```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from statsmodels.distributions.empirical_distribution import ECDF
```

```
df=pd.read_csv('/content/drive/MyDrive/EDA/netflix.csv', parse_dates=['date_added'])
```

```
df=df.rename(columns={"date_added":"date","duration":"time"})
```

```
df.head(5)
```

	show_id	type	title	director	cast	country	date	release_year	rating	time	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020.0	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021.0	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
		TV			Sami Bouajila, Tracy		2021-			1	Crime TV Shows,	To protect his family from a

```
import re
df['time'] = df['time'].apply(lambda x: int(re.search(r'\d+', str(x)).group()) if re.search(r'\d+', str(x)) else np.nan)
```

```
df.head(5)
```

	show_id	type	title	director	cast	country	date	release_year	rating
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020.0	PG-13
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021.0	TV-MA

```
df['time'].mean()
```

```
69.84688777828259
```

```
def conv(x):
  if x=='G':
    return 1
  elif x=='TV-Y':
    return 2
  elif x=='TV-Y7':
    return 3
  elif x=='TV-Y7-FV':
    return 4
  elif x=='PG':
    return 5
  elif x=='TV-PG':
    return 6
  elif  x== 'PG-13':
    return 7
  elif x=='TV-14':
    return 8
  elif x=='TV-MA':
    return 9
  elif  x=='R':
    return 10
  elif x=='NC-17':
    return 11
  elif x=='NR':
    return 12
  elif x=='UR':
    return 13
  else:
    return np.nan


df['rating']=df['rating'].apply(conv)


df.head(2)
```

	show_id	type	title	director	cast	country	date	release_year	rating
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020.0	7.0

```
df.duplicated()
```

```
0       False
1       False
2       False
3       False
4       False
        ...
8809    False
8810    False
8811    False
8812    False
8813     True
Length: 8814, dtype: bool
```

```
df.loc[df.duplicated()].index.values
```

```
array([8813])
```

```
#df.loc[df.duplicated(subset=['show_id'])].tail(20)


df=df.drop_duplicates(keep='first')


df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8813 entries, 0 to 8812
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   show_id       8813 non-null   object
 1   type          8808 non-null   object
 2   title         8807 non-null   object
 3   director      6173 non-null   object
 4   cast          7982 non-null   object
 5   country       7976 non-null   object
 6   date          8797 non-null   datetime64[ns]
 7   release_year  8807 non-null   float64
```

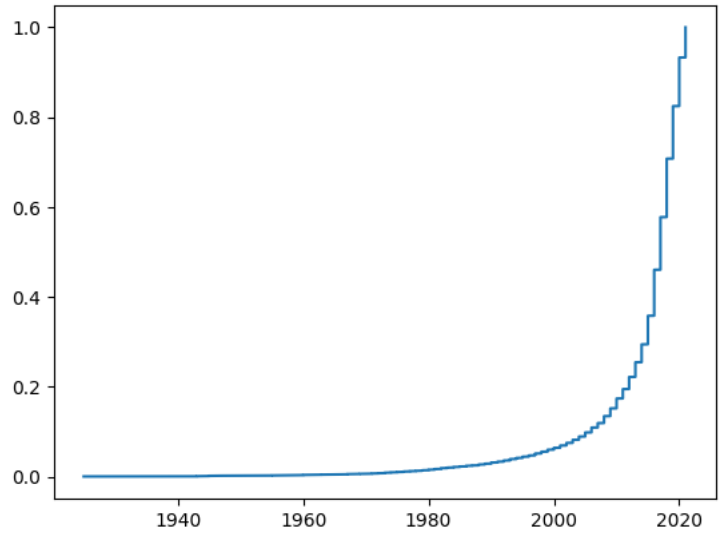
```
 8   rating        8580 non-null   float64
 9   time          8804 non-null   float64
 10  listed_in     8807 non-null   object
 11  description   8807 non-null   object
dtypes: datetime64[ns](1), float64(3), object(8)
memory usage: 895.1+ KB
```

```
df.loc[df.duplicated()]
```

show_id	type	title	director	cast	country	date	release_year	rating	time	lis

```
e=ECDF(df['release_year'])
plt.plot(e.x,e.y)
```

```
[<matplotlib.lines.Line2D at 0x7ebae82d0310>]
```




```
#Recent top 5 flim
df.sort_values(by=["rating",'release_year'],ascending=[False,True]).head(5)
```

	show_id	type	title	director	cast	country	date	release_year	ra
7058	s7059	Movie	Immoral Tales	Walerian Borowczyk	Lise Danvers, Fabrice Luchini, Charlotte Alexa...	France	2019-06-06	1974.0	
8790	s8791	Movie	You Don't Mess with the Zohan	Dennis Dugan	Adam Sandler, John Turturro, Emmanuelle Chriqu...	United States	2019-09-01	2008.0	

```
#df['numeric_val'] = df['duration'].apply(lambda x: int(re.search(r'\d+', str(x)).group()) if re.search(r'\d+', str(x)) else None)
```

```
df['release_year'].max()
```

```
2021.0
```

```
#df['release_year']=df["release_year"].fillna(0)
#df['release_year']=df['release_year'].astype(int)
```

```
# count of movies in year 2021
df.loc[df['release_year']=="2021"]
```

show_id	type	title	director	cast	country	date	release_year	rating	time	lis

```
#highest number of movies in all release_year?
ix=df['release_year'].value_counts().idxmax
ix
```

```
<bound method Series.idxmax of 2018.0    1147
2017.0    1032
2019.0    1030
2020.0     953
2016.0     902
          ...
1959.0       1
1925.0       1
1961.0       1
1947.0       1
1966.0       1
Name: release_year, Length: 74, dtype: int64>
```

```
df.loc[df["rating"]==13].head(5)
```

	show_id	type	title	director	cast	country	date	release_year	ra
7058	s7059	Movie	Immoral Tales	Walerian Borowczyk	Lise Danvers, Fabrice Luchini, Charlotte Alexa...	France	2019-06-06	1974.0	

```
df.loc[(df["country"]=='Japan') & (df["type"]=="Movie")& (df['rating']< 10)].head(2)
```

	show_id	type	title	director	cast	country	date	release_year	rati
51	s52	Movie	InuYasha the Movie 2: The Castle Beyond	Toshiya Shinohara	Kappei Yamaguchi, Satsuki Yukino, Mieko	Japan	2021-09-15	2002.0	

```
df.loc[df['rating'].isin([13])].nunique()
```

```
show_id         3
type            1
title           3
director        3
cast            3
country         3
date            3
release_year    3
rating          1
time            2
listed_in       2
description     3
dtype: int64
```

```
df.loc[df['rating'].isin([1])].nunique()
```

```
show_id         41
type             1
title           41
director        38
cast            40
country         12
date            28
release_year    27
rating           1
time            27
listed_in       17
description     41
dtype: int64
```

```
#find the highest rated movies year wise?
df1=df.sort_values(by=['release_year','rating'], ascending=[False,False])
df1.drop_duplicates(subset=['release_year'],keep='first').head(5)
```

	show_id	type	title	director	cast	country	date	release_year	ra
81	s82	Movie	Kate	Cedric Nicolas-Troyan	Mary Elizabeth Winstead, Jun Kunimura, Woody H...	United States	2021-09-10	2021.0	
721	s722	Movie	Rogue Warfare: Death of a Nation	Mike Gunther	Will Yun Lee, Jermaine Love, Rory	United States	2021-06-15	2020.0	

```
#productivity of a director= no of movies / total number of years
df2=df.loc[df['director']=='Rajiv Chilaka']
```

```
df2['release_year'].max()
```

```
2018.0
```

```
df2['release_year'].min()
```

```
2009.0
```

```
carrer_span=df2['release_year'].max()-df2['release_year'].min()
carrer_span
```

```
9.0
```

```
df2.shape[0]
```

```
19
```

```
productivity=df2.shape[0]/carrer_span
productivity
```

```
2.111111111111111
```

```
df.groupby(['director'])
```

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x7ebae923d4b0>
```

```
df.groupby(['director']).ngroups
```

```
4528
```

```
df['director'].nunique()
```

```
4528
```

```
df.groupby(['director']).groups
```

```
{'A. L. Vijay': [3537, 6078], 'A. Raajdheep': [2389], 'A. Salaam': [5549], 'A.R. Murugadoss': [4049, 4681], 'Aadish Keluskar':
[3602], 'Aamir Bashir': [5767], 'Aamir Khan': [1022], 'Aanand Rai': [2290], 'Aaron Burns': [7374], 'Aaron Hancox, Michael
McNamara': [6434], 'Aaron Hann, Mario Miscione': [5892], 'Aaron Lieber': [2778], 'Aaron Nee, Adam Nee': [6226], 'Aaron Sorkin':
[1842, 7479], 'Aaron Woodley': [2562], 'Aatmaram Dharne': [6439], 'Abba T. Makama': [2048, 6897], 'Abbas Alibhai Burmawalla, Mastan
Alibhai Burmawalla': [2285, 4601, 6107, 7212], 'Abbas Mustan': [2074], 'Abbas Tyrewala': [1018], 'Abby Epstein': [3828],
'Abdellatif Kechiche': [6338], 'Abdul Aziz Hashad': [6531], 'Abdulaziz Alshlahei': [436], 'Abel Ferrara': [8715], 'Abhay Chopra':
[4945], 'Abhijeet Deshpande': [2016], 'Abhijit Kokate, Srivinay Salian': [3729], 'Abhijit Panse': [2326, 2336], 'Abhinay Deo':
[1016, 3136, 5587], 'Abhishek Chaubey': [1939, 1942, 5651], 'Abhishek Kapoor': [3145, 4445, 4722], 'Abhishek Saxena': [7739],
'Abhishek Sharma': [3206, 4755, 8504], 'Abhishek Varman': [4712], 'Abir Sengupta': [1244], 'Abu Bakr Shawky': [8774], 'Achille
Brice': [704], 'Adam Alleca': [8080], 'Adam B. Stein, Zach Lipovsky': [2847], 'Adam Bhala Lough': [4156], 'Adam Bolt': [1911],
'Adam Collins, Luke Radford': [7686], 'Adam Davis, Jerry Kolber, Trey Nelson, Erich Sturm': [4135], 'Adam Del Giudice': [6817],
'Adam Deyoe': [5979], 'Adam Dubin': [5890], 'Adam Leon': [5524], 'Adam MacDonald': [6208], 'Adam Marino': [6282], 'Adam McKay':
[597, 8087, 8445], 'Adam Nimoy': [5693], 'Adam Randall': [5625], 'Adam Salky': [18], 'Adam Shankman': [6062, 6271, 6909], 'Adam
SjÃ¶berg': [7041], 'Adam Smith': [8627], 'Adam Wingard': [5318], 'Adam Wood': [6235], 'Adarsh Eshwarappa': [8012], 'Adekunle Nodash
Adejuyigbe': [2537], 'Adele K. Thomas, Richard Bailey': [8006], 'Adisorn Tresirikasem': [6230], 'Aditya Kripalani': [892, 8590],
'Aditya Sarpotdar': [3658, 7943], 'Aditya Vikram Sengupta': [3819], 'Adrian Murray': [8753], 'Adrian Teh': [1902, 2386, 4013],
'Adriana Trigiani': [2419], 'Adriano Rudiman': [893, 1054, 1193], 'Adrien Lagier, Ousmane Ly': [2675], 'AdriÃ¡n GarcÃa Bogliano':
[1146], 'Advait Chandan': [4975], 'Adze Ugah': [293, 2832, 3392], 'Afia Nathaniel': [6653], 'Afonso Poyart': [8108], 'Agasyah
Karim, Khalid Kashogi': [6268], 'Agnidev Chatterjee': [6564], 'AgustÃ Villaronga': [5373], 'Ah Loong': [6932], 'Ahishor Solomon':
[1000, 7137], 'Ahmad El-Badri': [2505, 2506, 6045, 8460], 'Ahmad Samir Farag': [7351], 'Ahmed Al-Badry': [2436, 2503, 8481], 'Ahmed
El Gendy': [7487], 'Ahmed Medhat': [2636], 'Ahmed Nader Galal': [2028, 2429, 2623, 2638], 'Ahmed Saleh': [2635], 'Ahmed Siddiqui':
[489], 'Ahmed Yousry, Hazem Fouda': [2614], 'Ahmed Zain': [2457, 3545], 'Ahmed Zein': [2463, 2464], 'Ahmet KatÄ±ksÄ±z': [3423],
'Ahn Byoung-wook': [2215], 'Ahsan Rahim': [4328], 'Aijaz Khan': [3822], 'Ainsley Gardiner, Briar Grace-Smith': [426], 'Aitor
Arregi, Jon GaraÃ±o': [4396], 'Aitor Arregi, Jon GaraÃ±o, Jose Mari Goenaga': [1728], 'Ajay Bahl': [6203], ...}
```

```
df.loc[[3537, 6078]]
```

	show_id	type	title	director	cast	country	date	release_year	ra
3537	s3538	Movie	Watchman	A. L. Vijay	G.V. Prakash Kumar, Samyuktha Hegde	India	2019-09-04	2019.0	

```
df.groupby(['director']).get_group('A. L. Vijay')
```

	show_id	type	title	director	cast	country	date	release_year	ra
3537	s3538	Movie	Watchman	A. L. Vijay	G.V. Prakash Kumar, Samyuktha Hegde	India	2019-09-04	2019.0	

```
#unsorted
df.groupby(['director'])['title'].count().sort_values(ascending=False)
```

```
director
Rajiv Chilaka                             19
RaÃºl Campos, Jan Suter                   18
Suhas Kadav                               16
Marcus Raboy                              16
Jay Karas                                 14
                                          ..
Jos Humphrey                               1
Jose Gomez                                 1
Jose Javier Reyes                          1
Joseduardo Giordano, Sergio Goyri Jr.      1
Khaled Youssef                             1
Name: title, Length: 4528, dtype: int64
```

```
#desc
df['director'].value_counts()
```

```
Rajiv Chilaka                     19
RaÃºl Campos, Jan Suter           18
Marcus Raboy                      16
Suhas Kadav                       16
Jay Karas                         14
                                  ..
Raymie Muzquiz, Stu Livingston     1
Joe Menendez                       1
Eric Bross                         1
Will Eisenberg                     1
Mozez Singh                        1
Name: director, Length: 4528, dtype: int64
```

```
df.groupby(['director'])['title'].count().sort_index(ascending=False)
```

```
director
Åženol SÃ¶nmez               2
Ã€lex Pastor, David Pastor    2
Ã‡agan Irmak                  1
Ã“skar ThÃ³r Axelsson         1
Ã–mer Faruk Sorak             2
                             ..
Aadish Keluskar               1
A.R. Murugadoss               2
A. Salaam                     1
A. Raajdheep                  1
A. L. Vijay                   2
Name: title, Length: 4528, dtype: int64
```

```
df.groupby(['director'])['release_year'].agg(['min','max'])
```

director	min	max
A. L. Vijay	2016.0	2019.0
A. Raajdheep	2020.0	2020.0
A. Salaam	1975.0	1975.0
A.R. Murugadoss	2017.0	2018.0
Aadish Keluskar	2018.0	2018.0
...	...	...
Ã–mer Faruk Sorak	2004.0	2011.0
Ã“skar ThÃ³r Axelsson	2017.0	2017.0
Ã‡agan Irmak	2005.0	2005.0
Ã€lex Pastor, David Pastor	2009.0	2020.0
Åženol SÃ¶nmez	2015.0	2019.0

4528 rows × 2 columns

```
df5=df.groupby(['director'])['release_year',"title"].aggregate({'release_year':[np.min,np.max],'title':["count"]})
```

```
<ipython-input-228-92a391649247>:1: FutureWarning: Indexing with multiple keys (implicitly converted to a tuple of keys) will be dep
  df5=df.groupby(['director'])['release_year',"title"].aggregate({'release_year':[np.min,np.max],'title':["count"]})
```

```
df5.columns
```

```
MultiIndex([('release_year',  'amin'),
            ('release_year',  'amax'),
            (       'title', 'count')],
           )
```

```
df5.columns=["_".join(i) for i in df5.columns]
```

```
df5.reset_index()
```

	director	release_year_amin	release_year_amax	title_count
0	A. L. Vijay	2016.0	2019.0	2
1	A. Raajdheep	2020.0	2020.0	1
2	A. Salaam	1975.0	1975.0	1
3	A.R. Murugadoss	2017.0	2018.0	2
4	Aadish Keluskar	2018.0	2018.0	1
...	...	...	...	...
4523	Ã–mer Faruk Sorak	2004.0	2011.0	2
4524	Ã“skar ThÃ³r Axelsson	2017.0	2017.0	1
4525	Ã‡agan Irmak	2005.0	2005.0	1
4526	Ã€lex Pastor, David Pastor	2009.0	2020.0	2
4527	Åženol SÃ¶nmez	2015.0	2019.0	2

4528 rows × 4 columns

```
df6=df.groupby(["director"])["release_year",'title'].agg(year_max=("release_year","max"),
                                                         year_min=("release_year","min"),
                                                         title_count=('title',"count")).reset_index()
df6
```

```
<ipython-input-239-e5f620cecad6>:1: FutureWarning: Indexing with multiple keys (impli
  df6=df.groupby(["director"])["release_year",'title'].agg(year_max=("release_year","
```

	director	year_max	year_min	title_count
0	A. L. Vijay	2019.0	2016.0	2
1	A. Raajdheep	2020.0	2020.0	1
2	A. Salaam	1975.0	1975.0	1
3	A.R. Murugadoss	2018.0	2017.0	2
4	Aadish Keluskar	2018.0	2018.0	1
...	...	...	...	...

```
df6["career_span"]=df6['year_max']-df6["year_min"]
```

```
df6
```

	director	year_max	year_min	title_count	career_span
0	A. L. Vijay	2019.0	2016.0	2	3.0
1	A. Raajdheep	2020.0	2020.0	1	0.0
2	A. Salaam	1975.0	1975.0	1	0.0
3	A.R. Murugadoss	2018.0	2017.0	2	1.0
4	Aadish Keluskar	2018.0	2018.0	1	0.0
...	...	...	...	...	...
4523	Ã–mer Faruk Sorak	2011.0	2004.0	2	7.0
4524	Ã“skar ThÃ³r Axelsson	2017.0	2017.0	1	0.0
4525	Ã‡agan Irmak	2005.0	2005.0	1	0.0
4526	Ã€lex Pastor, David Pastor	2020.0	2009.0	2	11.0
4527	Åženol SÃ¶nmez	2019.0	2015.0	2	4.0

4528 rows × 5 columns

```
df6['productivity']=df6['title_count']/df6['career_span']
```

```
df6
```

	director	year_max	year_min	title_count	career_span	productivity
0	A. L. Vijay	2019.0	2016.0	2	3.0	0.666667
1	A. Raajdheep	2020.0	2020.0	1	0.0	inf
2	A. Salaam	1975.0	1975.0	1	0.0	inf
3	A.R. Murugadoss	2018.0	2017.0	2	1.0	2.000000
4	Aadish Keluskar	2018.0	2018.0	1	0.0	inf
...	...	...	...	...	...	...
4523	Ã–mer Faruk Sorak	2011.0	2004.0	2	7.0	0.285714
4524	Ã“skar ThÃ³r Axelsson	2017.0	2017.0	1	0.0	inf
4525	Ã‡agan Irmak	2005.0	2005.0	1	0.0	inf
4526	Ã€lex Pastor, David Pastor	2020.0	2009.0	2	11.0	0.181818
4527	Åženol SÃ¶nmez	2019.0	2015.0	2	4.0	0.500000

```
df6.sort_values(by=['productivity'],ascending=False)
```