In [1]:
```
import pandas as pd
import numpy as np
import csv
import random as rd
```

In [2]:
```
df_movies = pd.read_csv("C:\\Users\\amang\\Downloads\\task2\\movies1.csv",sep=
df_movies.columns = ['Movie_IDs','Movie_Name','Genre']
df_movies.dropna(inplace=True)
df_movies = df_movies.replace('"','',regex=True)
df_movies.head()
```

Out[2]:

	Movie_IDs	Movie_Name	Genre
0	2	Jumanji (1995)	Adventure|Children's|Fantasy
1	3	Grumpier Old Men (1995)	Comedy|Romance
2	4	Waiting to Exhale (1995)	Comedy|Drama
3	5	Father of the Bride Part II (1995)	Comedy
4	6	Heat (1995)	Action|Crime|Thriller

In [3]:
```
df_rating = pd.read_csv("C:\\Users\\amang\\Downloads\\task2\\ratings1.csv",sep
df_rating.columns = ['ID','Movies_ID','Rating','Timestamp']
df_rating.dropna(inplace=True)
df_rating = df_rating.replace('"','',regex=True)
df_rating.head()
```

Out[3]:

	ID	Movies_ID	Rating	Timestamp
0	1	661	3	978302109
1	1	914	3	978301968
2	1	3408	4	978300275
3	1	2355	5	978824291
4	1	1197	3	978302268

In [4]:

```
df_user = pd.read_csv("C:\\Users\\amang\\Downloads\\task2\\users1.csv",sep='::
df_user.columns =['UserID','Gender','Age','Occupation','Zip-code']
df_user.dropna(inplace=True)
df_user = df_user.replace('"','',regex=True)
df_user.head()
```

Out[4]:

	UserID	Gender	Age	Occupation	Zip-code
0	2	M	56	16	70072
1	3	M	25	15	55117
2	4	M	45	7	02460
3	5	M	25	20	55455
4	6	F	50	9	55117

In [5]:

```
data = pd.concat([df_movies,df_rating,df_user],axis = 1)
data.head()
```

Out[5]:

	Movie_IDs	Movie_Name	Genre	ID	Movies_ID	Rating	Timestamp	UserI
0	2	Jumanji (1995)	Adventure|Children's|Fantasy	1	661	3	978302109	
1	3	Grumpier Old Men (1995)	Comedy|Romance	1	914	3	978301968	
2	4	Waiting to Exhale (1995)	Comedy|Drama	1	3408	4	978300275	
3	5	Father of the Bride Part II (1995)	Comedy	1	2355	5	978824291	
4	6	Heat (1995)	Action|Crime|Thriller	1	1197	3	978302268	

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

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000208 entries, 0 to 1000207
Data columns (total 12 columns):
 #   Column      Non-Null Count    Dtype  
---  ------      --------------    -----  
 0   Movie_IDs   3832 non-null     object 
 1   Movie_Name  3832 non-null     object 
 2   Genre       3832 non-null     object 
 3   ID          1000208 non-null  object 
 4   Movies_ID   1000208 non-null  int64  
 5   Rating      1000208 non-null  int64  
 6   Timestamp   1000208 non-null  object 
 7   UserID      6039 non-null     object 
 8   Gender      6039 non-null     object 
 9   Age         6039 non-null     float64
 10  Occupation  6039 non-null     float64
 11  Zip-code    6039 non-null     object 
dtypes: float64(2), int64(2), object(8)
memory usage: 99.2+ MB
```

In [7]:
```
data.describe(include = "all")
```

Out[7]:

	Movie_IDs	Movie_Name	Genre	ID	Movies_ID	Rating	Timestamp	UserI
count	3832	3832	3832	1000208	1.000208e+06	1.000208e+06	1000208	603
unique	3832	3832	301	6040	NaN	NaN	458455	603
top	2	Jumanji (1995)	Drama	4169	NaN	NaN	975528402	
freq	1	1	822	2314	NaN	NaN	30	
mean	NaN	NaN	NaN	NaN	1.865541e+03	3.581563e+00	NaN	Na
std	NaN	NaN	NaN	NaN	1.096041e+03	1.117102e+00	NaN	Na
min	NaN	NaN	NaN	NaN	1.000000e+00	1.000000e+00	NaN	Na
25%	NaN	NaN	NaN	NaN	1.030000e+03	3.000000e+00	NaN	Na
50%	NaN	NaN	NaN	NaN	1.835000e+03	4.000000e+00	NaN	Na
75%	NaN	NaN	NaN	NaN	2.770000e+03	4.000000e+00	NaN	Na
max	NaN	NaN	NaN	NaN	3.952000e+03	5.000000e+00	NaN	Na

In [8]:
```
data.isna()
```

Out[8]:

	Movie_IDs	Movie_Name	Genre	ID	Movies_ID	Rating	Timestamp	UserID	Gender
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...
1000203	True	True	True	False	False	False	False	True	True
1000204	True	True	True	False	False	False	False	True	True
1000205	True	True	True	False	False	False	False	True	True
1000206	True	True	True	False	False	False	False	True	True
1000207	True	True	True	False	False	False	False	True	True

1000208 rows × 12 columns

In [9]:
```
data.isna().sum()
```

Out[9]:
```
Movie_IDs     996376
Movie_Name    996376
Genre         996376
ID                 0
Movies_ID          0
Rating             0
Timestamp          0
UserID        994169
Gender        994169
Age           994169
Occupation    994169
Zip-code      994169
dtype: int64
```

In [10]:

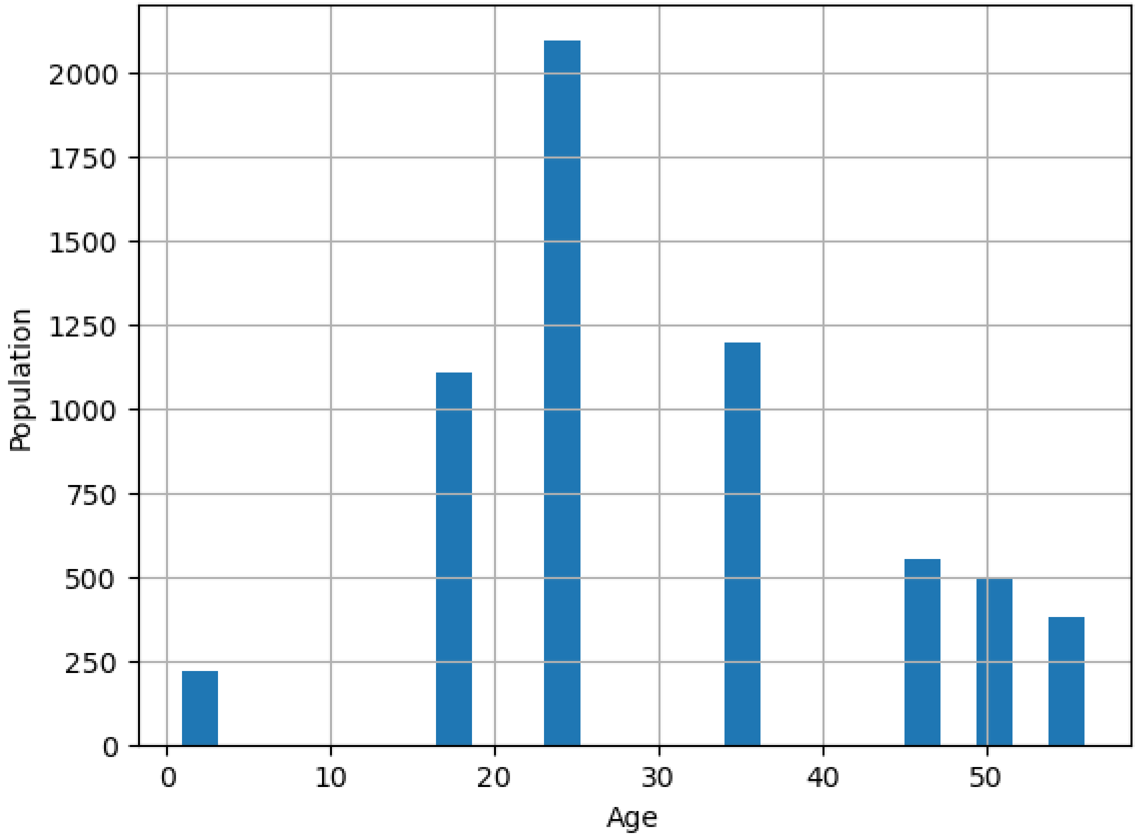
```
data.dropna(axis=0,inplace=True)
data.isna().sum()
```

Out[10]:

```
Movie_IDs     0
Movie_Name    0
Genre         0
ID            0
Movies_ID     0
Rating        0
Timestamp     0
UserID        0
Gender        0
Age           0
Occupation    0
Zip-code      0
dtype: int64
```

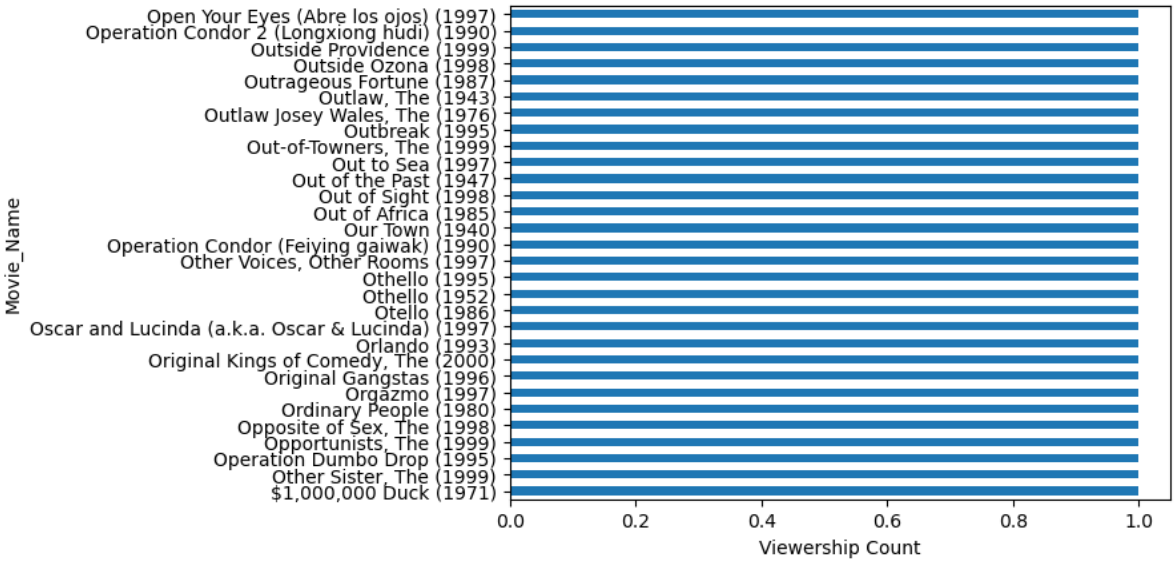
In [11]:

```
import matplotlib.pyplot as plt
df_user['Age'].hist(bins=25)
plt.xlabel('Age')
plt.ylabel('Population')
plt.show()
```



In [12]:
```
res = data.groupby("Movie_Name").size().sort_values(ascending=False)[:30]
plt.ylabel("Title")
plt.xlabel("Viewership Count")
res.plot(kind="barh")
```

Out[12]:
```
<AxesSubplot:xlabel='Viewership Count', ylabel='Movie_Name'>
```



In [13]:
```
data['Rating'].unique()
```

Out[13]:
```
array([3, 4, 5, 2, 1], dtype=int64)
```

In [14]:
```
df = data['Genre'].str.get_dummies(sep='|')
df.head()
```

Out[14]:

	Action	Adventure	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Fil N
0	0	1	0	1	0	0	0	0	1	
1	0	0	0	0	1	0	0	0	0	
2	0	0	0	0	1	0	0	1	0	
3	0	0	0	0	1	0	0	0	0	
4	1	0	0	0	0	1	0	0	0	

In [15]:
```
df = pd.concat((df,data['Rating']),axis=1)
df.head()
```

Out[15]:

	Action	Adventure	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Fil N
0	0	1	0	1	0	0	0	0	1	
1	0	0	0	0	1	0	0	0	0	
2	0	0	0	0	1	0	0	1	0	
3	0	0	0	0	1	0	0	0	0	
4	1	0	0	0	0	1	0	0	0	

In [43]:
```
df = pd.concat((df,data['Gender']),axis = 1)
df = pd.concat((df,data['Age']),axis = 1)
df.head()
df = df.loc[:,~df.columns.duplicated()]
df
```

Out[43]:

	Action	Adventure	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy
0	0	1	0	1	0	0	0	0	1
1	0	0	0	0	1	0	0	0	0
2	0	0	0	0	1	0	0	1	0
3	0	0	0	0	1	0	0	0	0
4	1	0	0	0	0	1	0	0	0
...	...	...	...	...	...	...	...	...	...
3877	0	0	0	0	1	0	0	0	0
3878	0	0	0	0	0	0	0	1	0
3879	0	0	0	0	0	0	0	1	0
3880	0	0	0	0	0	0	0	1	0
3881	0	0	0	0	0	0	0	1	0

3832 rows × 21 columns

In [17]:

```
from sklearn.model_selection import train_test_split
from sklearn.impute import SimpleImputer
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error,r2_score, mean_absolute_error
```

```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\__init__.py:155: UserWarnin
g: A NumPy version >=1.18.5 and <1.25.0 is required for this version of SciPy
(detected version 1.25.0
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

In [18]:

```
X = df.drop(["Rating","Age","Gender"],axis=1)
y = df["Rating"]
```

In [34]:

```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.4,random_stat
```

In [35]:

```
model=LinearRegression()
```

In [36]:

```
model.fit(X_train,y_train)
```

Out[36]:

```
LinearRegression()
```

In [37]:

```
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test,y_pred)
rmse = np.sqrt(mse)
print(mse)
print(rmse)
```

```
1.2071128036499075
1.098686854226402
```

#Decision tree

In [38]:

```
from sklearn.tree import DecisionTreeRegressor
model = DecisionTreeRegressor()
model.fit(X_train, y_train)
```

Out[38]:

```
DecisionTreeRegressor()
```

In [39]:

```
y_pred = model.predict(X_test)



mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Absolute Error: {mae}")
print(f"Mean Squared Error: {mse}")
print(f"R-squared: {r2}")
```

```
Mean Absolute Error: 0.939357165603892
Mean Squared Error: 1.3132902142011524
R-squared: -0.09512489832448923
```