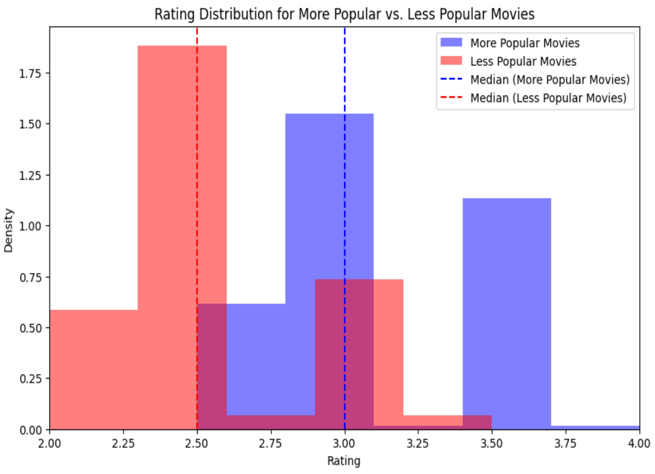
Data Analysis Project 1 Report

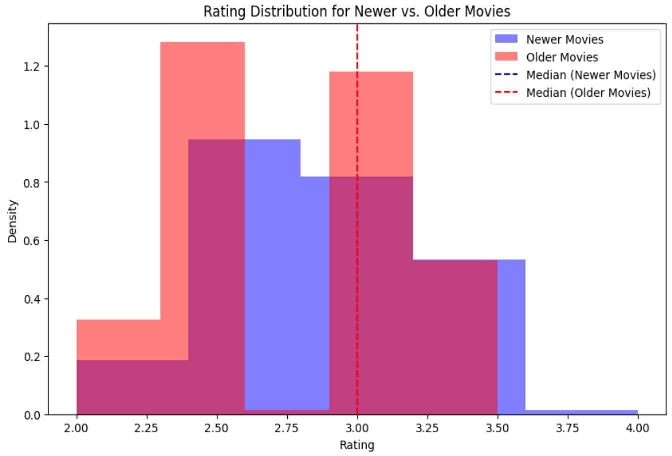
Group 9 – Smruti Nalawade, Carina Sun, Hongjiu Zhang

Question 1: We classified the movies into “more popular” and “less popular” groups based on the median number of ratings they received. To handle missing ratings (nan), we applied element-wise deletion, as not every user rated every movie. For each movie, we used the median of its available ratings as its representative rating and then conducted a one-sided **Mann-Whitney U-test** to compare movie ratings. We selected the Mann-Whitney U-test because it’s inappropriate to rely solely on the mean for movie ratings, given that we can’t assume consistent intervals between ratings (e.g., the difference between 2-3, 3-4, and 4-5 may not be uniform). For the hypotheses in the U-test for this question, we have:**H_0: More popular movies are not rated higher than less popular movies.H_1: More popular movies are rated higher than less popular movies.** The U-test produced a statistic of 33427.5 and a p-value of 9.929×10^{-35}. With such a low p-value, we have 99.5% confidence to conclude that more popular movies tend to receive higher ratings. Additionally, as illustrated in the figure on the right, we used histograms to visualize the rating distributions for both “more popular” and “less popular” movies, high-lighting their respective medians. The distributions in these groups further support our conclusion.



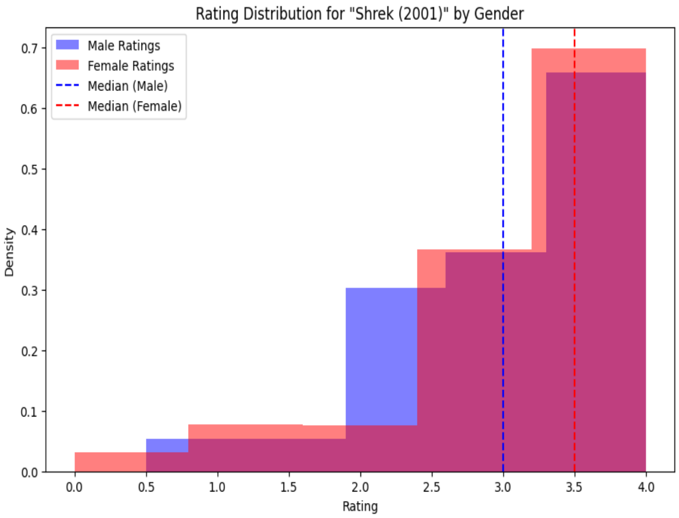
Question 2:

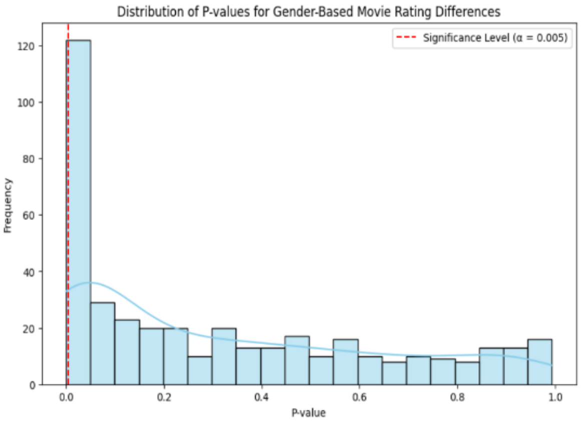
First, we extracted each movie's release year from the header and classified the movies into two groups—"newer movies" and "older movies"—based on the median year, 1999. We then **1. Removed missing values element- wise**, **2.** Used the median of each movie’s available ratings to **represent its rating** 3. Applied a **two-sided test Mann Whitney U- test** because we wanted to detect any difference in ratings between newer and older movies, without assuming which group might have higher ratings. For the hypotheses in the U-test, we defined: **H_0: Newer movies are not rated differently than older movies. H_1: Newer movies are rated differently than older movies.** The U-test yielded a statistic of 21061.0 and a p-value of 0.198, which is above the 0.005 significance threshold. Therefore, we cannot conclude that there is a significant difference in ratings between newer and older movies. Additionally, as shown in the figure on the right, the rating distributions for newer vs older movies mostly overlap, visually confirming that the difference between the groups is minimal, which further supports our conclusion.



Question 3:

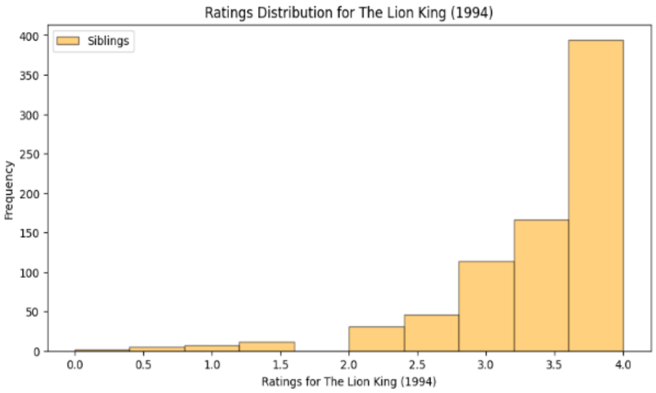
To compare the ratings of "Shrek" between female and male viewers, we began by selecting the ratings for "Shrek" and dividing them into two groups based on gender, while also removing any missing values. We plotted the histograms of ratings over “Shrek” of female and male viewer groups as shown in the figure on the right**.** We observed that neither group follows a normal distribution, which is expected as movie ratings are ordinal rather than continuous. Since the distributions of both groups are generally similar, the **Mann-Whitney U-test** is an appropriate choice for this question. We conducted a **two-sided test** to address the question of whether the two groups rate "Shrek" differently. The hypotheses were set as follows: **H_0:** There is no difference between the two groups' ratings on "Shrek"**; H_1:** There is a difference between the two groups' ratings. Using the **Mann-Whitney U-test**, we obtained a **p-value of 0.051**. Therefore, we could not conclude that there is a significant difference at the 0.005 significance level. Additionally, we observed that the **median rating** of female viewers was higher than that of male viewers. Then we also performed a **one-sided test** with **p-value 0.025**, which is still greater than 0.005, so we could not draw a significant conclusion from this test either. In conclusion, **female and male viewers do not rate differently**.

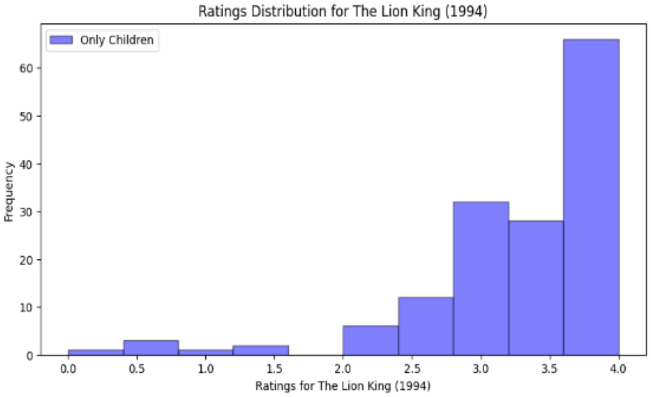


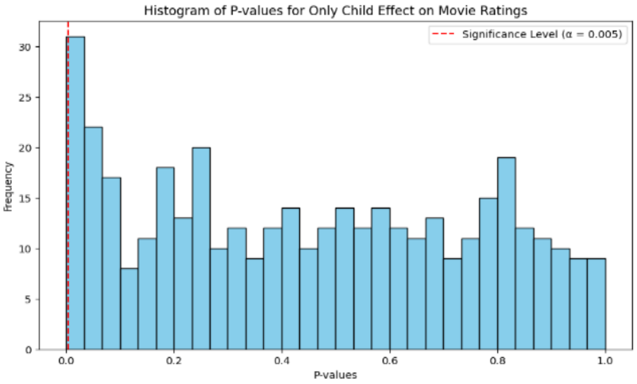


Question 4: For reasons similar to those mentioned earlier, we chose the Mann-Whitney U-test. Next, we isolated the data into 400 movie ratings and divided each movie rating into female and male groups. We used element-wise removal to deal with empty values because we were interested in the difference between two groups instead of the change about the same person. Finally, we found that 12.5 % of movies were rated differently by male and female viewers at significance level U = 0.005. Figure on the right shows the histogram of 400 p-values of significance tests.

Question 5: Here we are dealing with **one-sided** test directly. We assign the ratings over “The Lion King” to two groups according to the “Are you an only child” feature, remove the missing values, and turn ratings into a NumPy array for further modification. The figures below show that both groups do not follow normal distribution, thus, we turn to **Mann-Whitney U-test** again. In this question, **H0:** There is no difference between two groups’ ratings over “The Lion King” E.B. **H1:** People who are only children rate higher than people with siblings. Using Mann-Whitney U-test gives **test statistic to be 52929.0** and **p-value to be 0.978**. Thus, we failed to reject the null hypothesis under U = 0.005 significance level, and **we concluded that people who are only children do not enjoy “The Lion King” more than people with siblings**.

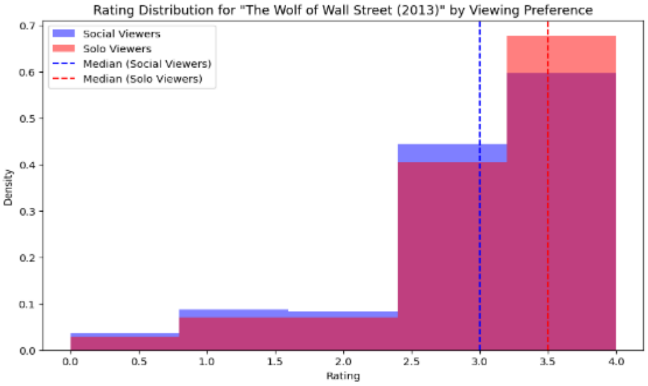




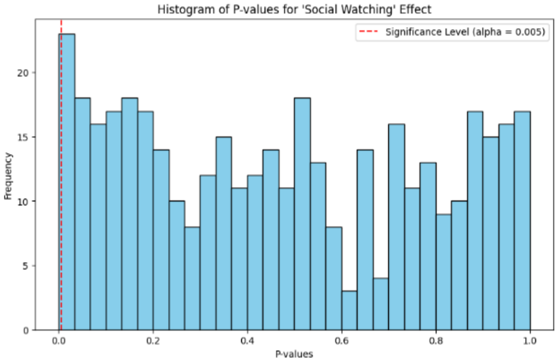


Question 6: For reasons similar to those mentioned earlier, we chose the Mann-Whitney U-test. We isolated the data into 400 movie-ratings and divided each movie rating into two groups. One group consisted of people who were the only child, and another group consisted of the people who had siblings. We used element-wise removal to deal with empty values. Finally, we found that 1.75% of movies exhibited an “only child effect” at significance level U = 0.005. Figure on the right shows the histogram of 400 p-values of significance tests.

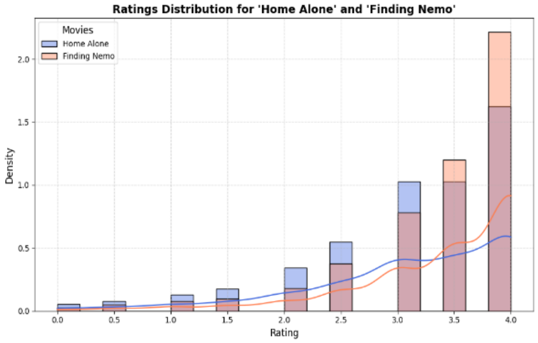
Question 7: For this question, we set up the hypotheses as follows: **H_0:** There is no difference between the ratings of "The Wolf of Wall Street" by social and solo viewers**.** **H_1:** People who prefer to watch movies socially rate "The Wolf of Wall Street" higher than those who prefer to watch alone. We conducted a **one-sided Mann-Whitney U-test** and obtained a test statistic of **49303.5** and a p-value of **0.112**. Since the p-value is greater than the significance level of 0.005, we failed to reject the null hypothesis. Therefore, we concluded that people who like to watch movies socially do not enjoy "The Wolf of Wall Street (2013)" more than those who prefer to watch alone.



Question 8: Similarly, we chose the **Mann-Whitney U-test**. We isolated the data into 400 movie ratings and divided each movie rating into two groups. One group contained people who liked to watch movies socially, and another group preferred watching alone. We used element-wise removal to deal with empty values. Finally, we found that **2.5%** of movies showed a "social watching" effect at significance level U = 0.005.



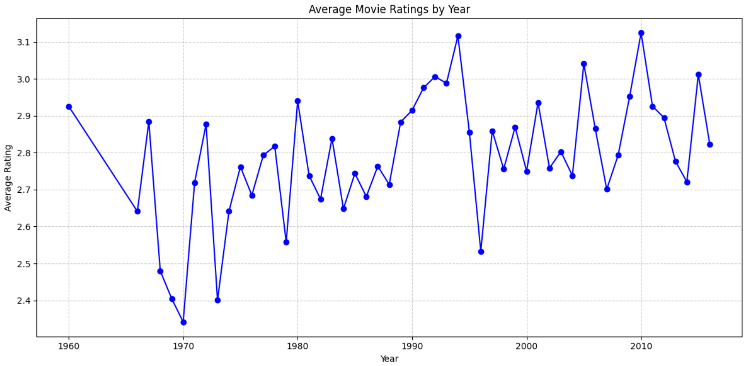
Question 9: We conducted a **Kolmogorov-Smirnov (KS)** test, as our objective was to compare the two continuous **distributions** of ratings for two movies. After an elementwise drop of any missing values, we proceeded with the test. The KS test yielded a p-value of **6.379 × 10⁻¹⁰**, which is significantly less than alpha = 0.005. Given this p-value, we are more than 99.5% confident that the distributions of ratings for the two movies are different.

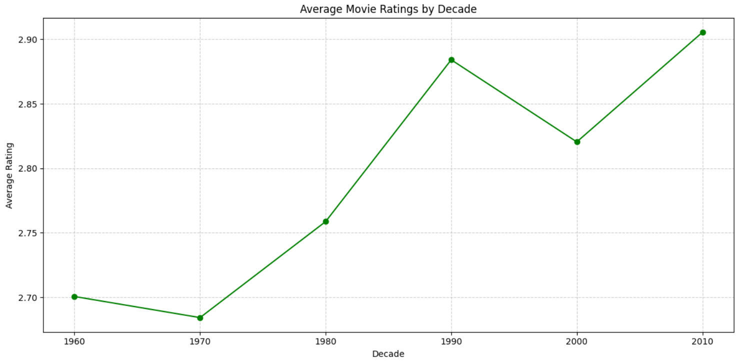


Question 10: To compare the ratings of movies within each franchise, we decided to use the **Kruskal-Wallis** test. This test is well-suited for data that cannot be accurately summarized by sample means and involves more than two groups, making it ideal for our needs. The Kruskal-Wallis test helps determine if there is a statistically significant difference in ratings among the movies within each franchise. The hypotheses for each franchise are as follows: **H0**: There is no significant difference in movie ratings within the franchise; **H1**: At least one movie within the franchise has a significantly different rating. The Kruskal-Wallis test was conducted, with 7 franchises' p value far below 0.005, we conclude that there is a statistically significant difference in ratings within those franchises, indicating inconsistency. Among these franchises, only the Harry Potter series has a consistent rating.

Extra Finding:

The trend of average movie ratings by year reveals fluctuating viewer preferences over time, with no clear upward or downward trend. This suggests that movie quality, as rated by audiences, is influenced by unique cultural, technological, or cinematic factors across years rather than improving or declining consistently.





To take a better look, let's group the years into decades. From the 1970s onward, there is a steady rise in average ratings, peaking in the 1990s at around 2.90. The 2000s show a slight drop in ratings, falling back down to around 2.80. This could be due to a more diverse range of movies, including some that were polarizing or less universally appreciated by viewers. The 2010s mark the highest point on the graph, with the average rating approaching 2.95. This rise could be attributed to advancements in technology, improved production values, and a greater focus on quality storytelling in recent years.

Movie Project

```
import pandas as pd
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import mannwhitneyu, kruskal, ks_2samp, ttest_ind,normaltest
import re
```

```
from google.colab import files
import io
uploaded = files.upload()
```

Choose Files movieReplicationSet.csv

- **movieReplicationSet.csv**(text/csv) - 1580654 bytes, last modified: 10/29/2024 - 100% done

```
Saving movieReplicationSet.csv to movieReplicationSet.csv
```

+ Code + Text

```
file_path = 'movieReplicationSet.csv'
df_raw = pd.read_csv(file_path)

df_raw.head()
```

	The Life of David Gale (2003)	Wing Commander (1999)	Django Unchained (2012)	Alien (1979)	Indiana Jones and the Last Crusade (1989)	Snatch (2000)	Rambo: First Blood Part II (1985)	Fargo (1996)	Let the Right One In (2008)	Black Swan (2010)	...	When watching a movie I cheer or shout or talk or curse at the screen	When watching a movie I feel like the things on the screen are happening to me	As a movie unfolds I start to have problems keeping track of events that happened earlier	em th on ins ha g fr ha g
0	NaN	NaN	4.0	NaN	3.0	NaN	NaN	NaN	NaN	NaN	...	1.0	6.0	2.0	
1	NaN	NaN	1.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	3.0	1.0	1.0	
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	5.0	4.0	3.0	
3	NaN	NaN	2.0	NaN	3.0	NaN	NaN	NaN	NaN	4.0	...	3.0	1.0	1.0	

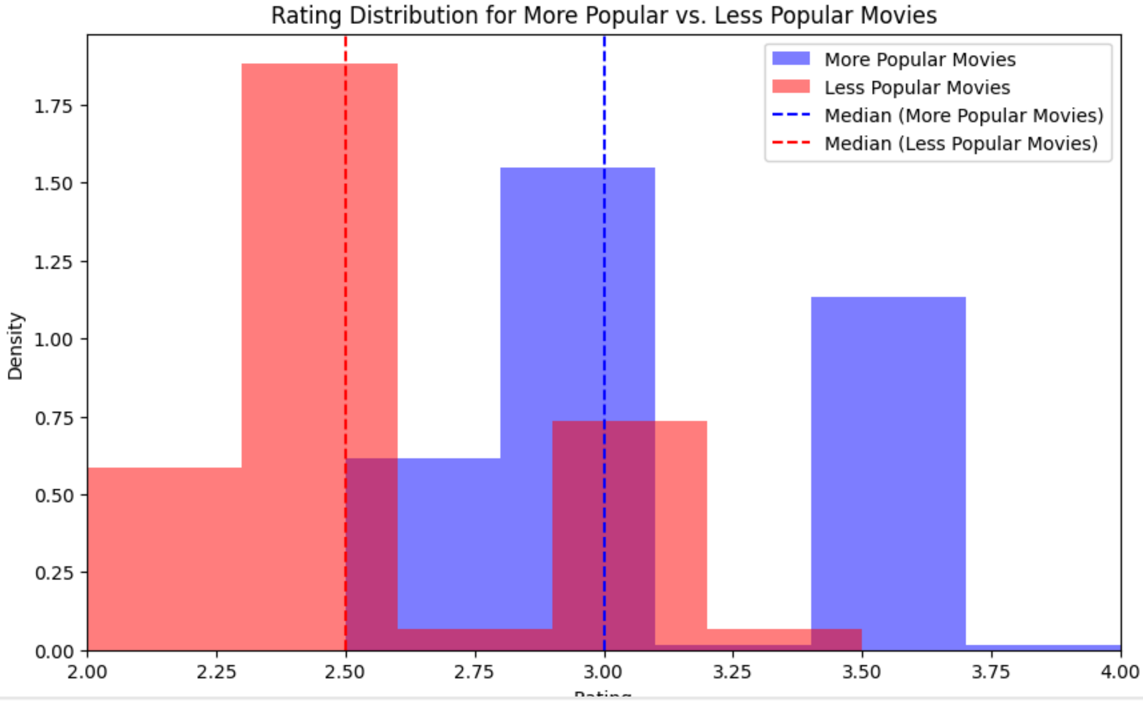
```
# Just delete the row with all null value
ratings_columns = df_raw.columns[:400]
df = df_raw.dropna(subset=ratings_columns, how='all')

print(df[ratings_columns].isnull().sum())
```

```
The Life of David Gale (2003)                  1020
Wing Commander (1999)                          1025
Django Unchained (2012)                         643
Alien (1979)                                    807
Indiana Jones and the Last Crusade (1989)       633
                                               ... 
Patton (1970)                                  1035
Anaconda (1997)                                 898
Twister (1996)                                  922
MacArthur (1977)                               1035
Look Who's Talking (1989)                       988
Length: 400, dtype: int64
```

1) Are movies that are more popular (operationalized as having more ratings) rated higher than movies that are less popular? [Hint: You can do a median-split of popularity to determine high vs. low popularity movies]

```
movie_popularity = df[ratings_columns].count()
median_popularity = movie_popularity.median()
popularity_index = movie_popularity.index
# do a median-split of popularity to determine high vs. low popularity movies
more_popular_movies = df.loc[:, popularity_index[movie_popularity > median_popularity]]
less_popular_movies = df.loc[:, popularity_index[movie_popularity <= median_popularity]]
more_popular_medians = more_popular_movies.median(axis=0)
less_popular_medians = less_popular_movies.median(axis=0)

# Apply Mann-Whitney U-test instead of T-test cause we want to compare the ratings
stat, p_value = mannwhitneyu(more_popular_medians, less_popular_medians, alternative='greater')

print(f"U-test statistic: {stat}")
print(f"p-value: {p_value}")

if p_value < 0.005:
    print("More popular movies are rated significantly higher than less popular ones.")
else:
    print("No significant difference in ratings between more and less popular movies.")

 # Show the plot
plt.figure(figsize=(10,6))
plt.hist(more_popular_medians, bins=5, alpha=0.5, label='More Popular Movies', color='blue', density=True)
plt.hist(less_popular_medians, bins=5, alpha=0.5, label='Less Popular Movies', color='red', density=True)
plt.axvline(more_popular_medians.median(), color='blue', linestyle='--', label='Median (More Popular Movies)')
plt.axvline(less_popular_medians.median(), color='red', linestyle='--', label='Median (Less Popular Movies)')
plt.title('Rating Distribution for More Popular vs. Less Popular Movies')
plt.xlabel('Rating')
plt.ylabel('Density')
plt.xlim(2, 4)
plt.legend()
plt.show()
```

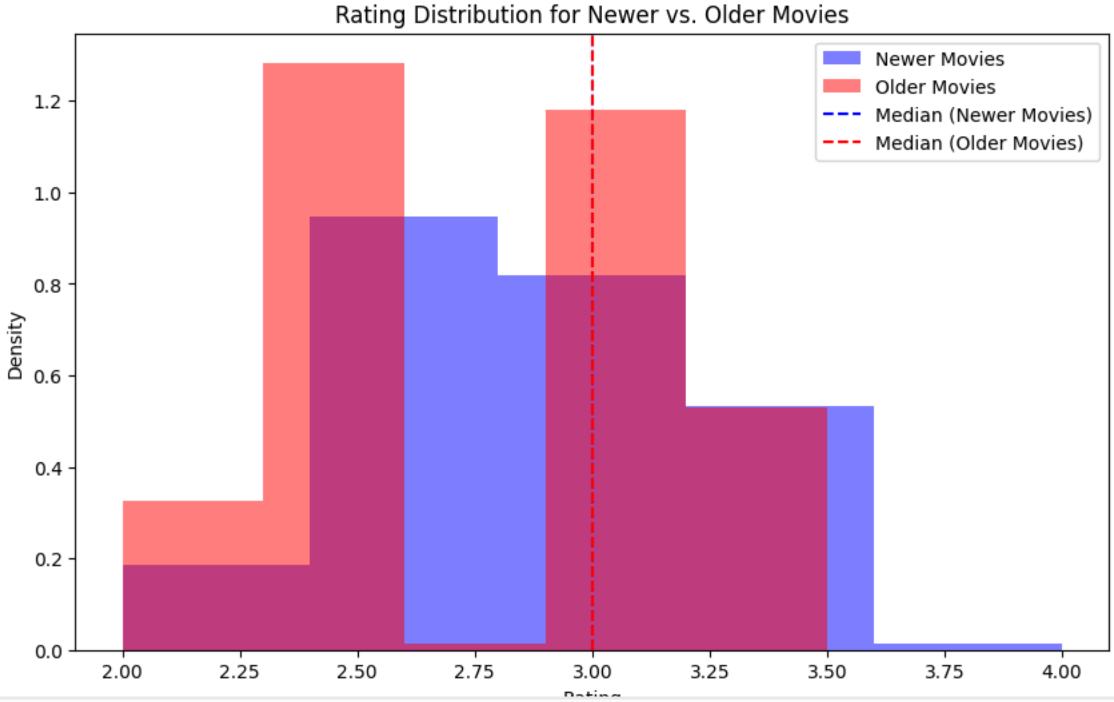
```
U-test statistic: 33427.5
p-value: 9.929258851707232e-35
More popular movies are rated significantly higher than less popular ones.
```



2) Are movies that are newer rated differently than movies that are older? [Hint: Do a median split of year of release to contrast movies in terms of whether they are old or new]

```
#extract the year and restore in a list first
movie_years = []
for movie in df.columns[:400]:
    match = re.search(r'\((\d{4})\)', movie)
    if match:
        year = int(match.group(1))
        movie_years.append(year)
    else:
        movie_years.append(None)

# Split the movies into newer and older movies based on the median year
median_year = np.median([year for year in movie_years if year is not None])
newer_movies = df.iloc[:, :400].loc[:, [year > median_year for year in movie_years]]
older_movies = df.iloc[:, :400].loc[:, [year <= median_year for year in movie_years]]
newer_movie_medians = newer_movies.median(axis=0)
older_movie_medians = older_movies.median(axis=0)
# Perform two-sided Mann-Whitney U-test
stat, p_value = mannwhitneyu(newer_movie_medians, older_movie_medians, alternative='two-sided')

# Results show
print(f"U-test statistic: {stat}")
print(f"p-value: {p_value}")
if p_value < 0.005:
    print("There is a significant difference in ratings between newer and older movies.")
else:
    print("There is no significant difference in ratings between newer and older movies.")
# Plot histograms for newer and older movies, and the medians for newer and older movies
plt.figure(figsize=(10,6))
plt.hist(newer_movie_medians, bins=5, alpha=0.5, label='Newer Movies', color='blue', density=True)
plt.hist(older_movie_medians, bins=5, alpha=0.5, label='Older Movies', color='red', density=True)
plt.axvline(newer_movie_medians.median(), color='blue', linestyle='--', label='Median (Newer Movies)')
plt.axvline(older_movie_medians.median(), color='red', linestyle='--', label='Median (Older Movies)')
plt.title('Rating Distribution for Newer vs. Older Movies')
plt.xlabel('Rating')
plt.ylabel('Density')
plt.legend()
plt.show()
```

```
U-test statistic: 21061.0
p-value: 0.19865156776112602
There is no significant difference in ratings between newer and older movies.
```

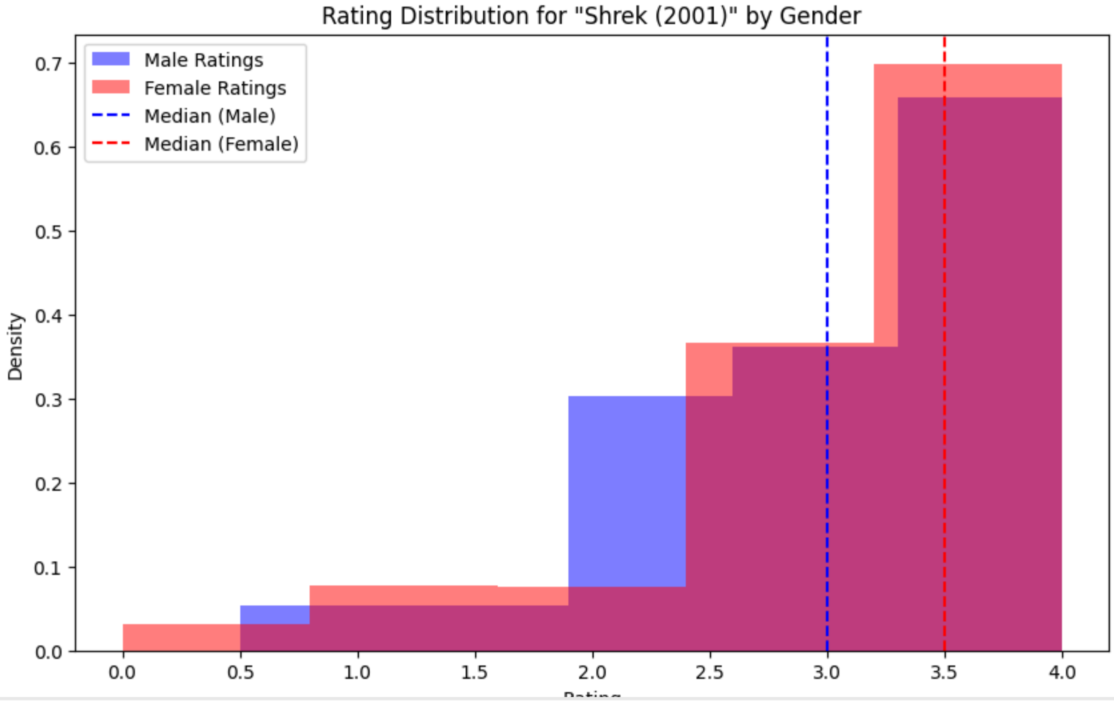


3) Is enjoyment of 'Shrek (2001)' gendered, i.e. do male and female viewers rate it differently?

```
# Extract the ratings for "Shrek (2001)" and drop missing values
shrek_ratings = df['Shrek (2001)'].dropna()
gender_column = df['Gender identity (1 = female; 2 = male; 3 = self-described)']
# Apply U test
male_ratings = shrek_ratings[gender_column == 2]
female_ratings = shrek_ratings[gender_column == 1]

stat, p_value = mannwhitneyu(female_ratings, male_ratings, alternative='two-sided')
```

```
# result show
print(f"U-test statistic: {stat}")
print(f"p-value: {p_value}")
if p_value < 0.005:
    print("There is a significant difference in how male and female viewers rate 'Shrek (2001)'.")
else:
    print("There is no significant difference in how male and female viewers rate 'Shrek (2001)'.")

plt.figure(figsize=(10,6))

# Plot histograms for male and female ratings
plt.hist(male_ratings, bins=5, alpha=0.5, label='Male Ratings', color='blue', density=True)
plt.hist(female_ratings, bins=5, alpha=0.5, label='Female Ratings', color='red', density=True)
plt.axvline(male_ratings.median(), color='blue', linestyle='--', label='Median (Male)')
plt.axvline(female_ratings.median(), color='red', linestyle='--', label='Median (Female)')
plt.title('Rating Distribution for "Shrek (2001)" by Gender')
plt.xlabel('Rating')
plt.ylabel('Density')
plt.legend()
plt.show()
```

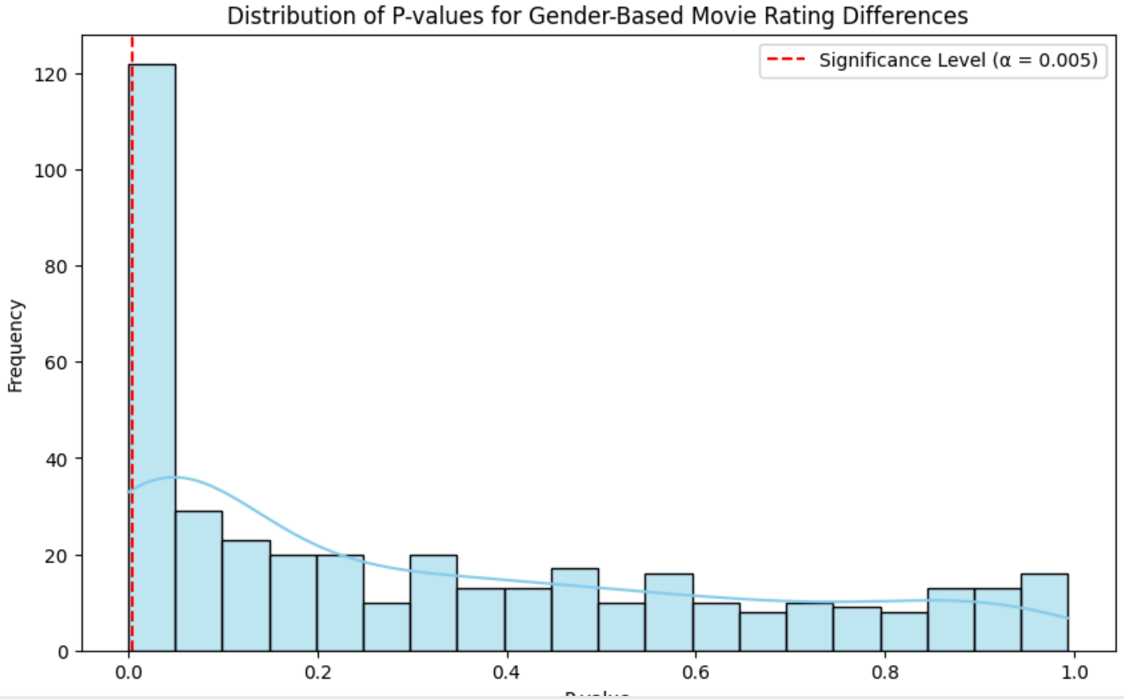
```
U-test statistic: 96830.5
p-value: 0.050536625925559006
There is no significant difference in how male and female viewers rate 'Shrek (2001)'.
```



4) What proportion of movies are rated differently by male and female viewers?

```
# Extract movie rating columns and gender column
movie_ratings = df.iloc[:, :400]
gender = df['Gender identity (1 = female; 2 = male; 3 = self-described)']

# Initialize a counter for movies rated differently by gender
diff_gender_count = 0
p_values = []
alpha = 0.005  # Significance level set to 0.005

# Loop through each movie column to test rating differences by gender
for col in movie_ratings.columns:
    male_ratings = movie_ratings[col][gender == 2].dropna()  # Male viewers
    female_ratings = movie_ratings[col][gender == 1].dropna()  # Female viewers

# Perform Mann-Whitney U test between male and female ratings
    u_stat, p_value = mannwhitneyu(male_ratings, female_ratings, alternative='two-sided')
    p_values.append(p_value)

    # Check if p-value is below significance level
    if p_value < alpha:
        diff_gender_count += 1

# Calculate proportion of movies rated differently by gender
proportion_diff_gender = diff_gender_count / movie_ratings.shape[1]
print(f"Proportion of movies rated differently by gender: {proportion_diff_gender:.3f}")

# Plot the histogram of p-values with significance level indicated by dashed lines
plt.figure(figsize=(10, 6))
sns.histplot(p_values, bins=20, kde=True, color="skyblue")
plt.axvline(x=alpha, color='red', linestyle='--', linewidth=1.5, label=f'Significance Level (α = {alpha})')
plt.title("Distribution of P-values for Gender-Based Movie Rating Differences")
plt.xlabel("P-value")
plt.ylabel("Frequency")
plt.legend()

plt.show()
```

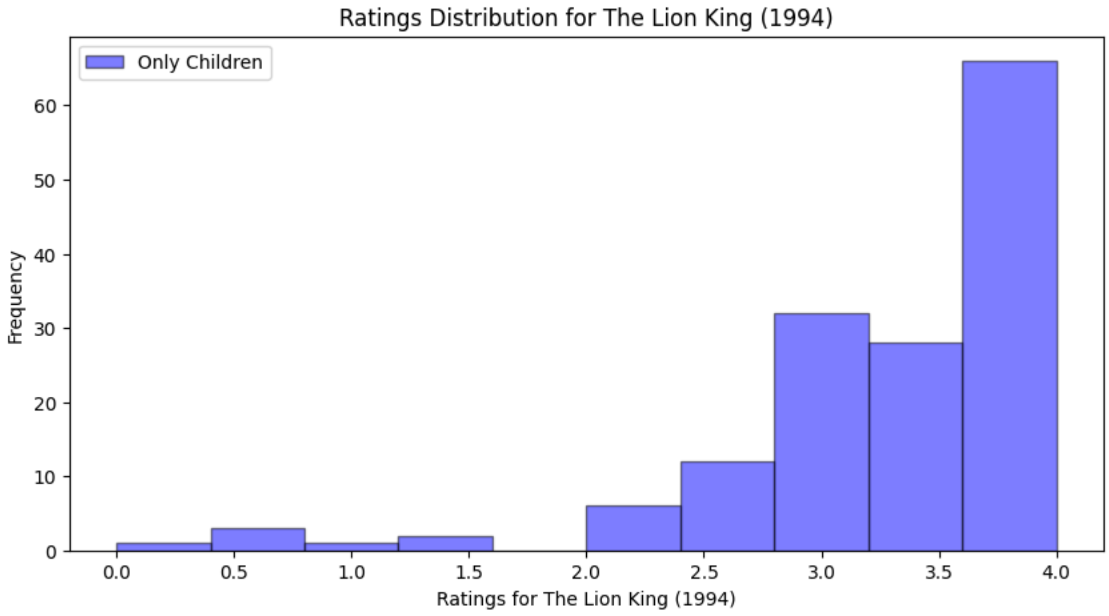
```
Proportion of movies rated differently by gender: 0.125
```

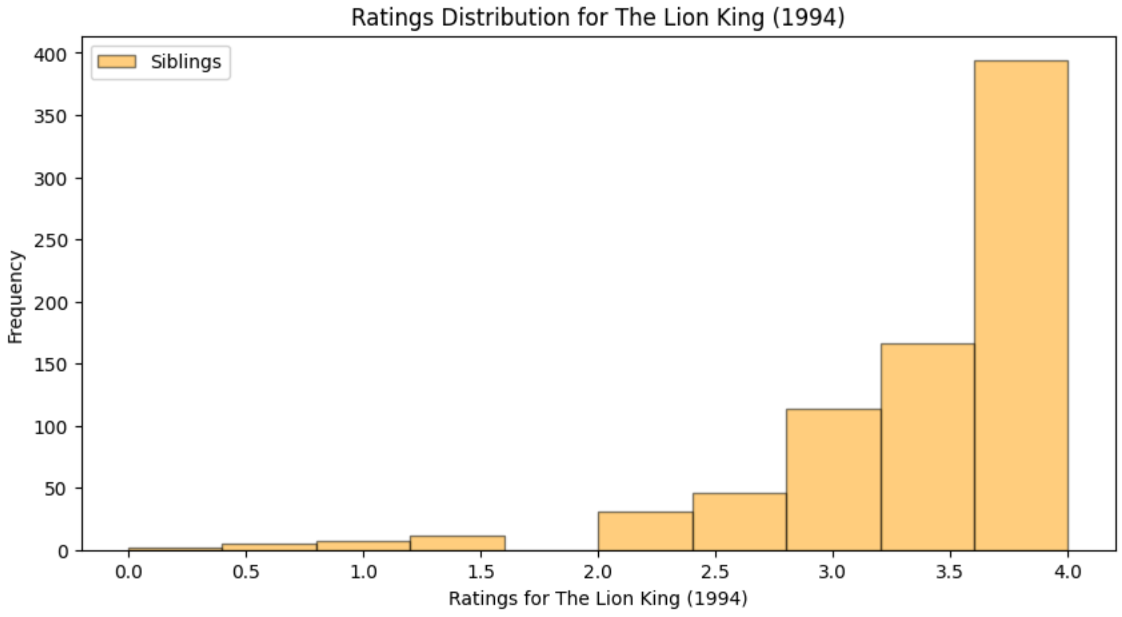


5) Do people who are only children enjoy ‘The Lion King (1994)’ more than people with siblings?

```
lion_king_ratings = df['The Lion King (1994)']
only_child_ratings = lion_king_ratings[df['Are you an only child? (1: Yes; 0: No; -1: Did not respond)'] == 1].dropna()
siblings_ratings = lion_king_ratings[df['Are you an only child? (1: Yes; 0: No; -1: Did not respond)'] == 0].dropna()


# Plotting the histogram for 'Only Children'
plt.figure(figsize=(10, 5))
plt.hist(only_child_ratings, alpha=0.5, label='Only Children', bins=10, color='blue', edgecolor='black')
plt.xlabel('Ratings for The Lion King (1994)')
plt.ylabel('Frequency')
plt.title('Ratings Distribution for The Lion King (1994)')
plt.legend()
plt.show()

# Plotting the histogram for 'Siblings'
plt.figure(figsize=(10, 5))
plt.hist(siblings_ratings, alpha=0.5, label='Siblings', bins=10, color='orange', edgecolor='black')
plt.xlabel('Ratings for The Lion King (1994)')
plt.ylabel('Frequency')
plt.title('Ratings Distribution for The Lion King (1994)')
plt.legend()
plt.show()

# Conduct the one-sided Mann-Whitney U Test
stat, p_value = mannwhitneyu(only_child_ratings, siblings_ratings, alternative='greater')
```

```
# Reporting results
print(f"Question 5: U-statistic = {stat}, p-value = {p_value}")
```

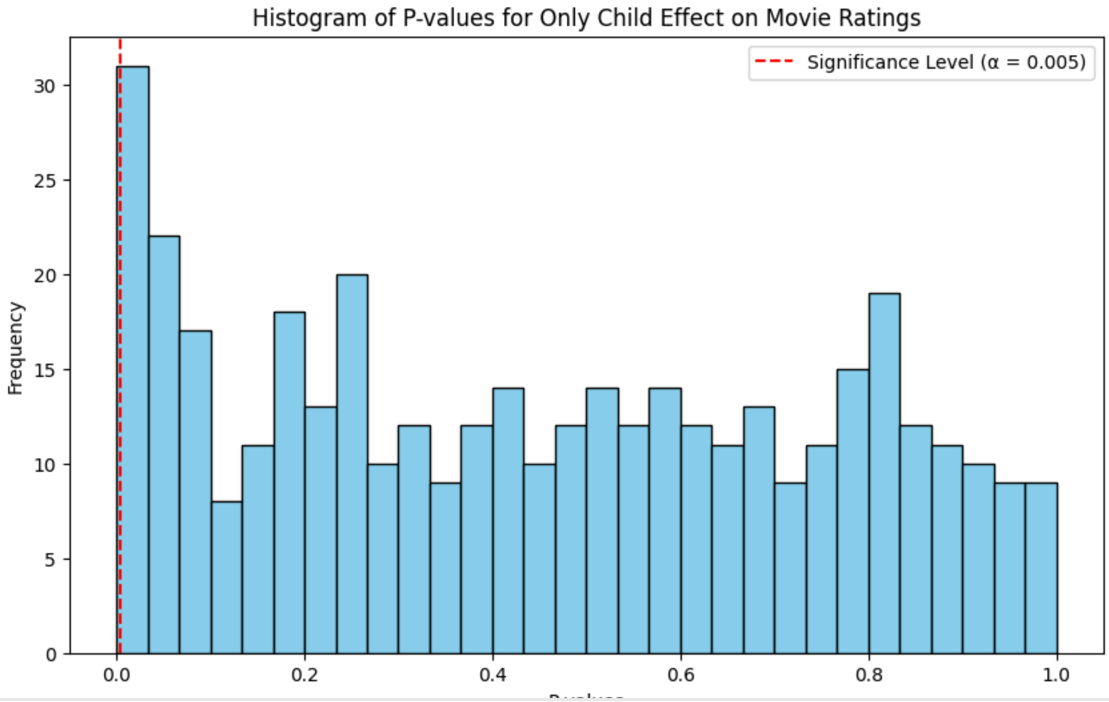




6) What proportion of movies exhibit an “only child effect”, i.e. are rated different by viewers with siblings vs. those without?

```
# Define column ranges for movies and only child status
movie_columns = df.columns[0:400]  # Assuming the first 400 columns are movie ratings
only_child_column = 'Are you an only child? (1: Yes; 0: No; -1: Did not respond)'

# Filter dataset for only child and non-only child ratings
only_child_ratings = df[df[only_child_column] == 1][movie_columns]
non_only_child_ratings = df[df[only_child_column] == 0][movie_columns]

# Initialize count for significant differences
significant_movies = 0

# Initialize variables for storing p-values
p_values = []
alpha = 0.005  # Significance level

# Perform Mann-Whitney U test on each movie
for movie in movie_columns:
    only_child_data = only_child_ratings[movie].dropna()
    non_only_child_data = non_only_child_ratings[movie].dropna()
    # Skip if there's insufficient data in either group
    if len(only_child_data) > 1 and len(non_only_child_data) > 1:
        # Perform Mann-Whitney U Test
        u_stat, p_value = mannwhitneyu(only_child_data, non_only_child_data, alternative='two-sided')

        # Check if p-value is below threshold
        if p_value < alpha:
            significant_movies += 1

        p_values.append(p_value)

# Calculate proportion of movies with significant only child effect
total_movies = len(movie_columns)
proportion_significant = significant_movies / total_movies
print("Proportion of movies with significant only child effect:", proportion_significant)

# Plot histogram of p-values
plt.figure(figsize=(10, 6))
plt.hist(p_values, bins=30, color='skyblue', edgecolor='black')
plt.axvline(x=alpha, color='red', linestyle='dashed', linewidth=1.5, label=f'Significance Level (α = {alpha})')
plt.xlabel("P-values")
plt.ylabel("Frequency")
plt.title("Histogram of P-values for Only Child Effect on Movie Ratings")
plt.legend()
plt.show()
```

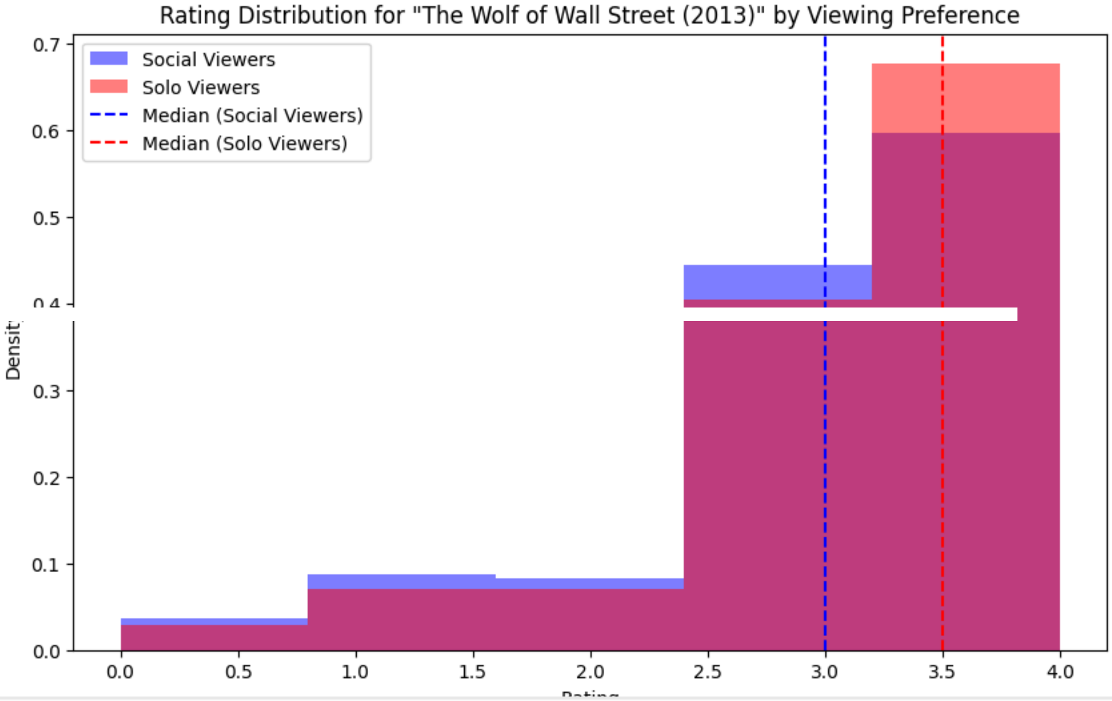
```
Proportion of movies with significant only child effect: 0.0175
```



7) Do people who like to watch movies socially enjoy ‘The Wolf of Wall Street (2013)’ more than those who prefer to watch them alone?

```
# Extract the ratings for "The Wolf of Wall Street (2013)" and drop missing values
wolf_ratings = df['The Wolf of Wall Street (2013)'].dropna()
social_watching = df['Movies are best enjoyed alone (1: Yes; 0: No; -1: Did not respond)']
social_watching = social_watching.astype(int)
social_ratings = wolf_ratings[social_watching == 0]
alone_ratings = wolf_ratings[social_watching == 1]
# Apply a two-sided Mann-Whitney U-test
stat, p_value = mannwhitneyu(social_ratings, alone_ratings, alternative='two-sided')

# result show
print(f"U-test statistic: {stat}")
print(f"p-value: {p_value}")
if p_value < 0.005:
    print("There is a significant difference in how people who like to watch movies socially and those who prefer to watch alone
else:
    print("There is no significant difference in how people who like to watch movies socially and those who prefer to watch alor
plt.figure(figsize=(10,6))
plt.hist(social_ratings, bins=5, alpha=0.5, label='Social Viewers', color='blue', density=True)
plt.hist(alone_ratings, bins=5, alpha=0.5, label='Solo Viewers', color='red', density=True)
plt.axvline(social_ratings.median(), color='blue', linestyle='--', label='Median (Social Viewers)')
plt.axvline(alone_ratings.median(), color='red', linestyle='--', label='Median (Solo Viewers)')
plt.title('Rating Distribution for "The Wolf of Wall Street (2013)" by Viewing Preference')
plt.xlabel('Rating')
plt.ylabel('Density')
plt.legend()
plt.show()
```

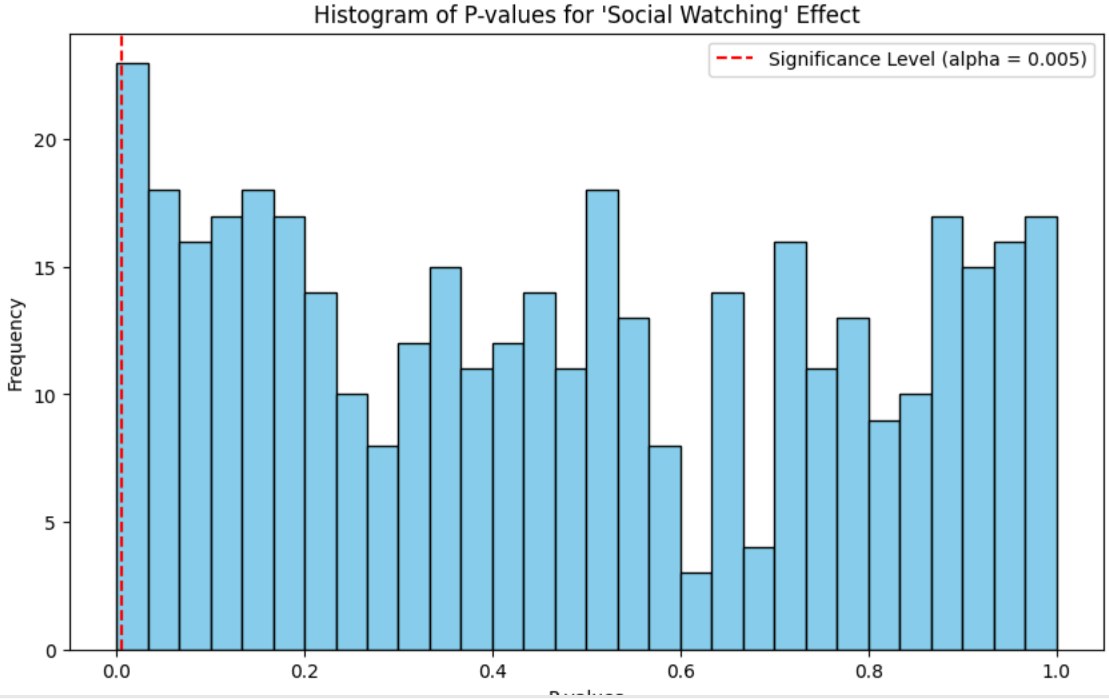
```
U-test statistic: 49303.5
p-value: 0.1127642933222891
There is no significant difference in how people who like to watch movies socially and those who prefer to watch alone rate
```



8) What proportion of movies exhibit such a “social watching” effect?

```
# Filter people by social preference
social_pref = df[df.columns[476]]

# Initialize count of movies with a significant difference
significant_movies_count = 0
alpha = 0.005
p_values = []

# Loop through each movie column (1 to 400)
for movie_col in df.columns[:400]:
    # Get ratings for both groups
    # prefers socially
    ratings_social = df[movie_col][social_pref == 0]
    # prefers alone
    ratings_alone = df[movie_col][social_pref == 1]

    # Perform U test
    t_value, p_value = mannwhitneyu(ratings_social.dropna(), ratings_alone.dropna())
    p_values.append(p_value)

    if p_value < alpha:
        significant_movies_count += 1

# Calculate proportion
proportion_social_effect = significant_movies_count / 400
print(f"Proportion of movies with a social watching effect: {proportion_social_effect}")

# Draw the graph
plt.figure(figsize=(10, 6))
plt.hist(p_values, bins=30, edgecolor='black', color='skyblue')
plt.axvline(x=alpha, color='red', linestyle='--', label=f'Significance Level (alpha = {alpha})')
plt.xlabel('P-values')
plt.ylabel('Frequency')
plt.title("Histogram of P-values for 'Social Watching' Effect")
plt.legend()
plt.show()
```

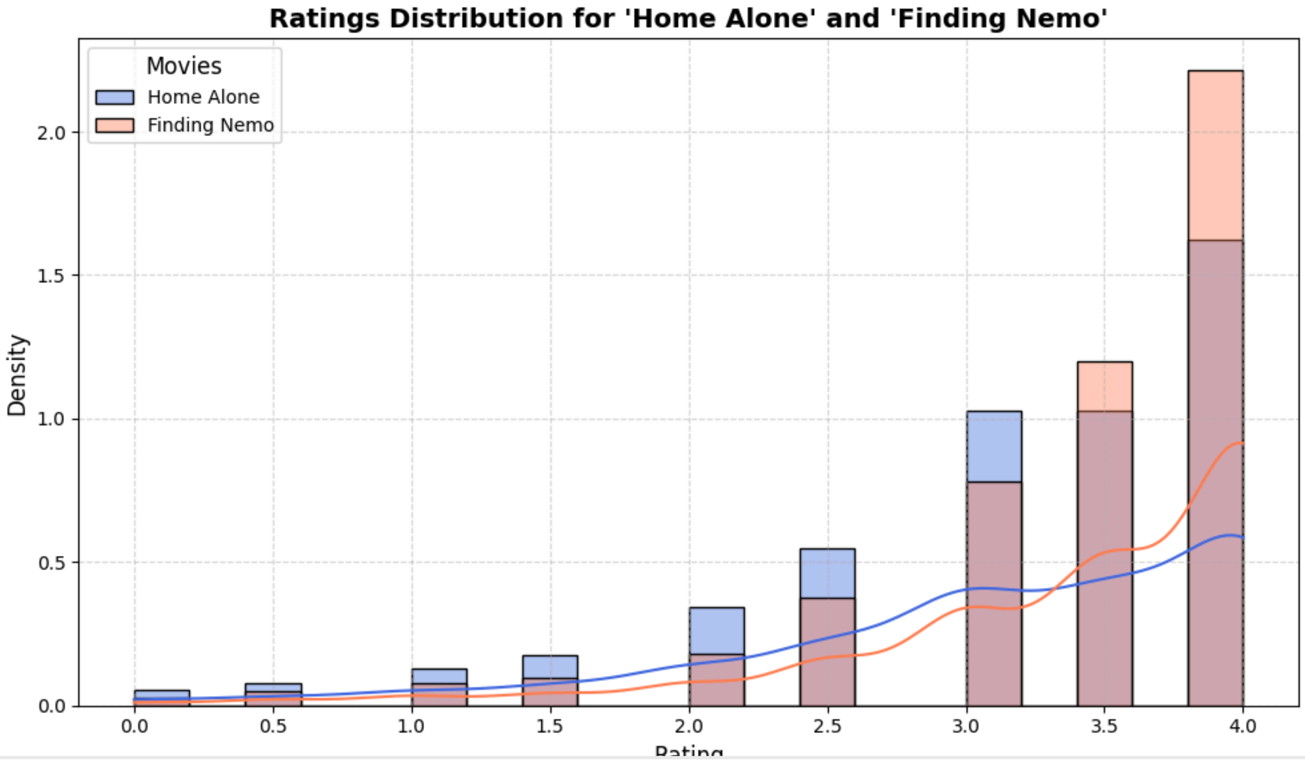
```
Proportion of movies with a social watching effect: 0.025
```



9) Is the ratings distribution of ‘Home Alone (1990)’ different than that of ‘Finding Nemo (2003)’?

```
home_alone_ratings = df['Home Alone (1990)'].dropna()
finding_nemo_ratings = df['Finding Nemo (2003)'].dropna()

# Perform KS test
_, p_value = ks_2samp(home_alone_ratings, finding_nemo_ratings)

# Print p
print(f"p-value: {p_value}")
alpha = 0.005

# Interpret results
if p_value < alpha:
    print("The ratings distributions of 'Home Alone (1990)' and 'Finding Nemo (2003)' are significantly different.")
else:
    print("The ratings distributions of 'Home Alone (1990)' and 'Finding Nemo (2003)' are NOT significantly different.")

# Draw the graph
plt.figure(figsize=(10, 6))
sns.histplot(home_alone_ratings, label='Home Alone', kde=True, color='royalblue', stat='density', bins=20, alpha=0.4, edgecolor=
sns.histplot(finding_nemo_ratings, label='Finding Nemo', kde=True, color='coral', stat='density', bins=20, alpha=0.4, edgecolor=

plt.xlabel('Rating', fontsize=12)
plt.ylabel('Density', fontsize=12)
plt.title("Ratings Distribution for 'Home Alone' and 'Finding Nemo'", fontsize=14, fontweight='bold')
plt.legend(title="Movies", fontsize=10, title_fontsize='12')
plt.grid(visible=True, linestyle='--', alpha=0.5)
plt.xticks(fontsize=10)
plt.yticks(fontsize=10)
plt.tight_layout()

plt.show()
```

```
p-value: 6.379397182836346e-10
The ratings distributions of 'Home Alone (1990)' and 'Finding Nemo (2003)' are significantly different.
```



10) There are ratings on movies from several franchises ([‘Star Wars’, ‘Harry Potter’, ‘The Matrix’, ‘Indiana Jones’, ‘Jurassic Park’, ‘Pirates of the Caribbean’, ‘Toy Story’, ‘Batman’]) in this dataset. How many of these are of inconsistent quality, as experienced by viewers? [Hint: You can use the keywords in quotation marks featured in this question to identify the movies that are part of each franchise] Extra Credit: Tell us something interesting and true (supported by a significance test of some kind) about the movies in this dataset that is not already covered by the questions above [for 5% of the grade score].

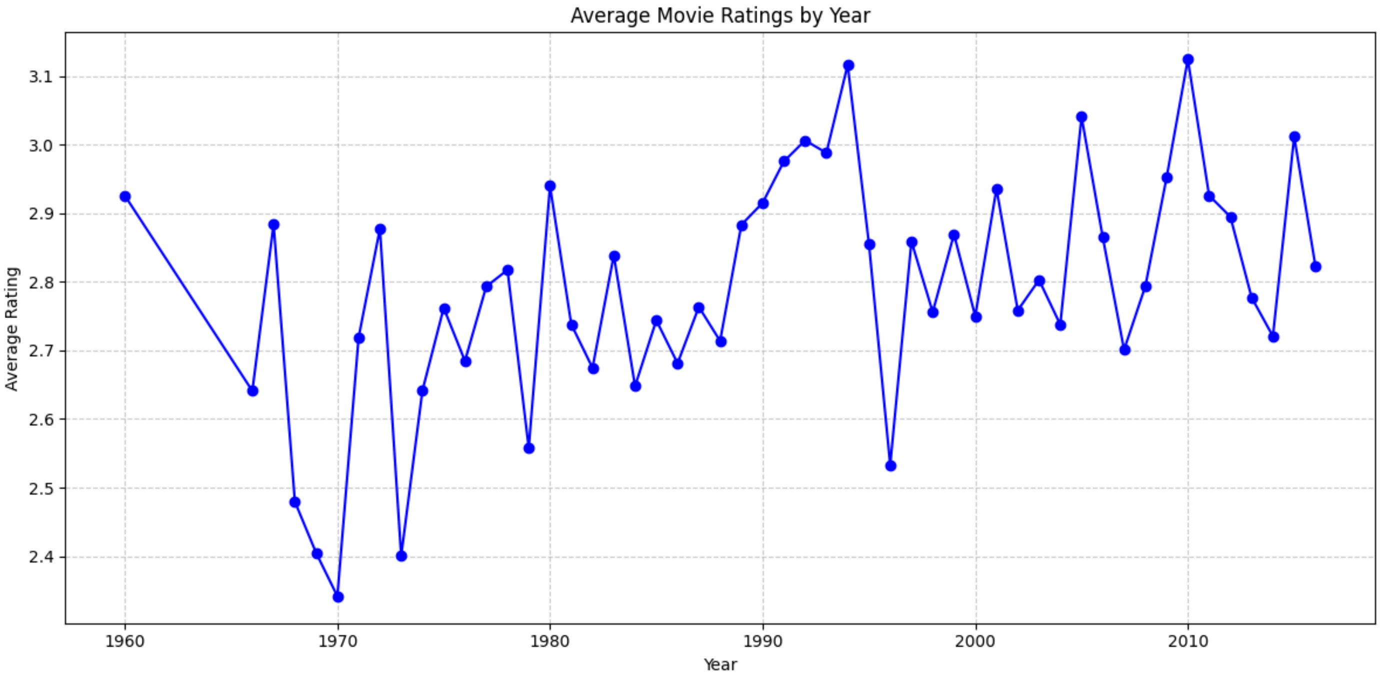
```
# Define franchise keywords
franchises = ['Star Wars', 'Harry Potter', 'The Matrix', 'Indiana Jones',
              'Jurassic Park', 'Pirates of the Caribbean', 'Toy Story', 'Batman']

# Significance threshold
alpha = 0.005
inconsistent_franchises_count = 0

# Check each franchise
for franchise in franchises:
    franchise_cols = [col for col in df.columns[:400] if franchise in col]

    if franchise_cols and len(franchise_cols) > 1:
        # drop element-wise
        franchise_ratings = [df[movie].dropna() for movie in franchise_cols]

        # Perform Kruskal-Wallis test
        stat, p_value = kruskal(*franchise_ratings)

        # Check p value vs. alpha
        if p_value < alpha:
            inconsistent_franchises_count += 1
            print(f"{franchise} has inconsistent quality with p-value {p_value:.4e}")

print(f"Number of inconsistent franchises: {inconsistent_franchises_count}")
```

```
Star Wars has inconsistent quality with p-value 8.0165e-48
The Matrix has inconsistent quality with p-value 3.1237e-11
Indiana Jones has inconsistent quality with p-value 6.2728e-10
Jurassic Park has inconsistent quality with p-value 7.6369e-11
Pirates of the Caribbean has inconsistent quality with p-value 3.2901e-05
Toy Story has inconsistent quality with p-value 5.0658e-06
Batman has inconsistent quality with p-value 4.2253e-42
Number of inconsistent franchises: 7
```

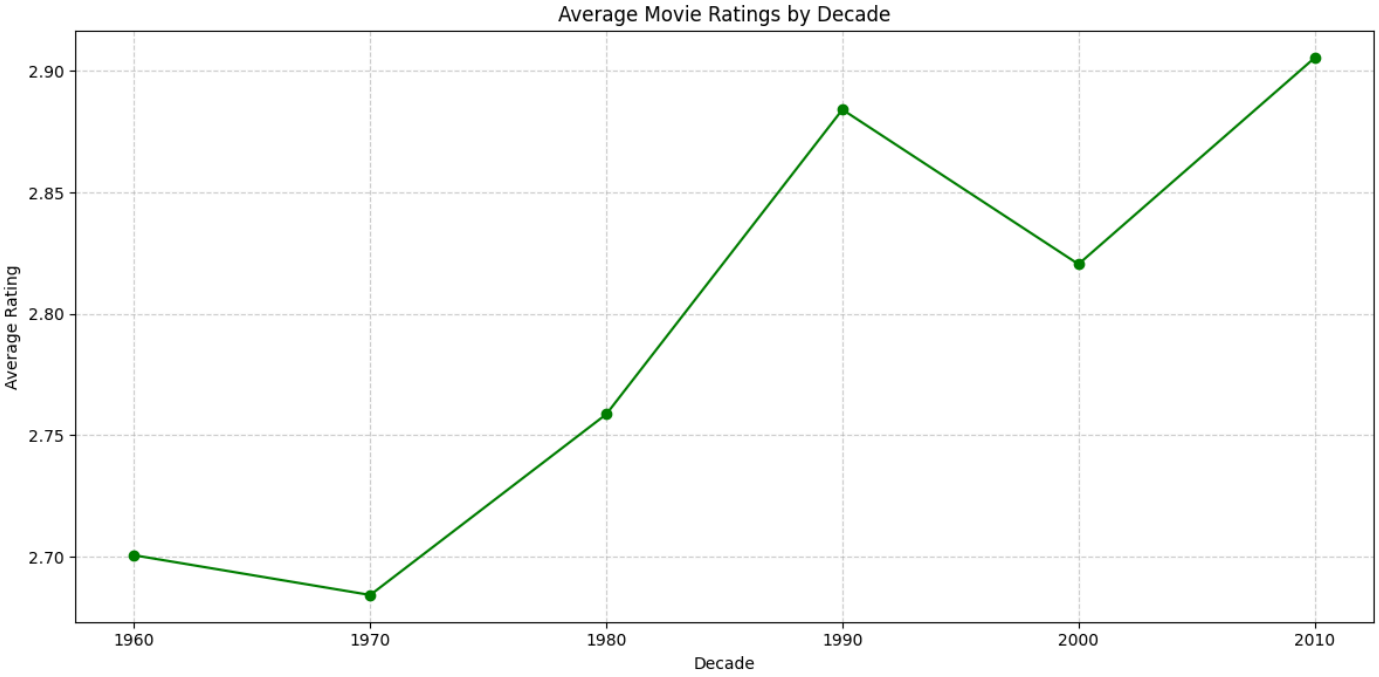
Extra Finding

We are curious how the ratings change across the years

```
ratings_by_year = {}
for col in df.columns[:400]:
  # Extract the year
    match = re.search(r'\((\d{4})\)', col)
    if match:
        year = int(match.group(1))
        if year not in ratings_by_year:
            ratings_by_year[year] = []
        ratings_by_year[year].append(df[col].dropna())  # Append non-missing ratings for each year

# Calculate the average rating per year
years = sorted(ratings_by_year.keys())
avg_ratings = [pd.concat(ratings_by_year[year]).mean() for year in years]

# Plot the trend of average ratings over years
plt.figure(figsize=(12, 6))
plt.plot(years, avg_ratings, marker='o', color='b', linestyle='-', linewidth=1.5)
plt.xlabel("Year")
plt.ylabel("Average Rating")
plt.title("Average Movie Ratings by Year")
plt.grid(visible=True, linestyle='--', alpha=0.6)
plt.tight_layout()
plt.show()
```



```
# Now group years into decades
ratings_by_decade = {}
for year in ratings_by_year:
# So, for a movie in 2005, it is one of the movies in 2000's
    decade = (year // 10) * 10
    if decade not in ratings_by_decade:
        ratings_by_decade[decade] = []
    ratings_by_decade[decade].extend(ratings_by_year[year])

# Calculate the average rating per decade
decades = sorted(ratings_by_decade.keys())
avg_ratings_decade = [pd.concat(ratings_by_decade[decade]).mean() for decade in decades]

# Plot the trend of average ratings by decade
plt.figure(figsize=(12, 6))
plt.figure(figsize=(12, 6))
plt.plot(decades, avg_ratings_decade, marker='o', color='g', linestyle='-', linewidth=1.5)
plt.xlabel("Decade")
plt.ylabel("Average Rating")
plt.title("Average Movie Ratings by Decade")
plt.grid(visible=True, linestyle='--', alpha=0.6)
plt.tight_layout()
plt.show()
```


Average Movie Ratings by Decade
Average Rating
Decade