□ IMDb Top 5000 TV Shows - Statistical Analysis

Mini Project – Statistics for Data Science

- **Course:** BCA Semester II
- **Objective:** Perform Univariate, Bivariate, and Multivariate Analysis on a real-world dataset using appropriate statistical tools and visualizations.

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

# Set style
sns.set(style="whitegrid")
plt.rcParams['figure.figsize'] = (10, 6)

# Load dataset
df = pd.read_csv(r"C:\Users\WINDOWS 11 PRO\Downloads\
imdb_top_5000_tv_shows.csv")
df.head()
```

```
      tconst      primaryTitle  startYear  endYear  rank  
averageRating  \
0  tt0903747      Breaking Bad       2008   2013.0     1  
9.5   
1  tt0185906  Band of Brothers       2001   2001.0     2  
9.4   
2  tt7366338         Chernobyl       2019   2019.0     3  
9.3   
3  tt0795176      Planet Earth       2006   2006.0     4  
9.4   
4  tt5491994   Planet Earth II       2016   2016.0     5  
9.4   

   numVotes                                          directors  \
0   2314919  Michelle MacLaren, Adam Bernstein, Vince Gilli...   
1    559518  David Frankel, Mikael Salomon, Tom Hanks, Davi...   
2    943168                                        Johan Renck   
3    226979                 Alastair Fothergill, Mark Linfield   
4    166038  Justin Anderson, Ed Charles, Fredi Devas, Chad...   

                                             writers  \
0  Vince Gilligan, Peter Gould, George Mastras, S...   
1  Stephen Ambrose, Erik Bork, E. Max Frye, Tom H...   
2                                        Craig Mazin   
3  David Attenborough, Vanessa Berlowitz, Alastai...   
```

```
4                                      Elizabeth White

                      genres  \
0    Crime, Drama, Thriller
1    Action, Drama, History
2  Drama, History, Thriller
3       Documentary, Family
4               Documentary

                                            IMDbLink  \
0  <a href="https://www.imdb.com/title/tt0903747"...
1  <a href="https://www.imdb.com/title/tt0185906"...
2  <a href="https://www.imdb.com/title/tt7366338"...
3  <a href="https://www.imdb.com/title/tt0795176"...
4  <a href="https://www.imdb.com/title/tt5491994"...

                                     Title_IMDb_Link
0  <a href="https://www.imdb.com/title/tt0903747"...
1  <a href="https://www.imdb.com/title/tt0185906"...
2  <a href="https://www.imdb.com/title/tt7366338"...
3  <a href="https://www.imdb.com/title/tt0795176"...
4  <a href="https://www.imdb.com/title/tt5491994"...

df

             primaryTitle  startYear  endYear  rank  averageRating
numVotes  \
0            Breaking Bad       2008     2013     1            9.5
2314919
1        Band of Brothers       2001     2001     2            9.4
559518
2               Chernobyl       2019     2019     3            9.3
943168
3            Planet Earth       2006     2006     4            9.4
226979
4         Planet Earth II       2016     2016     5            9.4
166038
...                   ...        ...      ...   ...            ...
...
4995  Rick Steves' Europe       2000     2024  4996            8.6
783
4996         Dynasties II       2022     2022  4997            8.6
783
4997       Muchachada nui       2007     2010  4998            7.9
783
4998               Empati       2022     2022  4999            8.9
781
4999    City Confidential       1998     2023  5000            8.6
781
```

```
                                              directors  \
0     Michelle MacLaren, Adam Bernstein, Vince Gilli...
1     David Frankel, Mikael Salomon, Tom Hanks, Davi...
2                                           Johan Renck
3                    Alastair Fothergill, Mark Linfield
4     Justin Anderson, Ed Charles, Fredi Devas, Chad...
...                                                 ...
4995                                     Simon Griffith
4996  Lydia Baines, Simon Blakeney, Felicity Lanches...
4997  Joaquín Reyes, Nacho Vigalondo, Helio Mira, Ko...
4998                                        Özcan Mavis
4999                       Scott Colthorp, Eric Futrell

                                                writers  \
0     Vince Gilligan, Peter Gould, George Mastras, S...
1     Stephen Ambrose, Erik Bork, E. Max Frye, Tom H...
2                                           Craig Mazin
3     David Attenborough, Vanessa Berlowitz, Alastai...
4                                       Elizabeth White
...                                                 ...
4995  Steve Cammarano, Cameron Hewitt, Gene Openshaw...
4996                                                  -
4997  Raúl Cimas, Julián López, Joaquín Reyes, Ernes...
4998                                  Egemen Alper Koca
4999  Matt Edens, Zak Weisfeld, Geoffrey Proud, Todd...

                        genres rating_bin  decade   main_genre
0       Crime, Drama, Thriller        Top    2000        Crime
1       Action, Drama, History        Top    2000       Action
2     Drama, History, Thriller        Top    2010        Drama
3          Documentary, Family        Top    2000  Documentary
4                  Documentary        Top    2010  Documentary
...                        ...        ...     ...          ...
4995   Documentary, Reality-TV        Top    2000  Documentary
4996               Documentary        Top    2020  Documentary
4997         Animation, Comedy       High    2000    Animation
4998                Reality-TV        Top    2020   Reality-TV
4999        Crime, Documentary        Top    1990        Crime

[5000 rows x 12 columns]
```

```
df.describe()
```

```
         startYear      endYear         rank  averageRating
numVotes  \
count  5000.000000  5000.000000  5000.000000    5000.000000
5.000000e+03
mean   2008.928200  2011.375800  2500.500000       8.003380
2.718001e+04
std      14.372653    13.494643  1443.520003       0.438594
```

```
9.033600e+04
min    1948.000000  1953.000000     1.000000       7.300000
7.130000e+02
25%    2003.000000  2006.750000  1250.750000       7.700000
2.102750e+03
50%    2014.000000  2016.000000  2500.500000       8.000000
5.203500e+03
75%    2019.000000  2021.000000  3750.250000       8.300000
1.704825e+04
max    2025.000000  2026.000000  5000.000000       9.600000
2.422280e+06

            decade
count  5000.000000
mean   2004.576000
std      14.896463
min    1940.000000
25%    2000.000000
50%    2010.000000
75%    2010.000000
max    2020.000000
```

```
# Data Cleaning
df.drop(columns=['tconst', 'IMDbLink', 'Title_IMDb_Link'],
inplace=True)
df['endYear'].fillna(df['startYear'], inplace=True)
df['endYear'] = df['endYear'].astype(int)
df['rating_bin'] = pd.cut(df['averageRating'], bins=[0, 6, 7.5, 8.5,
10], labels=["Low", "Medium", "High", "Top"])
df['decade'] = (df['startYear'] // 10) * 10
df['main_genre'] = df['genres'].str.split(',').str[0]
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 12 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   primaryTitle   5000 non-null   object
 1   startYear      5000 non-null   int64
 2   endYear        5000 non-null   int32
 3   rank           5000 non-null   int64
 4   averageRating  5000 non-null   float64
 5   numVotes       5000 non-null   int64
 6   directors      5000 non-null   object
 7   writers        5000 non-null   object
 8   genres         5000 non-null   object
 9   rating_bin     5000 non-null   category
 10  decade         5000 non-null   int64
 11  main_genre     5000 non-null   object
```

```
dtypes: category(1), float64(1), int32(1), int64(4), object(5)
memory usage: 415.4+ KB

C:\Users\WINDOWS 11 PRO\AppData\Local\Temp\
ipykernel_28532\4230260464.py:3: FutureWarning: A value is trying to
be set on a copy of a DataFrame or Series through chained assignment
using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.


  df['endYear'].fillna(df['startYear'], inplace=True)
```

```python
df.dropna(inplace=True)
print("After dropna (inplace=True):")
df.head()
```

```
After dropna (inplace=True):

       primaryTitle  startYear  endYear  rank  averageRating  numVotes
\
0      Breaking Bad       2008     2013     1            9.5   2314919

1  Band of Brothers       2001     2001     2            9.4    559518

2         Chernobyl       2019     2019     3            9.3    943168

3      Planet Earth       2006     2006     4            9.4    226979

4    Planet Earth II       2016     2016     5            9.4    166038


                                           directors  \
0  Michelle MacLaren, Adam Bernstein, Vince Gilli...
1  David Frankel, Mikael Salomon, Tom Hanks, Davi...
2                                        Johan Renck
3                 Alastair Fothergill, Mark Linfield
4  Justin Anderson, Ed Charles, Fredi Devas, Chad...

                                             writers  \
0  Vince Gilligan, Peter Gould, George Mastras, S...
1  Stephen Ambrose, Erik Bork, E. Max Frye, Tom H...
2                                        Craig Mazin
3  David Attenborough, Vanessa Berlowitz, Alastai...
4                                    Elizabeth White
```

```
                     genres rating_bin  decade   main_genre      PCA1
\
0    Crime, Drama, Thriller        Top    2000        Crime  3.468401

1    Action, Drama, History        Top    2000       Action -0.082651

2  Drama, History, Thriller        Top    2010        Drama  2.291523

3       Documentary, Family        Top    2000  Documentary -0.082008

4               Documentary        Top    2010  Documentary  0.819653


        PCA2
0 -15.662012
1  -5.986465
2  -7.737932
3  -4.093788
4  -3.632818
```

```
df.dropna(inplace=False)
print("After dropna (inplace=False):")
df.head()
```

```
After dropna (inplace=False):
```

```
       primaryTitle  startYear  endYear  rank  averageRating  numVotes
\
0      Breaking Bad       2008     2013     1            9.5   2314919

1  Band of Brothers       2001     2001     2            9.4    559518

2         Chernobyl       2019     2019     3            9.3    943168

3      Planet Earth       2006     2006     4            9.4    226979

4   Planet Earth II       2016     2016     5            9.4    166038


                                           directors  \
0  Michelle MacLaren, Adam Bernstein, Vince Gilli...
1  David Frankel, Mikael Salomon, Tom Hanks, Davi...
2                                        Johan Renck
3                 Alastair Fothergill, Mark Linfield
4  Justin Anderson, Ed Charles, Fredi Devas, Chad...

                                             writers  \
0  Vince Gilligan, Peter Gould, George Mastras, S...
1  Stephen Ambrose, Erik Bork, E. Max Frye, Tom H...
2                                        Craig Mazin
```

```
3  David Attenborough, Vanessa Berlowitz, Alastai...
4                                  Elizabeth White

                     genres rating_bin  decade   main_genre      PCA1
\
0    Crime, Drama, Thriller        Top    2000        Crime  3.468401

1    Action, Drama, History        Top    2000       Action -0.082651

2  Drama, History, Thriller        Top    2010        Drama  2.291523

3       Documentary, Family        Top    2000  Documentary -0.082008

4               Documentary        Top    2010  Documentary  0.819653


        PCA2
0 -15.662012
1  -5.986465
2  -7.737932
3  -4.093788
4  -3.632818
```

```
df.tail()
```

```
             primaryTitle  startYear  endYear  rank  averageRating
numVotes  \
4995  Rick Steves' Europe       2000     2024  4996            8.6
783
4996         Dynasties II       2022     2022  4997            8.6
783
4997       Muchachada nui       2007     2010  4998            7.9
783
4998               Empati       2022     2022  4999            8.9
781
4999    City Confidential       1998     2023  5000            8.6
781

                                              directors  \
4995                                     Simon Griffith
4996  Lydia Baines, Simon Blakeney, Felicity Lanches...
4997  Joaquín Reyes, Nacho Vigalondo, Helio Mira, Ko...
4998                                        Özcan Mavis
4999                       Scott Colthorp, Eric Futrell

                                                writers  \
4995  Steve Cammarano, Cameron Hewitt, Gene Openshaw...
4996                                                  -
4997  Raúl Cimas, Julián López, Joaquín Reyes, Ernes...
4998                                  Egemen Alper Koca
4999  Matt Edens, Zak Weisfeld, Geoffrey Proud, Todd...
```

```
                       genres rating_bin  decade   main_genre
4995  Documentary, Reality-TV        Top    2000  Documentary
4996              Documentary        Top    2020  Documentary
4997        Animation, Comedy       High    2000    Animation
4998               Reality-TV        Top    2020   Reality-TV
4999       Crime, Documentary        Top    1990        Crime
```

```
df.isnull().sum()
```

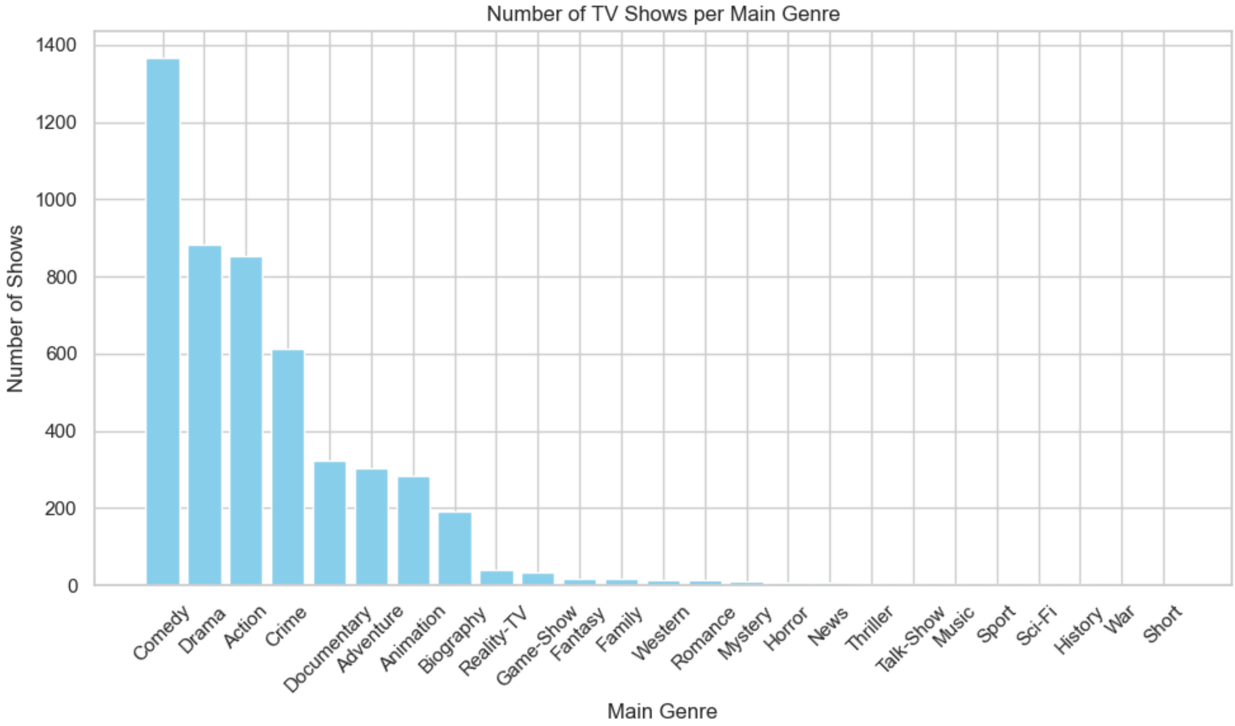
```
primaryTitle     0
startYear        0
endYear          0
rank             0
averageRating    0
numVotes         0
directors        0
writers          0
genres           0
rating_bin       0
decade           0
main_genre       0
dtype: int64
```

▯ Univariate Analysis

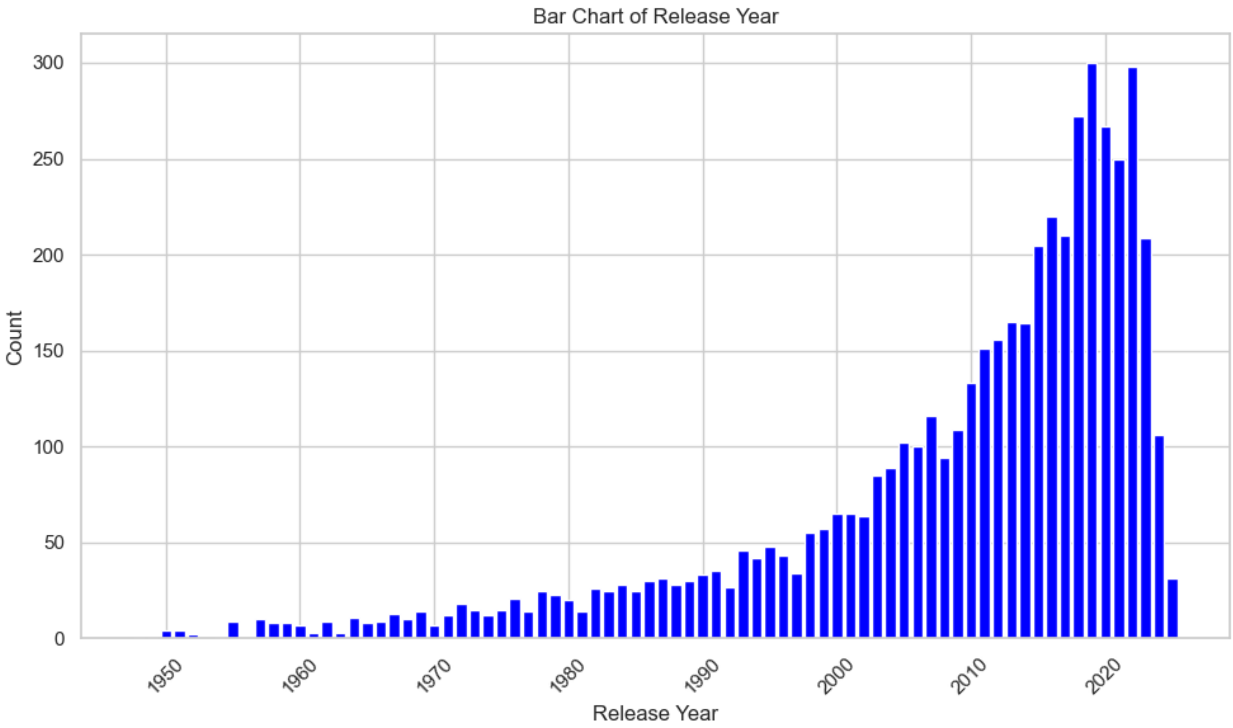
```
# Summary Statistics
df[['startYear', 'endYear', 'rank', 'averageRating',
'numVotes']].describe()
```

```
         startYear      endYear         rank  averageRating
numVotes
count  5000.000000  5000.000000  5000.000000    5000.000000
5.000000e+03
mean   2008.928200  2011.375800  2500.500000       8.003380
2.718001e+04
std      14.372653    13.494643  1443.520003       0.438594
9.033600e+04
min    1948.000000  1953.000000     1.000000       7.300000
7.130000e+02
25%    2003.000000  2006.750000  1250.750000       7.700000
2.102750e+03
50%    2014.000000  2016.000000  2500.500000       8.000000
5.203500e+03
75%    2019.000000  2021.000000  3750.250000       8.300000
1.704825e+04
max    2025.000000  2026.000000  5000.000000       9.600000
2.422280e+06
```

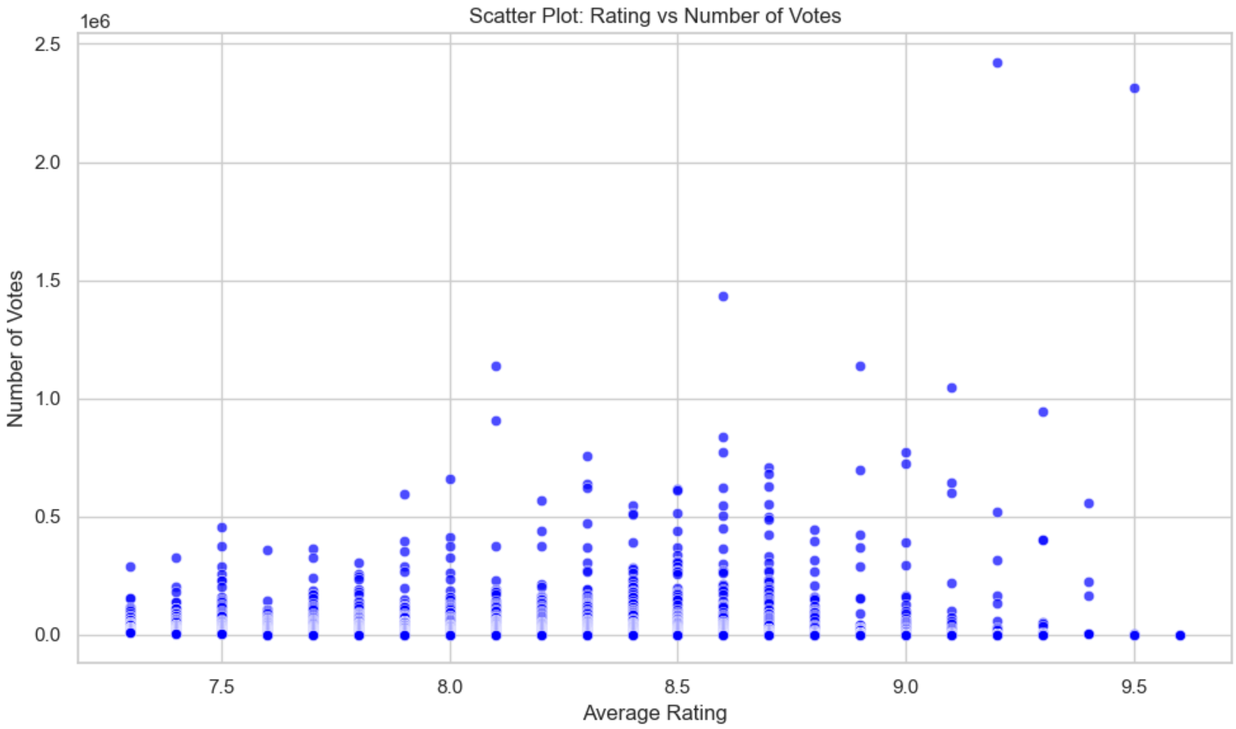
```python
import matplotlib.pyplot as plt

# Prepare data
df['main_genre'] = df['genres'].str.split(',').str[0]  # Extract main genre
genre_counts = df['main_genre'].value_counts()

# Plot
plt.bar(genre_counts.index, genre_counts.values, color='skyblue')
plt.xlabel('Main Genre')
plt.ylabel('Number of Shows')
plt.title('Number of TV Shows per Main Genre')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



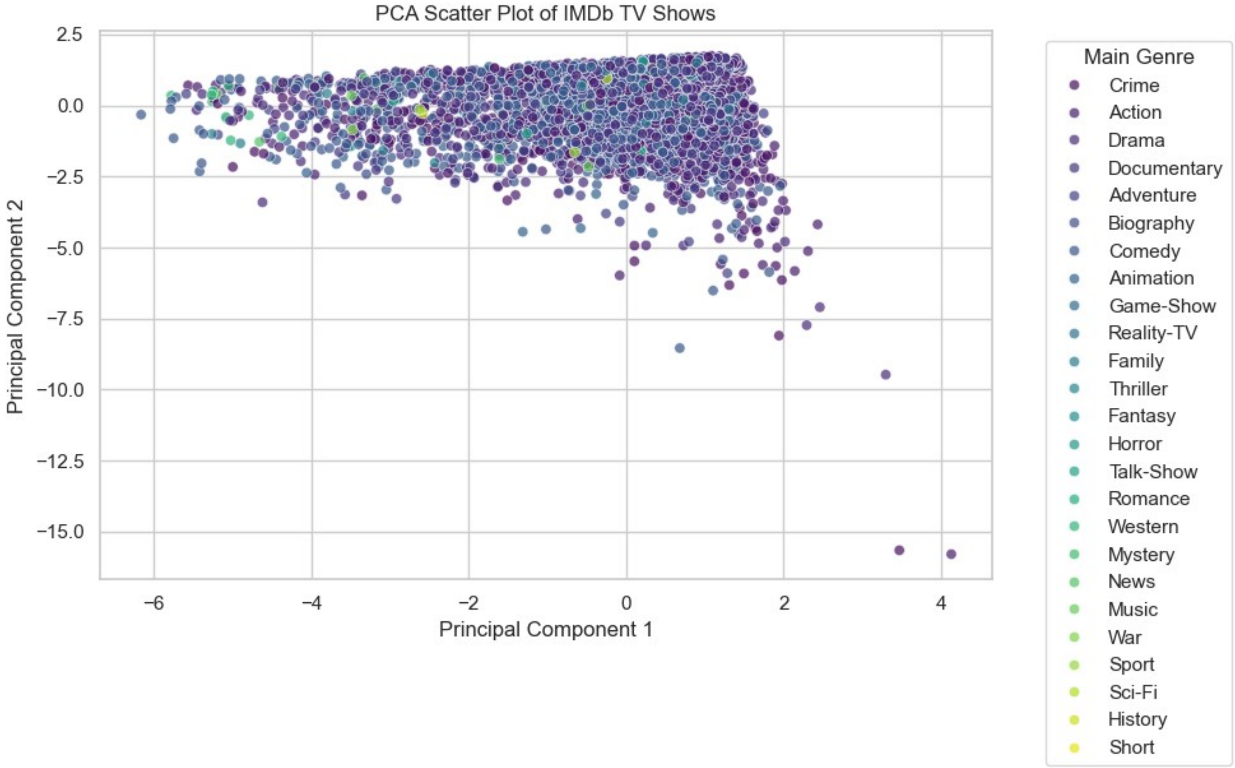
```python
import matplotlib.pyplot as plt

# Count number of shows by release (start) year
x = df['startYear'].value_counts().sort_index()

# Plot the bar chart
plt.bar(x.index, x.values, color='blue')
plt.xlabel('Release Year')
plt.ylabel('Count')
plt.title('Bar Chart of Release Year')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



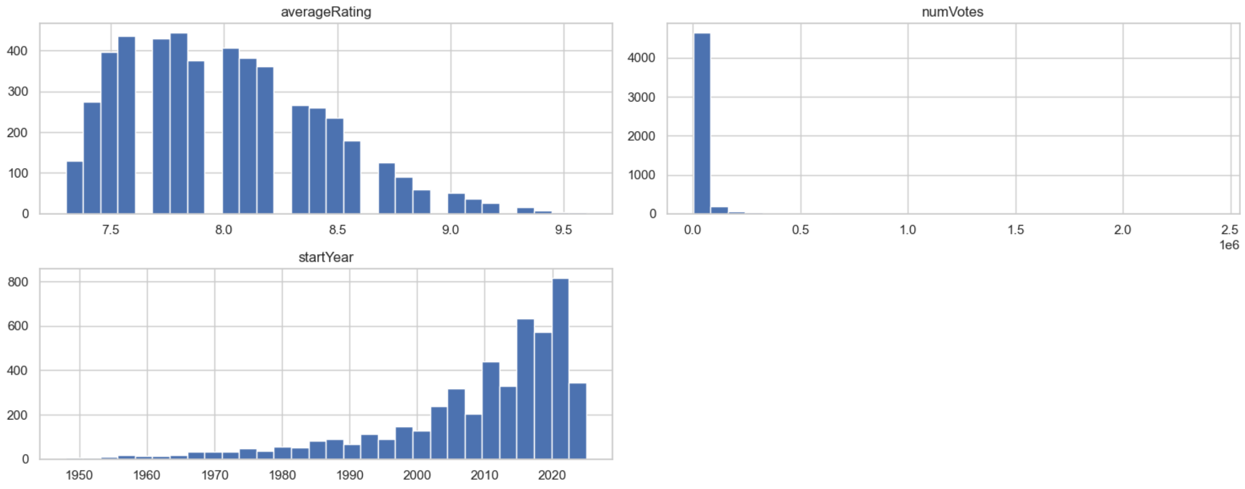
```
import matplotlib.pyplot as plt
import seaborn as sns

# Create the scatter plot
plt.figure(figsize=(10, 6))
sns.scatterplot(
    x=df['averageRating'],
    y=df['numVotes'],
    color='blue',
    alpha=0.7
)

plt.xlabel('Average Rating')
plt.ylabel('Number of Votes')
plt.title('Scatter Plot: Rating vs Number of Votes')
plt.tight_layout()
plt.show()
```



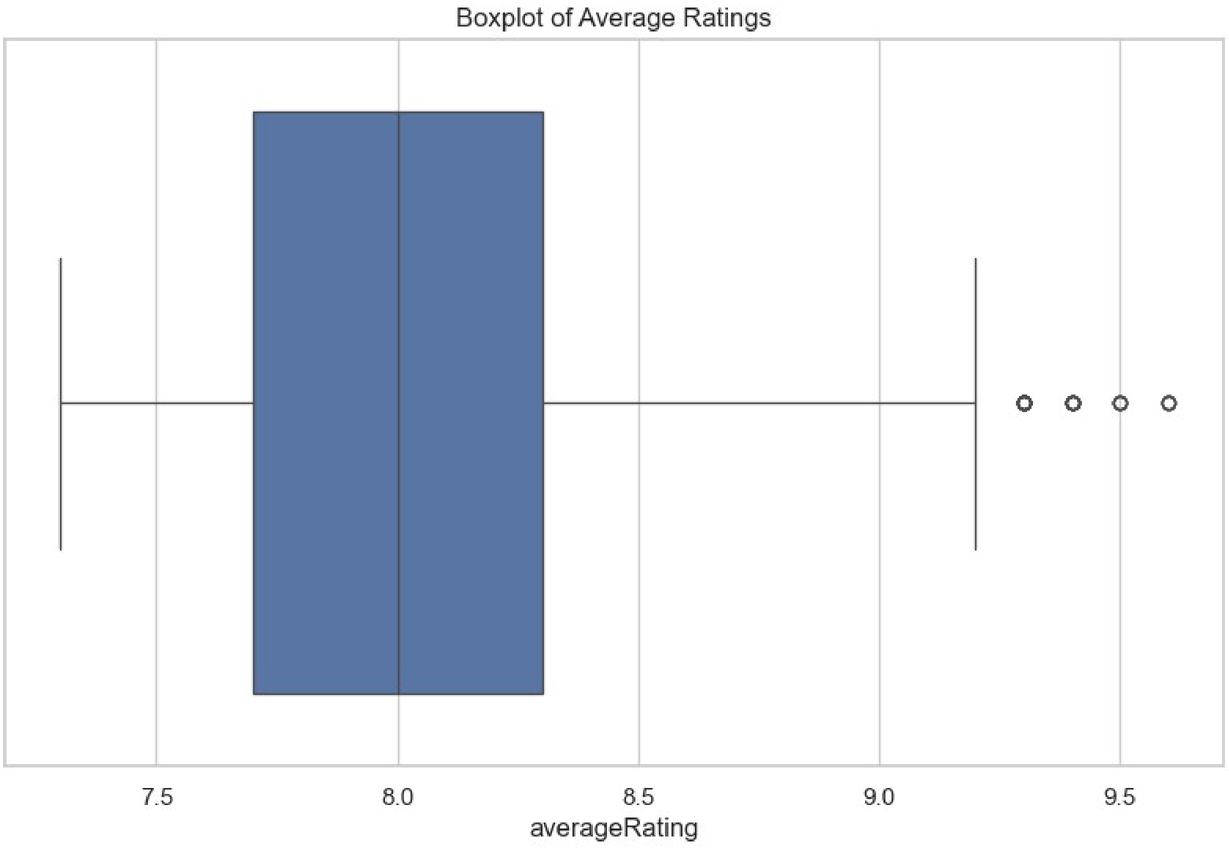
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

# Select numerical columns for PCA
numerical_columns = ['averageRating', 'numVotes', 'rank', 'startYear',
'endYear']
data_numeric = df[numerical_columns].dropna()

# Standardize the data
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data_numeric)

# Apply PCA
pca = PCA(n_components=2)
pca_result = pca.fit_transform(data_scaled)

# Add PCA results to the original DataFrame
df['PCA1'] = pca_result[:, 0]
df['PCA2'] = pca_result[:, 1]

# Scatter plot of PCA results colored by 'main_genre'
plt.figure(figsize=(10, 6))
sns.scatterplot(
    x=df['PCA1'],
    y=df['PCA2'],
    hue=df['main_genre'],
    palette='viridis',
    alpha=0.7
)

plt.title('PCA Scatter Plot of IMDb TV Shows')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.legend(title='Main Genre', bbox_to_anchor=(1.05, 1), loc='upper
left')
plt.tight_layout()
plt.show()
```



```
# Histograms
df[['averageRating', 'numVotes', 'startYear']].hist(bins=30,
figsize=(15, 6))
plt.tight_layout()
plt.show()
```

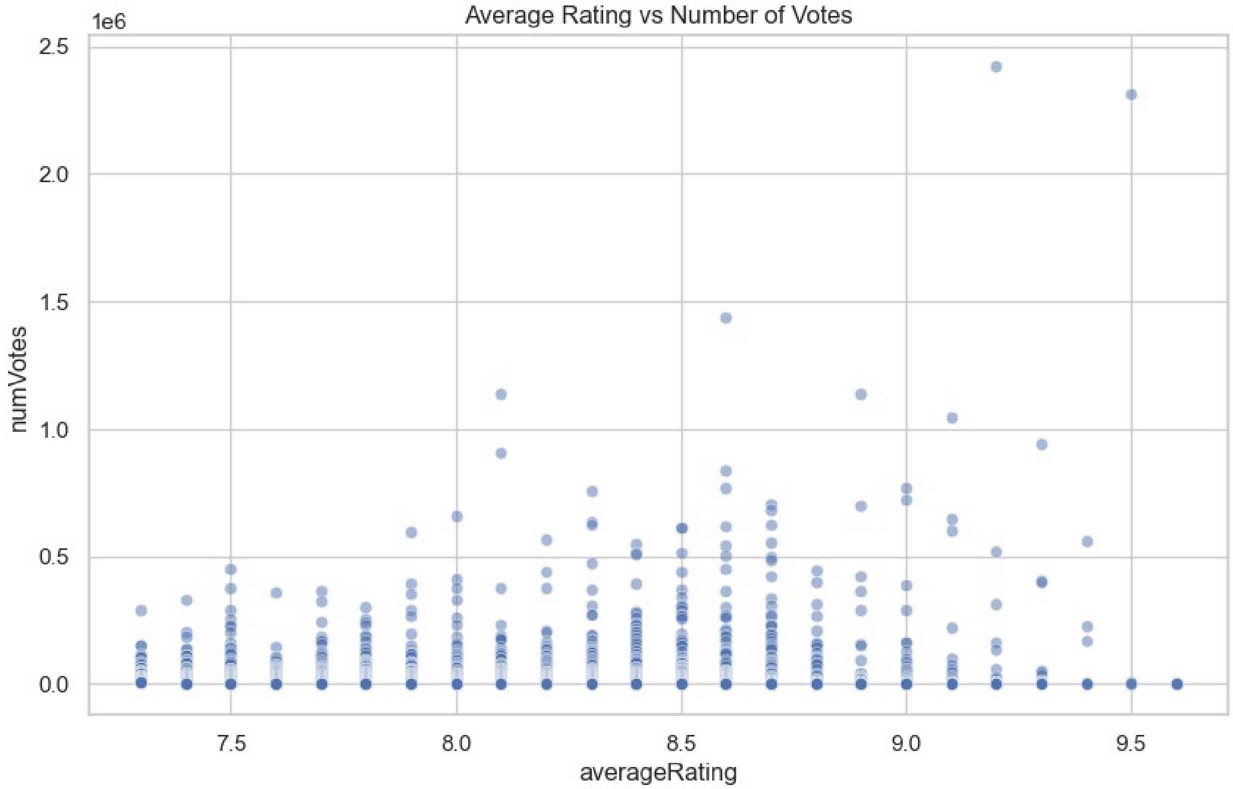


```python
# Boxplot for Average Ratings
sns.boxplot(x=df['averageRating'])
plt.title('Boxplot of Average Ratings')
plt.show()
```



Bivariate Analysis

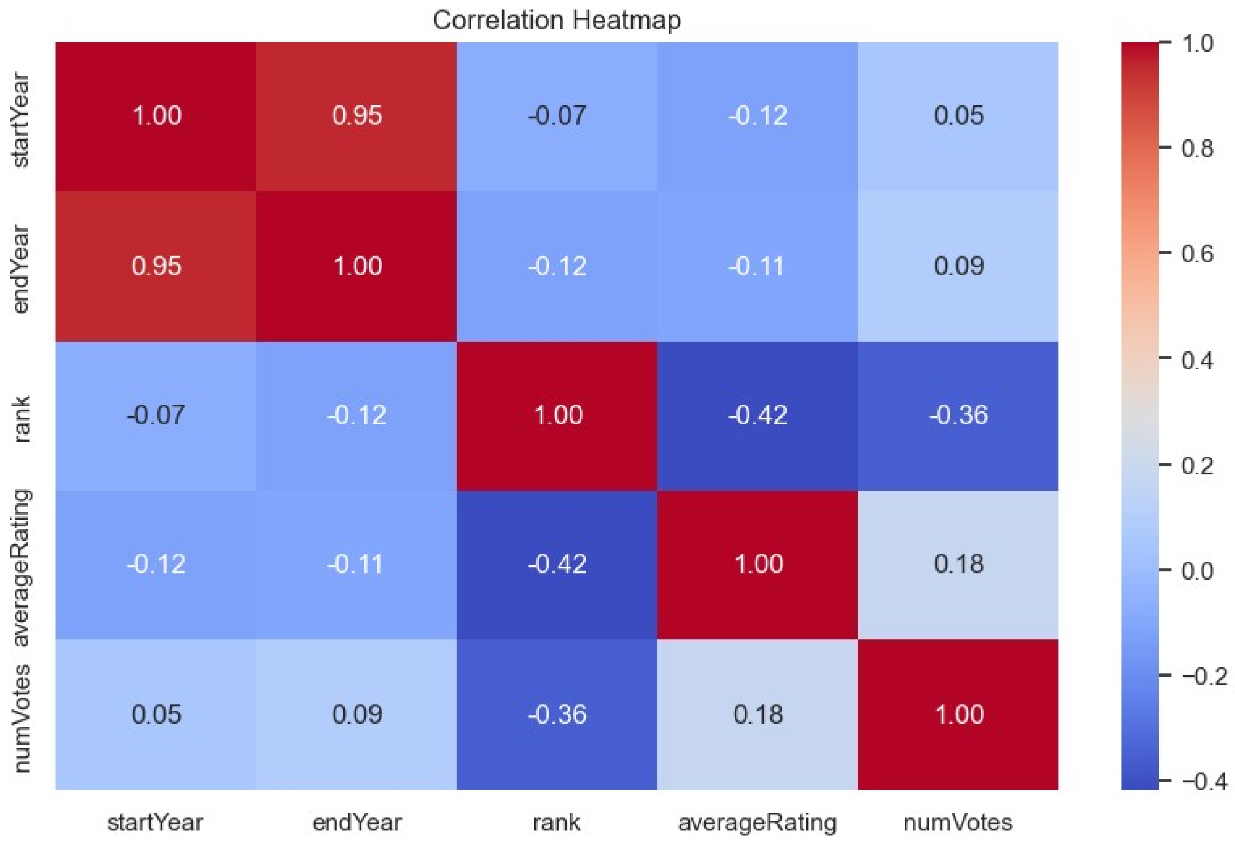
```python
# Scatter plot: Rating vs Votes
sns.scatterplot(data=df, x='averageRating', y='numVotes', alpha=0.5)
plt.title('Average Rating vs Number of Votes')
plt.show()
```



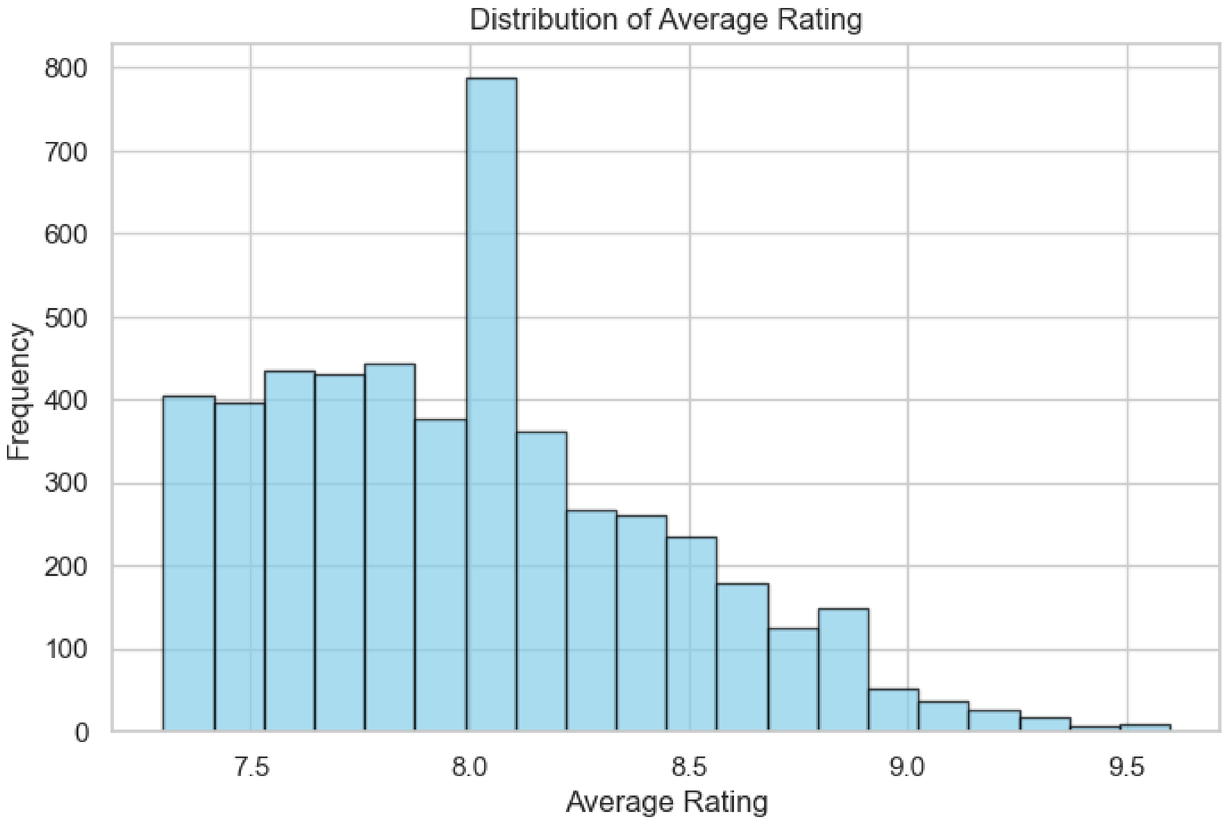
```python
# Handle missing values in the IMDb dataset

# Fill numerical columns with their mean
numerical_columns = df.select_dtypes(include=['int64',
'float64']).columns
for col in numerical_columns:
    if df[col].isnull().sum() > 0:
        df[col] = df[col].fillna(df[col].mean())

# Fill categorical columns with their mode
categorical_columns = df.select_dtypes(include=['object']).columns
for col in categorical_columns:
    if df[col].isnull().sum() > 0:
        df[col] = df[col].fillna(df[col].mode()[0])

# Verify if missing values are handled
print(df.isnull().sum())
```

```
primaryTitle     0
startYear        0
endYear          0
rank             0
averageRating    0
numVotes         0
directors        0
writers          0
genres           0
rating_bin       0
decade           0
main_genre       0
PCA1             0
PCA2             0
dtype: int64
```

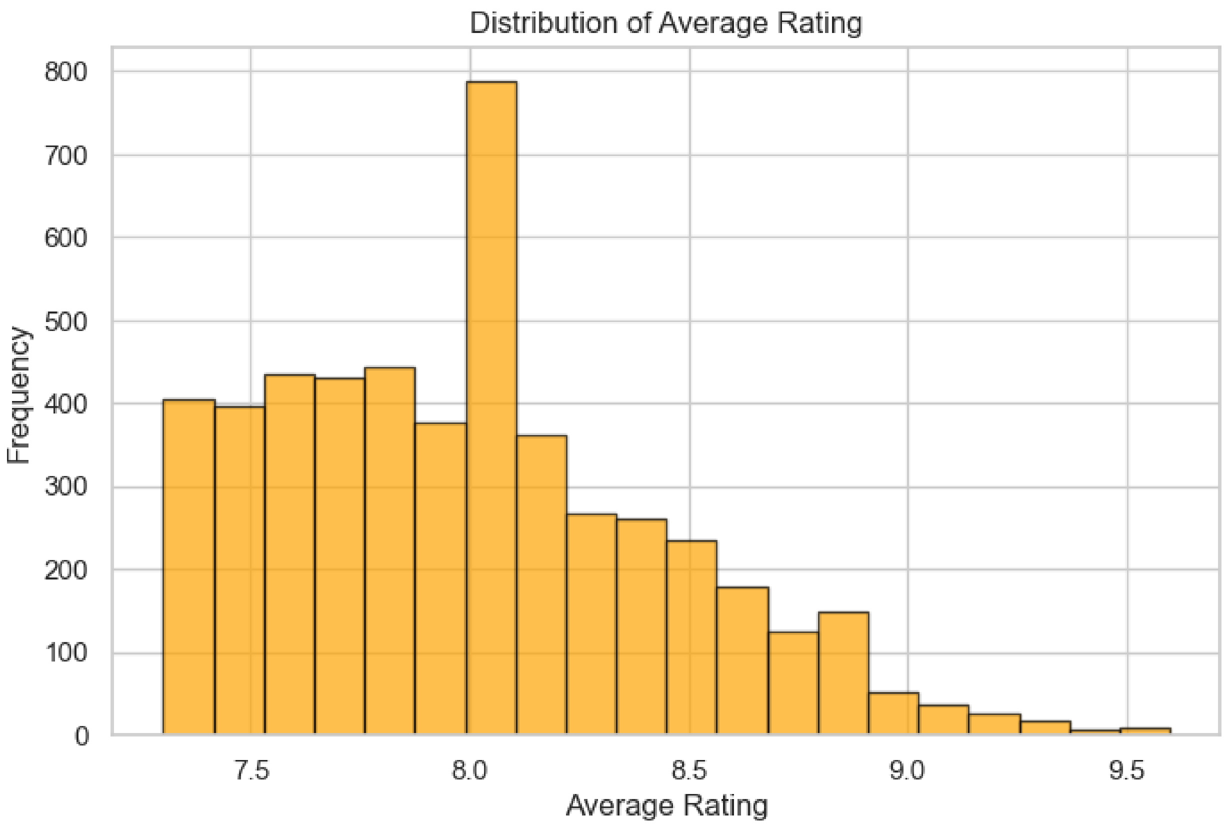
```
# Correlation Heatmap
corr = df[['startYear', 'endYear', 'rank', 'averageRating',
'numVotes']].corr()
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()
```



```
import matplotlib.pyplot as plt

# Plot histogram for 'averageRating'
plt.figure(figsize=(8, 5))
df['averageRating'].hist(bins=20, color='skyblue', edgecolor='black',
alpha=0.7)
plt.title('Distribution of Average Rating')
plt.xlabel('Average Rating')
plt.ylabel('Frequency')
plt.show()
```



```
import matplotlib.pyplot as plt

# Plot histogram for 'averageRating'
plt.figure(figsize=(8, 5))
df['averageRating'].hist(bins=20, color='orange', edgecolor='black',
alpha=0.7)
plt.title('Distribution of Average Rating')
plt.xlabel('Average Rating')
plt.ylabel('Frequency')
plt.show()
```



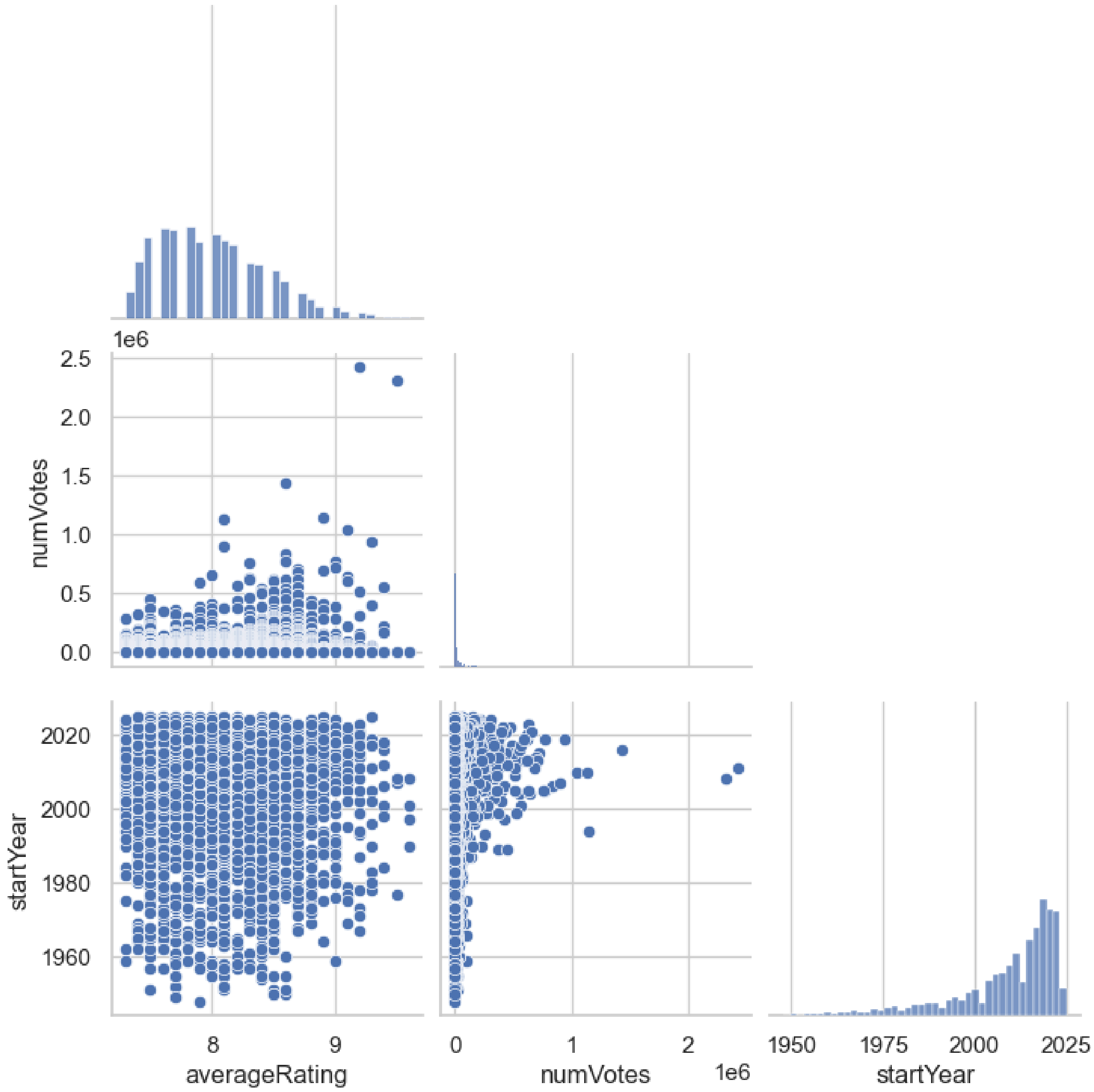
```
# Cross-tabulation: Genre vs Rating Bin
pd.crosstab(df['main_genre'], df['rating_bin'])
```

```
rating_bin   Medium  High  Top
main_genre
Action          178   595   80
Adventure        42   224   37
Animation        30   234   19
Biography        30   136   26
Comedy          186  1007  173
Crime           130   429   55
Documentary      14   209  100
Drama           169   629   85
Family            2    12    2
Fantasy           1    15    0
Game-Show         5    24    5
History           0     1    1
Horror            2     6    0
Music             1     3    1
Mystery           2     6    2
News              0     6    1
Reality-TV        4    33    4
Romance           2     8    2
Sci-Fi            1     1    0
```

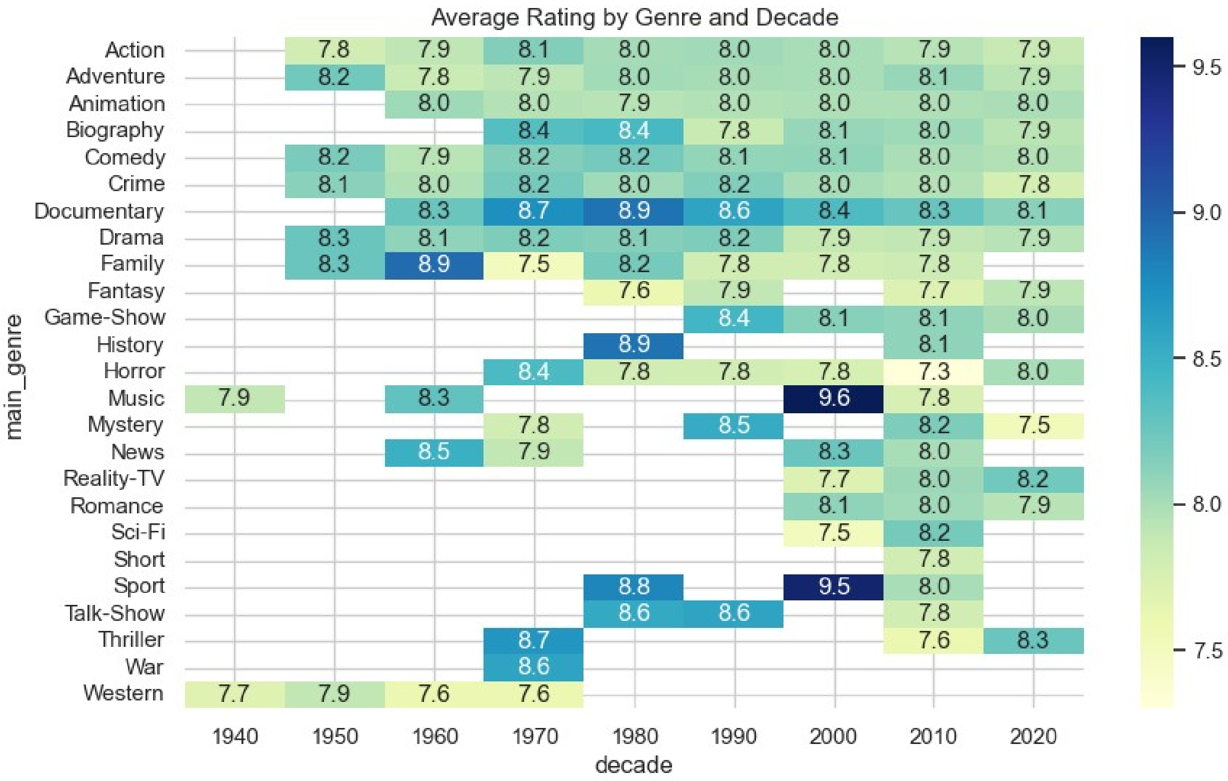
```
Short                0     1    0
Sport                0     1    2
Talk-Show            0     3    2
Thriller             0     3    2
War                  0     0    1
Western              2    13    0
```

Multivariate Analysis

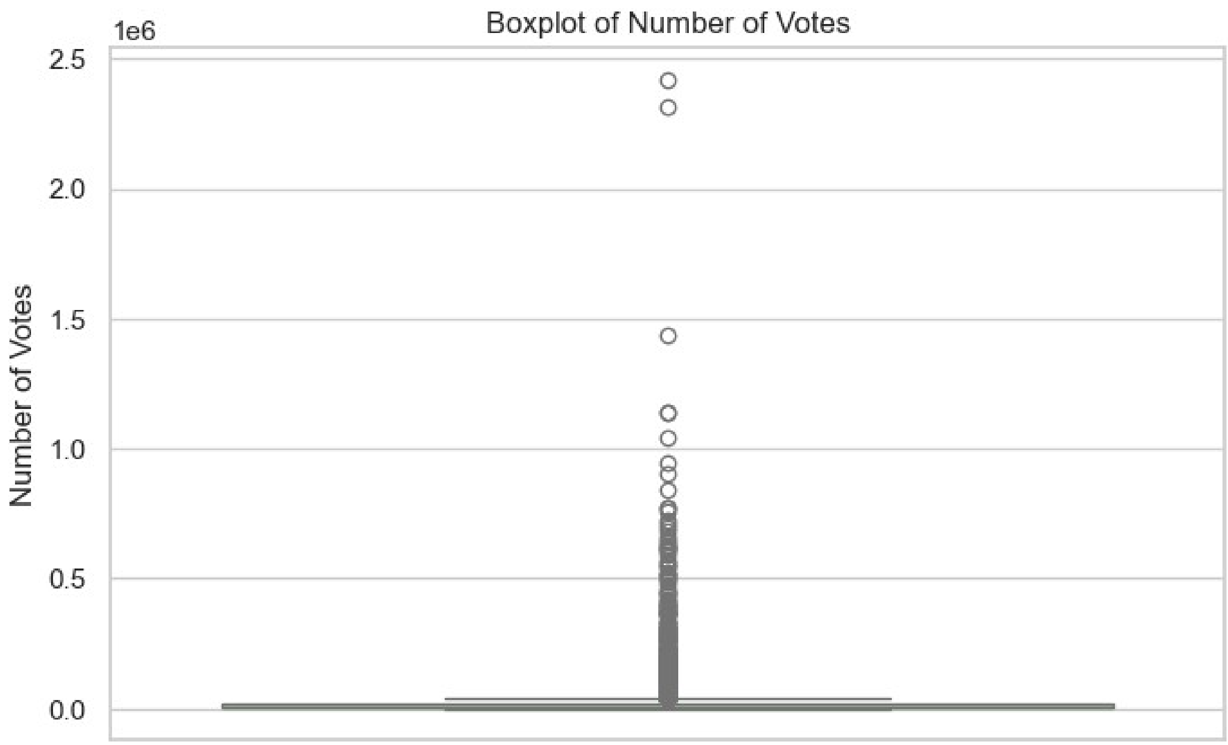
```
# Pairplot
sns.pairplot(df[['averageRating', 'numVotes', 'startYear']],
corner=True)
plt.show()
```



```
# Heatmap: Average Rating by Genre and Decade
pivot = df.pivot_table(values='averageRating', index='main_genre',
columns='decade', aggfunc='mean')
sns.heatmap(pivot, annot=True, cmap='YlGnBu', fmt=".1f")
plt.title("Average Rating by Genre and Decade")
plt.show()
```



```
import matplotlib.pyplot as plt
import seaborn as sns

# Boxplot for 'numVotes'
plt.figure(figsize=(8, 5))
sns.boxplot(y=df['numVotes'], color='lightgreen')
plt.title('Boxplot of Number of Votes')
plt.ylabel('Number of Votes')
plt.show()
```



```
# Save the cleaned dataset
df.to_csv('cleaned_imdb_dataset.csv', index=False)
print("\n□ Cleaned dataset saved as 'cleaned_imdb_dataset.csv'")


□ Cleaned dataset saved as 'cleaned_imdb_dataset.csv'
```