In [1]:
```python
import numpy as np
import pandas as pd

ratings_data = pd.read_csv("ratings.csv")
ratings_data.head()
```

Out[1]:

	userId	movieId	rating	timestamp
0	1	1	4.0	964982703
1	1	3	4.0	964981247
2	1	6	4.0	964982224
3	1	47	5.0	964983815
4	1	50	5.0	964982931

In [2]:
```python
movie_names = pd.read_csv("movies.csv")
movie_names.head()
```

Out[2]:

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

In [4]:
```python
movie_data = pd.merge(ratings_data, movie_names, on='movieId')
movie_data.head()
```

Out[4]:

	userId	movieId	rating	timestamp	title	genres
0	1	1	4.0	964982703	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	5	1	4.0	847434962	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
2	7	1	4.5	1106635946	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
3	15	1	2.5	1510577970	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
4	17	1	4.5	1305696483	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy

In [5]:
```python
movie_data.groupby('title')['rating'].mean().head()
```

Out[5]:
```
title
'71 (2014)                                4.0
'Hellboy': The Seeds of Creation (2004)   4.0
'Round Midnight (1986)                    3.5
'Salem's Lot (2004)                       5.0
'Til There Was You (1997)                 4.0
Name: rating, dtype: float64
```

In [6]:
```python
movie_data.groupby('title')['rating'].mean().sort_values(ascending=False).head()
```

Out[6]:
```
title
Karlson Returns (1970)                          5.0
Winter in Prostokvashino (1984)                 5.0
My Love (2006)                                  5.0
Sorority House Massacre II (1990)               5.0
Winnie the Pooh and the Day of Concern (1972)   5.0
Name: rating, dtype: float64
```

In [7]:
```python
movie_data.groupby('title')['rating'].count().sort_values(ascending=False).head()
```

Out[7]:
```
title
Forrest Gump (1994)                329
Shawshank Redemption, The (1994)   317
Pulp Fiction (1994)                307
Silence of the Lambs, The (1991)   279
Matrix, The (1999)                 278
Name: rating, dtype: int64
```

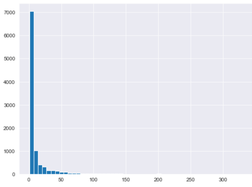
In [10]:
```python
ratings_mean_count['rating_counts'] = pd.DataFrame(movie_data.groupby('title')['rating'].count())
ratings_mean_count.head()
```

Out[10]:

	rating	rating_counts
title		
'71 (2014)	4.0	1
'Hellboy': The Seeds of Creation (2004)	4.0	1
'Round Midnight (1986)	3.5	2
'Salem's Lot (2004)	5.0	1
'Til There Was You (1997)	4.0	2

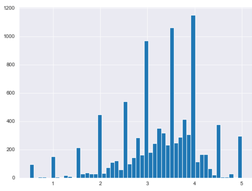
In [11]:
```python
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('dark')
%matplotlib inline

plt.figure(figsize=(8,6))
plt.rcParams['patch.force_edgecolor'] = True
ratings_mean_count['rating_counts'].hist(bins=50)
```

Out[11]:
```
<AxesSubplot:>
```



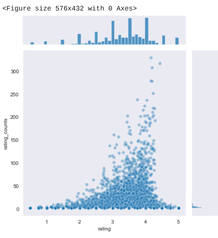
In [12]:
```python
plt.figure(figsize=(8,6))
plt.rcParams['patch.force_edgecolor'] = True
ratings_mean_count['rating'].hist(bins=50)
```

Out[12]:
```
<AxesSubplot:>
```



In [13]:
```python
plt.figure(figsize=(8,6))
plt.rcParams['patch.force_edgecolor'] = True
sns.jointplot(x='rating', y='rating_counts', data=ratings_mean_count, alpha=0.4)
```

Out[13]:
```
<seaborn.axisgrid.JointGrid at 0x1fc184737c0>
<Figure size 576x432 with 0 Axes>
```



In [14]:
```python
user_movie_rating = movie_data.pivot_table(index='userId', columns='title', values='rating')
user_movie_rating.head()
```

Out[14]:

title	'71 (2014)	'Hellboy': The Seeds of Creation (2004)	'Round Midnight (1986)	'Salem's Lot (2004)	'Til There Was You (1997)	'Tis the Season for Love (2015)	'burbs, The (1989)	'night Mother (1986)	(500) Days of Summer (2009)	*batteries not included (1987)	...	Zulu (2013)	[REC] (2007)	[REC]² (2009)	[REC]³ 3 Génesis (2012)	anohana: The Flower We Saw That Day - The Movie (2013)	eXistenZ (1999)	xXx (2002)	xXx: State of the Union (2005)	¡Three Amigos! (1986)	À nous la liberté (Freedom for Us) (1931)
userId																					
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	4.0	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

5 rows × 9719 columns

In [15]:
```python
forrest_gump_ratings = user_movie_rating['Forrest Gump (1994)']
forrest_gump_ratings.head()
```

Out[15]:
```
userId
1    4.0
2    NaN
3    NaN
4    NaN
5    NaN
Name: Forrest Gump (1994), dtype: float64
```

In [16]:
```python
movies_like_forest_gump = user_movie_rating.corrwith(forrest_gump_ratings)

corr_forrest_gump = pd.DataFrame(movies_like_forest_gump, columns=['Correlation'])
corr_forrest_gump.dropna(inplace=True)
corr_forrest_gump.head()
```

```
C:\Users\HP\anaconda3\lib\site-packages\numpy\lib\function_base.py:2551: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar)
C:\Users\HP\anaconda3\lib\site-packages\numpy\lib\function_base.py:2480: RuntimeWarning: divide by zero encountered in true_divide
  c *= np.true_divide(1, fact)
```

Out[16]:

	Correlation
title	
'burbs, The (1989)	0.197712
(500) Days of Summer (2009)	0.234095
***batteries not included (1987)**	0.892710
...And Justice for All (1979)	0.928571
10 Cent Pistol (2015)	-1.000000

In [17]:
```python
corr_forrest_gump.sort_values('Correlation', ascending=False).head(10)
```

Out[17]:

	Correlation
title	
Lost & Found (1999)	1.0
Century of the Self, The (2002)	1.0
The 5th Wave (2016)	1.0
Play Time (a.k.a. Playtime) (1967)	1.0
Memories (Memorias) (1995)	1.0
Playing God (1997)	1.0
Killers (2010)	1.0
Girl Walks Home Alone at Night, A (2014)	1.0
Tampopo (1985)	1.0
Cercle Rouge, Le (Red Circle, The) (1970)	1.0

In [18]:
```python
corr_forrest_gump = corr_forrest_gump.join(ratings_mean_count['rating_counts'])
corr_forrest_gump.head()
```

Out[18]:

	Correlation	rating_counts
title		
'burbs, The (1989)	0.197712	17
(500) Days of Summer (2009)	0.234095	42
***batteries not included (1987)**	0.892710	7
...And Justice for All (1979)	0.928571	3
10 Cent Pistol (2015)	-1.000000	2

In [19]:
```python
corr_forrest_gump[corr_forrest_gump ['rating_counts']>50].sort_values('Correlation', ascending=False).head()
```

Out[19]:

	Correlation	rating_counts
title		
Forrest Gump (1994)	1.000000	329
Mr. Holland's Opus (1995)	0.652144	80
Pocahontas (1995)	0.550118	68
Grumpier Old Men (1995)	0.534682	52
Caddyshack (1980)	0.520328	52

In [ ]: