Data Analysis Project 1

Question 1:
D: We first split our dataset into two groups(high/low popularity) over the median of the number of ratings, and we did an one-sided independent two-sample t-test. We made the assumptions that 1.The two groups are independent 2. Homoscedasticity 3. Normality in our data
Y: We believe assumption 1 and 2 are not violated because we do not think our simple split would introduce dependence relationships on ratings nor create a difference in variance among the groups. Finally we checked for normality, and based on the Q-Q Plot results in [Figure 1.1], we see that our data indeed follows normality decently except for the tails. Given we think the above assumptions are reasonable, the independent two-sample t-test is a suitable statistical test for comparing the means of our samples, and given its a parametric test, it's more powerful than alternative options like permutation tests. Also, even though the ratings(0-4) are ordinal and not intrinsically numerical, we think we can still treat it to be numerical if all we want is a relative comparison of the ratings(more/less popular).
F: We find a t-stat of 17.756 with a corresponding one-sided p-value of 1.13e-52.
Our t-stat represents the difference in mean ratings in units of standard deviation, and we calculated this with a degree of freedom = n1+n2-2 = 200+200-2 = 398. We also plotted out the distributions [Figure 1.2] and we can see that the distribution of group "high popularity" is indeed right shifted versus "low popularity".
A: Given that our p-value of 2.26965e-52 is much less than our significance level of 0.005, we reject the null hypothesis and conclude that there is a statistically significant difference in the average ratings of high popularity movies compared to low popularity movies, with the positive t-statistic indicates that high popularity movies have a higher average rating than low popularity movies.

Question 2:
D: We did a median split of release year into movie groups "old" and "new", and we did a two-sided independent two-sample t-test. We made the assumptions that 1.The two groups are independent 2. Homoscedasticity 3. Normality in our data
Y: We believe our samples are independent because they are split by release years, which really should not introduce any dependence in ratings, but we have quantified and tested the remaining two assumptions since they are not intuitively obvious. For homoscedasticity, we visually represented the spread with a boxplot [Figure 2.1] and since the boxes are approximately the same size and shape, this suggests that the homoscedasticity assumption is reasonable. For normality, we again plot a Q-Q plot[Figure2.2] and we observe that the data is overall normal with deviations in the tails. However, given the deviation is not drastic, and it is safe to assume normality in this case. The independent two-sample t-test is chosen as a suitable statistical test for comparing the means of two independent samples and is more powerful than non-parametric alternatives like permutation tests. Also, even though the ratings(0-4) are ordinal and not intrinsically numerical, we think we can still treat it to be numerical if all we want is a relative comparison of the ratings(more/less popular).
F: We find a t-stat of 1.605 with the corresponding p-value of 0.109. The t-statistic represents the difference in mean ratings between new and old movies in units of standard deviation. We calculated this with a degree of freedom = n1 + n2 - 2 = 203 + 197 - 2 = 398. We also plotted the distribution[Figure2.3] and can see that there is a big overlap between the two groups, suggesting that the ratings are similar.
A: Given that our p-value of 0.109 is greater than our significance level of 0.005, we fail to reject the null hypothesis and conclude that there is not a statistically significant difference in the average ratings of new and old movies, suggesting that the movie age does not have a significant impact on its average rating.

Question 3:
D: We split our data for Shrek into two groups based on gender, then conducted an independent two-sample t-test to compare the average ratings of male and female groups. We made the assumptions that 1.The two groups are independent 2. Homoscedasticity 3. Normality in our data
Y: We can assume that the two groups are independent because the gender of the viewer does not influence the ratings given by viewers of the other gender. From the boxplot[Figure 3.1], we can observe that the interquartile ranges for both groups are quite similar, which suggests that the variances of the two groups are approximately equal. Therefore, we have reasonable grounds to assume homoscedasticity.

Finally from the Q-Q plots[figure 3.2], we can see that both male and female ratings roughly follow a straight line, especially in the middle of the distribution, which suggests that the data is approximately normally distributed. The independent two-sample t-test is a suitable statistical test for comparing the means of two independent samples to determine if there is a statistically significant difference between them, as it provides more power as a parametric test compared to alternatives like permutation tests. Again, even though the ratings(0-4) are ordinal and not intrinsically numerical, we think we can still treat it to be numerical if all we want is a relative comparison of the ratings(if there exists a difference).
F: We found a t-statistic of -1.10 with a corresponding p-value of 0.27. The t-statistic provides a standardized measure of how far apart the sample means of our two groups are. We calculated it with df = n1 + n2 - 2 = 241 + 743 - 2 = 982. We also plotted the distribution[Figure 3.3] and can see that there is a big overlap of distribution patterns between the two groups, suggesting that the ratings are similar in distribution.
A: Given that our p-value of 0.27 is greater than our significance level of 0.005, we fail to reject the null hypothesis and conclude that there is not a statistically significant difference in the average ratings of 'Shrek (2001)' given by male and female viewers, and it is not gendered. We do have limitations in our assumptions as we assumed normality of our data based on the Q-Q plots. However, our data takes discrete values and is less strictly continuous, and therefore does not strictly meet the assumptions of normality. This is a limitation of our analysis, and it is important to be aware of it, as well as the tradeoff between more power from our parametric independent two-sample t-test versus less assumptions(of normality) needed for non-parametric options like Mann-Whitney U tests, which is also more suitable for ordinal data.

Question 4:
D: We performed an independent two-sample t-test for each movie to compare the average ratings given by male and female viewers. We then calculated the proportion of movies that had a statistically significant difference in average ratings between the two groups. We calculated results for both our regular alpha and adjusted our significance level to 0.005 / 400 = 0.0000125 using the Bonferroni correction to account for multiple comparisons.
Y: We chose the independent two-sample t-test because we are comparing the average ratings of two independent groups (male and female viewers), and we acknowledge that even though all three assumptions of independence, normality, homoscedasticity might not necessarily hold for each data group, the t-test is empirically known to be robust against violations of assumptions and we will use this test for such robustness and high statistical power, which we believe is crucial in multiple testing. For each movie, we also assumed that its ratings are continuous numerical data which can be reduced to means. The Bonferroni correction was used to control for the familywise error rate due to the large number of tests performed and to make sure that we are not rejecting simply by chance.
F: We found that without Bonferroni correction, approximately 11.5% (46 out of 400) of the movies had a statistically significant difference in average ratings between male and female viewers, whereas with Bonferroni correction, approximately 2.75% of the movies (11 out of 400) had a statistically significant difference.
A: We believe that when doing 400 tests, we should really be using Bonferroni correction. Given that our Bonferroni adjusted p-value, only 2.75% of the movies had a statistically significant difference in average ratings between male and female viewers, which suggests that gender probably does not play a significant role in how viewers rate movies.

Question 5:
D: We conducted an independent two-sample t-test to compare the average ratings of "The Lion King (1994)" between two groups of interest. We specified the 'greater' alternative in the t-test to perform a one-sided test, as we were interested in testing if people who are only children enjoy the movie more than people with siblings. We made the assumptions that 1. the two groups are independent, 2. the ratings are normally distributed within each group, and 3. homoscedasticity of variance of the two groups.
Y: We believe the assumption of independence holds in this case, as being an only child or having siblings does not affect another person's sibling status or their enjoyment of "The Lion King (1994)." To validate our other assumptions, we will need to check for homoscedasticity and normality. For homoscedasticity, we will perform Levene's test, and with a resulting Levene test statistic of 1.300 and Levene p-value of 0.254>0.005, we fail to reject the null and concludes that variances are

equal(homoscedastic) between the two groups. To check normality we again plot the Q-Q plot[Figure5.1], and we see that our data deviates from the normal distribution in the tails, but we will rely on the robustness of the t-test(and the fact that other assumptions hold), and we do not think it will significantly bias our results while providing high statistical power as a parametric method.
F: We conducted a one-sided independent two-sample t-test and found a t-statistic of -2.045 with a corresponding p-value of 0.9794. The negative t-statistic indicates that the average rating of "The Lion King (1994)" is lower for people who are only children compared to people with siblings. The degrees of freedom for this test is the sum of the sample sizes of the two groups minus 2, which is df = n1 + n2 − 2 = 151 + 786 - 2 = 935.
A: Given that our p-value of 0.9794 is greater than our significance level of 0.005, we do not have enough statistical evidence to reject the null hypothesis. Therefore, we conclude that people who are only children do not enjoy "The Lion King (1994)" more than people with siblings. However, it is important to note that our data consists of discrete values (ratings from 0 to 4) which we treated as continuous numerical values, and the Q-Q plots show some deviation from normality in the tails. While we relied on the empirical robustness of t-tests, this deviation is still a limitation of our analysis and may affect the validity of our t-test. In many cases where the normality assumption is violated, non-parametric tests such as the Mann-Whitney U test may be more appropriate.

Question 6:
D: We iterate over the 400 movies, and in each iteration, we run two-sided Mann-Whitney U test between two groups: the non-null movie ratings for this movie from viewers with siblings v.s. those without. We assume the two groups for each movie are independent, not normally distributed, non-categorical, and their sample means are not meaningful. For each movie, we make a null hypothesis that there is no difference between the two groups' movie ratings and we get the p-value from the Mann-Whitney U test. Then we count the number of times that the p-values have been lower than 0.005, and divide this number by the number of movies(400).
Y: It is unreasonable to reduce the movie ratings' datasets to their means, because the psychological distance between movie ratings from 0 to 1, 1 to 2, and so on are not identical. The data is non-categorical because movie ratings are not categories. We assume the data is not normally distributed because we are running the same test for so many (400) pairs of groups of data of different sizes and distributions and most of them are not normally distributed according to the result of normaltest from the Scipy package. We assume the two groups are independent because they came from different viewers (there is no overlap between viewers who have siblings and those without). Hence, we decided to use Mann-Whitney U test, which does not require normal distribution and meaningful sample means, and it is valid for testing independent groups of data which are non-categorical.
F: We found that in the 400 movies, there appears to be 7 movies that resulted in a p-value under 0.005 from their Mann-Whitney U tests, which means 7 of the 400 test results have rejected the null hypothesis that there is no difference between the two groups. Then we divide 7 by 400, and we get our result that 1.75% of the movies exhibit an "only children effect".
A: 1.75% of the movies exhibit "only children effect", i.e. are rated differently by viewers with siblings v.s. those without.

Question 7:
D: We dropped null values from the movie ratings of 'The Wolf of Wall Street (2013)' and then ran an one-sided Mann-Whitney U test on two groups: people prefer to watch movies alone v.s. those not. We assume the two groups of data are independent, not normally distributed, non-categorical, and their means are not meaningful. Our null hypothesis in the Mann-Whitney U test is that the movie ratings from viewers who like to watch movies socially is not greater than the movie ratings from those who prefer to watch alone.
Y: It is valid to assume these two groups are independent because their data come from different viewers; it is valid to assume that the data is non-categorical because movie ratings are not categories; it is valid to assume that these two groups' data are not normally distributed, because when we run normal test on both of the groups from Scipy package, the p-value is 1.33e-11<0.005 for the group from viewers who like to watch movie socially and 1.054e-19<0.005 for the group from viewers who like to watch movie along, which means both of them rejected the Scipy normal test null hypothesis that the group of data is normally distributed. It is also valid to assume their means are not meaningful, since the psychological

distance between adjacent movie ratings are not the same. Hence, we decided to use Mann-Whitney U test, which does not require normal distribution and meaningful sample means.
F: The p-value we obtained from Mann-Whitney U test is 0.944>0.005, so we failed to reject the null hypothesis. Hence, we conclude that the movie ratings from viewers who like to watch movies socially is not greater than the movie ratings from those who prefer to watch alone
A: People who like to watch movies socially do not socially enjoy 'The Wolf of Wall Street (2013)' more than those who prefer to watch them alone.

Question 8:
D: We iterate over the 400 movies, and in each iteration, we run two-sided Mann-Whitney U test between two groups: the non-null movie ratings for this movie from viewers who like to watch movies socially v.s. those who prefer to watch alone. We assume the two groups for each movie are independent, not normally distributed, non-categorical, and their sample means are not meaningful. For each movie, we make a null hypothesis that there is no difference between the two groups' movie ratings and we get the p-value from the Mann-Whitney U test. Then we count the number of times that the p-values have been lower than 0.005, and divide this number by the number of movies(400).
Y: It is unreasonable to reduce the movie ratings' datasets to their means, because the psychological distance between movie ratings are not identical. The data is non-categorical because movie ratings are not categories. We assume the data is not normally distributed because we are running the same test for so many (400) pairs of groups of data of different sizes and distributions and most of them are not normally distributed according to the result of normaltest from the Scipy package. We assume the two groups are independent because they came from different viewers (there is no overlap between viewers who prefer to watch socially and those who do not). Hence, we decided to use Mann-Whitney U test, which does not require normal distribution and meaningful sample means, and it is valid for testing independent groups of data which are non-categorical.
F: We found that in the 400 movies, there appears to be 10 movies that resulted in a p-value under 0.005 from their Mann-Whitney U tests, which means 10 of the 400 test results have rejected the null hypothesis that there is no difference between the two groups. Then we divide 10 by 400, and we get our result that 2.5% of the movies exhibit a "social watching effect".
A: 2.5% of the movies exhibit "social watching effect", i.e. are rated differently by viewers who prefer to watch movies socially v.s. those who do not.

Question 9:
D: We dropped null values from the movie ratings of 'Home Alone (1990)' and 'Finding Nemo (2003)', and then ran an Kolmogorov-Smirnov test for these two movies' ratings to test if their ratings are distributed differently. Our null hypothesis is that the ratings from Home Alone and Finding Nemo have the same underlying distribution.
Y: We use the Kolmogorov-Smirnov test because it can test whether the underlying distributions from two groups of data (two movies' ratings in this case) are the same. We also plotted a distribution comparison[Figure9.1], but we see that it is not visually obvious just from the histogram.
F: The p-value we obtained from the Kolmogorov-Sminov test is 6.38e-10<0.005, so we reject the null hypothesis that the ratings from Home Alone and Finding Nemo have the same underlying distribution.
A: The ratings distribution of 'Home Alone (1990)' is different from that of 'Finding Nemo (2003)'.

Question 10:
D: For each of the franchises, we only select the movie ratings from the viewers who have rated all of the movies within this franchise. Then we run Friedman test for those movie ratings, checking how many of the franchises are of inconsistent quality, i.e. how many of the franchises have their movies been rated differently by viewers. Our null hypothesis is that the franchise has its movies rated the same by viewers. We assume that the movie ratings are non-categorical data, it is unreasonable to reduce these data to their sample means, the samples are dependent, groups are random samples from population, and those dependent variables are measured at the ordinal or continuous level.
Y: We decided to use Friedman test for those franchises because it can test more than two groups and, unlike Kruskal-Wallis test or ANOVA, it does not require each group to be independent from others; instead, it assumes each group to be dependent. It is valid to assume that the groups are dependent, because we are testing the movie ratings given by the same group of viewers for each franchise, as we

have selected only the movie ratings from the viewers who have rated all of the movies within the franchise to ensure fairness. Moreover, movie ratings are non-categorical and it is unreasonable to reduce move ratings to their sample means, and the movie ratings are random samples from population as the viewers are randomly selected; the dependent variables are measured at the ordinal level, as movie ratings are ordinal variables. Hence, all of the assumptions for Friedman test are not violated, and thus it is valid to use Friedman test for this type of dependent, repeated measures of ordinal data, despite it losing some statistical power versus its parametric alternative which is ANOVA.
F: The p-value we obtained from the Star Wars franchise is 3.67e-57<0.005, so we reject the null hypothesis that the Star War franchise has its movies rated the same by the viewers. The p-value we obtained from the Harry Potter franchise is 0.0012<0.005, so we reject the null hypothesis that the Harry Potter franchise has its movies rated the same by the viewers. The p-value for The Matrix franchise is 2.07e-15<0.005, so we reject the null hypothesis that the The Matrix franchise has its movies rated the same by the viewers. The p-value for The Indiana Jones franchise is 5.54e-18<0.005, so we reject the null hypothesis that the Indiana Jones franchise franchise has its movies rated the same by the viewers. The p-value for the Jurassic Park franchise is 3.12e-17<0.005, so we reject the null hypothesis that the Jurassic Park franchise has its movies rated the same by the viewers. The p-value for the Pirates of the Caribbean franchise is 1.77e-05<0.005, so we reject the null hypothesis that the Pirates of the Caribbean franchise has its movies rated the same by the viewers. The p-value for the Toy Story franchise is 5.84e-13<0.005, so we reject the null hypothesis that the Toy Story franchise has its movies rated the same by the viewers. The p-value for the Batman franchise is 5.58e-23<0.005, so we reject the null hypothesis that the Toy Story franchise has its movies rated the same by the viewers. As a result, all 8 of those franchises have their movies rated differently by the viewers.
A: All 8 of those franchises are of inconsistent quality.

Extra Credit:
Question: We may have heard about the stereotype that male viewers do like Batman movies more than female viewers do. Is this true according to your tests?
D: We run an one-sided Mann-Whitney U test on the movie ratings from the three Batman movies (containing Batman keywords in their titles) from male viewers v.s. from female viewers. We assume the variables are independent, data is ordinal (not meaningful to reduce to sample mean) and non-categorical. Our null hypothesis is male viewers and female viewers give the same ratings for the Batman movies, and our alternative hypothesis is that male viewers give higher ratings than female viewers for the Batman movies.
Y: We decided to use Mann-Whitney U test because it does not require normal distribution and meaningful sample means. It is valid to assume the data is ordinal because movie rating are ordinal variables; it is valid to assume variables are independent because viewers cannot report themselves as both male and female; it is valid to assume data is non-categorical because movie ratings are not categories. Hence, all of the assumptions are not violated, and iit is valid to use Mann-Whitney U test in this case.
F:The p-value we obtained from the Mann-Whitney U test is 0.119>0.005, so we reject the null hypothesis in favor of the alternative hypothesis: male viewers do give higher ratings than female viewers for the Batman movies. And the p-value is surprisingly higher than the set threshold of 0.005, i.e. more than 20 times higher.
A: The stereotype is true: male viewers do like Batman movies more than female viewers do.

Appendix

Figure1.1:

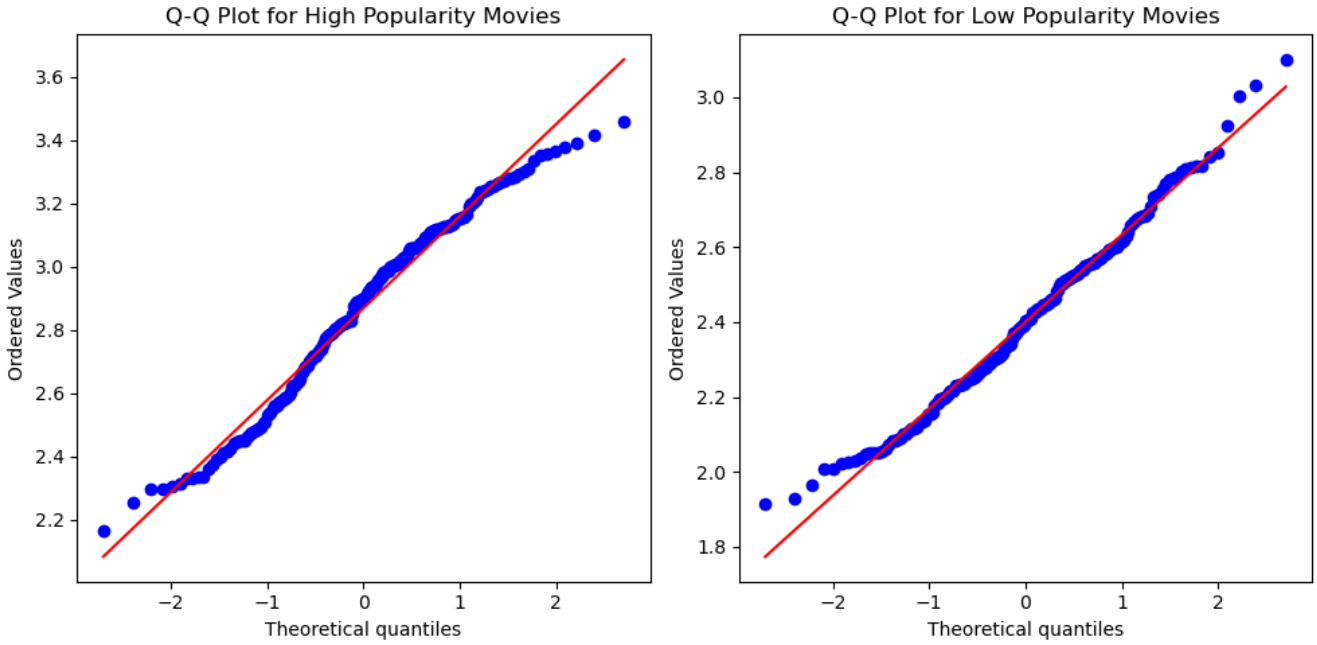


Figure1.2:

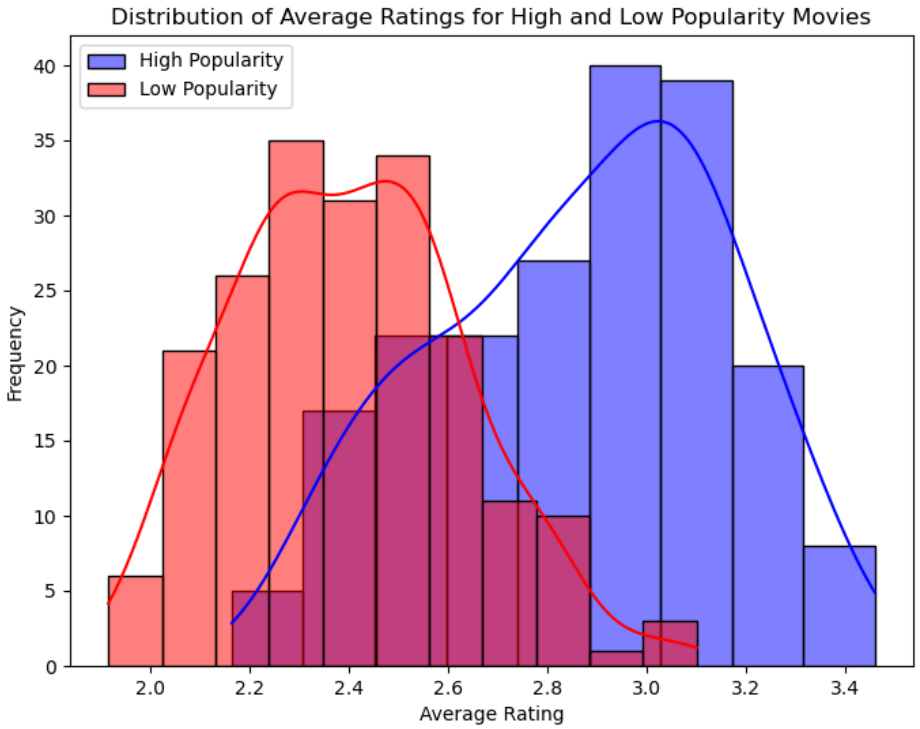


Figure2.1:

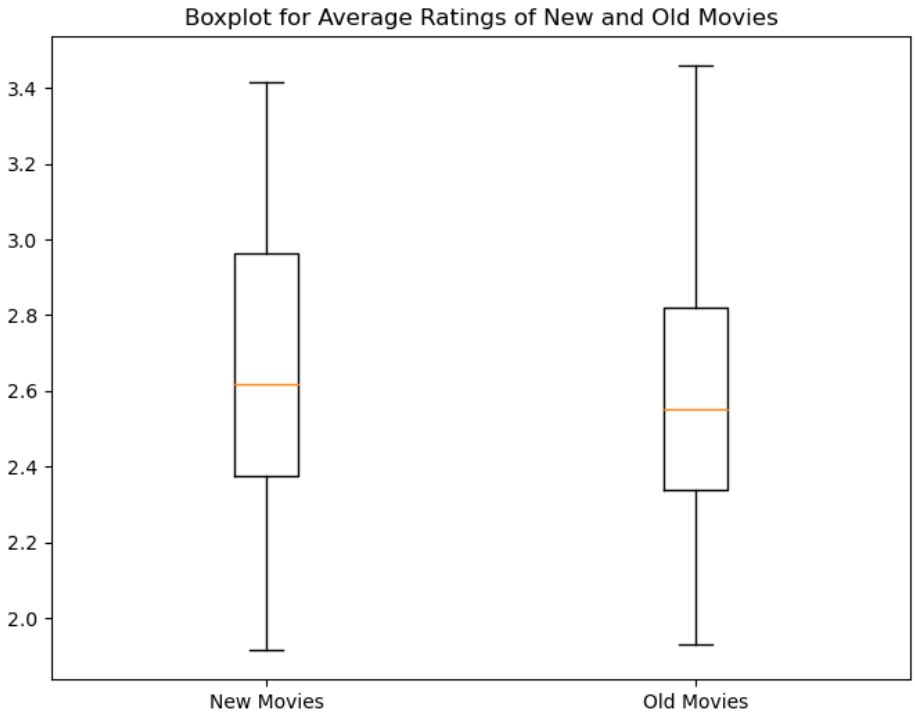


Figure2.2:

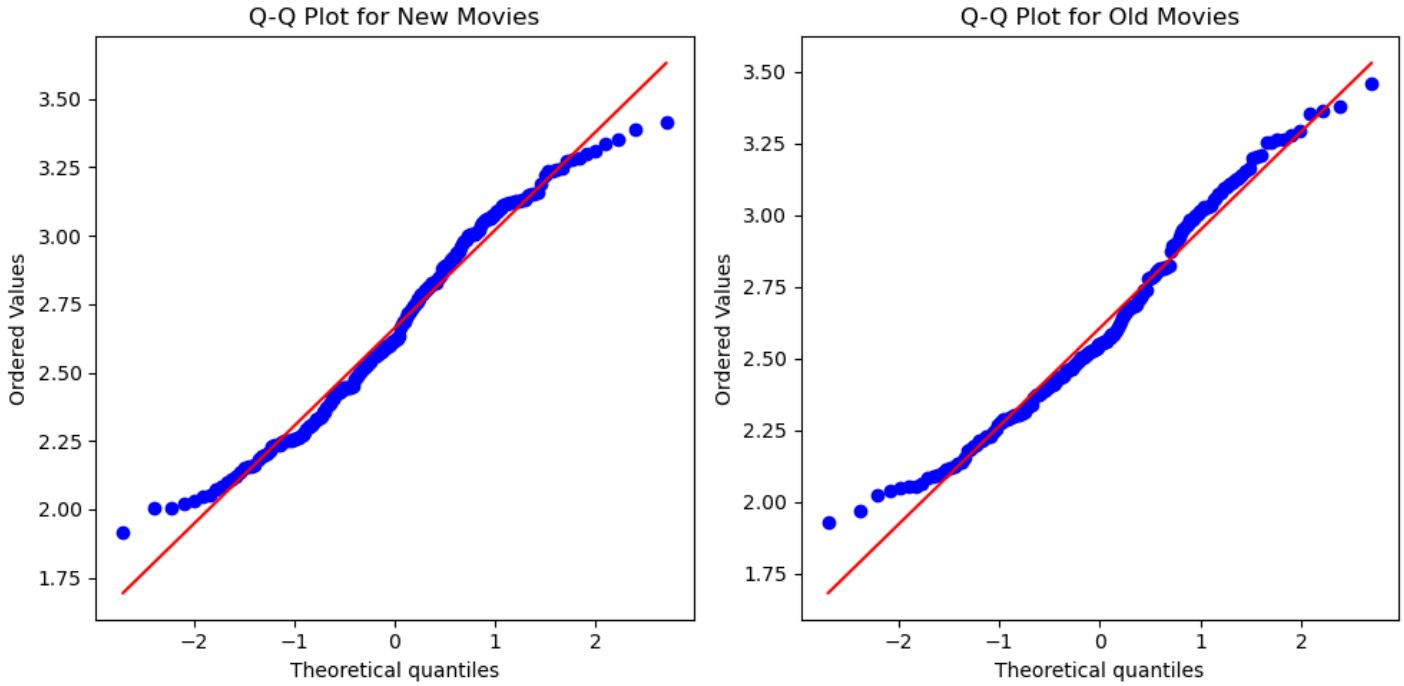


Figure2.3:

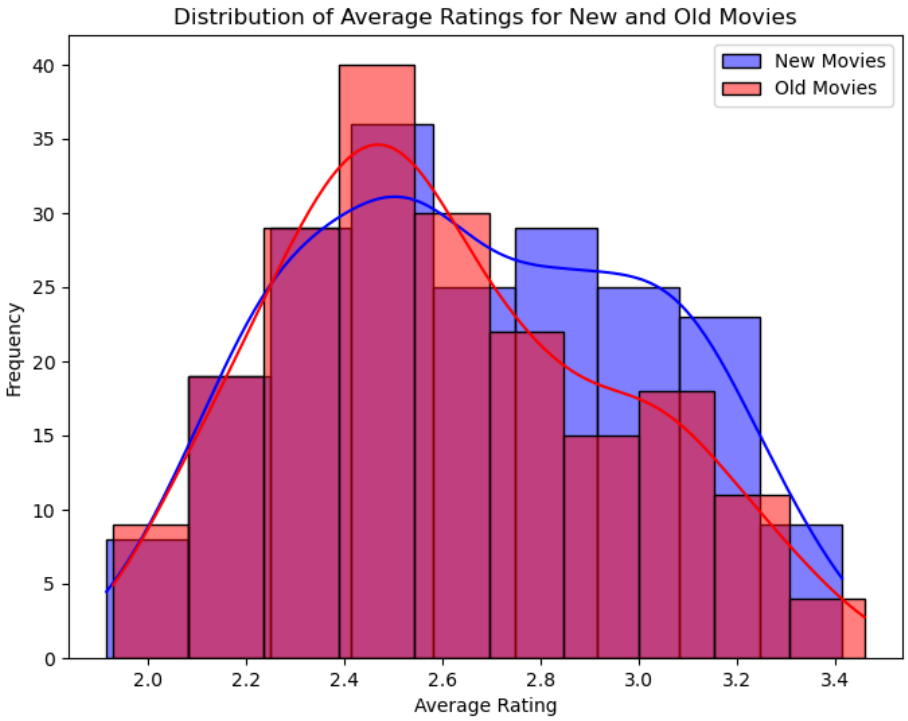


Figure3.1

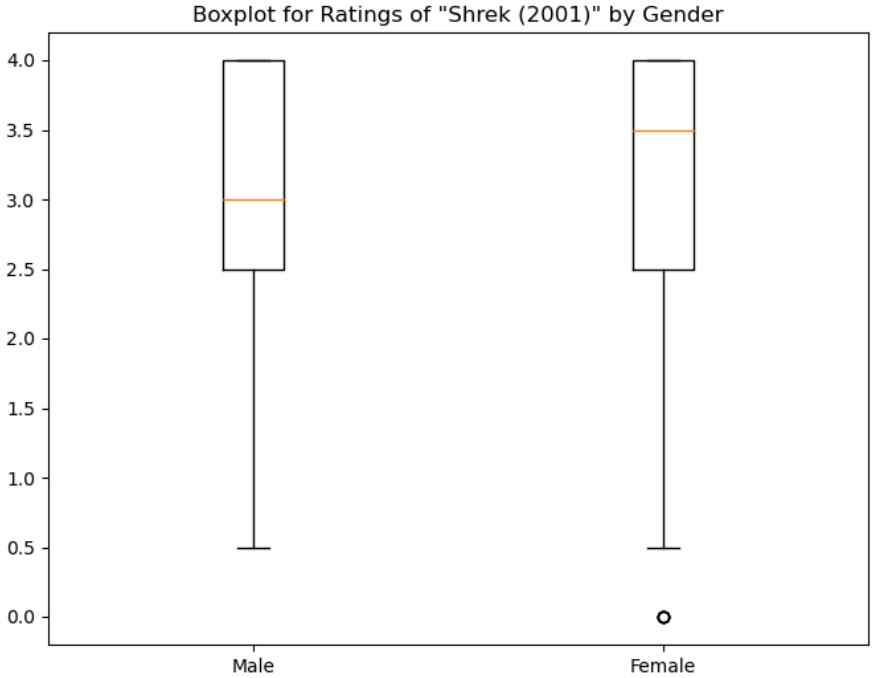


Figure3.2

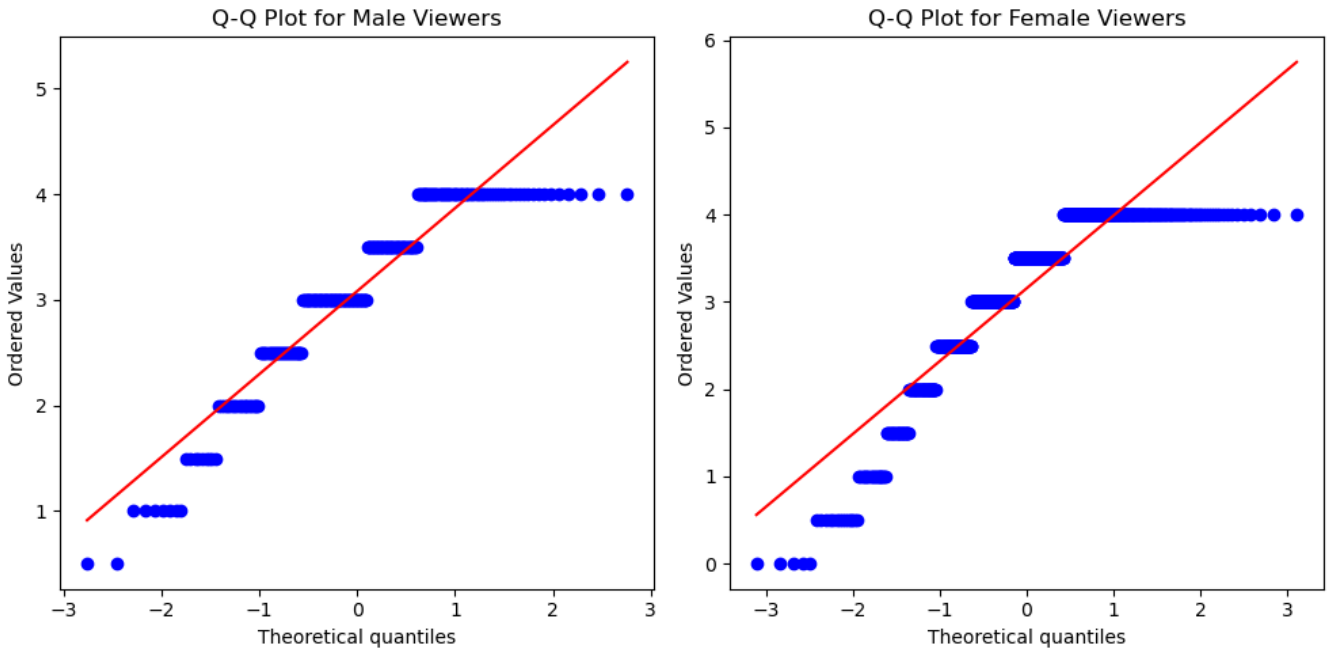


Figure3.3

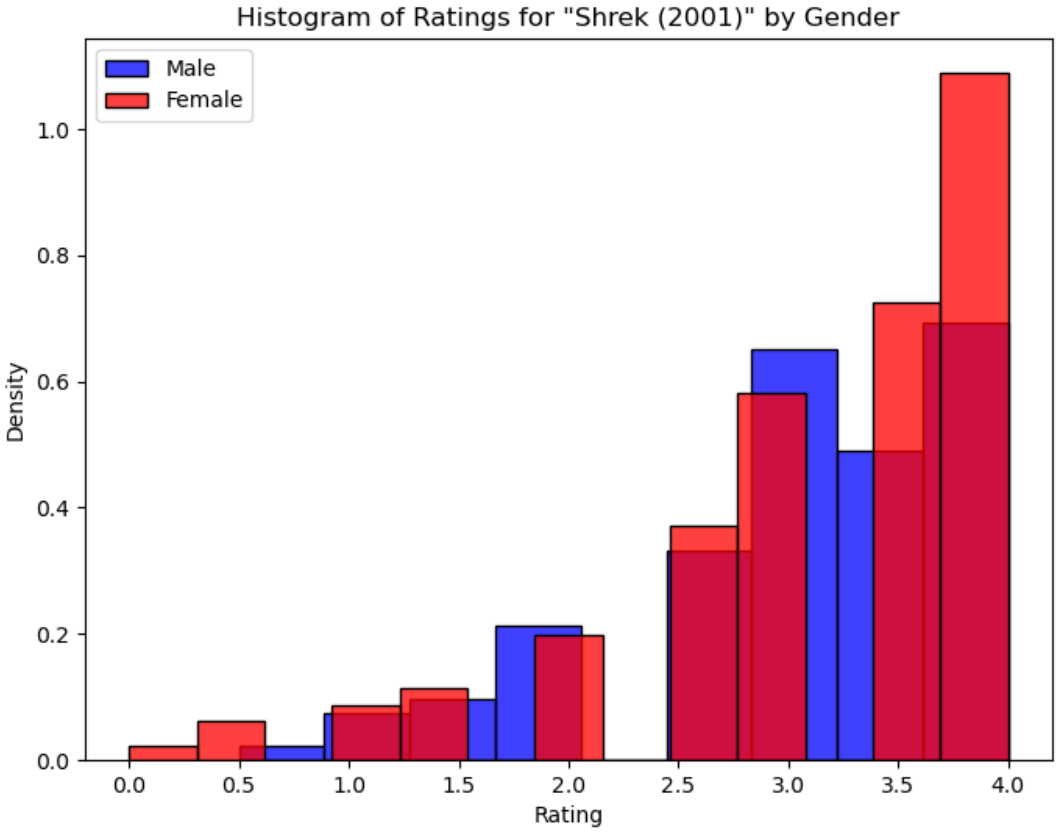


Figure 5.1

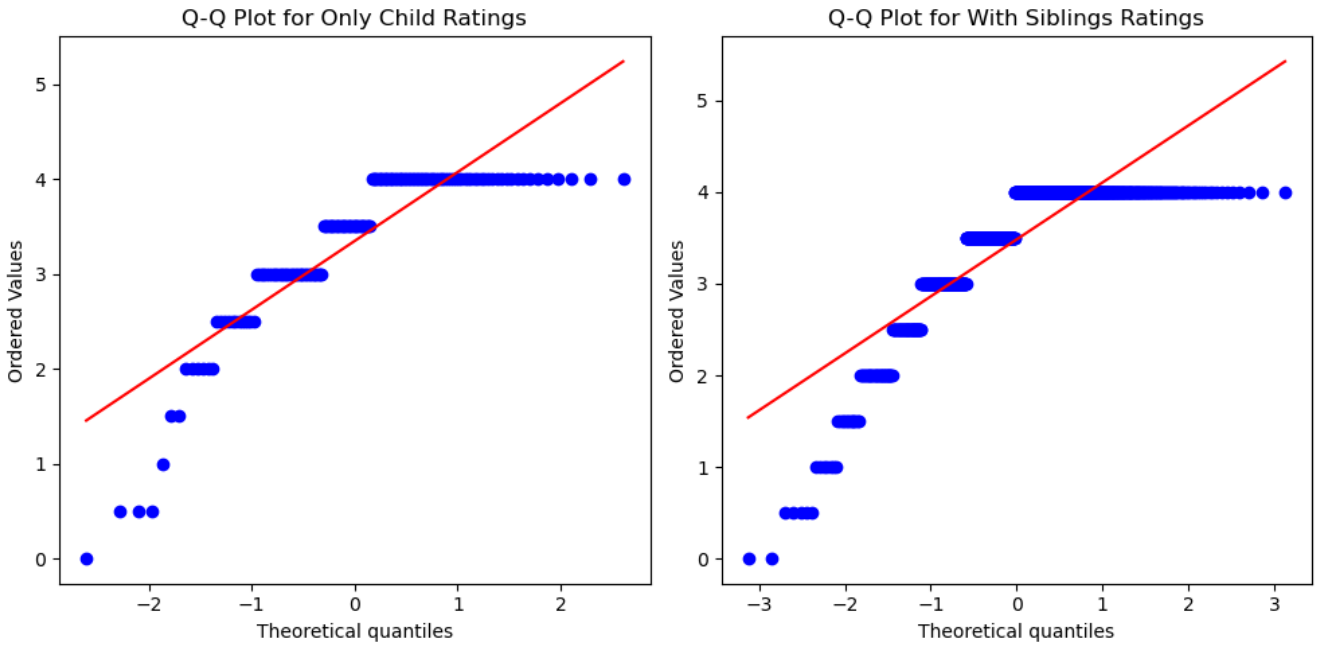
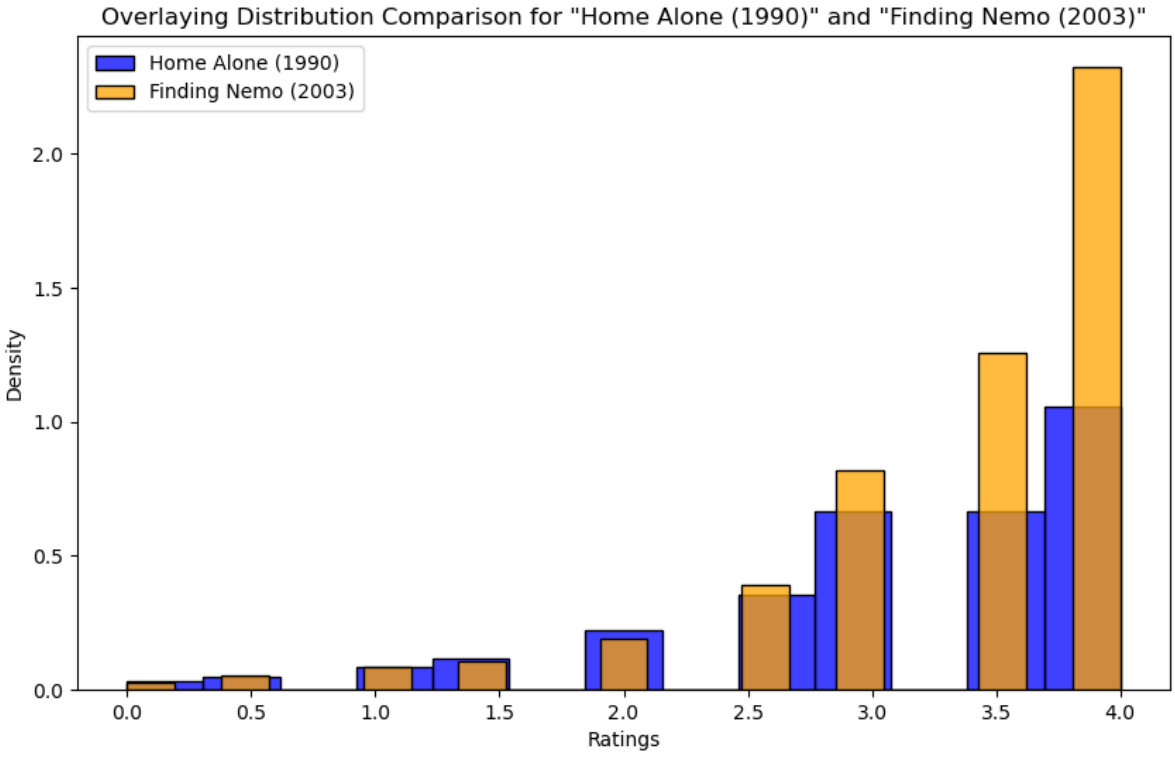


Figure 9.1



Code:
https://drive.google.com/drive/folders/1YR8QbEqBfGzLnxTcmg5c-qceL2PsX7Br?usp=sharing
PDF of the notebook is also attached at the end of this document

In [ ]:
```
import pandas as pd
import numpy as np
df = pd.read_csv("movieReplicationSet.csv")
```

In [ ]:
```
df.head(3)
```

Out[ ]:

	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)	Blac Swa (201
0	NaN	NaN	4.0	NaN	3.0	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	1.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

3 rows × 477 columns

In [ ]:
```
df.iloc[:, 474:].head(3)
```

Out[ ]:

	Gender identity (1 = female; 2 = male; 3 = self-described)	Are you an only child? (1: Yes; 0: No; -1: Did not respond)	Movies are best enjoyed alone (1: Yes; 0: No; -1: Did not respond)
0	1.0	0	1
1	1.0	0	0
2	1.0	1	0

Q1 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]

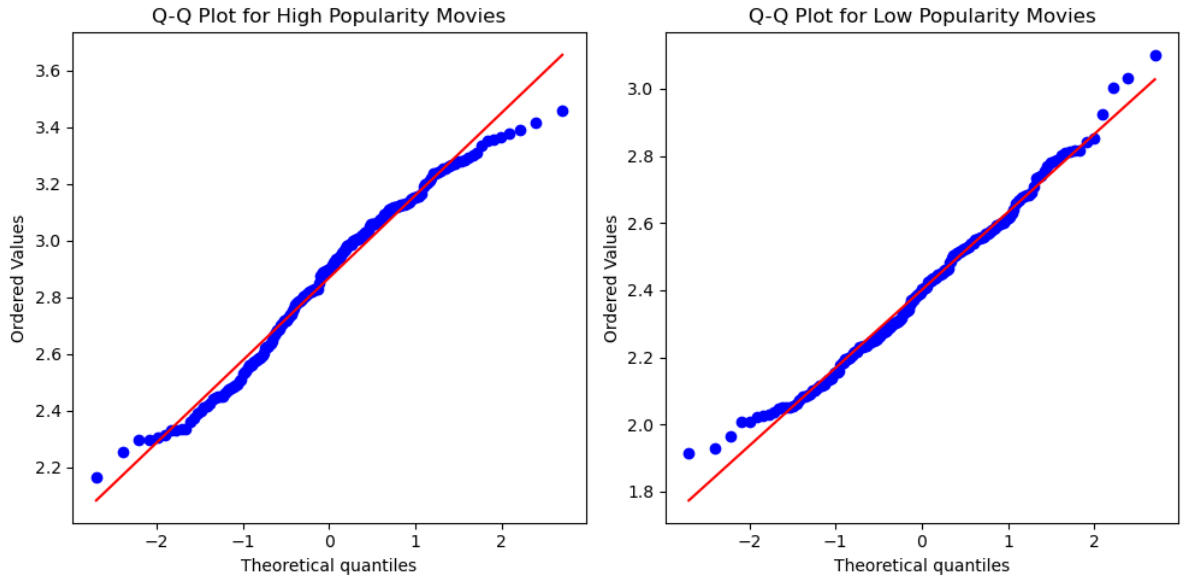
In [ ]:
```
median_ratings = df.iloc[:, :400].count(axis=0).median()
high_popularity_movies = df.iloc[:, :400].count(axis=0) >= median_ratings
low_popularity_movies = df.iloc[:, :400].count(axis=0) < median_ratings
high_popularity_indices = high_popularity_movies[high_popularity_movies].index
low_popularity_indices = low_popularity_movies[low_popularity_movies].index
high_df = df[high_popularity_indices]
low_df = df[low_popularity_indices]
low_df.head()
```

Out[ ]:

	The Life of David Gale (2003)	Wing Commander (1999)	Snatch (2000)	Rambo: First Blood Part II (1985)	Let the Right One In (2008)	The Machinist (2004)	Brazil (1985)	Change of Habit (1969)	Night of the Living Dead (1968)	Man on Fire (2004)
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	0.5	NaN	NaN	NaN	NaN	NaN	NaN

5 rows × 200 columns

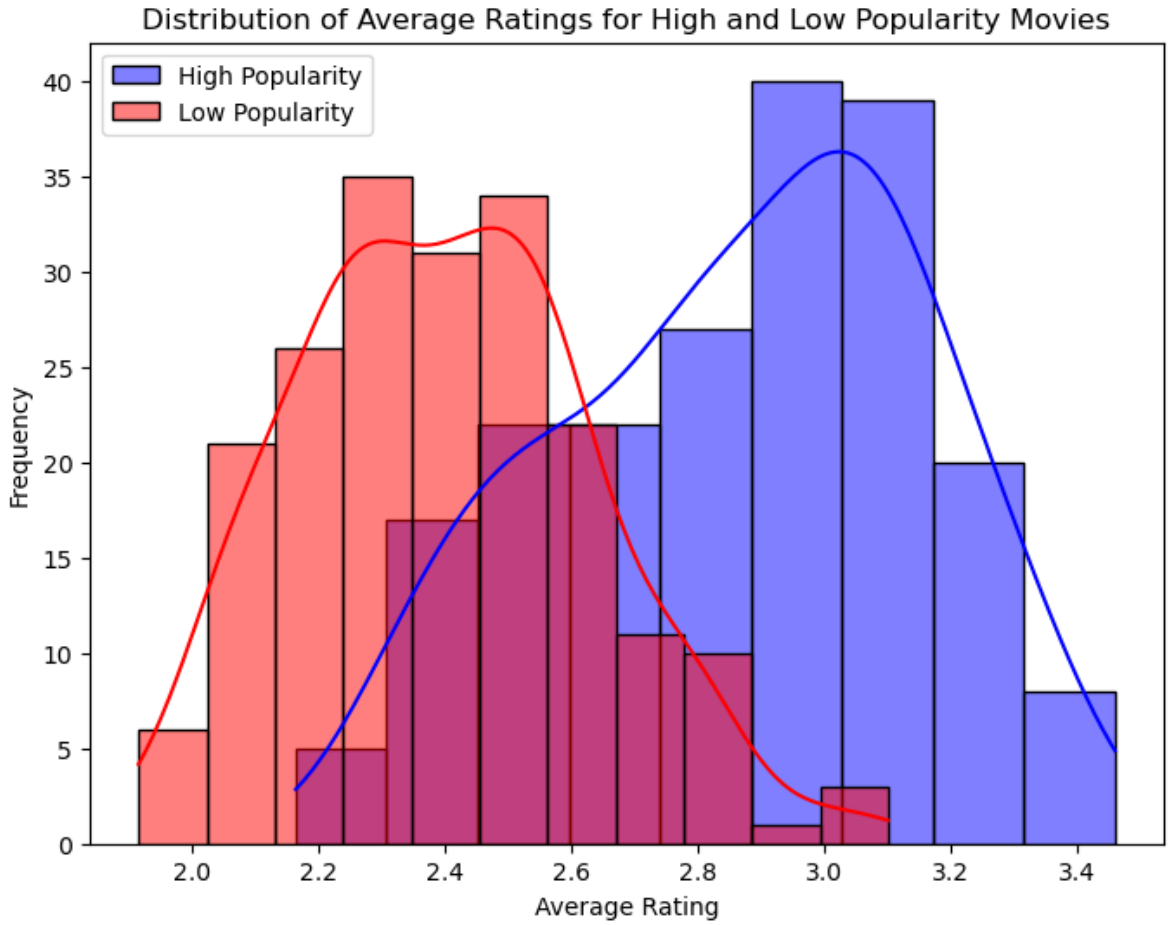
In [ ]:
```
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import probplot

fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(10, 5))

#high popularity movies
probplot(high_df.mean(axis=0).dropna(), plot=axes[0])
axes[0].set_title('Q-Q Plot for High Popularity Movies')
# low popularity movies
probplot(low_df.mean(axis=0).dropna(), plot=axes[1])
axes[1].set_title('Q-Q Plot for Low Popularity Movies')

plt.tight_layout()
plt.savefig('q1_qq_plots.png')
plt.show()
```



In [ ]:
```
from scipy.stats import ttest_ind
high_avg_ratings = high_df.mean(axis=0).dropna()
low_avg_ratings = low_df.mean(axis=0).dropna()

t_stat, p_value = ttest_ind(high_avg_ratings, low_avg_ratings)
t_stat, p_value
```

Out[ ]: (17.7560492698737, 2.2696530276564846e-52)

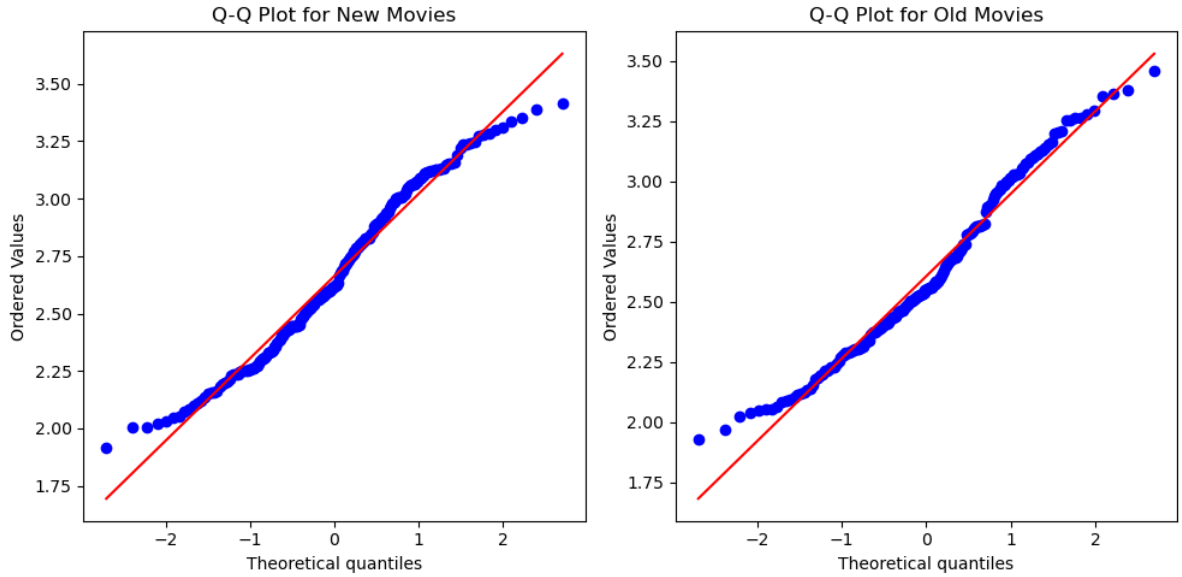
In [ ]:
```
fig, ax = plt.subplots(figsize=(8, 6))


sns.histplot(high_avg_ratings, kde=True, color='blue', label='High Pop
ularity', ax=ax)
sns.histplot(low_avg_ratings, kde=True, color='red', label='Low Popula
rity', ax=ax)

ax.legend()
ax.set_title('Distribution of Average Ratings for High and Low Popular
ity Movies')
ax.set_xlabel('Average Rating')
ax.set_ylabel('Frequency')
plt.savefig('q1_distribution_plots.png')
plt.show()
```



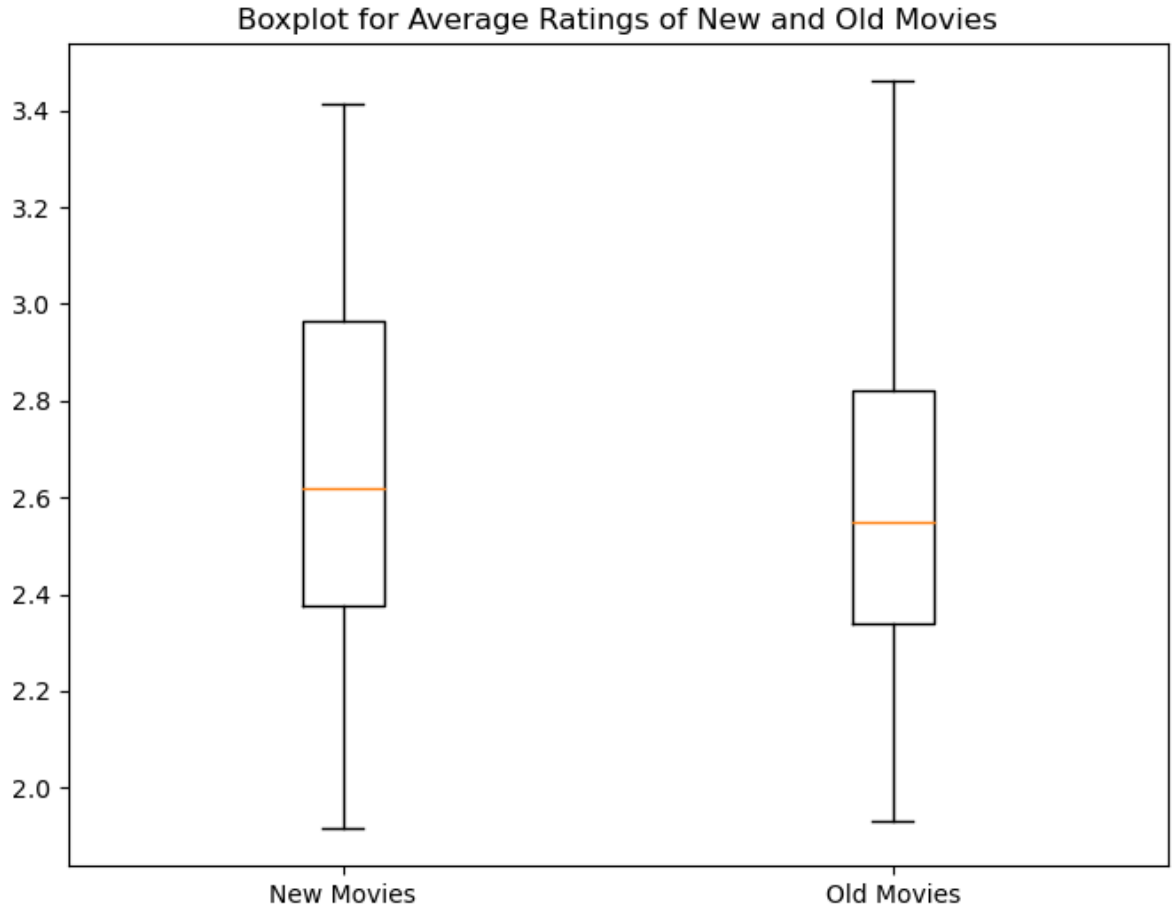
Question 2 Are movies that are newer rated differently than movies that are older? [Hint: Do a median split of year of release to contrast movies in terms of whether they are old or new]

In [ ]:
```
years = df.columns[:400].str.extract(r'\((\d{4})\)')[0].astype(float)
median_year = years.median()
new_movies = years >= median_year
old_movies = years < median_year


new_movies_indices = new_movies[new_movies].index
old_movies_indices = old_movies[old_movies].index
new_movies_titles = df.columns[new_movies_indices]
old_movies_titles = df.columns[old_movies_indices]
new_df = df[new_movies_titles]
old_df = df[old_movies_titles]
```

In [ ]:
```
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(10, 5))

probplot(new_df.mean(axis=0).dropna(), plot=axes[0])
axes[0].set_title('Q-Q Plot for New Movies')

probplot(old_df.mean(axis=0).dropna(), plot=axes[1])
axes[1].set_title('Q-Q Plot for Old Movies')
plt.tight_layout()
plt.savefig('q2_qq_plots.png')
plt.show()
```



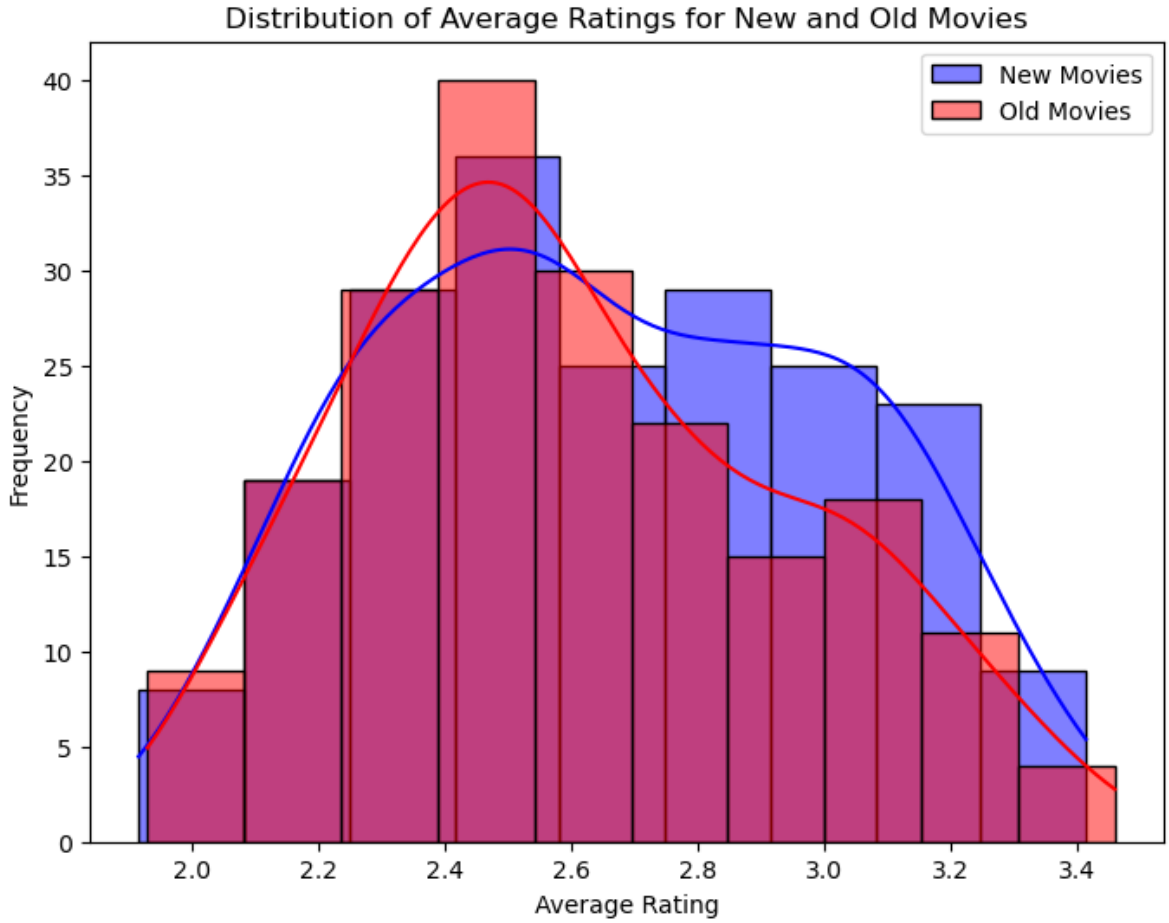
In [ ]:
```
# Boxplot for average ratings comparison
fig, ax = plt.subplots(figsize=(8, 6))

data = [new_df.mean(axis=0).dropna(), old_df.mean(axis=0).dropna()]
labels = ['New Movies', 'Old Movies']

ax.boxplot(data, labels=labels)
ax.set_title('Boxplot for Average Ratings of New and Old Movies')
plt.savefig('q2_boxplot.png')
plt.show()
```



In [ ]:
```
t_stat, p_value = ttest_ind(new_df.mean(axis=0).dropna(), old_df.mean
(axis=0).dropna())

t_stat, p_value
```

Out[ ]: (1.605479609469478, 0.10918141397982746)

In [ ]:
```
# Plot the distributions of average ratings for new and old movies
fig, ax = plt.subplots(figsize=(8, 6))

sns.histplot(new_df.mean(axis=0).dropna(), kde=True, label='New Movies', color='blue', ax=ax)
sns.histplot(old_df.mean(axis=0).dropna(), kde=True, label='Old Movies', color='red', ax=ax)

ax.set_title('Distribution of Average Ratings for New and Old Movies')
ax.set_xlabel('Average Rating')
ax.set_ylabel('Frequency')
ax.legend()
plt.savefig('q2_distribution.png')
plt.show()
```

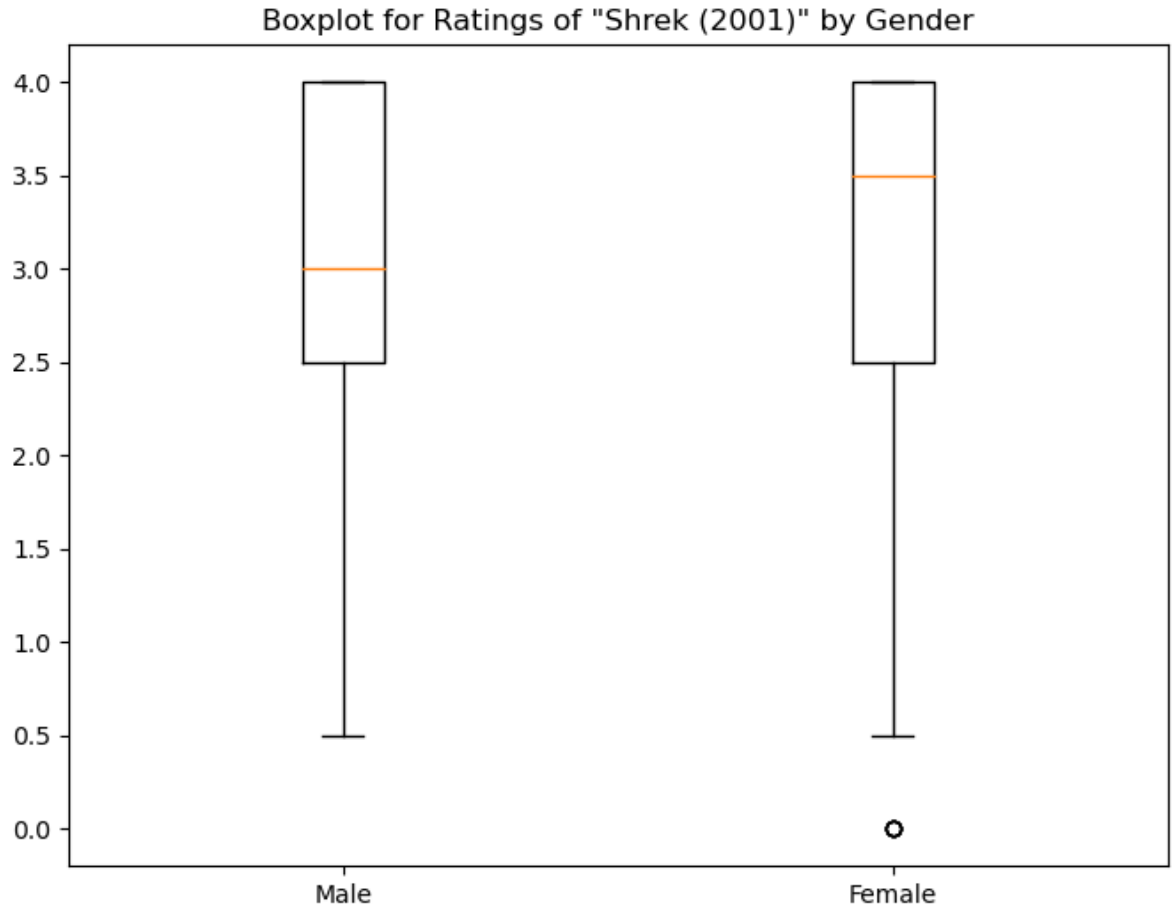


Q3 Is enjoyment of 'Shrek (2001)' gendered, i.e. do male and female viewers rate it differently?

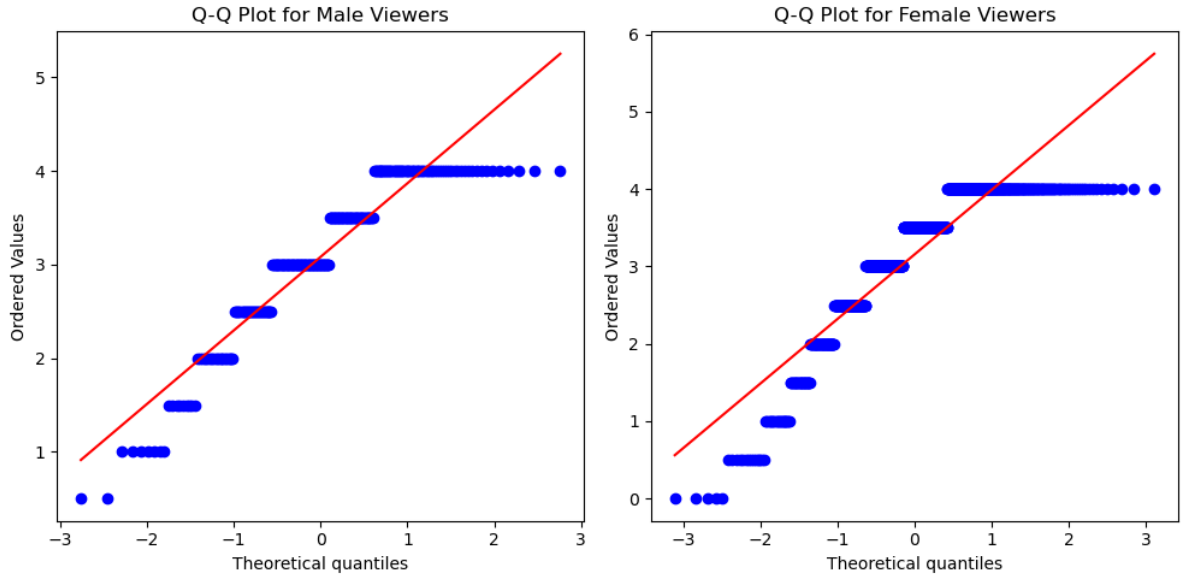
In [ ]:
```
shrek_ratings = df['Shrek (2001)']
gender_data = df['Gender identity (1 = female; 2 = male; 3 = self-described)']
male_ratings = shrek_ratings[gender_data == 2].dropna()
female_ratings = shrek_ratings[gender_data == 1].dropna()

#independent two-sample t-test
t_stat, p_value = ttest_ind(male_ratings, female_ratings)
t_stat, p_value
```

Out[ ]: (-1.1016699726285888, 0.27087511813734183)

In [ ]:
```
# Boxplot for average ratings comparison
fig, ax = plt.subplots(figsize=(8, 6))

data = [male_ratings, female_ratings]
labels = ['Male', 'Female']

ax.boxplot(data, labels=labels)
ax.set_title('Boxplot for Ratings of "Shrek (2001)" by Gender')
plt.savefig('q3_boxplot.png')
plt.show()
```



In [ ]:
```
# Q-Q plots for normality
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(10, 5))
probplot(male_ratings, plot=axes[0])
axes[0].set_title('Q-Q Plot for Male Viewers')
probplot(female_ratings, plot=axes[1])
axes[1].set_title('Q-Q Plot for Female Viewers')
plt.tight_layout()
plt.savefig('q3_qq_plots.png')
plt.show()
```

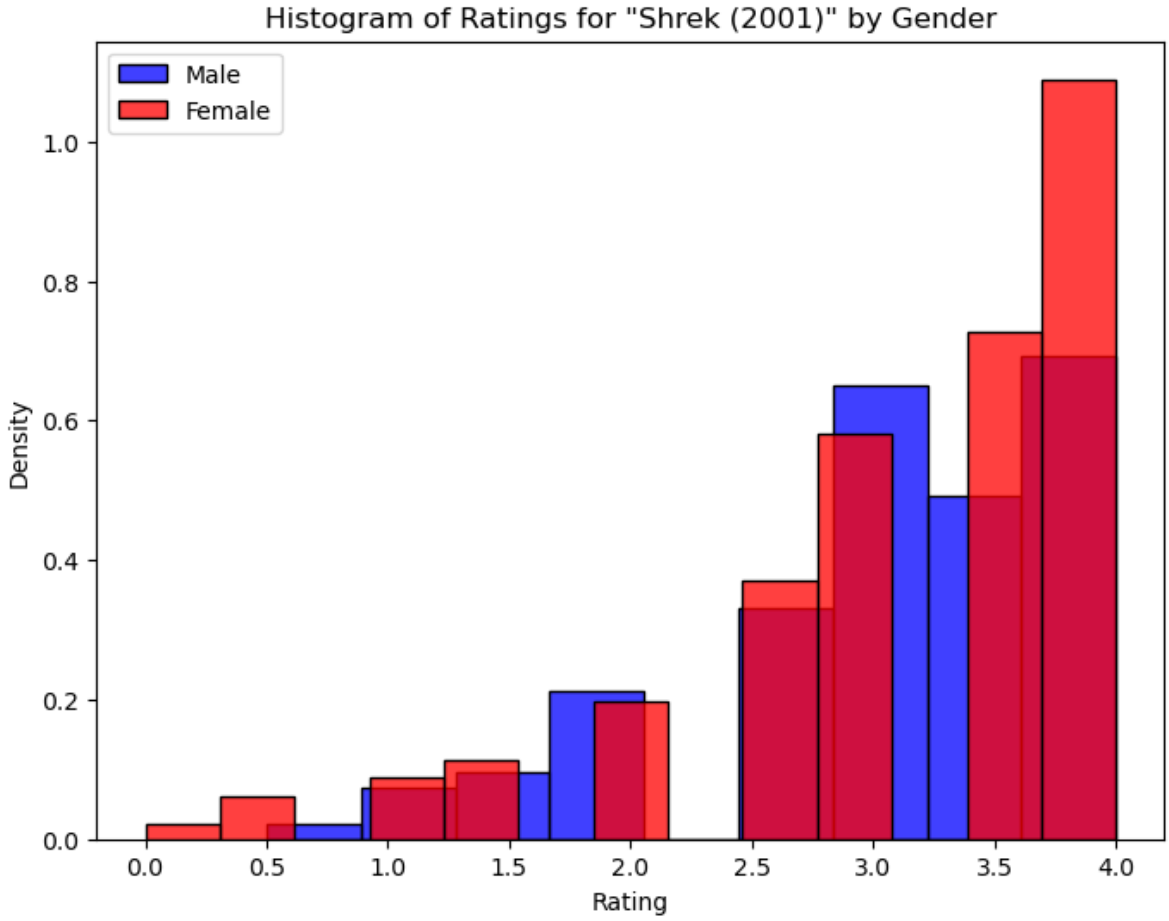


In [ ]:
```
n_male = len(male_ratings)
n_female = len(female_ratings)

d_f = n_male + n_female - 2
print(f'n_male = {n_male} , n_female = {n_female}')
d_f
```

```
n_male = 241 , n_female = 743
```

Out[ ]:
```
982
```

In [ ]:
```
# Plot the histograms of ratings for 'Shrek (2001)' for male and femal
e viewers
fig, ax = plt.subplots(figsize=(8, 6))

sns.histplot(male_ratings, kde=False, label='Male', color='blue', stat
="density", ax=ax)
sns.histplot(female_ratings, kde=False, label='Female', color='red', s
tat="density", ax=ax)

ax.set_title('Histogram of Ratings for "Shrek (2001)" by Gender')
ax.set_xlabel('Rating')
ax.set_ylabel('Density')
ax.legend()
plt.savefig('q3_distribution.png')
plt.show()
```



In [ ]:
```
df
```

Out[ ]:

	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)	E S (2
0	NaN	NaN	4.0	NaN	3.0	NaN	NaN	NaN	NaN	N
1	NaN	NaN	1.5	NaN	NaN	NaN	NaN	NaN	NaN	N
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
3	NaN	NaN	2.0	NaN	3.0	NaN	NaN	NaN	NaN	4
4	NaN	NaN	3.5	NaN	0.5	NaN	0.5	1.0	NaN	0
...	...	...	...	...	...	...	...	...	...	..
1092	NaN	NaN	NaN	NaN	3.5	NaN	NaN	NaN	NaN	N
1093	3.0	4.0	NaN	NaN	4.0	4.0	2.5	NaN	3.5	3
1094	NaN	NaN	NaN	NaN	NaN	NaN	NaN	3.5	NaN	N
1095	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
1096	NaN	NaN	4.0	NaN	2.5	NaN	NaN	3.0	NaN	3

1097 rows × 477 columns

Q4 What proportion of movies are rated differently by male and female viewers?

In [ ]:
```
significant_movies = []

# Adjusted significance level for multiple comparisons (Bonferroni cor
rection)
alpha = 0.005 / 400

for movie in df.columns[:400]:

    movie_ratings = df[movie]
    male_ratings = movie_ratings[gender_data == 2].dropna()
    female_ratings = movie_ratings[gender_data == 1].dropna()

    if len(male_ratings) > 0 and len(female_ratings) > 0:
        t_stat, p_value = ttest_ind(male_ratings, female_ratings)

        if p_value < alpha:
            significant_movies.append(movie)

proportion_different = len(significant_movies) / 400
proportion_different, significant_movies[:10]
```

Out[ ]:
```
(0.0275,
 ['13 Going on 30 (2004)',
  'The Proposal (2009)',
  'Ghostbusters (2016)',
  '10 Things I Hate About You (1999)',
  'Beauty and the Beauty (1991)',
  'Grease (1978)',
  'Harry Potter and the Deathly Hallows: Part 2 (2011)',
  'Chicago (2002)',
  'Bend it Like Beckham (2002)',
  'The Wolf of Wall Street (2013)'])
```

In [ ]:
```
len(significant_movies)
```

Out[ ]:
```
11
```

In [ ]:
```
significant_movies = []

# not Adjusted significance level for multiple comparisons
alpha = 0.005

for movie in df.columns[:400]:

    movie_ratings = df[movie]
    male_ratings = movie_ratings[gender_data == 2].dropna()
    female_ratings = movie_ratings[gender_data == 1].dropna()

    if len(male_ratings) > 0 and len(female_ratings) > 0:
        t_stat, p_value = ttest_ind(male_ratings, female_ratings)

        if p_value < alpha:
            significant_movies.append(movie)

proportion_different = len(significant_movies) / 400
proportion_different, significant_movies[:10]
```

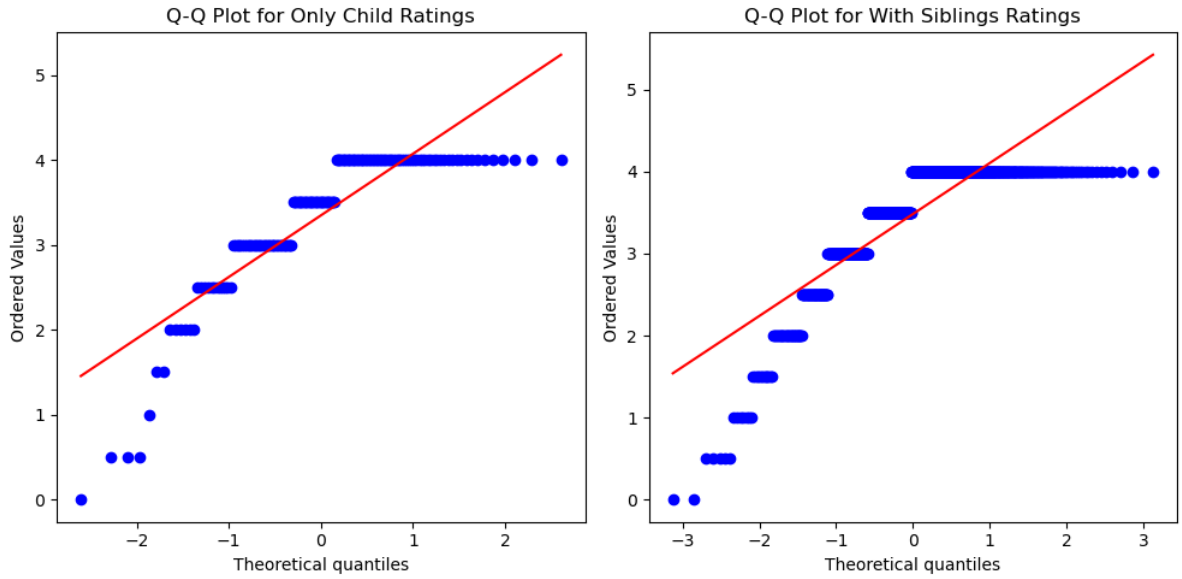
Out[ ]:
```
(0.115,
 ['Django Unchained (2012)',
  'Alien (1979)',
  'Star Wars: Episode IV - A New Hope (1977)',
  '13 Going on 30 (2004)',
  'Sorority Boys (2002)',
  'Inglorious Bastards (2009)',
  'Clueless (1995)',
  'The Exorcist (1973)',
  'Funny Girl (1968)',
  'The Thing (1982)'])
```

Q5 Do people who are only children enjoy 'The Lion King (1994)' more than people with siblings?

In [ ]:
```
lion_king_ratings = df['The Lion King (1994)']
only_child_data = df['Are you an only child? (1: Yes; 0: No; -1: Did n
ot respond)']
only_child_ratings = lion_king_ratings[only_child_data == 1].dropna()
with_siblings_ratings = lion_king_ratings[(only_child_data == 0) | (on
ly_child_data == -1)].dropna()

t_stat_lion_king, p_value_lion_king = ttest_ind(only_child_ratings, wi
th_siblings_ratings, alternative = 'greater')
t_stat_lion_king, p_value_lion_king
```

Out[ ]:
```
(-2.0450787705974394, 0.979436700331301)
```

In [ ]:
```
from scipy.stats import levene, probplot
import matplotlib.pyplot as plt

# Conduct Levene's test for homoscedasticity
levene_stat, levene_p_value = levene(only_child_ratings, with_siblings
_ratings)

# Q-Q plots for normality
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(10, 5))

probplot(only_child_ratings, plot=axes[0])
axes[0].set_title('Q-Q Plot for Only Child Ratings')

probplot(with_siblings_ratings, plot=axes[1])
axes[1].set_title('Q-Q Plot for With Siblings Ratings')

plt.tight_layout()
plt.savefig('q5_qq.png')
plt.show()


print('Levene\'s test statistic:', levene_stat)
print('Levene\'s test p-value:', levene_p_value)
```




```
Levene's test statistic: 1.3004327241596456
Levene's test p-value: 0.25442621663113313
```

In [ ]:
```
degf = len(only_child_ratings) + len(with_siblings_ratings) - 2
degf
```

Out[ ]: 935

In [ ]:
```
len(only_child_ratings)
```

Out[ ]: 151

In [ ]:
```
len(with_siblings_ratings)
```

Out[ ]: 786

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

In [ ]:
```
import numpy as np
import pandas as pd
from scipy.stats import bootstrap, permutation_test, mannwhitneyu, tte
st_rel, normaltest, kstest, kruskal, friedmanchisquare,f_oneway
```

In [ ]:
```
datas = pd.read_csv("MovieReplicationSet.csv")
```

In [ ]:
```
datas.iloc[:,475]
```

Out[ ]:
```
0       0
1       0
2       1
3       0
4       1
       ..
1092    0
1093    0
1094    0
1095    0
1096    0
Name: Are you an only child? (1: Yes; 0: No; -1: Did not respond), Len
gth: 1097, dtype: int64
```

In [ ]:
```
#problem_6
significant_count = 0
for i in range(400):#iterate over each of the 400 movies
    only_child = []
    have_siblings = []
    for j in range(0,1097):
        if pd.isna(datas.iloc[j,i]) or (datas.iloc[j,475]==-1):
            pass
        else:#has answer in both movie rating and only child
            if datas.iloc[j,475]==1:#only child
                only_child.append(datas.iloc[j,i])
            else:#have siblings
                have_siblings.append(datas.iloc[j,i])

    #test_data = (np.array(only_child),np.array(have_siblings))
    #p_test = permutation_test(test_data, test_stat_func, n_resamples=
int(1e4), random_state=69420)
    pvalue = mannwhitneyu(only_child,have_siblings).pvalue
    if pvalue<0.005:
        significant_count+=1
```

In [ ]:
```
significant_count
```

Out[ ]: 7

In [ ]:
```
7/400
```

Out[ ]: 0.0175

Q7 Do people who like to watch movies socially enjoy 'The Wolf of Wall Street (2013)' more than those who prefer to watch them alone?

In [ ]:
```
#problem 7
datas.columns[476]
```

Out[ ]: 'Movies are best enjoyed alone (1: Yes; 0: No; -1: Did not respond)'

In [ ]:
```
matrix_1 = datas.loc[:,['The Wolf of Wall Street (2013)','Movies are best enjoyed alone (1: Yes; 0: No; -1: Did not respond)']].dropna().to_numpy()
```

In [ ]:
```
matrix_1
```

Out[ ]:
```
array([[4. , 1. ],
       [3. , 1. ],
       [2.5, 0. ],
       ...,
       [3.5, 0. ],
       [2. , 0. ],
       [4. , 1. ]])
```

In [ ]:
```
social = []#0
alone = []#1
for i in matrix_1:
    if i[1]==0:
        social.append(i[0])
    elif i[1]==1:
        alone.append(i[0])
```

In [ ]:
```
pvalue = mannwhitneyu(social,alone,alternative="greater").pvalue
```

In [ ]:
```
pvalue
```

Out[ ]: 0.9436657996253056

In [ ]:
```
p = normaltest(alone).pvalue
```

In [ ]:
```
p
```

Out[ ]: 1.0549049474094136e-19

Q8 What proportion of movies exhibit such a “social watching” effect?

In [ ]:
```
#problem_8
significant_count = 0
for i in range(400):#iterate over each of the 400 movies
    social = []
    alone = []
    for j in range(0,1097):
        if (pd.isna(datas.iloc[j,i])) or (datas.iloc[j,476]==-1):
            pass
        else:#has answer in both movie rating and social
            if datas.iloc[j,476]==1:#alone
                alone.append(datas.iloc[j,i])
            else:#social
                social.append(datas.iloc[j,i])
    pvalue = mannwhitneyu(social,alone).pvalue
    if pvalue<0.005:
        significant_count+=1
```

In [ ]:
```
significant_count
```

Out[ ]:
```
10
```

In [ ]:
```
10/400
```

Out[ ]:
```
0.025
```

Q9 Is the ratings distribution of ‘Home Alone (1990)’ different than that of ‘Finding Nemo (2003)’?

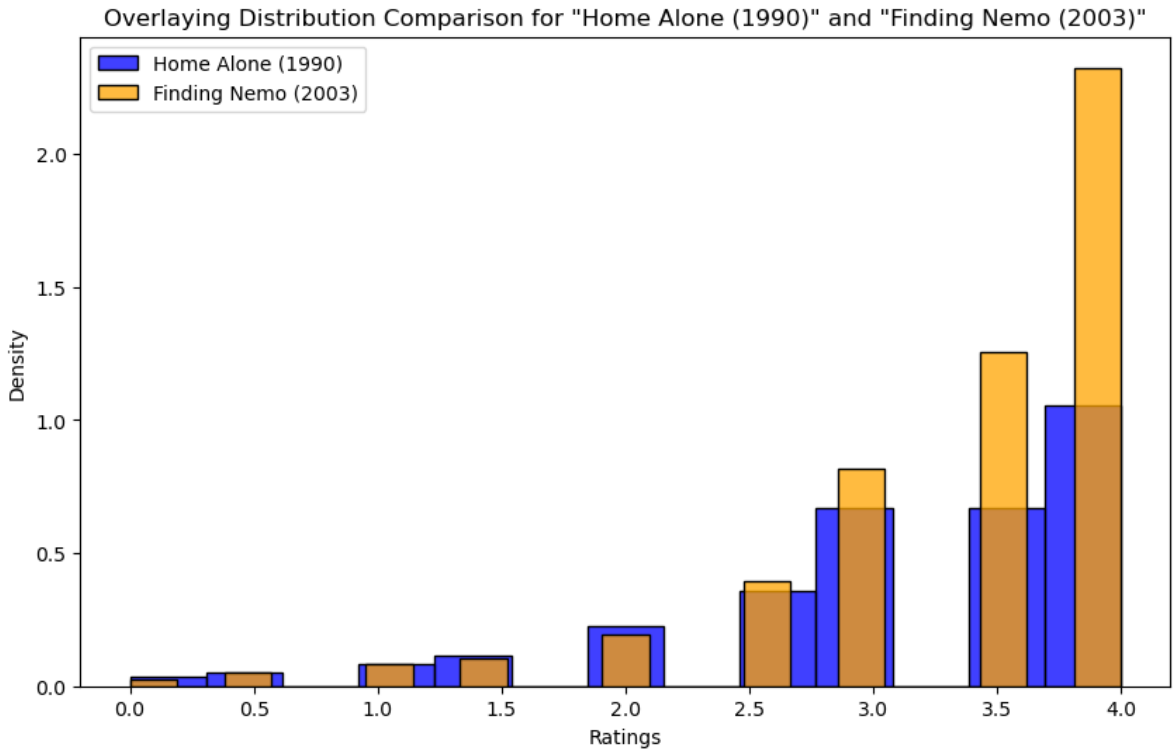
In [ ]:
```
#problem 9
home_alone = datas['Home Alone (1990)'].dropna()
```

In [ ]:
```
finding_nemo = datas['Finding Nemo (2003)'].dropna()
```

In [ ]:
```
kstest(home_alone,finding_nemo)
```

Out[ ]:
```
KstestResult(statistic=0.15269080020897632, pvalue=6.379397182836346e-
10, statistic_location=3.0, statistic_sign=1)
```

In [ ]:
```
home_alone_ratings = df['Home Alone (1990)'].dropna()
finding_nemo_ratings = df['Finding Nemo (2003)'].dropna()
fig, ax = plt.subplots(figsize=(10, 6))
sns.histplot(home_alone_ratings, kde=False, label='Home Alone (1990)',
color='blue', stat='density', ax=ax)
sns.histplot(finding_nemo_ratings, kde=False, label='Finding Nemo (200
3)', color='orange', stat='density', ax=ax)

plt.title('Overlaying Distribution Comparison for "Home Alone (1990)"
and "Finding Nemo (2003)"')
plt.xlabel('Ratings')
plt.ylabel('Density')
plt.legend()
plt.savefig('q9_distribution_plots.png')
plt.show()
```



Q10 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]

In [ ]:
```
#problem 10
sw_series = datas.filter(regex = 'Star Wars')
sw_series.columns
```

Out[ ]:
```
Index(['Star Wars: Episode IV - A New Hope (1977)',
       'Star Wars: Episode II - Attack of the Clones (2002)',
       'Star Wars: Episode V - The Empire Strikes Back (1980)',
       'Star Wars: Episode 1 - The Phantom Menace (1999)',
       'Star Wars: Episode VII - The Force Awakens (2015)',
       'Star Wars: Episode VI - The Return of the Jedi (1983)'],
      dtype='object')
```

In [ ]:
```
sw = datas[['Star Wars: Episode IV - A New Hope (1977)','Star Wars: Ep
isode II - Attack of the Clones (2002)',
       'Star Wars: Episode V - The Empire Strikes Back (1980)',
       'Star Wars: Episode 1 - The Phantom Menace (1999)',
       'Star Wars: Episode VII - The Force Awakens (2015)',
       'Star Wars: Episode VI - The Return of the Jedi (1983)']].dropn
a().to_numpy()
```

In [ ]:
```
sw
```

Out[ ]:
```
array([[4. , 0. , 4. , 0. , 2.5, 4. ],
       [4. , 3.5, 4. , 4. , 4. , 4. ],
       [3. , 4. , 4. , 4. , 4. , 4. ],
       ...,
       [3.5, 3.5, 4. , 3.5, 3.5, 4. ],
       [4. , 3. , 3.5, 4. , 4. , 4. ],
       [3. , 1.5, 3. , 2.5, 4. , 4. ]])
```

In [ ]:
```
sw1 = sw[:,0]
sw2 = sw[:,1]
sw3 = sw[:,2]
sw4 = sw[:,3]
sw5 = sw[:,4]
sw6 = sw[:,5]
```

In [ ]:
```
sw_series.columns
```

Out[ ]:
```
Index(['Star Wars: Episode IV - A New Hope (1977)',
       'Star Wars: Episode II - Attack of the Clones (2002)',
       'Star Wars: Episode V - The Empire Strikes Back (1980)',
       'Star Wars: Episode 1 - The Phantom Menace (1999)',
       'Star Wars: Episode VII - The Force Awakens (2015)',
       'Star Wars: Episode VI - The Return of the Jedi (1983)'],
      dtype='object')
```

In [ ]:
```
friedmanchisquare(sw1, sw2, sw3, sw4, sw5, sw6)
```

Out[ ]:
```
FriedmanchisquareResult(statistic=274.10740469208264, pvalue=3.6709439
947980374e-57)
```

In [ ]:
```
hp_series = datas.filter(regex = 'Harry Potter')
```

In [ ]:
```
hp_series.columns
```

Out[ ]:
```
Index(['Harry Potter and the Sorcerer's Stone (2001)',
       'Harry Potter and the Deathly Hallows: Part 2 (2011)',
       'Harry Potter and the Goblet of Fire (2005)',
       'Harry Potter and the Chamber of Secrets (2002)'],
      dtype='object')
```

In [ ]:
```
hp = datas[['Harry Potter and the Sorcerer\'s Stone (2001)',
       'Harry Potter and the Deathly Hallows: Part 2 (2011)',
       'Harry Potter and the Goblet of Fire (2005)',
       'Harry Potter and the Chamber of Secrets (2002)']].dropna().to_
numpy()
```

In [ ]:
```
hp1 = hp[:,0]
hp2 = hp[:,1]
hp3 = hp[:,2]
hp4 = hp[:,3]
```

In [ ]:
```
friedmanchisquare(hp1, hp2, hp3, hp4)
```

Out[ ]:
```
FriedmanchisquareResult(statistic=15.918421052632318, pvalue=0.0011785
008446512004)
```

In [ ]:
```
ma_series = datas.filter(regex = 'The Matrix')
```

In [ ]:
```
ma_series.columns
```

Out[ ]:
```
Index(['The Matrix Revolutions (2003)', 'The Matrix Reloaded (2003)',
       'The Matrix (1999)'],
      dtype='object')
```

In [ ]:
```
ma = datas[['The Matrix Revolutions (2003)', 'The Matrix Reloaded (200
3)',
       'The Matrix (1999)']].dropna().to_numpy()
```

In [ ]:
```
ma1 = ma[:,0]
ma2 = ma[:,1]
ma3 = ma[:,2]
```

In [ ]:
```
friedmanchisquare(ma1,ma2,ma3)
```

Out[ ]:
```
FriedmanchisquareResult(statistic=67.62078651685432, pvalue=2.07172819
64718423e-15)
```

In [ ]:
```
ij_series = datas.filter(regex = 'Indiana')
ij_series.columns
```

Out[ ]:
```
Index(['Indiana Jones and the Last Crusade (1989)',
       'Indiana Jones and the Temple of Doom (1984)',
       'Indiana Jones and the Raiders of the Lost Ark (1981)',
       'Indiana Jones and the Kingdom of the Crystal Skull (2008)'],
      dtype='object')
```

In [ ]:
```
ij = datas[['Indiana Jones and the Last Crusade (1989)',
       'Indiana Jones and the Temple of Doom (1984)',
       'Indiana Jones and the Raiders of the Lost Ark (1981)',
       'Indiana Jones and the Kingdom of the Crystal Skull (2008)']].dropna().to_numpy()
```

In [ ]:
```
ij1 = ij[:,0]
ij2 = ij[:,1]
ij3 = ij[:,2]
ij4 = ij[:,3]
```

In [ ]:
```
friedmanchisquare(ij1,ij2,ij3,ij4)
```

Out[ ]:
```
FriedmanchisquareResult(statistic=83.4644655847796, pvalue=5.542750833909764e-18)
```

In [ ]:
```
jp_series = datas.filter(regex = 'Jurassic')
jp_series.columns
```

Out[ ]:
```
Index(['The Lost World: Jurassic Park (1997)', 'Jurassic Park III (2001)',
       'Jurassic Park (1993)'],
      dtype='object')
```

In [ ]:
```
jp = datas[['The Lost World: Jurassic Park (1997)', 'Jurassic Park III (2001)',
       'Jurassic Park (1993)']].dropna().to_numpy()
```

In [ ]:
```
jp1 = jp[:,0]
jp2 = jp[:,1]
jp3 = jp[:,2]
```

In [ ]:
```
friedmanchisquare(jp1,jp2,jp3)
```

Out[ ]:
```
FriedmanchisquareResult(statistic=76.01454545454507, pvalue=3.116385552875769e-17)
```

In [ ]:
```
pc_series = datas.filter(regex = 'Pirates of')
pc_series.columns
```

Out[ ]:
```
Index(['Pirates of the Caribbean: Dead Man's Chest (2006)',
       'Pirates of the Caribbean: At World's End (2007)',
       'Pirates of the Caribbean: The Curse of the Black Pearl (2003)'],
      dtype='object')
```

In [ ]:
```
pc = datas[['Pirates of the Caribbean: Dead Man\'s Chest (2006)',
       'Pirates of the Caribbean: At World\'s End (2007)',
       'Pirates of the Caribbean: The Curse of the Black Pearl (200
3)']].dropna().to_numpy()
pc
```

Out[ ]:
```
array([[2. , 1. , 3. ],
       [2. , 3. , 3. ],
       [4. , 4. , 4. ],
       ...,
       [3.5, 3.5, 4. ],
       [3.5, 3.5, 3.5],
       [2.5, 3.5, 4. ]])
```

In [ ]:
```
pc1 = pc[:,0]
pc2 = pc[:,1]
pc3 = pc[:,2]
friedmanchisquare(pc1,pc2,pc3)
```

Out[ ]:
```
FriedmanchisquareResult(statistic=21.88005780346822, pvalue=1.77339637
38836908e-05)
```

In [ ]:
```
ts_series = datas.filter(regex = 'Toy Story')
ts_series.columns
```

Out[ ]:
```
Index(['Toy Story 2 (1999)', 'Toy Story 3 (2010)', 'Toy Story (199
5)'], dtype='object')
```

In [ ]:
```
ts = datas[['Toy Story 2 (1999)', 'Toy Story 3 (2010)', 'Toy Story (19
95)']].dropna().to_numpy()
ts
```

Out[ ]:
```
array([[3. , 3. , 4. ],
       [1. , 4. , 3. ],
       [3. , 3. , 3. ],
       ...,
       [3. , 3.5, 3. ],
       [2.5, 3.5, 3. ],
       [4. , 4. , 4. ]])
```

In [ ]:
```
ts1 = ts[:,0]
ts2 = ts[:,1]
ts3 = ts[:,2]
```

In [ ]:
```
friedmanchisquare(ts1,ts2,ts3)
```

Out[ ]:
```
FriedmanchisquareResult(statistic=56.338080495357836, pvalue=5.8390374
39450979e-13)
```

In [ ]:
```
bm_series = datas.filter(regex = 'Batman')
bm_series.columns
```

Out[ ]:
```
Index(['Batman & Robin (1997)', 'Batman (1989)',
       'Batman: The Dark Knight (2008)'],
      dtype='object')
```

In [ ]:
```
bm = datas[['Batman & Robin (1997)', 'Batman (1989)',
       'Batman: The Dark Knight (2008)']].dropna().to_numpy()
```

In [ ]:
```
bm1= bm[:,0]
bm2 = bm[:,1]
bm3 = bm[:,2]
```

In [ ]:
```
friedmanchisquare(bm1,bm2,bm3)
```

Out[ ]:
```
FriedmanchisquareResult(statistic=102.47941176470552, pvalue=5.5831395
2192982e-23)
```

In [ ]:
```
datas.columns.get_loc('Batman & Robin (1997)')
```

Out[ ]:
```
46
```

In [ ]:
```
datas.columns.get_loc('Batman (1989)')
```

Out[ ]:
```
181
```

In [ ]:
```
datas.columns.get_loc('Batman: The Dark Knight (2008)')
```

Out[ ]:
```
235
```

In [ ]:
```
datas.columns[474]
```

Out[ ]:
```
'Gender identity (1 = female; 2 = male; 3 = self-described)'
```

Extra Credit: We may have heard about the stereotype that male viewers do like Batman movies more than female viewers do. Is this true according to your tests?

In [ ]:
```
#extra credit question
male = []
female = []
for i in [46,181,235]:#iterate over three Batman movies
    for j in range(0,1097):
        if (pd.isna(datas.iloc[j,i])) or (datas.iloc[j,474]==3):
            pass
        else:#identified as male or female and has watched batman
            if datas.iloc[j,474]==1:#female
                female.append(datas.iloc[j,i])
            else:#male
                male.append(datas.iloc[j,i])
```

In [ ]:
```
mannwhitneyu(male,female,alternative="greater").pvalue
```

Out[ ]:
```
0.11866990804380168
```