Users (u.user), Movies (u.item), Ratings (u.data)

```
import pandas as pd
import matplotlib.pyplot as plt
```

Read data from ml-100k/u.user

```
column_Users = 'userId | age | gender | occupation | zip_code'.split('
| ')
#Read data from files and put them in Users
Users = pd.read_table('./ml-100k/ml-100k/u.user', names=column_Users,
sep='|')
#Show only the first 10 rows
# Users =
#-----------------------
Users.head(10)
```

	userId	age	gender	occupation	zip_code
0	1	24	M	technician	85711
1	2	53	F	other	94043
2	3	23	M	writer	32067
3	4	24	M	technician	43537
4	5	33	F	other	15213
5	6	42	M	executive	98101
6	7	57	M	administrator	91344
7	8	36	M	administrator	05201
8	9	29	M	student	01002
9	10	53	M	lawyer	90703

Read from ml-100k/u.item

```
column_Movies = ['movieId','movieTitle','releaseDate',
'videoReleaseDate', 'IMDbURL','unknown', 'Action',

'Adventure','Animation','Childrens','Comedy','Crime','Documentary',
'Drama', 'Fantasy',
                'Film-Noir','Horror','Musical', 'Mystery',
'Romance','Sci-Fi','Thriller','War','Western']

#May need encoding option when opening the csv file where encoding =
"ISO-8859-1"
# May need to define engine='python'

#Read data from files and put them in Movies
Movies = pd.read_table('./ml-100k/ml-100k/u.item',
names=column_Movies, sep='|', encoding='ISO-8859-1')
#Show only the first 10 rows
#Movies =
```

```
#-----------------------------------------------
Movies.head(10)
```

```
   movieId                                          movieTitle
releaseDate  \
0        1                                Toy Story (1995)  01-Jan-
1995
1        2                                GoldenEye (1995)  01-Jan-
1995
2        3                               Four Rooms (1995)  01-Jan-
1995
3        4                               Get Shorty (1995)  01-Jan-
1995
4        5                                  Copycat (1995)  01-Jan-
1995
5        6  Shanghai Triad (Yao a yao yao dao waipo qiao) ...  01-Jan-
1995
6        7                             Twelve Monkeys (1995)  01-Jan-
1995
7        8                                     Babe (1995)  01-Jan-
1995
8        9                           Dead Man Walking (1995)  01-Jan-
1995
9       10                               Richard III (1995)  22-Jan-
1996

   videoReleaseDate                                                IMDbURL
\
0               NaN  http://us.imdb.com/M/title-exact?Toy%20Story%2...

1               NaN  http://us.imdb.com/M/title-exact?GoldenEye%20(...

2               NaN  http://us.imdb.com/M/title-exact?Four%20Rooms%...

3               NaN  http://us.imdb.com/M/title-exact?Get%20Shorty%...

4               NaN  http://us.imdb.com/M/title-exact?Copycat%20(1995)

5               NaN  http://us.imdb.com/Title?Yao+a+yao+yao+dao+wai...

6               NaN  http://us.imdb.com/M/title-exact?Twelve%20Monk...

7               NaN     http://us.imdb.com/M/title-exact?Babe%20(1995)

8               NaN  http://us.imdb.com/M/title-exact?Dead%20Man%20...

9               NaN  http://us.imdb.com/M/title-exact?Richard%20III...


   unknown  Action  Adventure  Animation  Childrens  ...  Fantasy
Film-Noir  \
```

```
0         0         0           0           1           1  ...         0
0
1         0         1           1           0           0  ...         0
0
2         0         0           0           0           0  ...         0
0
3         0         1           0           0           0  ...         0
0
4         0         0           0           0           0  ...         0
0
5         0         0           0           0           0  ...         0
0
6         0         0           0           0           0  ...         0
0
7         0         0           0           0           1  ...         0
0
8         0         0           0           0           0  ...         0
0
9         0         0           0           0           0  ...         0
0

   Horror  Musical  Mystery  Romance  Sci-Fi  Thriller  War  Western
0       0        0        0        0       0         0    0        0
1       0        0        0        0       0         1    0        0
2       0        0        0        0       0         1    0        0
3       0        0        0        0       0         0    0        0
4       0        0        0        0       0         1    0        0
5       0        0        0        0       0         0    0        0
6       0        0        0        0       1         0    0        0
7       0        0        0        0       0         0    0        0
8       0        0        0        0       0         0    0        0
9       0        0        0        0       0         0    1        0

[10 rows x 24 columns]
```

Read data from ml-100k/u.data

```
column_Ratings = [ 'userId', 'movieId', 'rating', 'timestamp']
#Read data from files and put them in Ratings
Ratings = pd.read_table('./ml-100k/ml-100k/u.data',
names=column_Ratings, sep='\t')
#Show only the first 10 rows
#Ratings =
#-----------------------------------------------
Ratings.head(10)

   userId  movieId  rating  timestamp
0     196      242       3  881250949
1     186      302       3  891717742
```

```
2      22      377       1  878887116
3     244       51       2  880606923
4     166      346       1  886397596
5     298      474       4  884182806
6     115      265       2  881171488
7     253      465       5  891628467
8     305      451       3  886324817
9       6       86       3  883603013
```

```
#Check data in Users
#is any row NULL ?
#-----------------------
Users.isna().sum()
```

```
user id       0
age           0
gender        0
occupation    0
zip code      0
dtype: int64
```

```
#Check data in Movies
#is any row NULL ?
#-----------------------
Movies.isna().sum()
```

```
movieId                0
movieTitle             0
releaseDate            1
videoReleaseDate    1682
IMDbURL                3
unknown                0
Action                 0
Adventure              0
Animation              0
Childrens              0
Comedy                 0
Crime                  0
Documentary            0
Drama                  0
Fantasy                0
Film-Noir              0
Horror                 0
Musical                0
Mystery                0
Romance                0
Sci-Fi                 0
Thriller               0
War                    0
```

```
Western                  0
dtype: int64
```

releaseDate 1 videoReleaseDate 1682 IMDbURL 3

```
#Check data in Ratings
#is any row NULL ?
#-----------------------
Ratings.isna().sum()
```

```
userId       0
movieId      0
rating       0
timestamp    0
dtype: int64
```

```
#Fix dirty data
#---------------------
# Dropping the entire videoReleaseDate column since it is fully
missing
Movies = Movies.drop('videoReleaseDate', axis=1)
# Show the first 10 records after cleansing
Movies.head(10)
```

```
   movieId                                          movieTitle
releaseDate  \
0        1                                Toy Story (1995)  01-Jan-
1995
1        2                                GoldenEye (1995)  01-Jan-
1995
2        3                               Four Rooms (1995)  01-Jan-
1995
3        4                               Get Shorty (1995)  01-Jan-
1995
4        5                                  Copycat (1995)  01-Jan-
1995
5        6  Shanghai Triad (Yao a yao yao dao waipo qiao) ...  01-Jan-
1995
6        7                            Twelve Monkeys (1995)  01-Jan-
1995
7        8                                    Babe (1995)  01-Jan-
1995
8        9                          Dead Man Walking (1995)  01-Jan-
1995
9       10                               Richard III (1995)  22-Jan-
1996

                                           IMDbURL  unknown  Action
\
0  http://us.imdb.com/M/title-exact?Toy%20Story%2...        0       0
```

```
1  http://us.imdb.com/M/title-exact?GoldenEye%20(...         0       1

2  http://us.imdb.com/M/title-exact?Four%20Rooms%...         0       0

3  http://us.imdb.com/M/title-exact?Get%20Shorty%...         0       1

4  http://us.imdb.com/M/title-exact?Copycat%20(1995)         0       0

5  http://us.imdb.com/Title?Yao+a+yao+yao+dao+wai...         0       0

6  http://us.imdb.com/M/title-exact?Twelve%20Monk...         0       0

7     http://us.imdb.com/M/title-exact?Babe%20(1995)         0       0

8  http://us.imdb.com/M/title-exact?Dead%20Man%20...         0       0

9  http://us.imdb.com/M/title-exact?Richard%20III...         0       0


   Adventure  Animation  Childrens  Comedy  ...  Fantasy  Film-Noir
Horror  \
0          0          1          1       1  ...        0          0
0
1          1          0          0       0  ...        0          0
0
2          0          0          0       0  ...        0          0
0
3          0          0          0       1  ...        0          0
0
4          0          0          0       0  ...        0          0
0
5          0          0          0       0  ...        0          0
0
6          0          0          0       0  ...        0          0
0
7          0          0          1       1  ...        0          0
0
8          0          0          0       0  ...        0          0
0
9          0          0          0       0  ...        0          0
0

   Musical  Mystery  Romance  Sci-Fi  Thriller  War  Western
0        0        0        0       0         0    0        0
1        0        0        0       0         1    0        0
2        0        0        0       0         1    0        0
3        0        0        0       0         0    0        0
4        0        0        0       0         1    0        0
5        0        0        0       0         0    0        0
```

```
6         0         0         0         1         0    0         0
7         0         0         0         0         0    0         0
8         0         0         0         0         0    0         0
9         0         0         0         0         0    1         0

[10 rows x 23 columns]
```

```
#Recheck data in Movies table
#-----------------------------------
Movies.isna().sum()
```

```
movieId        0
movieTitle     0
releaseDate    1
IMDbURL        3
unknown        0
Action         0
Adventure      0
Animation      0
Childrens      0
Comedy         0
Crime          0
Documentary    0
Drama          0
Fantasy        0
Film-Noir      0
Horror         0
Musical        0
Mystery        0
Romance        0
Sci-Fi         0
Thriller       0
War            0
Western        0
dtype: int64
```

```
#From the previous cell, if some columns are still Null, show how many
records in each column are still null?
#-----------------------------------
Movies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1682 entries, 0 to 1681
Data columns (total 23 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   movieId      1682 non-null   int64
 1   movieTitle   1682 non-null   object
 2   releaseDate  1681 non-null   object
 3   IMDbURL      1679 non-null   object
```

```
 4   unknown      1682 non-null   int64
 5   Action       1682 non-null   int64
 6   Adventure    1682 non-null   int64
 7   Animation    1682 non-null   int64
 8   Childrens    1682 non-null   int64
 9   Comedy       1682 non-null   int64
 10  Crime        1682 non-null   int64
 11  Documentary  1682 non-null   int64
 12  Drama        1682 non-null   int64
 13  Fantasy      1682 non-null   int64
 14  Film-Noir    1682 non-null   int64
 15  Horror       1682 non-null   int64
 16  Musical      1682 non-null   int64
 17  Mystery      1682 non-null   int64
 18  Romance      1682 non-null   int64
 19  Sci-Fi       1682 non-null   int64
 20  Thriller     1682 non-null   int64
 21  War          1682 non-null   int64
 22  Western      1682 non-null   int64
dtypes: int64(20), object(3)
memory usage: 302.4+ KB
```

```
#Show a number of rows in Movies dataFrame
#-----------------------------------------
Movies.shape
```

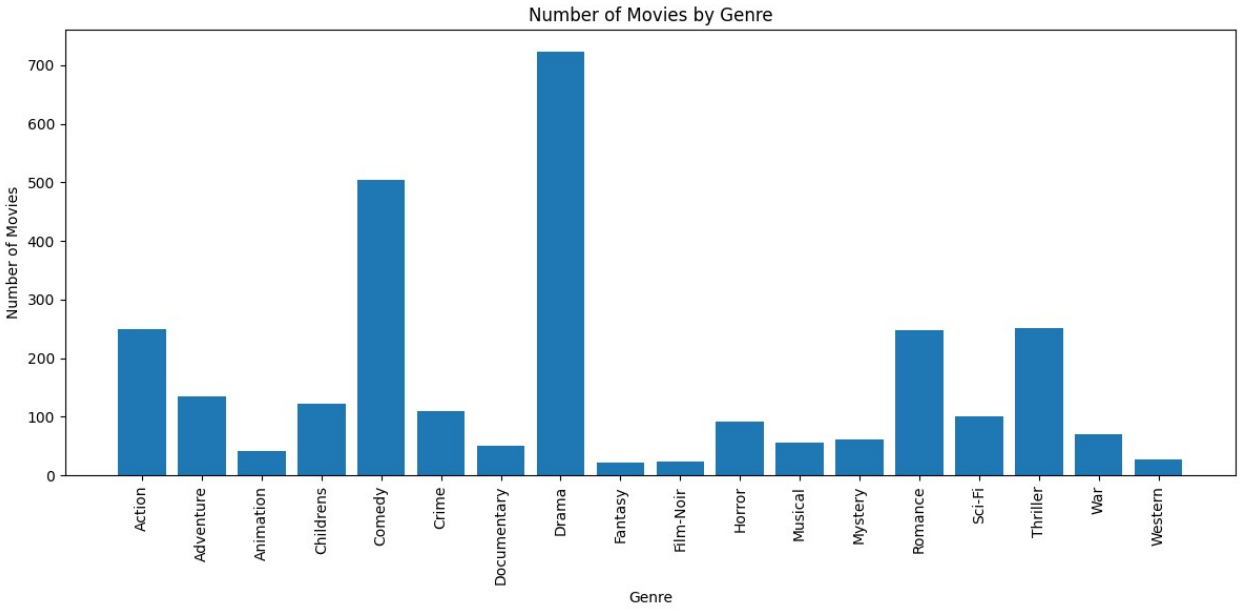
```
(1682, 23)
```

```
#Handle rows that are null in the Movies datafram.  Justify your
answer.
#-------------------------------------------------------------
Movies = Movies.dropna()

# We have dropped the missing values since they are only 3 row data
Movies.shape
```

```
(1679, 23)
```

```
genre_list =
['Action','Adventure','Animation','Childrens','Comedy','Crime','Docume
ntary', 'Drama', 'Fantasy',
                    'Film-Noir','Horror','Musical', 'Mystery',
'Romance','Sci-Fi','Thriller','War','Western']
#Count a number of movies for each genre as shown in the genre_list
per_genre_count = Movies[genre_list].sum()

#Plot graph to show a number of movies for each genre
#-----------------------------------------------------
plt.figure(figsize=(12,6))
plt.bar(range(len(per_genre_count)), [count for genre, count in
zip(genre_list, per_genre_count)])
```

```
plt.xticks(range(len(per_genre_count)), [genre for genre, count in
zip(genre_list, per_genre_count)], rotation=90)
plt.title('Number of Movies by Genre')
plt.xlabel('Genre')
plt.ylabel('Number of Movies')
plt.tight_layout()
plt.show()
```




```
#Find users who are less than 25 years old and are female
#-------------------------------------------------------
mask = (Users.age < 25) & (Users.gender == 'F')

Users[mask]
```

	userId	age	gender	occupation	zip_code
23	24	21	F	artist	94533
34	35	20	F	homemaker	42459
35	36	19	F	student	93117
48	49	23	F	student	76111
51	52	18	F	student	55105
..	...	...	...	...	...
886	887	14	F	student	27249
903	904	17	F	student	61073
916	917	22	F	student	20006
920	921	20	F	student	98801
924	925	18	F	salesman	49036

[68 rows x 5 columns]

```
#=-----------------------------
joined_usr_rat = pd.merge(Users, Ratings)
joined_usr_rat.groupby(Users.userId)['rating'].mean()
```

```
userId
1.0      4.0
2.0      3.0
3.0      4.0
4.0      4.0
5.0      4.0
        ...
939.0    4.0
940.0    5.0
941.0    4.0
942.0    5.0
943.0    3.0
Name: rating, Length: 943, dtype: float64
```

```
#----------------------------------
mask = (Users['occupation'] == 'programmer') & (Users['gender'] ==
'M')

Users[mask].age.mean()
```

```
33.21666666666667
```

```
Movies.releaseDate = pd.to_datetime(Movies.releaseDate)

#----------------------------------------
joined_mov_rat = pd.merge(Movies, Ratings, on='movieId')

mask = (joined_mov_rat.releaseDate >= '1996-01-01')
# .sort_values(by='rating', ascending=False)[['movieTitle', 'rating',
'releaseDate']]
joined_mov_rat =
joined_mov_rat[mask].groupby('movieTitle').rating.mean().reset_index()
.sort_values(by='rating', ascending=False)
joined_mov_rat.head()
```

```
                        movieTitle  rating
525      Santa with Muscles (1996)     5.0
560  Someone Else's America (1995)     5.0
573                Star Kid (1997)     5.0
11             Aiqing wansui (1994)     5.0
492              Prefontaine (1997)     5.0
```

```
#-------------------------------
joined_pearson = pd.merge(Users, Ratings)

joined_pearson['age'].corr(joined_pearson['rating'], method='pearson')
```

```
0.054460397809808034
```

:Justification:

So, I merged two dataframes based on userId key, then calculated correlation based on age to rating using pearson method.