Movie Data Analysis Project Report

Kunkun Hu, Jiaran Peng

In this movie data analysis project, I mainly researched 11 questions and conducted various types of hypothesis tests depending on the context of the questions. The alpha value I will use is 0.005. Due to the length of this report, I will not explicitly write each null and alternative hypothesis. I will use both Welch's and Student's t-tests for most cases since the defination the Homogeneity of variance is not clear. Using both tests allows me to make a safer decision. For dealing with missing values, I decided to use element-wise removal of missing values instead of row-wise removal of missing values. Therefore, all of my tests will be independent test instead of dependent test. The main reason is I want to keep as much information and data as possible. Using row-wise removal will lose roughly twice as much data as element-wise removal. That may be a potential limitation of my tests since I didn't check the inter-individual variability for each test.

Question 1: Whether more popular movies are rated higher than less popular movies. I divided the movies into popular and unpopular groups based on the median number of ratings. Then, I computed the mean rating scores of each movie and the variance for the two groups. Since I don't know the underlying distribution and my research question is whether A is bigger than B, I decided to use Welch's independent one-sided t-test. Welch's t-test's assumptions are that data are independently and normally distributed, corresponding with their density distribution. Finally, I got the t-statistics = 17.76 and p-value = 4.76e-52. Since the p-value is smaller than 0.005, we can reject the null hypothesis and announce that popular movies are rated higher than less popular movies. I selected Welch's test instead of the Student's t-test because of the variance difference between these two groups (0.09, 0.05).

Question 2: Whether newer movies are rated differently than older movies. I split the dataset into two groups based on the median value of the release year. I compared the variance of these two groups, which are 0.13 and 0.12. I conducted both Student's independent one-side t-test and Welch's independent one-side t-test for safety. The only difference between these two tests is that the Student's t-test assumes homogeneity of variance. Finally, both tests got a 1.6 t-statistics and 0.11 p-value, which is way higher than 0.005. We fail to reject the null hypothesis that newer and older movies are rated similarly.

Question 3: Whether the enjoyment of 'Shrek (2001)' is gendered. Similarly, I split the dataset into male and female groups. Since the variance of these two groups are 0.82 and 0.68. I still chose to conduct both Welch's independent t-test and Student's independent t-test. The only difference is I will conduct a two-sided t-test since it asks whether male and female viewers rate it differently. The assumptions of these two tests are stated above. Both tests gave me t-statistics of around 1.1 and a p-value of around 0.25. We fail to reject the null hypothesis.

Question 4: What proportion of movies are rated differently by male and female viewers. I conducted a Welch's independent two-sided t-test for each movie. I did not choose Student's t-test

because it is hard to define how small the variance difference is enough for the assumption homogeneity of variance. Finally, I calculated the proportion of significant p-values, which is 11.25% which means around 11.25% of movies are rated differently by male and female viewers.

Question 5: Do people who are only children enjoy 'The Lion King (1994)' more than people with siblings? I split the dataset into two groups based on whether the person is an only child. The approach is similar to the above, and I conducted both Welch's independent one-side t-test and Student's independent one-side t-test. The corresponding t-statistics are -1.88 and -2.05. P-values are roughly 0.97 for both tests. We fail to reject the null hypothesis.

Question 6: The proportion of movies that exhibit an "only child effect". Like question 4, I conducted a Welch's independent two-sided t-test for each movie in two groups. The proportion of significant p-value is 0.015, which means the only child effect barely exists.

Question 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? I first split the dataset into two groups based on whether movies are best enjoyed alone. I conducted both Welch's independent one-side t-test and Student's independent one-side t-test. The t-statistics are both -1.55, and p-values are 0.94. We fail to reject the null hypothesis.

Question 8: What proportion of movies exhibit such a "social watching" effect. Similar to question 4, I conducted a two-sided Welch's independent t-test for each movie after splitting. The proportion of significant p-value is 0.015, which means social watching barely exists.

Question 9: Whether 'Home Alone (1990)' rating distribution differs from 'Finding Nemo (2003)'? I used a two-sided K-S test to compare the distribution difference. The assumption is that data is independently sampled from a continuous distribution without missing values. The bigger the sample size is, the more powerful the K-S test will be. After addressing the missing values, the statistic is 0.15 with a corresponding p-value 6.38e-10. We can firmly reject the null hypothesis and announce that the rates of these two movies are from different distributions.

Question 10: Whether several franchises have consistent quality. I first collected the data of each franchise using the element-wise approach. Then, I conducted a one-way ANOVA test for each franchise to test whether each movie has a similar rating. The assumptions one-way ANOVA makes are data sampled randomly and independently. Each category forms a normal distribution with similar variance and mutual exclusions. After conducting the one-way ANOVA test, I found that only Harry Potter franchise has a consistent quality with a p-value<0.005. The limitation of this test is I needed to check the assumption of normality and homogeneity of variance.

For the extra credit, I studied whether people who like to drive fast will give higher ratings for the movie "The Fast and the Furious (2001)". I split the data into two groups based on whether they like driving fast. Then, I compared their variance and dropped the missing values. Finally, I conducted both one side Student's independent t-test and one side Welch's independent t-test. The t-statistics are both 0.7 with corresponding p-value = 0.24. We fail to reject the null hypothesis since the p-value is way bigger than 0.005.

Data analysis project 1:

Hypothesis testing of movie ratings data

Introduction to Data Science (DS-GA1001)

Code by: Kunkun Hu(kh3492@nyu.edu (mailto:kh3492@nyu.edu)), Jiaran Peng(jp7238@nyu.edu (mailto:jp7238@nyu.edu))

Date: 10-06-23

In [4]:
```
# importing necessary libraries
import random
import numpy as np
import pandas as pd
import scipy.stats as stats
from scipy.stats import bootstrap, permutation_test,ks_2samp
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
```

In [5]:
```
# importing data
df = pd.read_csv('movieReplicationSet.csv')
df.head()
```

Out[5]:

	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)	...	w a c s c
0	NaN	NaN	4.0	NaN	3.0	NaN	NaN	NaN	NaN	NaN	...	

In [6]:
```
df.dtypes
```

Out[6]:
```
The Life of David Gale (2003)
float64
Wing Commander (1999)
float64
Django Unchained (2012)
float64
Alien (1979)
float64
Indiana Jones and the Last Crusade (1989)
float64

...
Movies change my position on social economic or political issues
float64
When watching movies things get so intense that I have to stop watching
float64
Gender identity (1 = female; 2 = male; 3 = self-described)
float64
Are you an only child? (1: Yes; 0: No; -1: Did not respond)
int64
Movies are best enjoyed alone (1: Yes; 0: No; -1: Did not respond)
int64
Length: 477, dtype: object
```

question 1: Are movies that are more popular (operationalized as having more ratings) rated higher than movies that are less popular?

In [7]:
```
# data wrangling
movies = df.iloc[:,0:400]
average_ratings = movies.mean(axis=0)
num_ratings = movies.notna().sum(axis=0)
# Create a new DataFrame with the desired format
movie_ratings_df = pd.DataFrame({'Average_Rating': average_ratings, 'Num_

# Set the index of the new DataFrame to be the movie names
movie_ratings_df.index = movies.columns
# find the median number of ratings
median_Num_Ratings = np.median(movie_ratings_df["Num_Ratings"])

movie_ratings_df["popularity"] = (movie_ratings_df["Num_Ratings"]>median_
movie_ratings_df.head()
```

Out[7]:

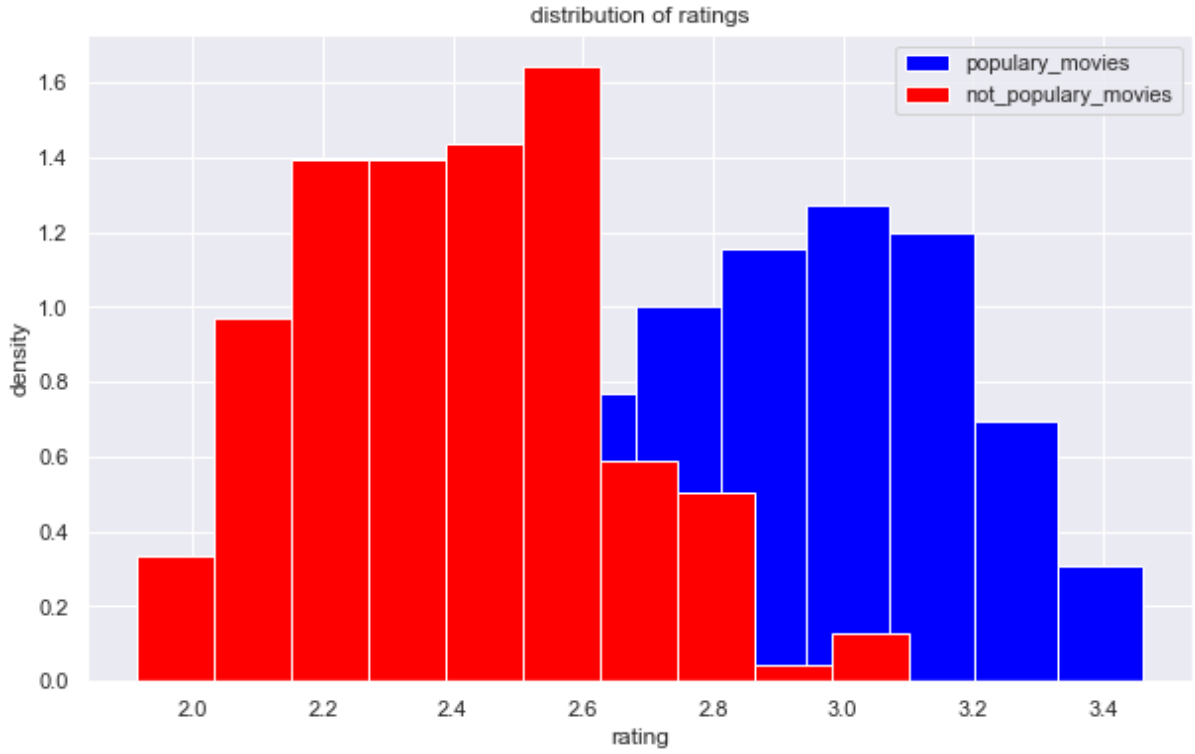
	Average_Rating	Num_Ratings	popularity
The Life of David Gale (2003)	2.151316	76	0
Wing Commander (1999)	2.021127	71	0
Django Unchained (2012)	3.153422	453	1
Alien (1979)	2.707612	289	1
Indiana Jones and the Last Crusade (1989)	2.778618	463	1

In [8]:
```
# filter out the two categories
filt = movie_ratings_df["popularity"] ==1

populary_movies = movie_ratings_df.loc[filt,"Average_Rating"]
not_popularly_movies = movie_ratings_df.loc[~filt,"Average_Rating"]
```

In [9]:
```
print(popularly_movies.var(),not_popularly_movies.var())
```

```
0.08500243213090868 0.05357798836736422
```

In [114]:
```
# plot the distribution of two categories
plt.figure(figsize=(10,6))
plt.hist(popularly_movies, label='popularly_movies', color='blue',density =
plt.hist(not_popularly_movies, label='not_popularly_movies',color='red',den
plt.title('distribution of ratings')
plt.xlabel('rating')
plt.ylabel('density')
plt.legend(loc='upper right')
plt.show()
```



In [115]:
```
print(popular_movies.var(),not_popular_movies.var())
# Welch's one side t-test
print("Welch's one side t-test:",stats.ttest_ind(popular_movies, not_pop
                                                 equal_var = False,alter
# student one side t-test
print("student's one side t-test:", stats.ttest_ind(popular_movies, not_
                                                    equal_var = True,alte
```

```
0.08500243213090868 0.05357798836736422
Welch's one side t-test: Ttest_indResult(statistic=17.7560492698737, pv
alue=4.768468295426728e-52)
student's one side t-test: Ttest_indResult(statistic=17.7560492698737,
pvalue=1.1348265138282423e-52)
```

In [116]:
```
def test_stat_func(x,y):
    return np.mean(x) - np.mean(y)

test_data = (popular_movies, not_popular_movies)
p_test = permutation_test(test_data, test_stat_func, n_resamples=int(1e4)
print('Test stat:', p_test.statistic)
print('P-val:',p_test.pvalue)
```

```
Test stat: 0.4673930830804105
P-val: 9.999000099990002e-05
```

reject H0 since pvalue <0.005 for all three tests we have conducted

question 2: Are movies that are newer rated differently than movies that are older?

In [117]:
```
# creating a new column that classify the movie by their released year de
movie_ratings_df["year_released"] = movie_ratings_df.index.map(lambda x:
movie_ratings_df["new_or_old"] = (movie_ratings_df["year_released"]>=np.m
movie_ratings_df.head()
```

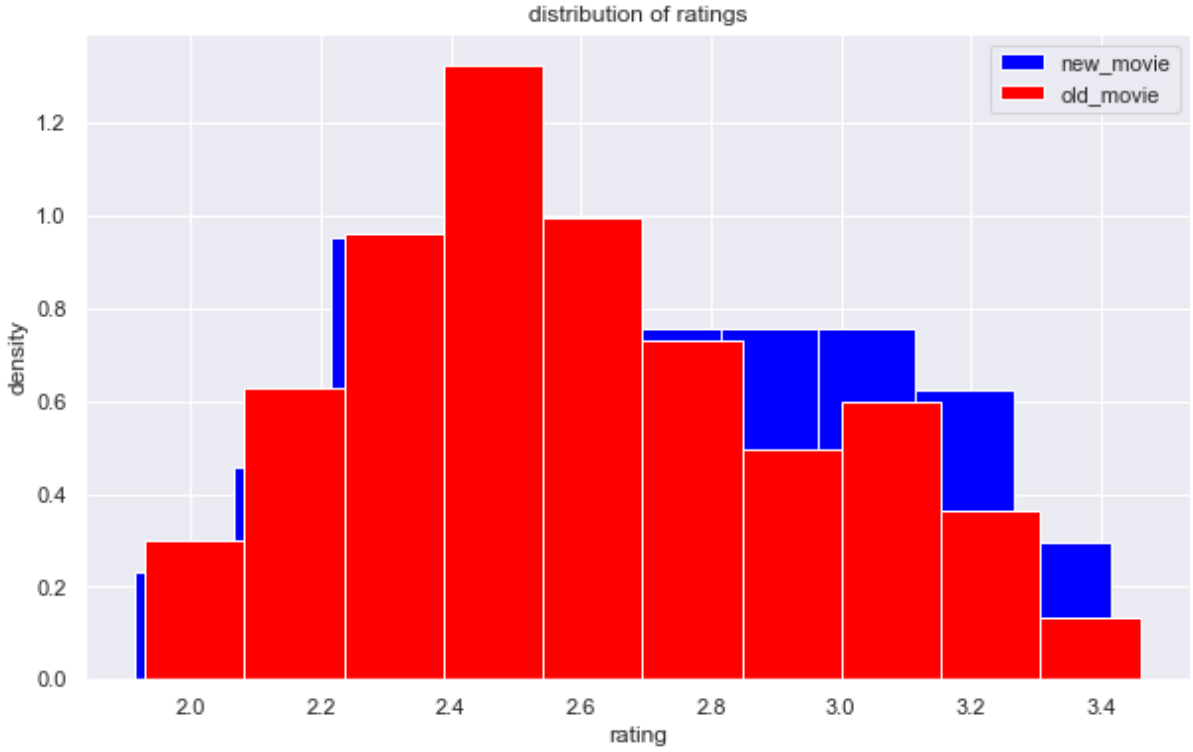
Out[117]:

	Average_Rating	Num_Ratings	popularity	year_released	new_or_old
The Life of David Gale (2003)	2.151316	76	0	2003	1
Wing Commander (1999)	2.021127	71	0	1999	1
Django Unchained (2012)	3.153422	453	1	2012	1
Alien (1979)	2.707612	289	1	1979	0
Indiana Jones and the Last Crusade (1989)	2.778618	463	1	1989	0

In [118]:
```
new_movie = movie_ratings_df[movie_ratings_df["new_or_old"]==1]["Average_
old_movie = movie_ratings_df[movie_ratings_df["new_or_old"]==0]["Average_
print(len(new_movie))
print(len(old_movie))
```

```
203
197
```

In [119]:
```
# plot the distribution of two categories
plt.figure(figsize=(10,6))
plt.hist(new_movie, label='new_movie', color='blue',density = "True")
plt.hist(old_movie, label='old_movie',color='red',density = "True")
plt.title('distribution of ratings')
plt.xlabel('rating')
plt.ylabel('density')
plt.legend(loc='upper right')
plt.show()
```



In [120]:
```
# check variance within two groups
print(new_movie.var(),old_movie.var())
```

```
0.1285073626847248 0.1180950341944459
```

In [121]:
```
# in that case we do two side test since our alternative hypothesis is wh
# instead of one is bigger than another
# Welch's two side t-test
print("Welch's two side t-test:",stats.ttest_ind(new_movie, old_movie, eq
# student two side t-test
print("student's two side t-test:",stats.ttest_ind(new_movie, old_movie,
```

```
Welch's two side t-test: Ttest_indResult(statistic=1.6064993124617366,
pvalue=0.10895741235786588)
student's two side t-test: Ttest_indResult(statistic=1.605479609469478,
pvalue=0.10918141397982746)
```

In [122]:
```
# permutation test
test_data = (new_movie, old_movie)
p_test = permutation_test(test_data, test_stat_func, n_resamples=int(1e4)
print('Test stat:', p_test.statistic)
print('P-val:',p_test.pvalue)
```

```
Test stat: 0.05639952980994467
P-val: 0.100989901009899
```

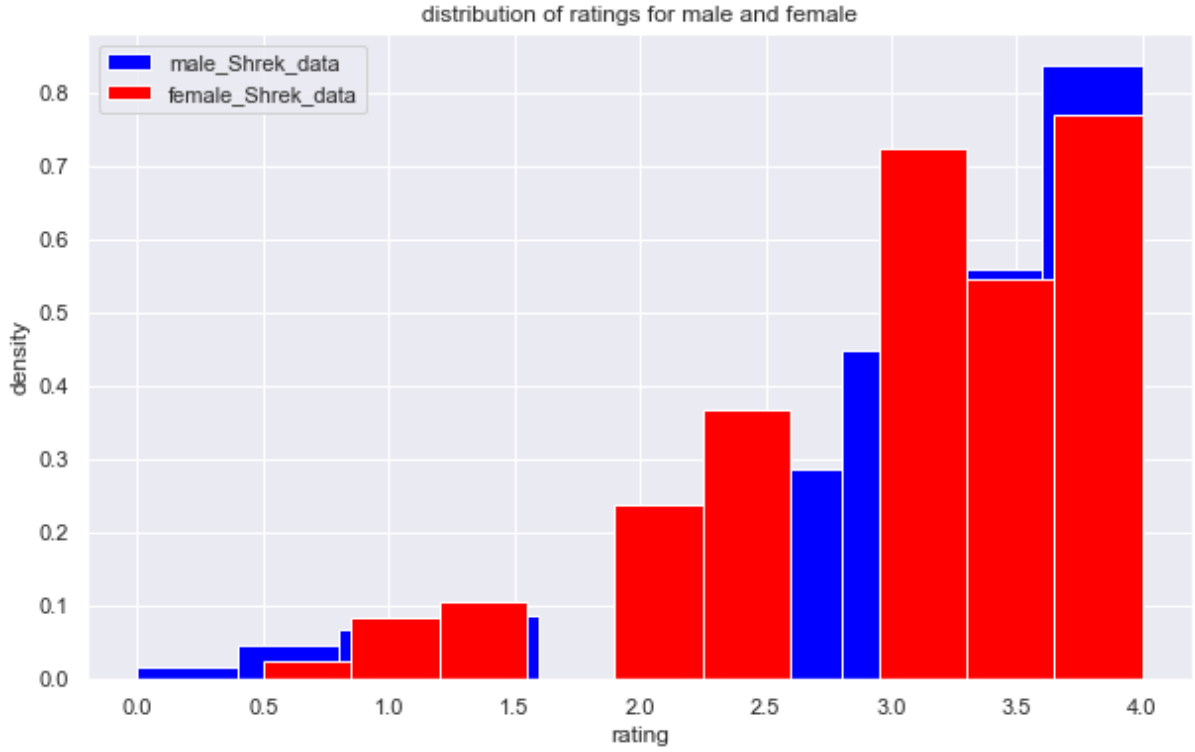
All three tests have corresponding result with the distribution of our plot. We fail rejecting the Null hypothesis and decide there is not significant difference between old movies and new moives in terms of rating based on our data.

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

In [123]:
```
male_filt =df["Gender identity (1 = female; 2 = male; 3 = self-described)
male_Shrek_data  = df.loc[male_filt,"Shrek (2001)"].dropna().to_numpy()
female_filt = df["Gender identity (1 = female; 2 = male; 3 = self-describ
female_Shrek_data = df.loc[female_filt,"Shrek (2001)"].dropna().to_numpy(
print(len(male_Shrek_data),len(female_Shrek_data))
```

```
743 241
```

In [124]:
```
# plot the density distribution of two categories
plt.figure(figsize=(10,6))
plt.hist(male_Shrek_data, label='male_Shrek_data', color='blue',density =
plt.hist(female_Shrek_data, label='female_Shrek_data',color='red',density
plt.title('distribution of ratings for male and female')
plt.xlabel('rating')
plt.ylabel('density')
plt.legend(loc='upper left')
plt.show()
```



In [125]:
```
# check variance within two groups
print(male_Shrek_data.var(),female_Shrek_data.var())
```

```
0.8207206244373233 0.6777603691396499
```

In [126]:
```
# in that case we do two side test since our alternative hypothesis is wh
# instead of one is bigger than another
# Welch's two side t-test
print("Welch's two side t-test:",stats.ttest_ind(male_Shrek_data, female_
# student two side t-test
print("student's two side t-test:",stats.ttest_ind(male_Shrek_data, fema
# two side permutation test for mean difference
test_data = (male_Shrek_data, male_Shrek_data)
p_test = permutation_test(test_data, test_stat_func, n_resamples=int(1e4)
print('Test stat for permutation test:', p_test.statistic)
print('P-val for permutation test:',p_test.pvalue)
```

```
Welch's two side t-test: Ttest_indResult(statistic=1.1558907155973421,
pvalue=0.24834907946281018)
student's two side t-test: Ttest_indResult(statistic=1.101669972628588
8, pvalue=0.27087511813734183)
Test stat for permutation test: 0.0
P-val for permutation test: 1.0
```

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

In [127]:
```
# planning do Welch's two side t-test for each movie and calculate the pr
significant_case = 0
alpha = 0.005
for movie in movies.columns:
    male_data  = df.loc[male_filt,movie].dropna().to_numpy()
    female_data = df.loc[female_filt,movie].dropna().to_numpy()
    p_value =stats.ttest_ind(male_data, female_data, equal_var = False).p
    if p_value<alpha:
        significant_case+=1
print(significant_case/len(movies.columns))
```

```
0.1125
```

did 400 two side Welch's t-test with alpha = 0.005. got the result of 11.25% movies out of 400 are significantly different rated between female and male

question 5: Do people who are only children enjoy 'The Lion King (1994)' more than people with siblings?

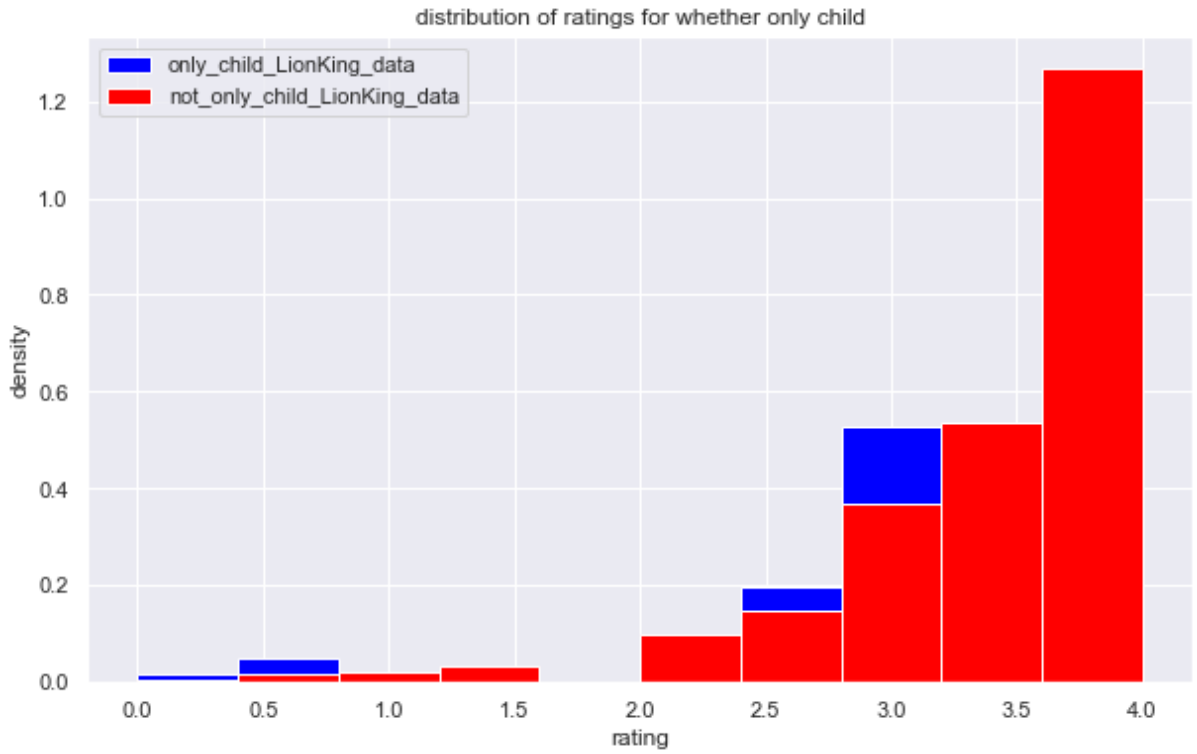
In [128]:
```
only_child_filt =df["Are you an only child? (1: Yes; 0: No; -1: Did not r
not_only_child_filt = df["Are you an only child? (1: Yes; 0: No; -1: Did

only_child_LionKing_data  = df.loc[only_child_filt,"The Lion King (1994)"

not_only_child_LionKing_data = df.loc[not_only_child_filt,"The Lion King
print(len(only_child_LionKing_data),len(not_only_child_LionKing_data))
```

```
151 776
```

In [129]:
```
# check variance within two groups
print(only_child_LionKing_data.var(),not_only_child_LionKing_data.var())
```

```
0.6622297267663699 0.5151384312891911
```

In [130]:
```
# plot the density distribution of two categories
plt.figure(figsize=(10,6))
plt.hist(only_child_LionKing_data, label='only_child_LionKing_data', colo
plt.hist(not_only_child_LionKing_data, label='not_only_child_LionKing_dat
plt.title('distribution of ratings for whether only child')
plt.xlabel('rating')
plt.ylabel('density')
plt.legend(loc='upper left')
plt.show()
```



In [131]:
```
# check variance within two groups
print(only_child_LionKing_data.var(),not_only_child_LionKing_data.var())
```

```
0.6622297267663699 0.5151384312891911
```

In [132]:
```
# in that case we do one side test since our alternative hypothesis is wh
# Welch's one side t-test
print("Welch's one side t-test:",stats.ttest_ind(only_child_LionKing_data
                                                 alternative='greater',eq

# student one side t-test
print("student's one side t-test:",stats.ttest_ind(only_child_LionKing_da
                                                   alternative='greater',

# one side permutation test for mean difference
test_data = (only_child_LionKing_data, not_only_child_LionKing_data)
p_test = permutation_test(test_data, test_stat_func,alternative='greater
print('Test stat for permutation test:', p_test.statistic)
print('P-val for permutation test:',p_test.pvalue)
```

```
Welch's one side t-test: Ttest_indResult(statistic=-1.884028409511613,
pvalue=0.9694855681322363)
student's one side t-test: Ttest_indResult(statistic=-2.05388899605898
6, pvalue=0.9798664723686588)
Test stat for permutation test: -0.1342766436813001
P-val for permutation test: 0.9811018898110189
```

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

In [133]:
```
significant_case = 0
for movie in movies.columns:
    only_child_data  = df.loc[only_child_filt,movie].dropna().to_numpy()
    not_only_child_data = df.loc[not_only_child_filt,movie].dropna().to_n
    p_value =stats.ttest_ind(only_child_data, not_only_child_data, equal_
    if p_value<alpha:
        significant_case+=1
print(significant_case/len(movies.columns))
```

```
0.015
```

Conducted 400 two sided Welch's t-test with alpha = 0.005. only 1.5% showed the only child effect.

question 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?

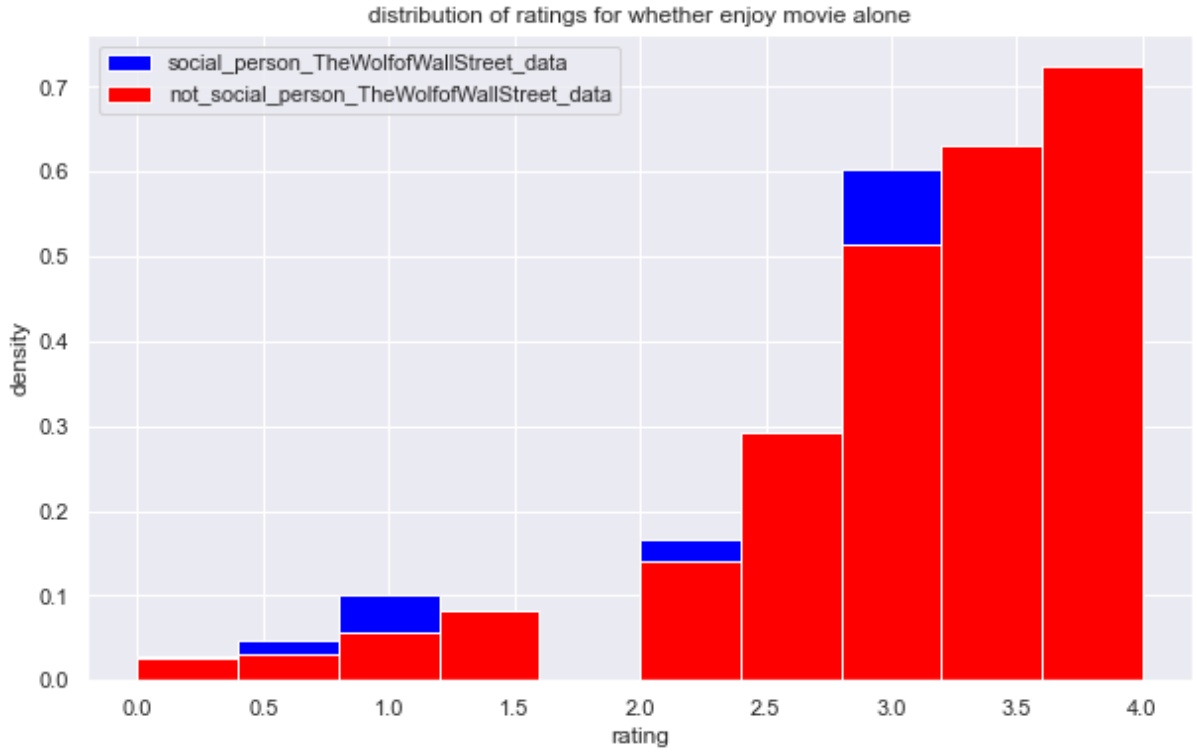
In [134]:
```
not_social_filt =df["Movies are best enjoyed alone (1: Yes; 0: No; -1: Di
social_filt = df["Movies are best enjoyed alone (1: Yes; 0: No; -1: Did n

social_person_TheWolfofWallStreet_data  = df.loc[social_filt,"The Wolf of
not_social_person_TheWolfofWallStreet_data  = df.loc[not_social_filt,"The
print(len(social_person_TheWolfofWallStreet_data),len(not_social_person_T
```

```
270 393
```

In [135]:
```
# check variance within two groups
print(social_person_TheWolfofWallStreet_data.var(),not_social_person_TheW
```

```
0.8451851851851853 0.754776657666932
```

In [136]:
```
# plot the density distribution of two categories
plt.figure(figsize=(10,6))
plt.hist(social_person_TheWolfofWallStreet_data, label='social_person_The
         color='blue',density = "True")
plt.hist(not_social_person_TheWolfofWallStreet_data, label='not_social_pe
         color='red',density = "True")
plt.title('distribution of ratings for whether enjoy movie alone')
plt.xlabel('rating')
plt.ylabel('density')
plt.legend(loc='upper left')
plt.show()
```



In [137]:
```
# in that case we do one side test since our alternative hypothesis is wh
# Welch's one side t-test
print("Welch's one side t-test:",stats.ttest_ind(social_person_TheWolfofW
            not_social_person_TheWolfofWallStreet_data, alternative='grea

# student one side t-test
print("student's one side t-test:",stats.ttest_ind(social_person_TheWolfo
            not_social_person_TheWolfofWallStreet_data,alternative='great

# one side permutation test for mean difference
test_data = (social_person_TheWolfofWallStreet_data, not_social_person_Th
p_test = permutation_test(test_data, test_stat_func,alternative='greater
print('Test stat for permutation test:', p_test.statistic)
print('P-val for permutation test:',p_test.pvalue)
```

```
Welch's one side t-test: Ttest_indResult(statistic=-1.5513309472217705,
pvalue=0.9393044802498963)
student's one side t-test: Ttest_indResult(statistic=-1.56787387450499
4, pvalue=0.9413054316716771)
Test stat for permutation test: -0.11043256997455497
P-val for permutation test: 0.942005799420058
```

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

In [138]:
```
# in that case of social watching effect,
# I assume it means there is a different rate between watching alone and
# So I used two-sided welch's t-test
significant_case = 0
for movie in movies.columns:
    only_child_data  = df.loc[social_filt,movie].dropna().to_numpy()
    not_only_child_data = df.loc[not_social_filt,movie].dropna().to_numpy
    p_value =stats.ttest_ind(only_child_data, not_only_child_data, equal_
    if p_value<alpha:
        significant_case+=1
print(significant_case/len(movies.columns))
```

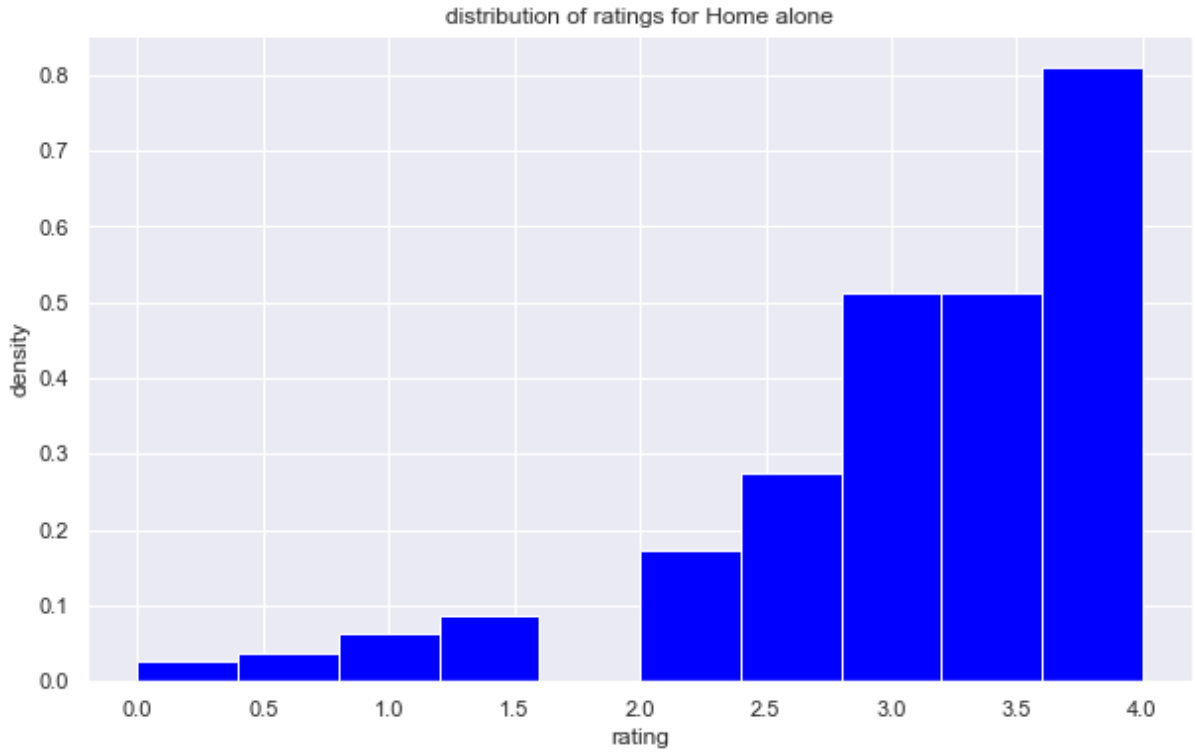
```
0.015
```

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

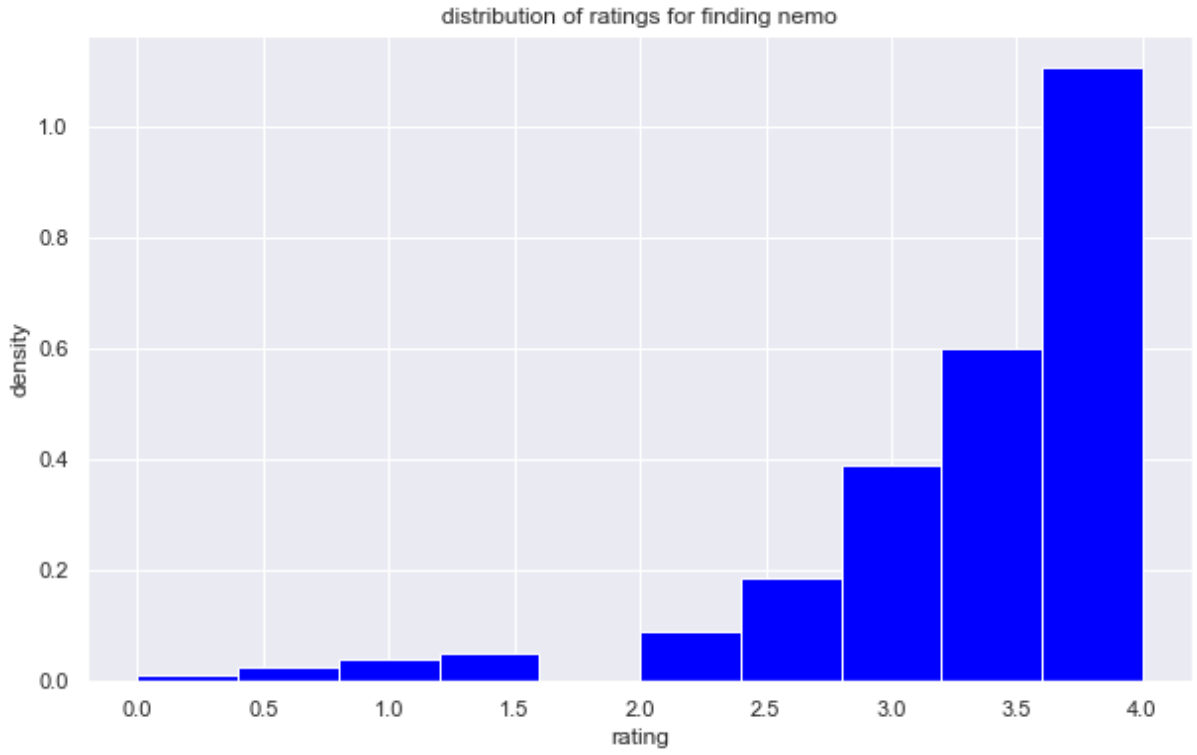
In [139]:
```
home_alone_data = df.loc[:,"Home Alone (1990)"].dropna().to_numpy()
finding_nemo_data = df.loc[:,"Finding Nemo (2003)"].dropna().to_numpy()
print(len(home_alone_data),len(finding_nemo_data))
```

```
857 1014
```

In [140]:
```
# plot the density distribution of two movies
plt.figure(figsize=(10,6))
plt.hist(home_alone_data,  color='blue',density = "True")
plt.title('distribution of ratings for Home alone')
plt.xlabel('rating')
plt.ylabel('density')
plt.show()
```



In [141]:
```
plt.figure(figsize=(10,6))
plt.hist(finding_nemo_data,color='blue',density = "True")
plt.title('distribution of ratings for finding nemo')
plt.xlabel('rating')
plt.ylabel('density')
plt.show()
```



In [142]:
```
# conduct a two-side ks two samples test to determine whether from the sa
statistic, p_value = ks_2samp(home_alone_data, finding_nemo_data)
print(ks_2samp(home_alone_data, finding_nemo_data))
```

```
KstestResult(statistic=0.15269080020897632, pvalue=6.379397182836346e-1
0, statistic_location=3.0, statistic_sign=1)
```

question 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]

In [143]:
```
Movie_list = ["Star Wars", "Harry Potter", "The Matrix", "Indiana Jones",
              "Pirates of the Caribbean", "Toy Story", "Batman"]
```

In [146]:
```
# element-wise approach using one way anova
for movie in Movie_list:
    movie_df = movies.loc[:,movies.columns.str.contains(movie)]
    data_list = list()
    for column in movie_df.columns:
        column = df.loc[:,column].dropna().to_numpy()
        data_list.append(column)
    f_statistic, p_value = stats.f_oneway(*data_list)
    if p_value<0.005:
        print(movie," franchises has a inconsistent quality")
    else:
        print(movie," franchises has a consistent quality")
```

```
Star Wars  franchises has a inconsistent quality
Harry Potter  franchises has a consistent quality
The Matrix  franchises has a inconsistent quality
Indiana Jones  franchises has a inconsistent quality
Jurassic Park  franchises has a inconsistent quality
Pirates of the Caribbean  franchises has a inconsistent quality
Toy Story  franchises has a inconsistent quality
Batman  franchises has a inconsistent quality
```

it looks like only Harry Potter franchises has a consistent quality.

extra credit

Does people who like drive fast will give higher rating for the movie " The Fast and the Furious (2001)"?

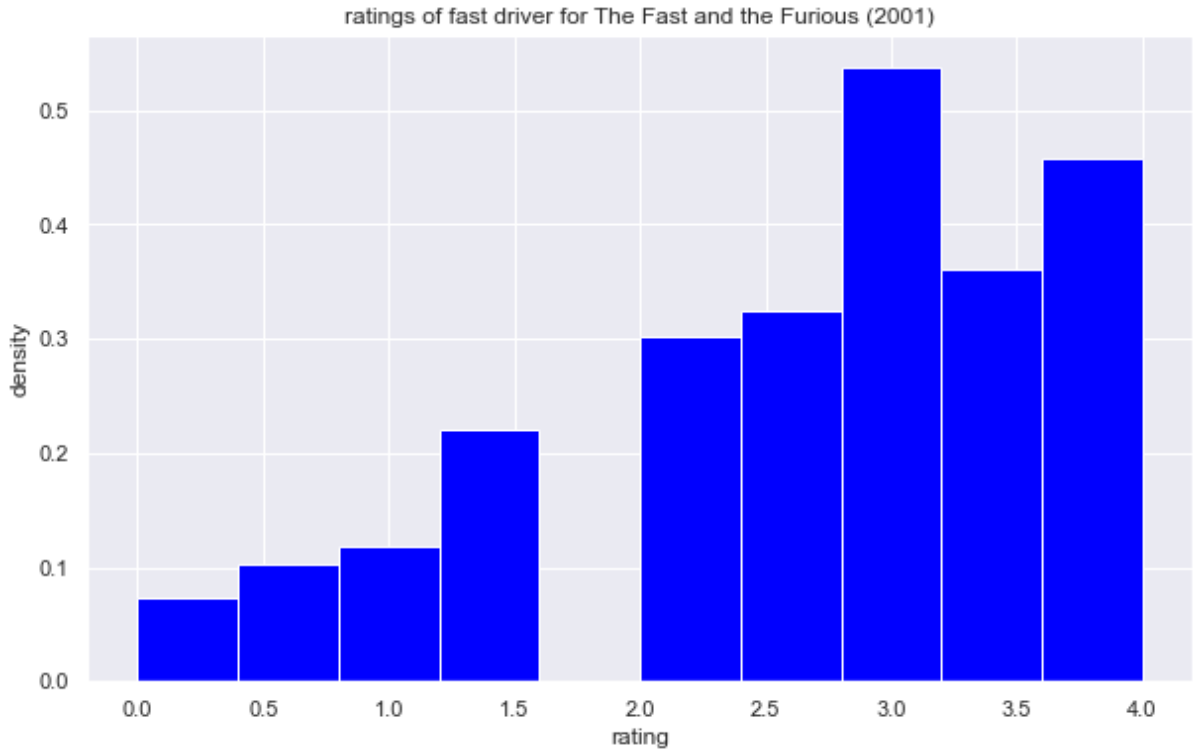
In [168]:
```
not_fast_driver_filt =df["I enjoy driving fast"]<=3
fast_driver_filt = df["I enjoy driving fast"]>3

not_fast_driver_data  = df.loc[not_fast_driver_filt,"The Fast and the Fu
fast_driver_data  = df.loc[fast_driver_filt,"The Fast and the Furious (20
print(len(fast_driver_data),len(not_fast_driver_data))
```

```
339 232
```

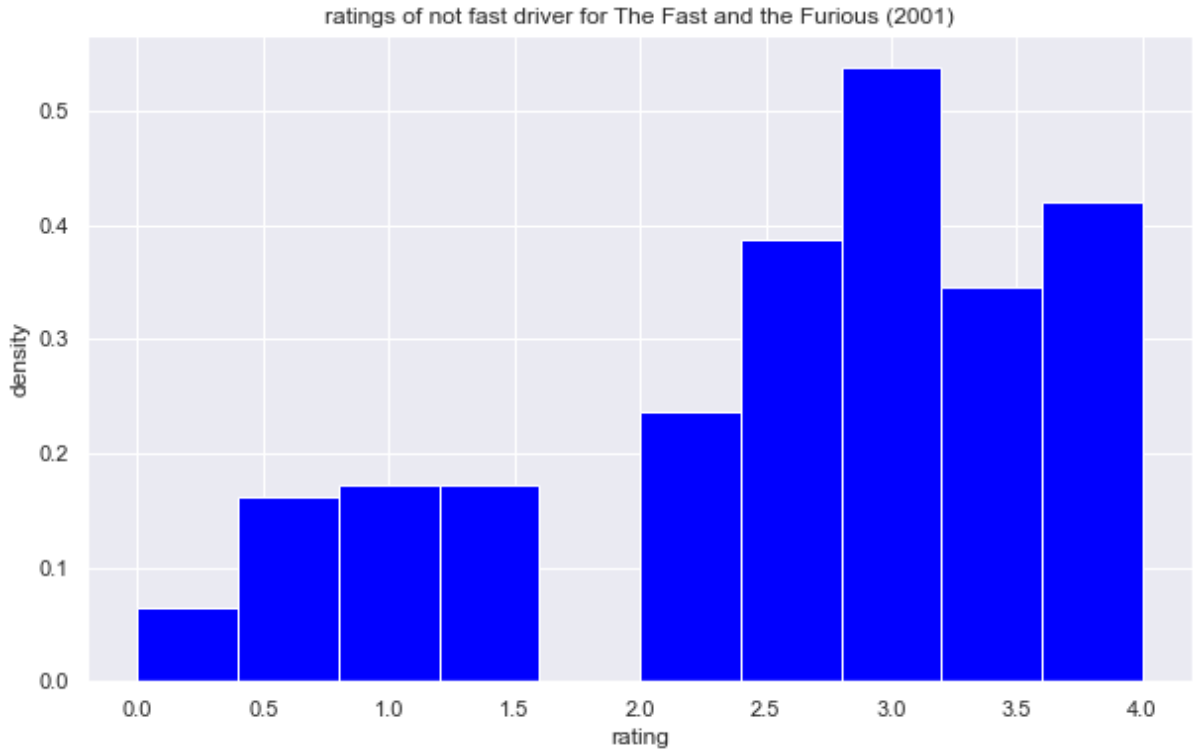
In [169]:
```
# check variance within two groups
print(fast_driver_data.var(),not_fast_driver_data.var())
```

```
1.1617328425614117 1.231081859393579
```

In [174]:
```
plt.figure(figsize=(10,6))
plt.hist(fast_driver_data,color='blue',density = "True")
plt.title('ratings of fast driver for The Fast and the Furious (2001)')
plt.xlabel('rating')
plt.ylabel('density')
plt.show()
```



In [175]:
```
plt.figure(figsize=(10,6))
plt.hist(not_fast_driver_data,color='blue',density = "True")
plt.title('ratings of not fast driver for The Fast and the Furious (2001)
plt.xlabel('rating')
plt.ylabel('density')
plt.show()
```



In [170]:
```
# in that case we do one side test since our alternative hypothesis is wh
# Welch's one side t-test
print("Welch's one side t-test:",stats.ttest_ind(fast_driver_data,
            not_fast_driver_data, alternative='greater',equal_var = False

# student one side t-test
print("student's one side t-test:",stats.ttest_ind(fast_driver_data,
            not_fast_driver_data,alternative='greater', equal_var = True)

# one side permutation test for mean difference
test_data = (fast_driver_data, not_fast_driver_data)
p_test = permutation_test(test_data, test_stat_func,alternative='greater
print('Test stat for permutation test:', p_test.statistic)
print('P-val for permutation test:',p_test.pvalue)
```

```
Welch's one side t-test: Ttest_indResult(statistic=0.7090812695416734,
pvalue=0.23930678495554936)
student's one side t-test: Ttest_indResult(statistic=0.713040942706280
7, pvalue=0.23805637034315508)
Test stat for permutation test: 0.06639075373817516
P-val for permutation test: 0.2410758924107589
```