Combined Exercise 2

Total Marks: 100 marks (Part 1: **`40 Marks`** and Part 2: **`60 Marks`**)

Allowed libraries to Use:

- Pandas
- NumPy
- Matplotlib (For visualization should use only Matplotlib) and Graph should include x-axis label, y-axis label and Title

Note:

1. Please submit pdf file along with ipynb file. You need to show the basis of your calculation or inference.

Part 1 (40 Marks)

The following questions needs to be answered for the dataset in the below link

http://files.grouplens.org/datasets/movielens/ml-1m.zip (http://files.grouplens.org/datasets/movielens/ml-1m.zip)

This dataset mainly contains three `.dat` files and data available in this file is in a tabular format and delimited with a :: as a separator.

For more details please go through the "README" file which is available in the dataset.

Note: Load the dataset into `CSV file` by using pandas libray into three different dataframe stated (users, ratings and movies)

Questions:

1. Find out the total number of movies, total number of ratings and total number of users_who_rated for movies (9 marks)

2. Visualize the distribution of overall rating by users (3 marks)

3. Visualize the users rating distribution (3 marks)

4. Genre distribution as a pie chart (10 marks)
ALthough there can be mutiple genre assigned to one movie. We'll assume that first Genre is the primary. Pie chart can be created based on that first Genre

5. List out top 15 ranked movies (consider only those movies which are rated by atleast 100 users) (15 marks)

PART 1 Question 1 Answer

In [ ]:

```
1 import pandas as pd
```

In [220]:

```
1 movies = pd.read_csv('movies.dat', sep="::", names = ['MovieID','Title','Genres'])
2 ratings = pd.read_csv('ratings.dat', sep="::", names = ['UserID','MovieID','Rating','Ti
3 users = pd.read_csv('ratings.dat', sep="::", names = ['UserID','Gender','Age','Occupati
```

executed in 10.2s, finished 11:53:38 2020-11-30

```
C:\Users\jymch\anaconda3\lib\site-packages\ipykernel_launcher.py:1: ParserWa
rning: Falling back to the 'python' engine because the 'c' engine does not s
upport regex separators (separators > 1 char and different from '\s+' are in
terpreted as regex); you can avoid this warning by specifying engine='pytho
n'.
  """Entry point for launching an IPython kernel.
C:\Users\jymch\anaconda3\lib\site-packages\ipykernel_launcher.py:2: ParserWa
rning: Falling back to the 'python' engine because the 'c' engine does not s
upport regex separators (separators > 1 char and different from '\s+' are in
terpreted as regex); you can avoid this warning by specifying engine='pytho
n'.

C:\Users\jymch\anaconda3\lib\site-packages\ipykernel_launcher.py:3: ParserWa
rning: Falling back to the 'python' engine because the 'c' engine does not s
upport regex separators (separators > 1 char and different from '\s+' are in
terpreted as regex); you can avoid this warning by specifying engine='pytho
n'.
  This is separate from the ipykernel package so we can avoid doing imports
until
```

In [41]:

```
print(movies.isnull().sum())
print("--")
print(movies.nunique())
print("--")
print(ratings.isnull().sum())
print("--")
print(ratings.nunique())
print("--")
print(users.isnull().sum())
print("--")
print(users.nunique())
print("--")
print(len(ratings), len(users),len(movies))
```

executed in 1.02s, finished 09:52:23 2020-11-30

```
MovieID       0
Title         0
Genres        0
2          2025
3          3347
4          3768
5          3868
6          3882
dtype: int64
--
MovieID    3883
Title      3883
Genres       18
2            17
3            15
4            13
5             6
6             1
dtype: int64
--
UserID       0
MovieID      0
Rating       0
Timestamp    0
dtype: int64
--
UserID         6040
MovieID        3706
Rating            5
Timestamp    458455
dtype: int64
--
UserID              0
Gender              0
Age                 0
Occupation          0
Zip-code      1000209
dtype: int64
--
UserID          6040
Gender          3706
Age                5
Occupation    458455
```

```
Zip-code           0
dtype: int64
--
1000209 1000209 3883
```

Since there is no NaN or any cell with null values and the length of these dataframes are shown using len(dataframe) code, the total number of movies is 3883, total number of ratings is 1000209 and the total number of users_who_rated for movies is 6040.

PART 1 Question 2 Answer

In [25]:

```
1 import matplotlib.pyplot as plt
```

executed in 1.51s, finished 09:33:14 2020-11-30

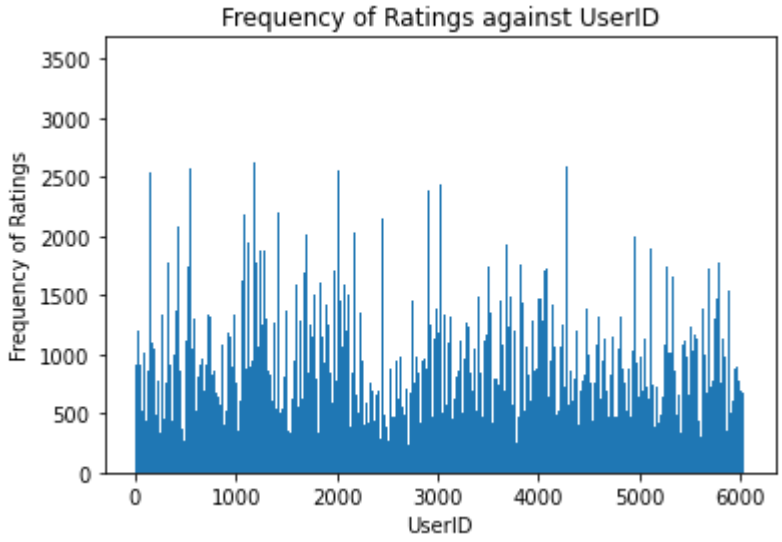
In [221]:

```
1 #usersrating = ratings.groupby(['UserID']).count()
2 plt.hist(ratings['UserID'],bins=1000)
3 plt.title("Frequency of Ratings against UserID")
4 plt.xlabel("UserID")
5 plt.ylabel("Frequency of Ratings")
```

executed in 1.59s, finished 11:53:43 2020-11-30

Out[221]:

```
Text(0, 0.5, 'Frequency of Ratings')
```



PART 1 Question 3 Answer - Each user rated how many movies?

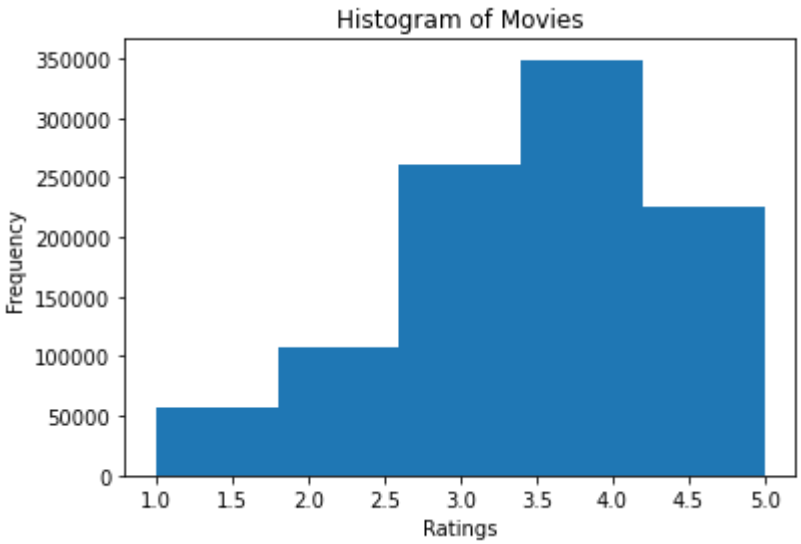
In [235]:

```
plt.hist(ratings['Rating'].astype(int),bins=5)
plt.xlabel("Ratings")
plt.ylabel("Frequency")
plt.title("Histogram of Movies")
```

executed in 176ms, finished 11:56:40 2020-11-30

Out[235]:

```
Text(0.5, 1.0, 'Histogram of Movies')
```



PART 1 Question 4 Answer

In [212]:

```
movies['Genres'].str.split("|",expand=True)
movies[['Genres','2','3','4','5','6']] = movies.Genres.str.split("|",expand=True)
```

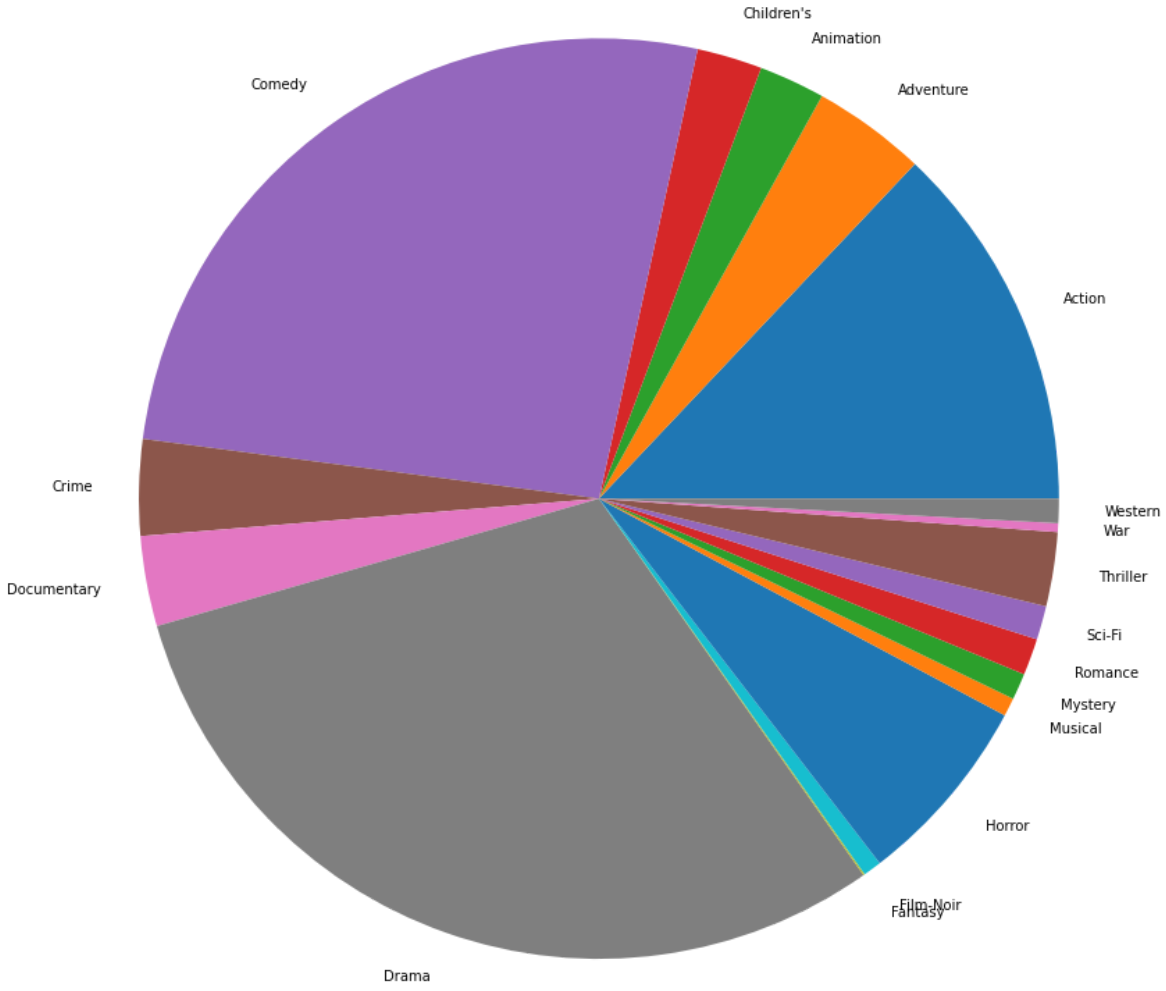
executed in 39ms, finished 11:50:15 2020-11-30

In [213]:

```
moviesGGB = movies.groupby(['Genres']).count()

#plt.pie(movies.groupby(['Genres']).count())
fig = plt.figure(figsize =(12,12))
plt.pie(moviesGGB['MovieID'], labels = moviesGGB.index)
plt.tight_layout()
```

executed in 166ms, finished 11:50:24 2020-11-30



PART 1 Question 5 Answer

In [94]:

```
movies['MovieID'].astype(int)

ratingsGB = ratings.groupby(['MovieID'])['UserID'].nunique()

print(ratingsGB.nlargest(15).index.tolist()) # List of MovieID with the 15 highest user

movies.loc[movies['MovieID'].isin(ratingsGB.nlargest(15).index)][['MovieID','Title']]
```

executed in 564ms, finished 10:32:57 2020-11-30

```
[2858, 260, 1196, 1210, 480, 2028, 589, 2571, 1270, 593, 1580, 1198, 608, 27
62, 110]
```

Out[94]:

	MovieID	Title
108	110	Braveheart (1995)
257	260	Star Wars: Episode IV - A New Hope (1977)
476	480	Jurassic Park (1993)
585	589	Terminator 2: Judgment Day (1991)
589	593	Silence of the Lambs, The (1991)
604	608	Fargo (1996)
1178	1196	Star Wars: Episode V - The Empire Strikes Back...
1180	1198	Raiders of the Lost Ark (1981)
1192	1210	Star Wars: Episode VI - Return of the Jedi (1983)
1250	1270	Back to the Future (1985)
1539	1580	Men in Black (1997)
1959	2028	Saving Private Ryan (1998)
2502	2571	Matrix, The (1999)
2693	2762	Sixth Sense, The (1999)
2789	2858	American Beauty (1999)

Part 2 (60 Marks)

The following questions needs to be answered for the dataset in the below link:

https://grouplens.org/datasets/movielens/100k/ (https://grouplens.org/datasets/movielens/100k/)

This dataset contains multiple files and data in this file is a tab separated. But you have to use only three files as listed below

1. u.data (The full u data set, 100000 ratings by 943 users on 1682 items. Each user has rated at least 20 movies)
2. u.item (Information about the items (movies))
3. u.user (Demographic information about the users)

For more information of the dataset pleasse go through the README file.

Instructions:

1. Load the dataset into `CSV file` by using pandas libray as a three different dataframe stated (users, ratings and movies)
2. Merge all these three dataframe into single dataframe as a `movielens` for furthur prosessing

Questions:

Each of the questions carry 10 marks.

1. Among Male doctors in the age group 23- 30 (10 Marks)

- a. What genre of the movies they have rated 5 and count the no. of ratings for each genere?
- b. What genre of the movies they have rated above 3 and count the no. of ratings for each genere?
- c. What genre of the movies they have rated below 4 and count the no. of ratings for each genere?
- d. Write your inferences based on above

2. Among Female healthcare professionals in the age group 23- 30 (10 Marks)

- a. What genre of the movies they have rated 5 and count the no. of ratings for each genere?
- b. What genre of the movies they have rated above 3 and count the no. of ratings for each genere?
- c. What genre of the movies they have rated below 4 and count the no. of ratings for each genere?
- d. Write your inferences based on above

3. Among Male doctors in the age group 35 – 50 (10 Marks)

- a. What genre of the movies they have rated 5 and count the no. of ratings for each genere?
- b. What genre of the movies they have rated above 3 and count the no. of ratings for each genere?
- c. What genre of the movies they have rated below 4 and count the no. of ratings for each genere?
- d. Write your inferences based on above

4. Among Female healthcare professionals in the age group 35 - 50 (10 Marks)

- a. What genre of the movies they have rated 5 and count the no. of ratings for each genere?
- b. What genre of the movies they have rated above 3 and count the no. of ratings for each genere?
- c. What genre of the movies they have rated below 4 and count the no. of ratings for each genere?
- d. Write your inferences based on above

In [214]:

```
import pandas

u_cols = ['user','age','sex','occupation','zip_code']
users = pd.read_csv('u.user', sep='|', names=u_cols,encoding='latin-1')

r_cols = ['user','movie','rating','unix_timestamp']
ratings = pd.read_csv('u.data', sep='\t',names=r_cols,encoding = 'latin-1')

m_cols = ['movie','title','release_date','video_release_date','imdb_url','unknown','Act
movies = pd.read_csv('u.item',sep='|',names=m_cols,encoding='latin-1')

occupation = pd.read_csv("u.occupation")
genres1 = ['Action','Animation',"Children's", 'Comedy','Crime','Documentary','Drama','F
```

executed in 204ms, finished 11:50:43 2020-11-30

Part 2 Question 1a

In [215]:

```
# Question 1a
usersdoc = users[users['occupation'].isin(['doctor'])]
usersdoc = usersdoc[23<usersdoc['age']]
usersdoc = usersdoc[usersdoc['age']<30]
usersdoc = usersdoc[usersdoc['sex']=='M']

rateusers = ratings.loc[ratings['user'].isin(usersdoc['user'])]
rate5docs = rateusers[rateusers['rating']==5]
movGenres = movies[movies.index.isin(rate5docs['movie'])][genres1]
movGenres.loc['Total']= movGenres.sum()
movGenres.iloc[[-1],::]
```

executed in 30ms, finished 11:50:46 2020-11-30

Out[215]:

	Action	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horro
Total	6	0	2	15	3	0	24	0	1	:

Part 2 Question 1b

In [216]:

```
# Question 1b
usersdoc = users[users['occupation'].isin(['doctor'])]
usersdoc = usersdoc[23<usersdoc['age']]
usersdoc = usersdoc[usersdoc['age']<30]
usersdoc = usersdoc[usersdoc['sex']=='M']

rateusers = ratings.loc[ratings['user'].isin(usersdoc['user'])]
rate5docs = rateusers[rateusers['rating']>3]
movGenres = movies[movies.index.isin(rate5docs['movie'])][genres1]
movGenres.loc['Total']= movGenres.sum()
movGenres.iloc[[-1],::]
```

executed in 37ms, finished 11:50:48 2020-11-30

Out[216]:

	Action	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horro
Total	14	6	6	61	10	6	88	0	5	

Part 2 Question 1c

In [217]:

```
# Question 1c
usersdoc = users[users['occupation'].isin(['doctor'])]
usersdoc = usersdoc[23<usersdoc['age']]
usersdoc = usersdoc[usersdoc['age']<30]
usersdoc = usersdoc[usersdoc['sex']=='M']

rateusers = ratings.loc[ratings['user'].isin(usersdoc['user'])]
rate5docs = rateusers[rateusers['rating']<4]
movGenres = movies[movies.index.isin(rate5docs['movie'])][genres1]
movGenres.loc['Total']= movGenres.sum()
movGenres.iloc[[-1],::]
```

executed in 25ms, finished 11:50:50 2020-11-30

Out[217]:

	Action	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horro
Total	18	12	13	59	10	5	56	5	0	

Part 2 Question 1d

Young male doctors like to watch drama, followed by comedy.

```
# Part 2 Question 2a
```

In [218]:

```
# Question 2a

usersdoc = users[users['occupation'].isin(['healthcare'])]
usersdoc = usersdoc[23<usersdoc['age']]
usersdoc = usersdoc[usersdoc['age']<30]
usersdoc = usersdoc[usersdoc['sex']=='F']

rateusers = ratings.loc[ratings['user'].isin(usersdoc['user'])]
rate5docs = rateusers[rateusers['rating']==5]
movGenres = movies[movies.index.isin(rate5docs['movie'])][genres1]
movGenres.loc['Total']= movGenres.sum()
movGenres.iloc[[-1],::]
```

executed in 30ms, finished 11:50:53 2020-11-30

Out[218]:

	Action	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horro
Total	2	0	0	5	2	0	7	0	0	

Part 2 Question 2b

In [202]:

```
# Question 2b

usersdoc = users[users['occupation'].isin(['healthcare'])]
usersdoc = usersdoc[23<usersdoc['age']]
usersdoc = usersdoc[usersdoc['age']<30]
usersdoc = usersdoc[usersdoc['sex']=='F']

rateusers = ratings.loc[ratings['user'].isin(usersdoc['user'])]
rate5docs = rateusers[rateusers['rating']>3]
movGenres = movies[movies.index.isin(rate5docs['movie'])][genres1]
movGenres.loc['Total']= movGenres.sum()
movGenres.iloc[[-1],::]
```

executed in 27ms, finished 11:30:49 2020-11-30

Out[202]:

	Action	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horro
Total	13	4	6	37	11	1	49	3	0	

Part 2 Question 2c

In [203]:

```
# Question 2c

usersdoc = users[users['occupation'].isin(['healthcare'])]
usersdoc = usersdoc[23<usersdoc['age']]
usersdoc = usersdoc[usersdoc['age']<30]
usersdoc = usersdoc[usersdoc['sex']=='F']

rateusers = ratings.loc[ratings['user'].isin(usersdoc['user'])]
rate5docs = rateusers[rateusers['rating']<4]
movGenres = movies[movies.index.isin(rate5docs['movie'])][genres1]
movGenres.loc['Total']= movGenres.sum()
movGenres.iloc[[-1],::]
```

executed in 31ms, finished 11:31:07 2020-11-30

Out[203]:

	Action	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horro
Total	11	2	5	27	6	0	45	2	0	

Part 2 Question 2d

Young female healthcare workers like to watch drama, followed by some comedy.

Part 3 Question 3a

In [204]:

```
# Question 3a
usersdoc = users[users['occupation'].isin(['doctor'])]
usersdoc = usersdoc[35<usersdoc['age']]
usersdoc = usersdoc[usersdoc['age']<50]
usersdoc = usersdoc[usersdoc['sex']=='M']

rateusers = ratings.loc[ratings['user'].isin(usersdoc['user'])]
rate5docs = rateusers[rateusers['rating']==5]
movGenres = movies[movies.index.isin(rate5docs['movie'])][genres1]
movGenres.loc['Total']= movGenres.sum()
movGenres.iloc[[-1],::]
```

executed in 34ms, finished 11:31:20 2020-11-30

Out[204]:

	Action	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horro
Total	2	0	0	10	7	1	16	0	1	

Part 2 Question 3b

In [219]:

```
# Question 3b

usersdoc = users[users['occupation'].isin(['doctor'])]
usersdoc = usersdoc[35<usersdoc['age']]
usersdoc = usersdoc[usersdoc['age']<50]
usersdoc = usersdoc[usersdoc['sex']=='M']

rateusers = ratings.loc[ratings['user'].isin(usersdoc['user'])]
rate5docs = rateusers[rateusers['rating']>3]
movGenres = movies[movies.index.isin(rate5docs['movie'])][genres1]
movGenres.loc['Total']= movGenres.sum()
movGenres.iloc[[-1],::]
```

executed in 25ms, finished 11:51:57 2020-11-30

Out[219]:

	Action	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horro
Total	12	1	2	22	10	1	45	0	2	

In [206]:

```
# Question 3c

usersdoc = users[users['occupation'].isin(['doctor'])]
usersdoc = usersdoc[35<usersdoc['age']]
usersdoc = usersdoc[usersdoc['age']<50]
usersdoc = usersdoc[usersdoc['sex']=='M']

rateusers = ratings.loc[ratings['user'].isin(usersdoc['user'])]
rate5docs = rateusers[rateusers['rating']<4]
movGenres = movies[movies.index.isin(rate5docs['movie'])][genres1]
movGenres.loc['Total']= movGenres.sum()
movGenres.iloc[[-1],::]
```

executed in 37ms, finished 11:31:48 2020-11-30

Out[206]:

	Action	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horro
Total	2	1	2	8	2	0	7	1	0	

Question 3d

Older male doctors like to watch drama too and some thriller. They don't like to watch comedies as much as young male doctors do.

Part 2 Question 4a

In [207]:

```
# Question 4a

usersdoc = users[users['occupation'].isin(['healthcare'])]
usersdoc = usersdoc[35<usersdoc['age']]
usersdoc = usersdoc[usersdoc['age']<50]
usersdoc = usersdoc[usersdoc['sex']=='F']

rateusers = ratings.loc[ratings['user'].isin(usersdoc['user'])]
rate5docs = rateusers[rateusers['rating']==5]
movGenres = movies[movies.index.isin(rate5docs['movie'])][genres1]
movGenres.loc['Total']= movGenres.sum()
movGenres.iloc[[-1],::]
```

executed in 30ms, finished 11:34:49 2020-11-30

Out[207]:

	Action	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horro
Total	13	6	10	16	6	1	40	1	1	

Part 2 Question 4b

In [208]:

```
# Question 4b

usersdoc = users[users['occupation'].isin(['healthcare'])]
usersdoc = usersdoc[35<usersdoc['age']]
usersdoc = usersdoc[usersdoc['age']<50]
usersdoc = usersdoc[usersdoc['sex']=='F']

rateusers = ratings.loc[ratings['user'].isin(usersdoc['user'])]
rate5docs = rateusers[rateusers['rating']>3]
movGenres = movies[movies.index.isin(rate5docs['movie'])][genres1]
movGenres.loc['Total']= movGenres.sum()
movGenres.iloc[[-1],::]
```

executed in 28ms, finished 11:35:06 2020-11-30

Out[208]:

	Action	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horro
Total	54	17	29	63	19	5	129	3	7	

Part 2 Question 4c

In [209]:

```
# Question 4c

usersdoc = users[users['occupation'].isin(['healthcare'])]
usersdoc = usersdoc[35<usersdoc['age']]
usersdoc = usersdoc[usersdoc['age']<50]
usersdoc = usersdoc[usersdoc['sex']=='F']

rateusers = ratings.loc[ratings['user'].isin(usersdoc['user'])]
rate5docs = rateusers[rateusers['rating']<4]
movGenres = movies[movies.index.isin(rate5docs['movie'])][genres1]
movGenres.loc['Total']= movGenres.sum()
movGenres.iloc[[-1],::]
```

executed in 23ms, finished 11:35:20 2020-11-30

Out[209]:

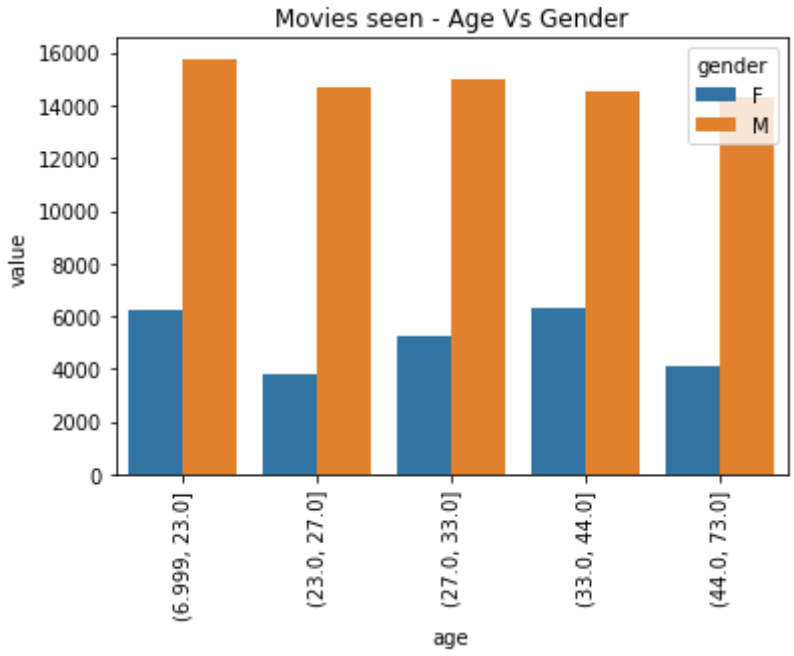
	Action	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horro
Total	50	19	46	84	16	2	101	7	3	

Question 4d

Older female healthcare workers tend to watch drama, followed by romance and thriller. Similar to their younger counterpart, older healthcare workers like watch drama but they don't watch as much comedies as their younger counterpart.

5. For each of the following graphs, write your inferences:(20 Marks)

1. Age-wise Vs Gender distribution of the users who rated for the movies (10 Marks)



Cross tab with count

Cross tab with count

age, gender	F	M
(6.999, 23.0]	6235	15760
(23.0, 27.0]	3808	14687
(27.0, 33.0]	5276	14977
(33.0, 44.0]	6340	14503
(44.0, 73.0]	4081	14333

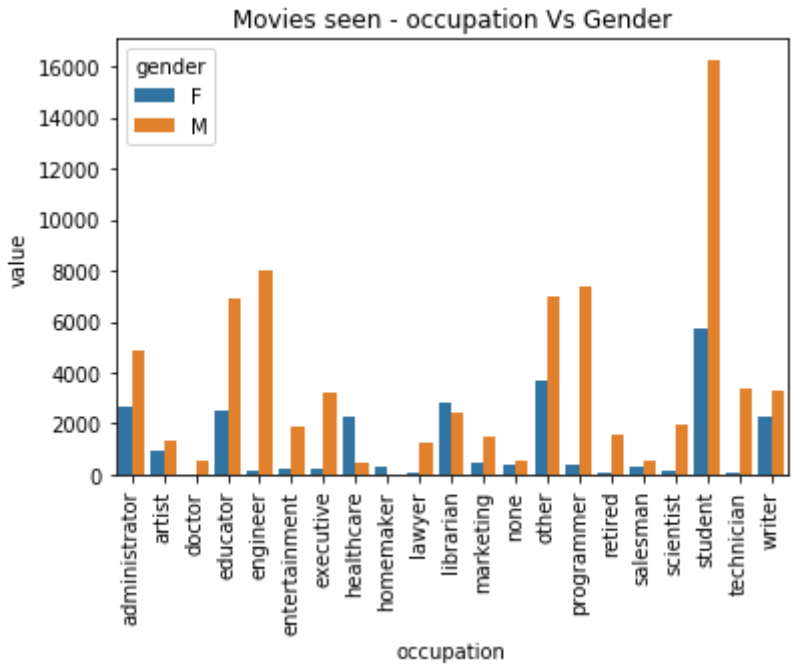
Cross tab with row %

age, gender	F	M
(6.999, 23.0]	28.35	71.65
(23.0, 27.0]	20.59	79.41
(27.0, 33.0]	26.05	73.95
(33.0, 44.0]	30.42	69.58
(44.0, 73.0]	22.16	77.84

ANSWER:

Male movie-goers tend to rate movies than female movie-goers, the ratio is approximately 3:1 where males are 3 times more likely to rate a movie than females.

2. Occupation-wise Vs Gender distribution of the users who rated for the movies (10 Marks)



Cross tab with count

Occupation	Gender Female	Gender Male
administrator	2654	4825
artist	971	1337
doctor	0	540

Occupation	Gender Female	Gender Male
educator	2537	6905
engineer	145	8030
entertainment	225	1870
executive	221	3182
healthcare	2307	497
homemaker	269	30
lawyer	69	1276
librarian	2860	2413
marketing	442	1508
none	365	536
other	3665	6998
programmer	419	7382
retired	71	1538
salesman	339	517
scientist	139	1919
student	5696	16261
technician	108	3398
writer	2238	3298

Cross tab with row %

Occupation	Gender F%	Gender M%
administrator	35.49	64.51
artist	42.07	57.93
doctor	0.00	100.00
educator	26.87	73.13
engineer	1.77	98.23
entertainment	10.74	89.26
executive	6.49	93.51
healthcare	82.28	17.72
homemaker	89.97	10.03
lawyer	5.13	94.87
librarian	54.24	45.76
marketing	22.67	77.33
none	40.51	59.49
other	34.37	65.63
programmer	5.37	94.63
retired	4.41	95.59
salesman	39.60	60.40
scientist	6.75	93.25

Occupation	Gender F%	Gender M%
student	25.94	74.06
technician	3.08	96.92
writer	40.43	59.57

ANSWER:

Conclusions are difficult to be drawn from this data (i.e. statements like, 'what is the probability of someone belonging to a particular occupation and of a particular gender leaving a movie rating?').

Data pertaining to the occupations like lawyer, doctor and scientist is to be taken lightly as the base rate of a random person drawn from the larger population having these occupations is low (i.e. there are not many people who are lawyer or doctor or scientist a priori), consequently, the samples taken within the dataset belonging to these occupations may not be representative of the people belonging to these occupations in the population. In order to more accurately draw conclusions from this dataset (e.g. such as the claim 'female lawyers are less probable to rate a movie') or verify hypotheses gathered from this data, one would need to use the Bayes rule, taking into consideration, the prior distribution for each of these occupations.

Similarly, comparisons between different genders in some occupations like engineer, programmer and the like have to be done using the Bayes rule, taking into account the prior distribution of genders in these occupation. In the absence of these prior distributions, it will be difficult to draw conclusions on how genders-occupation pair influences the likelihood of someone belonging to a particular gender-occupation pair leaving a rating for a movie.

However, at first glance, it does seem likely (i.e. high likelihood) that for some occupations, males are predominantly more likely to leave a rating than females. Yet, as explained above, to conclude rigorously, one will need more information on the prior distribution for each occupation and for each gender.

In [ ]:

```
1
```