsplayers_WithMixedNconst[['Rank','Name_IM
df_StarActors_With_nconst = df_StarActors_With_nconst.drop_duplicates()
df_StarActors_With_nconst = df_StarActors_With_nconst.dropna(subset=["Name"]
```

In [ ]:

```
df_StarActors_With_nconst.to_excel("df_StarActors_With_nconst.xlsx")
```

In [ ]:

```
#DF 2 NCONST AND TCONST
df_StarActors_With_nconst_tconst = df_starsplayers_WithMixedNconst[['Rank',
       'Link', 'Contribution', 'BirthdayClean', 'nameyear', 'nconst', 'tcons
df_StarActors_With_nconst_tconst = df_StarActors_With_nconst_tconst.drop_dup
df_StarActors_With_nconst_tconst = df_StarActors_With_nconst_tconst.dropna(s
```

In [ ]:

```
df_StarActors_With_nconst_tconst.to_excel("df_StarActors_With_nconst_tconst.
```

In [ ]:

```

```

In [ ]:

```
#________________________CODE FOR CHECKING THE ABOVE FOR ERRORS,ETC
```

In [ ]:

```
df_StarActors_With_nconst[["Name","Contribution","nconst"]].sort_values(by="
```

In [ ]:

```
# df_StarActors_With_nconst_tconst[["Name","Contribution","nconst","tconst"]

df_StarActors_With_nconst_tconst[["Name","Contribution","tconst", "Movies"]]

```

In [ ]:

```
#_________________________END______________________________________________
```

In [ ]:

```

```

In [ ]:

```
#_________________________insert tconstinto KnownforMoviesTable____________
```

In [129]:

```
df_starsplayers_knownForMovie["IMDB_XREF"] = df_starsplayers_knownForMovie["
df_starsplayers_knownForMovie["ReleaseYear"] = df_starsplayers_knownForMovie

df_starsplayers_knownForMovie["ReleaseYear"] = df_starsplayers_knownForMovie

df_starsplayers_knownForMovie["title_linktoIMDB"] = df_starsplayers_knownFor

df_starsplayers_knownForMovie["titleyear_titlelinktoIMDB"] = df_starsplayers
```

In [131]:

```python
df_IMDB_MovieCatalog["primaryTitle"] = df_IMDB_MovieCatalog["primaryTitle"].
df_IMDB_MovieCatalog["titleyear1"] = df_IMDB_MovieCatalog["primaryTitle"] +

# df_IMDB_MovieCatalog["titleyear"] = df_IMDB_MovieCatalog.apply(lambda row
```

In [133]:

```python
df_IMDB_MovieCatalog_Short = df_IMDB_MovieCatalog[[ "primaryTitle","titleyea
```

In [135]:

```python
df_KnownForMerged = df_starsplayers_knownForMovie.merge(df_IMDB_MovieCatalog
```

In [136]:

```python
df_KnownForMerged.to_excel("df_StarActors_KnownForWithTconst.xlsx")
```

In [ ]:

In [77]:

```python
#________________________________USED TO UPDATE TITLE MANUALLY in Star Play D
df_KnownForMerged_nulls = df_KnownForMerged[["title_linktoIMDB","titleyear1"
df_KnownForMerged_nulls = df_KnownForMerged_nulls.drop_duplicates()
```

In [31]:

```python
# code for checking for errors
df_IMDB_MovieCatalog[["titleyear1","tconst"]][df_IMDB_MovieCatalog["titleyea
```

...

In [ ]:

```python
#___________________________________END________________________
```

In [ ]:

In [ ]:

```python
#________________________Remove Dups FROM Star Players Table... just asssume
```

In [ ]:

```python
df_Stars_no_dups = df_starsplayers.sort_values(by=["Name", "Contribution"])
df_Stars_no_dups1 = df_Stars_no_dups.drop_duplicates(subset="Name", keep="fi
```

In [ ]:

```python
df_Stars_no_dups1.to_excel("df_StarActors_NoDups.xlsx")
```