**Python**

1)Importing the required libraries

import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

import matplotlib.pyplot as plt

import seaborn as sns

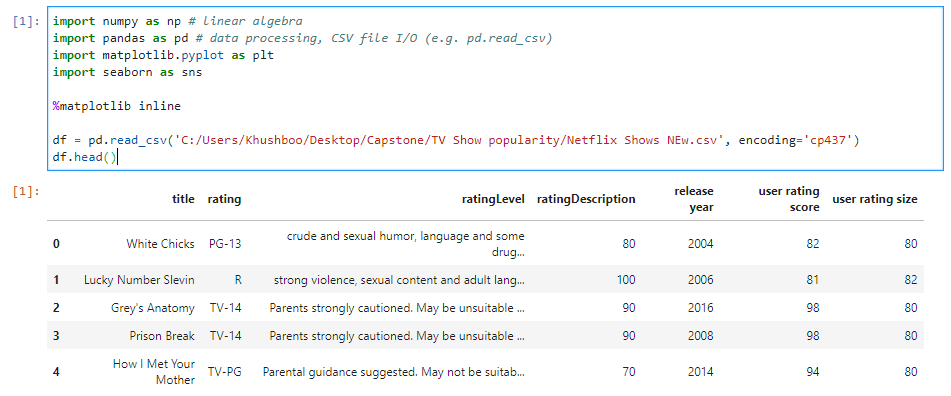
%matplotlib inline

2)Adding the file in Annaconda Navigator

df = pd.read\_csv('C:/Users/Khushboo/Desktop/Capstone/TV Show popularity/Netflix Shows Python.csv', encoding='cp437')

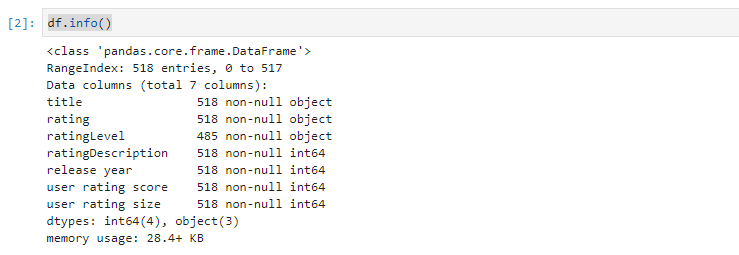
3)Reading top 4 entries of the file

df.head()



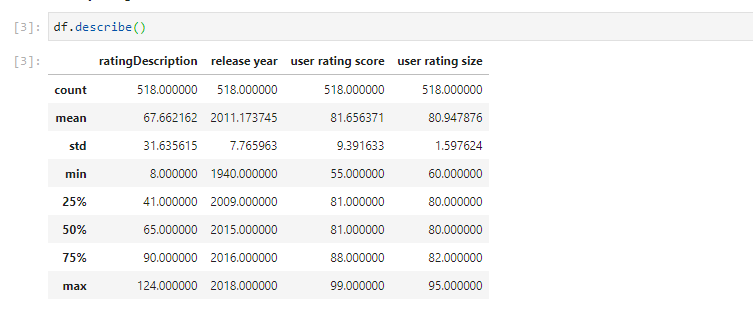
4)Reading how many data and columns are there in the file

df.info()



5)Calculating mean,total count and standard deviation of the values.

df.describe()

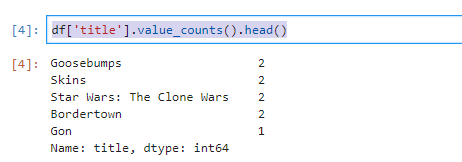


6)Checking for duplicate titles

The oldest release data was in 1940 and the most recent is from this year (2017).

First lets check to see if there are any duplicate titles.

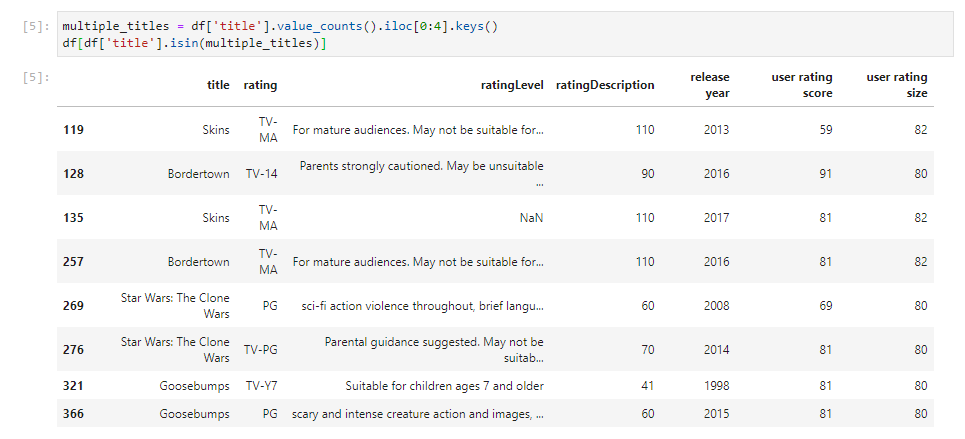
df['title'].value\_counts().head()



7)Showing the whole data of the duplicate titles

multiple\_titles = df['title'].value\_counts().iloc[0:4].keys()

df[df['title'].isin(multiple\_titles)]



8)Finding relationships among the data

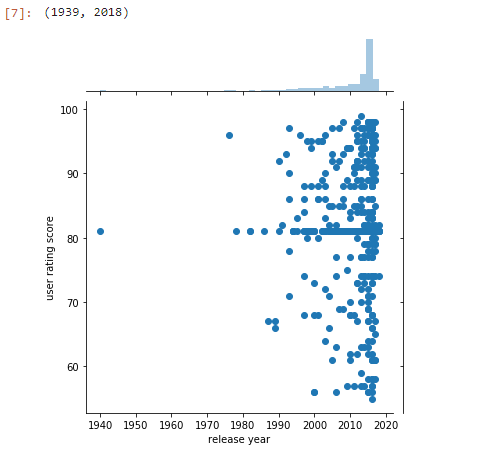
Over half of the entries in the original database were duplicates.

a) I will investigate whether there is any relationship between the release year the the user rating score.

sns.jointplot(data=df, y='user rating score', x='release year')

plt.xlim(1939, 2018)





There does not appear be a relationship between the release date and user score.

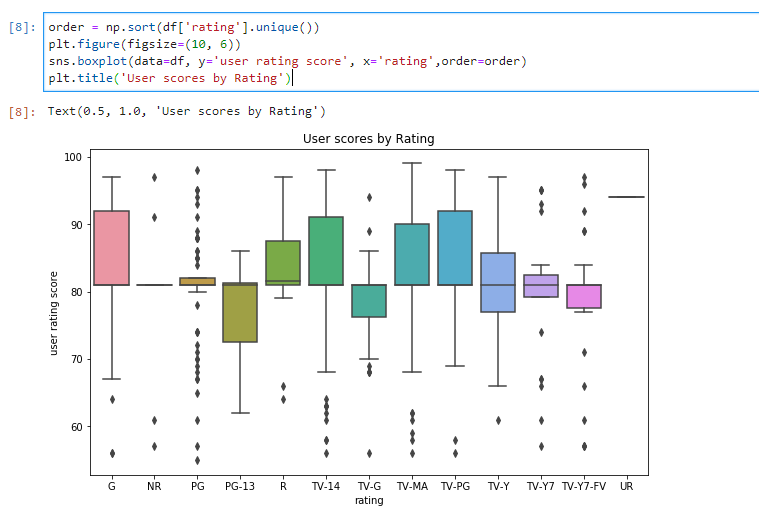
b)Next will be users ratings by ratings

order = np.sort(df['rating'].unique())

plt.figure(figsize=(10, 6))

sns.boxplot(data=df, y='user rating score', x='rating',order=order)

plt.title('User scores by Rating')



9)Combining ratings into age groups

Information from [Netflix Official Site](https://help.netflix.com/en/node/2064)

* Ratings for Little Kids: G, TV-Y, TV-G.
* Ratings for Older Kids: PG, TV-Y7, TV-Y7-FV, TV-PG.
* Ratings for Teens: PG-13, TV-14.
* Ratings for Adults: R, NC-17, NR, UR, TV-MA.

Ratings will be combined into age groups.

def age\_group(rating):

little = ['G','TV-Y','TV-G']

older = ['PG', 'TV-Y7', 'TV-Y7-FV', 'TV-PG']

teens = ['PG-13', 'TV-14']

adult = ['R', 'NC-17', 'NR', 'UR', 'TV-MA']

if rating in little:

return 'Little Kids'

elif rating in older:

return 'Older Kids'

elif rating in teens:

return 'Teens'

elif rating in adult:

return 'Adults'

else:

return 'Missing'

df['age\_group'] = df['rating'].apply(age\_group)

df.head()

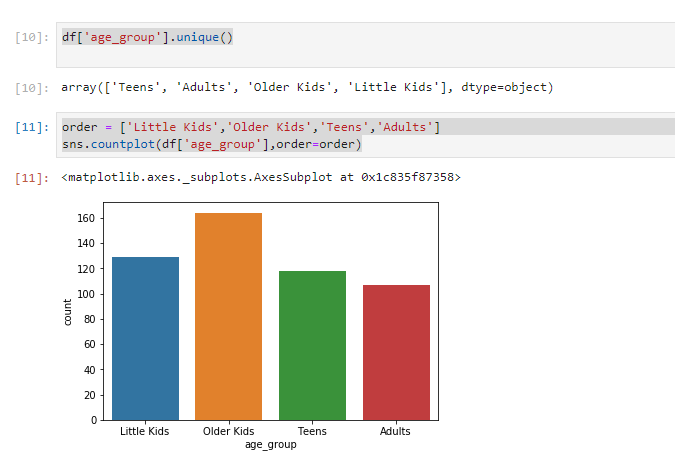


10)Checking for missing ratings and then plotting them in graph by age group.

df['age\_group'].unique()

order = ['Little Kids','Older Kids','Teens','Adults']

sns.countplot(df['age\_group'],order=order)

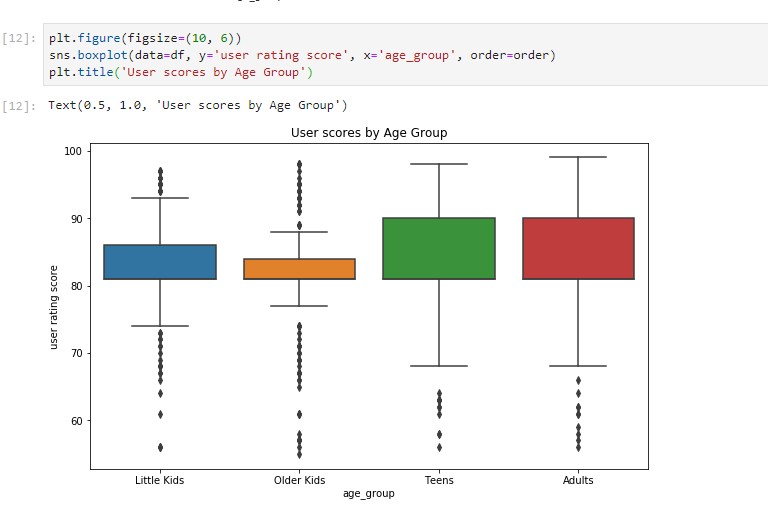


11)Checking for relationship between user scores and age group.

plt.figure(figsize=(10, 6))

sns.boxplot(data=df, y='user rating score', x='age\_group', order=order)

plt.title('User scores by Age Group')



12)Finding highest and lowest rated shows by age group

print('Highest and lowest user rated shows by age group:')

for group in order:

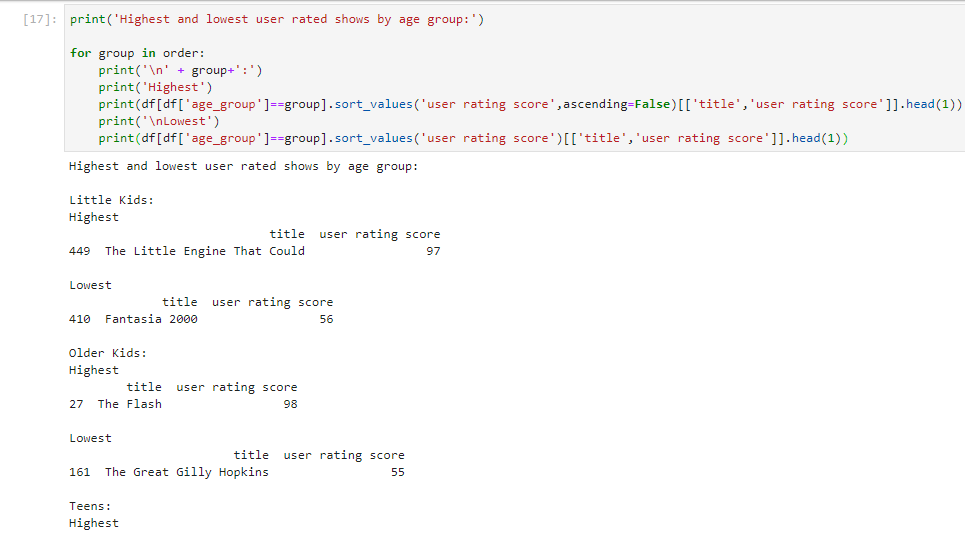
print('\n' + group+':')

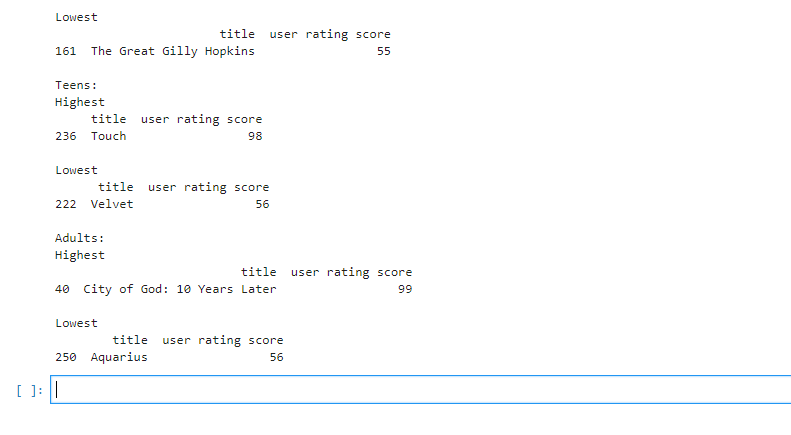
print('Highest')

print(df[df['age\_group']==group].sort\_values('user rating score',ascending=False)[['title','user rating score']].head(1))

print('\nLowest')

print(df[df['age\_group']==group].sort\_values('user rating score')[['title','user rating score']].head(1))





13)Finding how many shows telecast each year

year=data.groupby("release year")['title'].count().reset\_index().sort\_values(by='release year',ascending=False).reset\_index(drop=True)

year.columns=['release\_year','No\_of\_release\_showss']

year



