EDS Assignmnet-5

Group Members:

Tuhinansh Sharma (265)

Tushar Najawan (266)

Atharva Pawar (279)

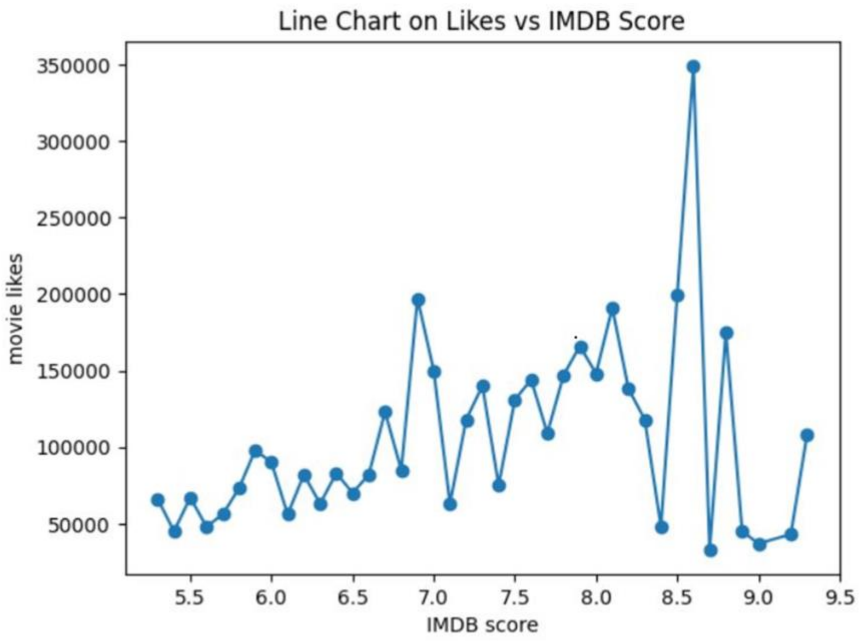
```
import pandas as pd
impor
t
matpl
otlib
.pypl
ot as
plt
impor
```

1. Line Graph: Likes v/s IMDB Rating

Code:

```
df1=df.g
roupby('
imdb_sco
re').max
()df1 =
df1.tail
(40)

#plot the graph
```

Output:

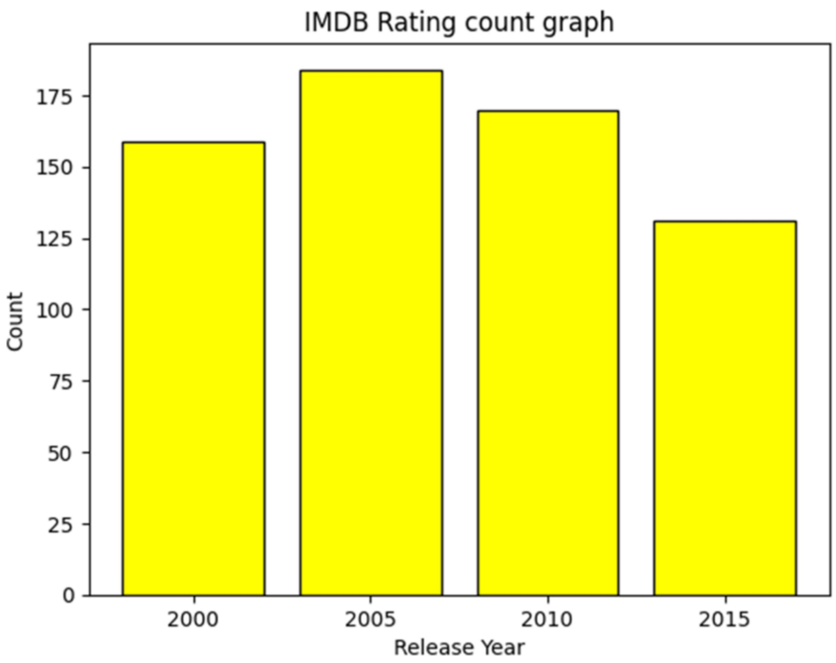


2. Vertical Bar Graph: IMDB Rating Bar Graph

Code:

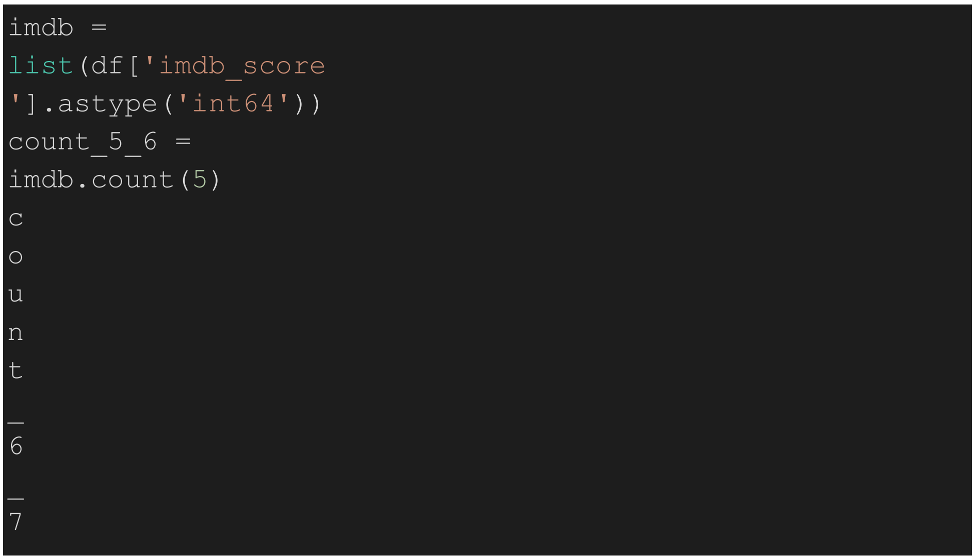
```
year                =
list(df['title_year
'].astype('int64'))
count_2000          =
year.count(2000)
cou
nt_
200
5 =
yea
r.c
oun
t(2
005
```

Output:



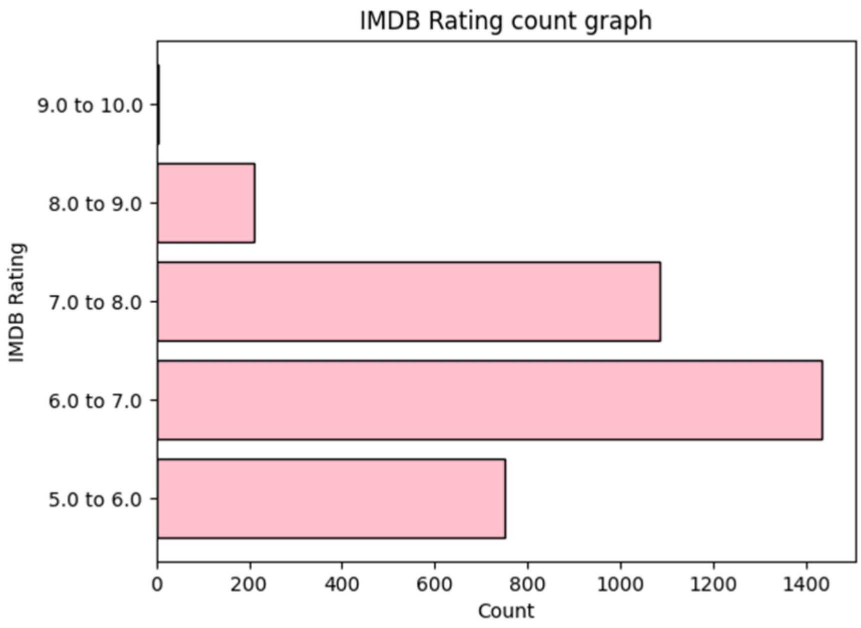
3. Horizontal Bar Graph: IMDB Rating Bar Graph

Code:



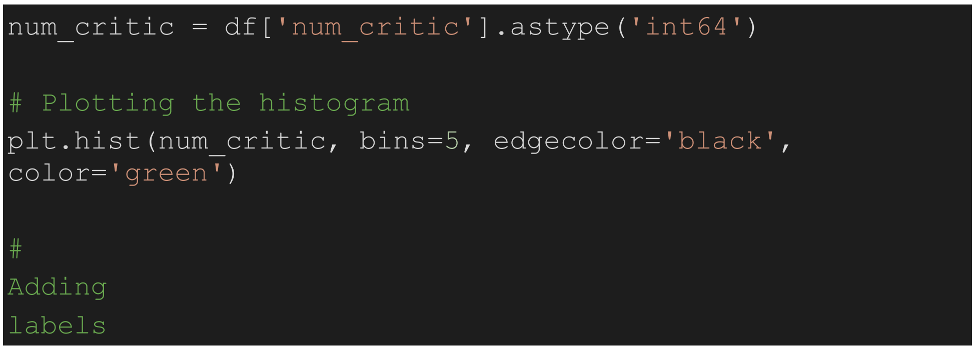
```
imdb =
list(df['imdb_score
'].astype('int64'))
count_5_6 =
imdb.count(5)
c
o
u
n
t
_
6
_
7
```

Output:



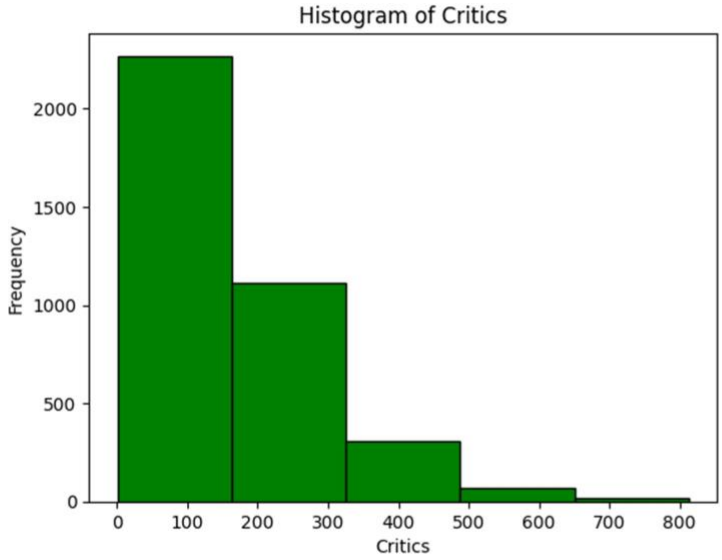
4. Histogram Graph: Critics histogram graph

Code:



```
num_critic = df['num_critic'].astype('int64')

# Plotting the histogram
plt.hist(num_critic, bins=5, edgecolor='black',
color='green')

#
Adding
labels
```

Output:

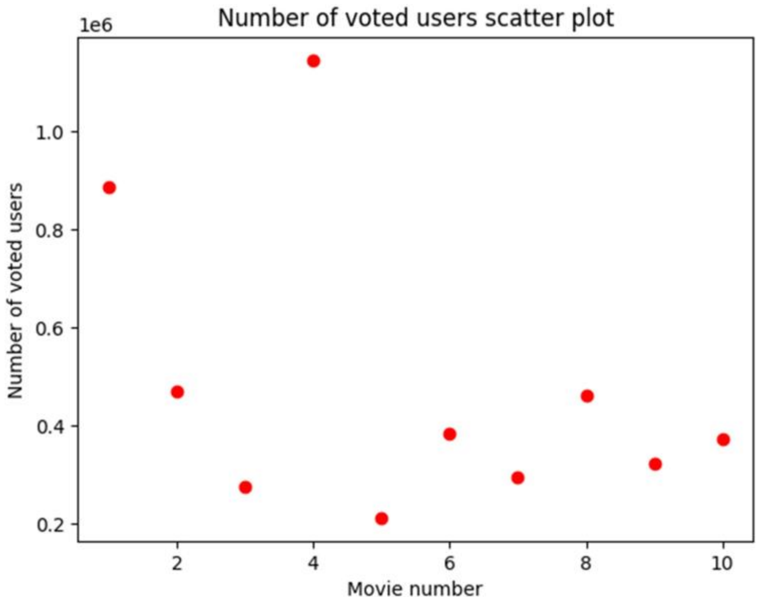


5. Scatter Graph: Number of voted user scatter graph

Code:

```
movie_number = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
num_voted_user =
(df['num_voted_users'][0:10]).astype('int64')

# Plotting the scatter plot
plt.scatter(movie_number, num_voted_user, color='red')

# Adding
labels
and title
```

Output:



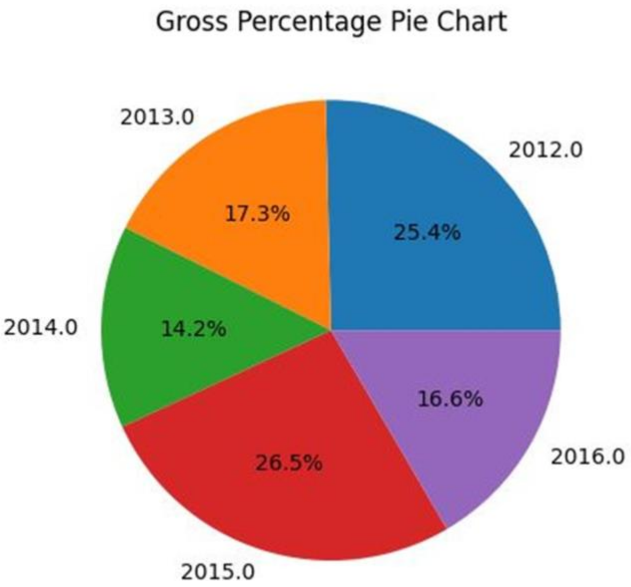
5. Pie Graph: Gross Percentage Pie Chart

Code:

```
df1 =
df.groupby
('title_ye
ar').max()
df1 =
df1.tail(5
)
```

```
#
```

Output:



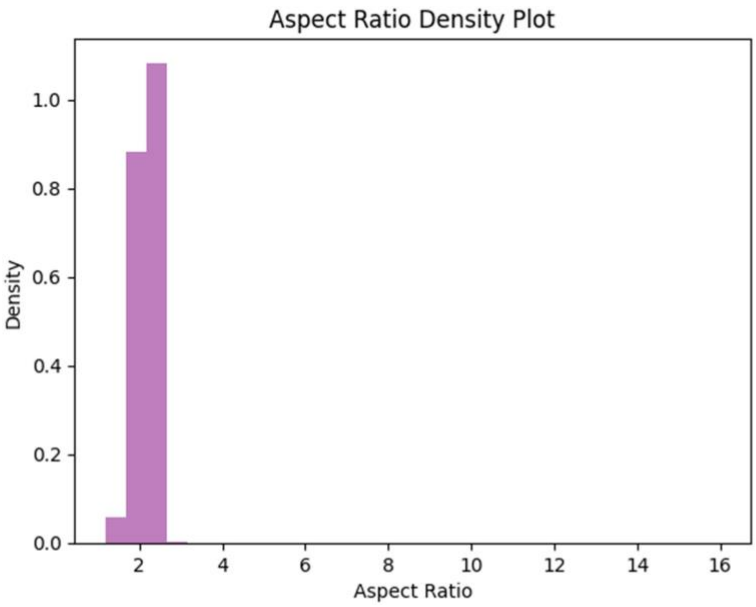
---

6. Density Graph: Aspect ratio density graph

Code:

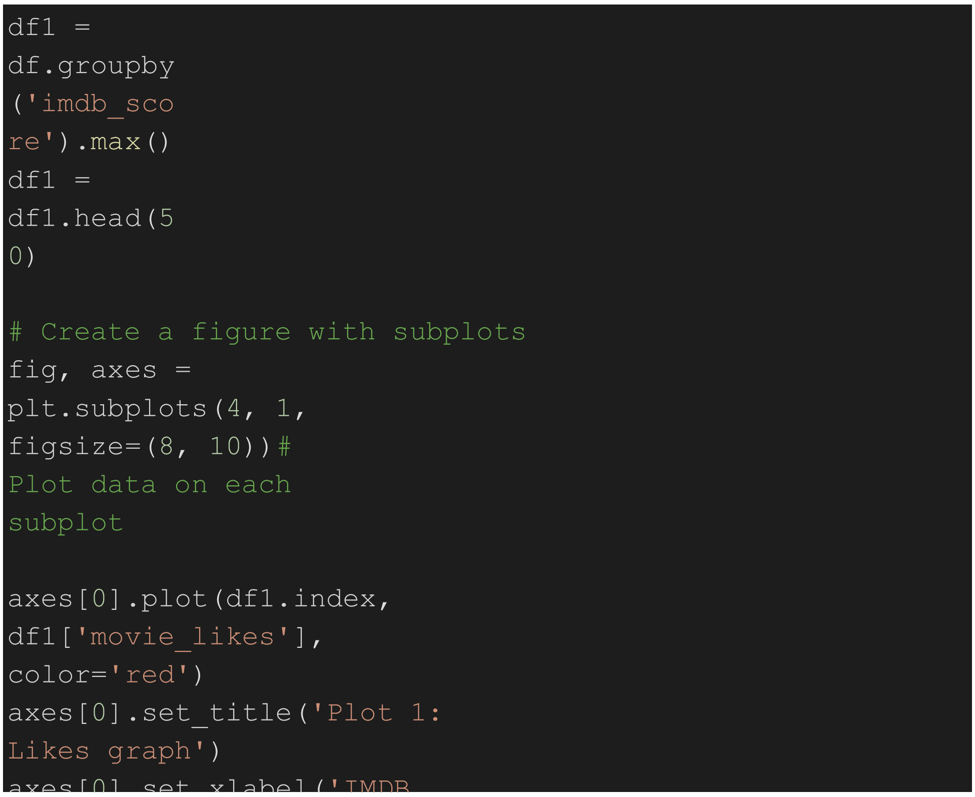
```
aspect_ratio = df['aspect_ratio'].astype(float)

# Create a density plot
plt.hist(aspect_ratio, density=True, bins=30, alpha=0.5,
color='purple')

# Add labels
and title
plt.xlabel('
```

Output:



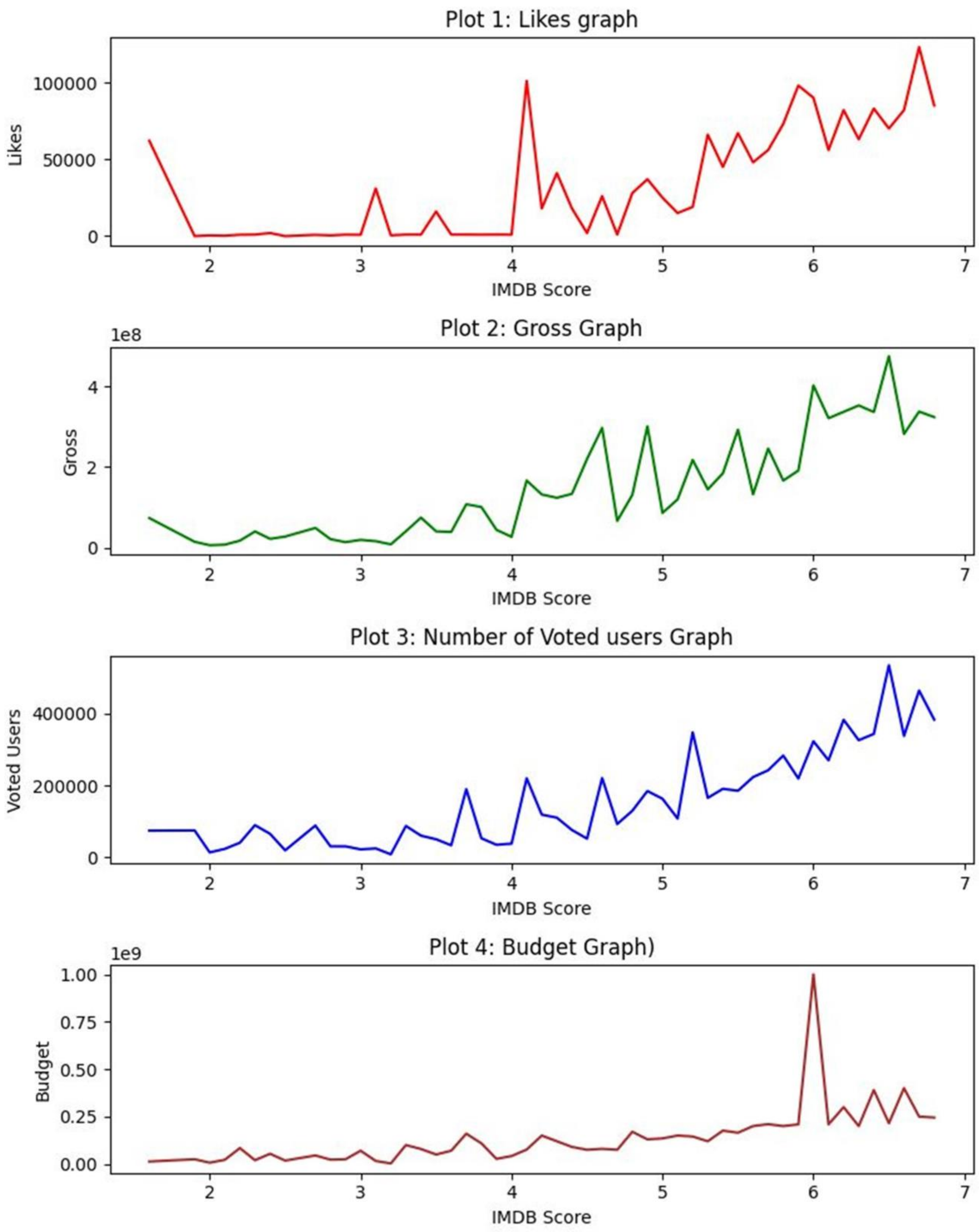
7. Group Plot: Likes graph, Gross graph, IMDB rating graph, Critic graph

Code:



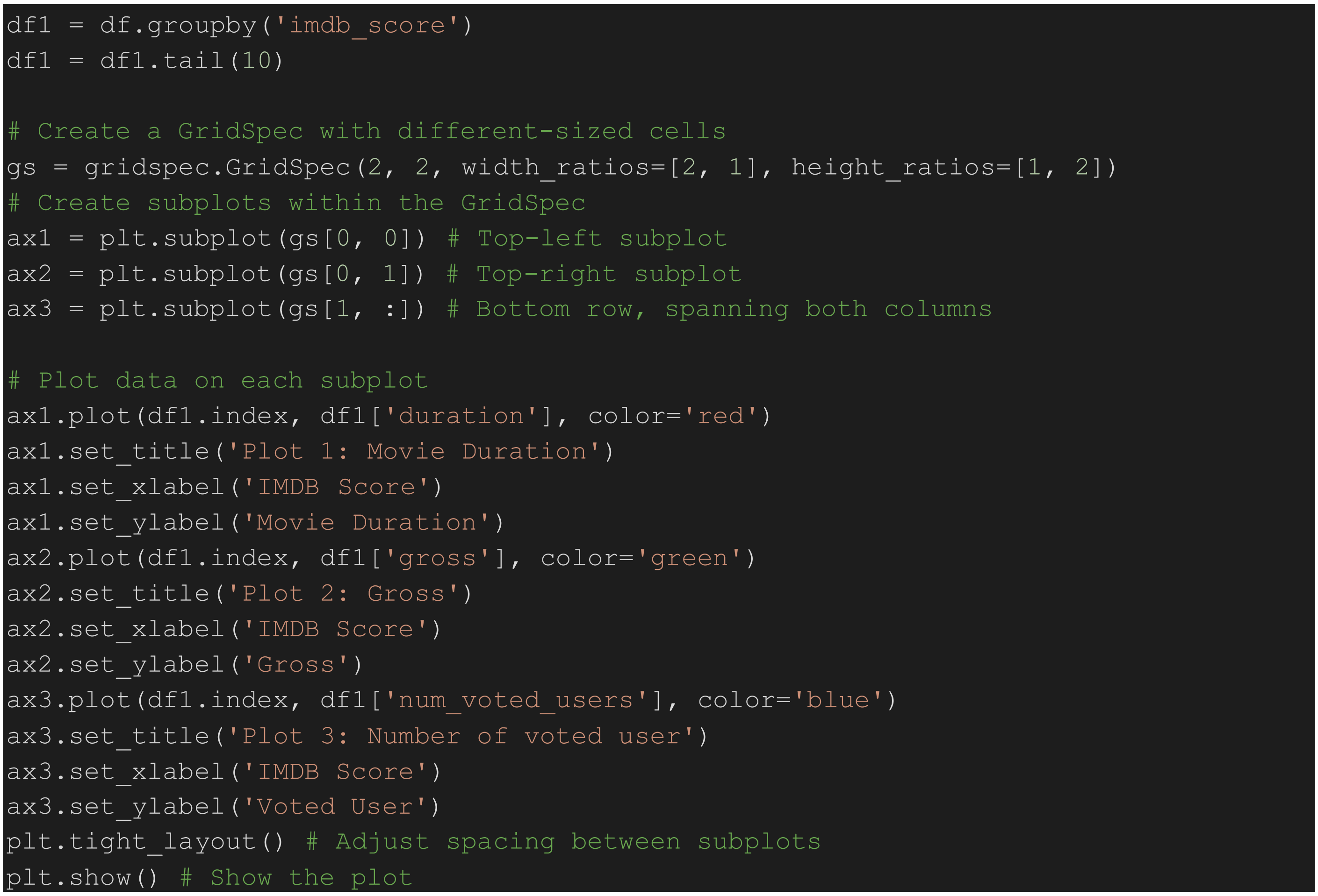
```
df1 =
df.groupby
('imdb_sco
re').max()
df1 =
df1.head(5
0)

# Create a figure with subplots
fig, axes =
plt.subplots(4, 1,
figsize=(8, 10))#
Plot data on each
subplot

axes[0].plot(df1.index,
df1['movie_likes'],
color='red')
axes[0].set_title('Plot 1:
Likes graph')
axes[0].set_xlabel('IMDB
plt.show()
```

Output:



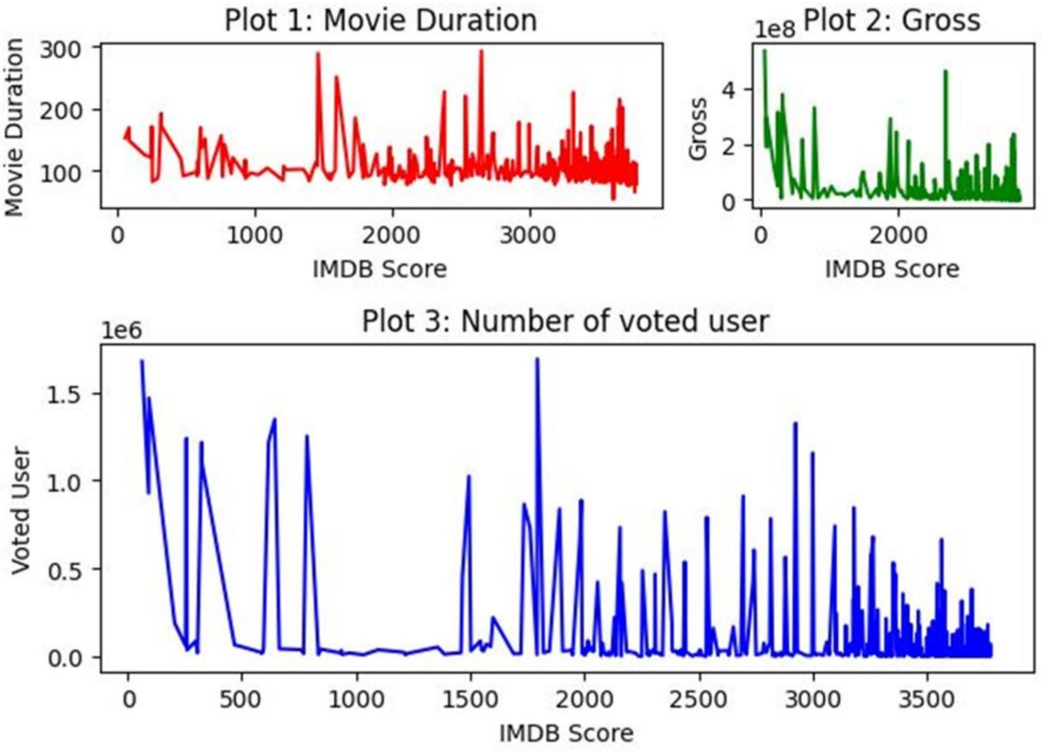
8. Subplot Graph: Movie duration, gross and number of voted user graphs

Code:



```
df1 = df.groupby('imdb_score')
df1 = df1.tail(10)

# Create a GridSpec with different-sized cells
gs = gridspec.GridSpec(2, 2, width_ratios=[2, 1], height_ratios=[1, 2])
# Create subplots within the GridSpec
ax1 = plt.subplot(gs[0, 0]) # Top-left subplot
ax2 = plt.subplot(gs[0, 1]) # Top-right subplot
ax3 = plt.subplot(gs[1, :]) # Bottom row, spanning both columns

# Plot data on each subplot
ax1.plot(df1.index, df1['duration'], color='red')
ax1.set_title('Plot 1: Movie Duration')
ax1.set_xlabel('IMDB Score')
ax1.set_ylabel('Movie Duration')
ax2.plot(df1.index, df1['gross'], color='green')
ax2.set_title('Plot 2: Gross')
ax2.set_xlabel('IMDB Score')
ax2.set_ylabel('Gross')
ax3.plot(df1.index, df1['num_voted_users'], color='blue')
ax3.set_title('Plot 3: Number of voted user')
ax3.set_xlabel('IMDB Score')
ax3.set_ylabel('Voted User')
plt.tight_layout() # Adjust spacing between subplots
plt.show() # Show the plot
```

Output:

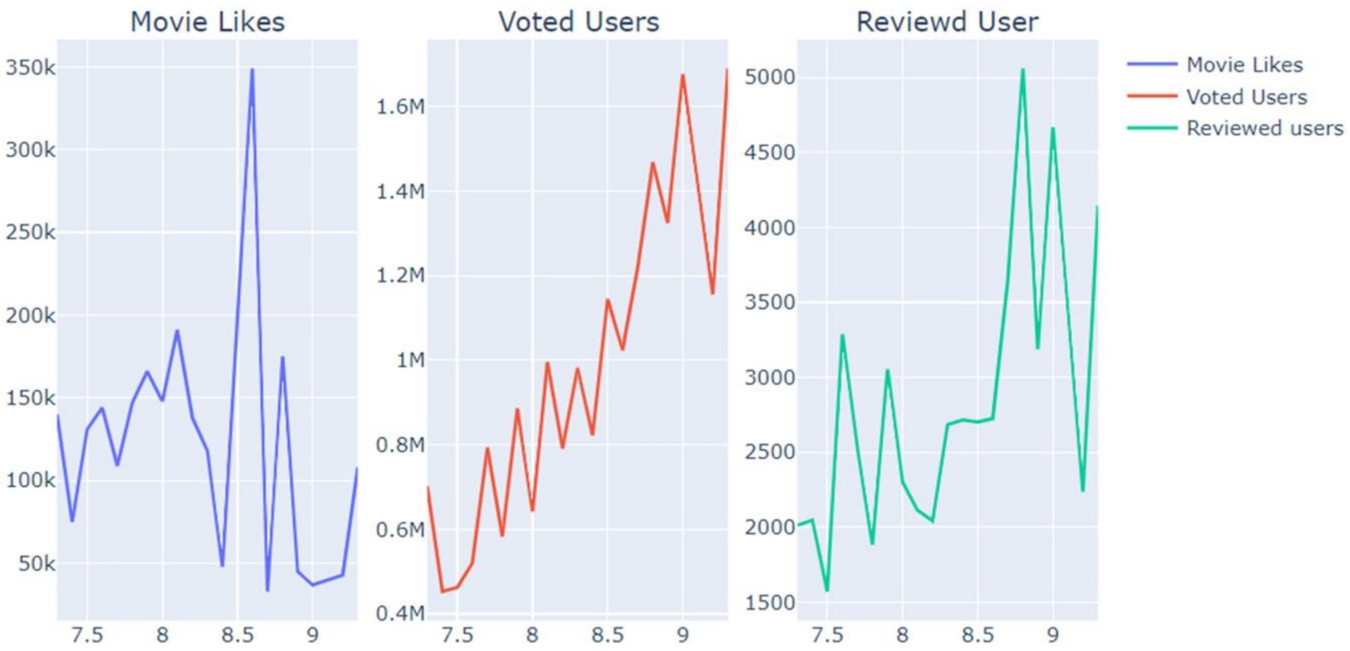


9. Panel Graph: Movie likes, voted users and reviewed users

Code:

```
# Create
data for
the
panels
df1=df.g
roupby('
imdb_sco
re').max
()df1 =
df1.tail
(20)

# Create subplots with panels
```

Output:



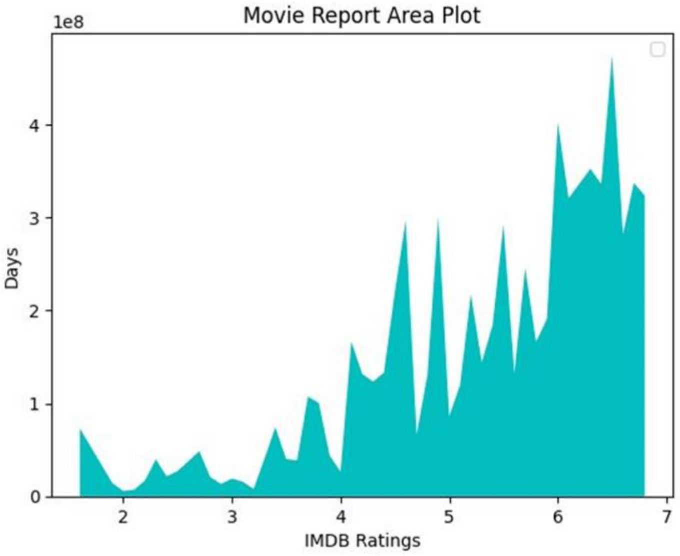
10. Area Graph: Movie likes, voted users and reviewed users

Code:

```
df1 =
df.groupby
```

```
plt.stackplot(df1.index, df1['gross'],
df1['movie_likes'], colors = ['c', 'g'])
plt.xlabel('IMDB Ratings')
plt.ylabe
l('Movie
Likes')
```

Output:


Movie Report Area Plot
1e8
Days
IMDB Ratings