summary

November 7, 2022

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]

I used the median of counts of movie ratings as threshold and split movies into high vs low popularity movies (two groups). Since the movie ratings are meaningless if reduced to mean and we are testing two sample, I used the Mann Whiteney U test instead of t-test to test whether there is a difference in frequency of ratings between high popularity movies and low popularity moviews. For question 1 to 8, I used the Mann Whitney U test for the same reason, but according to different test circumstance I used different hypothesis. For question 1, the null hypothesis is that high popularity movies would are not higher than low popularity movies. Under $\alpha = 0.005$ We get a p-value that is much less than 0.005, so there is evidence that popular movies have higher ratings than movies that are not that popular.

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]

I also do a median split to determine older, newer movies. The median of release year of movies in this dataset is 1999, so I splited them, into movie released before 1999 as the group of older movies and movies released after (including) 1999 as the group of newer movies. Again, for movie ratings, it is more reasonable to use a Mann Whitney U test to detect difference in frequency than difference in mean. The null hypothesis of this test is that ratings of older movies and newer movies are the same. The p value is really small (1.2849216001533932e-06), at $\alpha = 0.005$ significance level, we rejected the null hypothesis that movies released in recent years are rated the same as movies released earlier.

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

I compared the Shrek ratings by male and female using the Mannu Whitney U test. Since we want to test whether the two group of rating has difference, the null hypothesis is that males rating and females rating are the same (no difference). The althernative hypothesis is that there is a difference between male/female ratings. The p-value is 0.050536625925559006, at 0.005 significance level, we failed to reject that Shrek (2001) is rated differently by male and female.

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

Using the Mann Whitney U test as in question 3 for all the movies, under the null that movies are rated the same by male and female, there are 12.5% of the movies have a p-value that is less than 0.005, so under $\alpha = 0.005$, approximately 12.5% of the movies are rated differently by male and female.

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

I compared The Lion King 1994 ratings between those who are single child and those who with siblings using the Mann Whitney U test. The null hypothesis is that rating by single child are not greater (i.e. less or equal) than rating by people with siblings and the alternative hypothsis is that single child enjoy the movie more. At 0.005 significance level, we fail reject the null hypothesis (p = 0.978419092554931). As a result, we have no evidence that rating of only child is greater than rating of who with siblings.

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

Using the same test and hypothesis for question 6 on each movies, and under 0.005 significance level, we find that 10% of the cases we generated a p-value that is less than 0.005. So under $\alpha = 0.005$, there are 1.75% of movies have "only child effect".

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?

Since it is comparing I filtered out who enjoy watching movie socially and who enjoy watching alone and used the Mann Whitney U test to test whether the ratings of who like to watch movies socially is higher than those who like to watch movies alone. The null hypothesis is that the rating of who like to watch socially is less or equal to the rating of who like to watch moview alone. The p-value of this test is 0.9436657996253056, which fail to reject the null under 0.005 significance level. We have no evidence to believe people who like to watch movies socially enjoy 'The Wolf of Wall Street (2013)' more than those who prefer to watch them alone.

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

By performing the same test in question 7 on every movie in the dataset, 2.5% of the movies generated a p-value less than 0.005. So under the 0.005 significance level, 2.5% of the movies exhibit such a "social watching" effect.

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

This is still two-sample test scenario, but we want to compare whether the two distributions of samples came from are the same. Instead of Mann Whitney U test, I used the KS (Kolmogorov-Smirnov) test, it is a kind of goodness-of-fit test to compare distributions. I filtered out the movie ratings of Home Alone and Finding Nemo as the two samples and conducted the KS test. The null hypothesis is that the distributions of these two samples are the same. The alternative hypothesis is that they are not the same. The p-value is 6.379397182836346e-10, which reject the null hypothesis that the two distribution are the same, under 0.005 significance level. So we have evidence to say the ratings distribution of 'Home Alone (1990)' different than that of 'Finding Nemo (2003)'.

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]

I first filtered out all the franchise movies, and there are more than two movies for each franchise, so we cannot use Mann-Whitney again. And we cannot use normal one-way anova, since the data is movie rating. I found that Kruskal-Wallis H-test is a non-parametric version of ANOVA, which is to test the median for 2 or more samples, which may have different sample sizes. I also find the

Friedman test is a appriopriate in this circumstance but it does not support different sample size for each sample, so I finally choose Kruskal-Wallis H test.

The null hypothesis of the test is that all movies in the franchise have consistent quality (the median of the ratings have no difference). Using the KW H-test on the 8 franchise movies, I found that only the Harry Potter Series generated a p-value higher than 0.005, that is saying, under 0.005 significance level, We fail to reject that Harry Potter Series has consistent quality (p = 0.34331950837289205), while for the other 7 franchise movies, we rejeceted that their quality is consistent. Inconclusion, 7 out of 8 franchise movies have inconsistent quanlity.

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].

I tried to find out whether newer released moview are more popular than older movies (have more ratings). I would use the median split as in question 2. For this two sample circumstance, I would use the two-sample t-test, since I only care about count of rating, so I only need to test whether there is a difference in the mean of counts. And I assume the rating counts are independent, so I would use the independent t-test.

By spliting the movie ratings into those before 1999 and after 1999, I used the independent t-test and rejected the null that the count of newer movie ratings are less or equal to the count of older movie ratings under 0.005 significance level (p = 4.2443663905911793e-05). That it is to say, we have evidence to believe that newer movies are more popular, or younger generation movie viewers are more likely to give ratings of movie.

Concerns on this approach is that the rating of same person might be correlated, so it is more reasonable to use correlated t-test. But that might lose too much data, dropping all na row-wisely, I only got two rows left, and it is meaningless to do a test on two data point.

0.0.1 Appendix

[ ]:
```
# q1
import pandas as pd
import numpy as np
from scipy import stats
df = pd.read_csv('movieReplicationSet.csv')
df1 = df.iloc[:,:400]
pop = df1.loc[:,df1.isna().sum() < df1.isna().sum().median()]
not_pop = df1.loc[:,df1.isna().sum() >= df1.isna().sum().median()]
size = pop.shape[1]
array_pop = np.empty(0)
for i in range(size):
    a = pop.iloc[:,i].dropna()
    array_pop = np.concatenate((array_pop, a), axis=None)
array_not_pop = np.empty(0)
for i in range(400-size):
    b = not_pop.iloc[:,i].dropna()
    array_not_pop = np.concatenate((array_not_pop, b), axis=None)
```

```
# simple summary
# use mannwhitney u test
u,p = stats.mannwhitneyu(array_pop, array_not_pop, alternative="greater")
print(u, p)
```

```
1242808144.5 0.0
```

[ ]:
```
# q2
# select movie names
df1.columns[0]
list = []
# get the release year from the name
for i in range(400):
    list.append(int(df1.columns[i][-5:-1]))
median_year = np.median(list)
# split by release year
newer_movies = df1.iloc[:,list >= median_year]
older_movies = df1.iloc[:,list < median_year]
size = newer_movies.shape[1]
array_newer = np.empty(0)
for i in range(size):
    a = newer_movies.iloc[:,i].dropna()
    array_newer = np.concatenate((array_newer, a), axis=None)
array_older = np.empty(0)
for i in range(400-size):
    b = older_movies.iloc[:,i].dropna()
    array_older = np.concatenate((array_older, b), axis=None)
u,p = stats.mannwhitneyu(array_older, array_newer, alternative="two-sided")
p
```

[ ]:
```
1.2849216001533932e-06
```

[ ]:
```
# q3
female = df.loc[df.iloc[:,-3] == 1, :].loc[:, "Shrek (2001)"].dropna()
male = df.loc[df.iloc[:,-3] == 2, :].loc[:, "Shrek (2001)"].dropna()
u,p = stats.mannwhitneyu(female, male, alternative = "two-sided")
p
```

[ ]:
```
0.050536625925559006
```

[ ]:
```
# q4
sum = 0
for i in range(400):
    female = df.loc[df.iloc[:,-3] == 1, :].iloc[:,i].dropna()
    male = df.loc[df.iloc[:,-3] == 2, :].iloc[:,i].dropna()
    u,p = stats.mannwhitneyu(female, male, alternative = "two-sided")
    if p < 0.005:
        sum += 1
```

```
str(sum/400*100) + '%'
```

[ ]: '30.5%'

[ ]:
```
# q5
only_child = df.loc[df.iloc[:,-2] == 1, :].loc[ : , "The Lion King (1994)"].
 ↪dropna()
not_only_child = df.loc[df.iloc[:,-2] == 0, :].loc[ : , "The Lion King (1994)"].
 ↪dropna()
u,p = stats.mannwhitneyu(only_child, not_only_child, alternative = "greater")
p
```

[ ]: 0.978419092554931

[ ]:
```
# q6
sum = 0
for i in range(400):
    only_child  = df.loc[df.iloc[:,-2] == 1, :].iloc[:, i].dropna()
    not_only_child = df.loc[df.iloc[:,-2] == 0, :].iloc[:, i].dropna()
    u,p = stats.mannwhitneyu(only_child, not_only_child, alternative =␣
 ↪"two-sided")
    if p < 0.005:
        sum += 1
sum/400
```

[ ]: 0.0175

[ ]:
```
# q7
alone = df.loc[df.iloc[:,-1] == 1, :].loc[ : , "The Wolf of Wall Street␣
 ↪(2013)"].dropna()
socially = df.loc[df.iloc[:,-1] == 0, :].loc[ : , "The Wolf of Wall Street␣
 ↪(2013)"].dropna()
u,p = stats.mannwhitneyu(socially, alone, alternative = "greater")
p
```

[ ]: 0.9436657996253056

[ ]:
```
# q8
sum = 0
for i in range(400):
    alone  = df.loc[df.iloc[:,-1] == 1, :].iloc[:,i].dropna()
    socially = df.loc[df.iloc[:,-1] == 0, :].iloc[:,i].dropna()
    u,p = stats.mannwhitneyu(alone, socially, alternative = "two-sided")
    if p < 0.005:
        sum += 1
sum/400
```

[ ]:
```
0.025
```

[ ]:
```
# q9
home_alone = df.loc[:,"Home Alone (1990)"].dropna()
finding_nemo = df.loc[:,"Finding Nemo (2003)"].dropna()
d, p = stats.ks_2samp(home_alone, finding_nemo)
p
```

[ ]:
```
6.379397182836346e-10
```

[ ]:
```
# q10
def func(s,x):
    return s in x
franchises = ["Star Wars", "Harry Potter", "The Matrix", "Indiana Jones",␣
 ↪"Jurassic Park", "Pirates of the Caribbean", "Toy Story", "Batman"]
list = [[False]*len(df1.columns) for i in range(len(franchises))]
for j in range(len(franchises)):
    for i in range(len(df1.columns)):
        list[j][i] = (func(franchises[j], df1.columns[i]))
```

[ ]:
```
# Star Wars
star_wars = df1.loc[:,list[0]]
h,p = stats.kruskal(star_wars.iloc[:,0].dropna(),
star_wars.iloc[:, 1].dropna(),
star_wars.iloc[:, 2].dropna(),
star_wars.iloc[:, 3].dropna(),
star_wars.iloc[:, 4].dropna(),
star_wars.iloc[:, 5].dropna())
p
```

[ ]:
```
8.01647736660335e-48
```

[ ]:
```
# Harry Potter
harry_potter = df1.loc[:,list[1]]
h,p = stats.kruskal(harry_potter.iloc[:,0].dropna(),
harry_potter.iloc[:, 1].dropna(),
harry_potter.iloc[:, 2].dropna(),
harry_potter.iloc[:, 3].dropna())
p
```

[ ]:
```
0.34331950837289205
```

[ ]:
```
# The Matrix
the_matrix = df1.loc[:,list[2]]
h,p = stats.kruskal(the_matrix.iloc[:,0].dropna(),
the_matrix.iloc[:,1].dropna(),
the_matrix.iloc[:,2].dropna())
p
```

[ ]: 3.1236517880781424e-11

[ ]:
```
# Indiana Jones
indiana = df1.loc[:,list[3]]
h,p = stats.kruskal(indiana.iloc[:,0].dropna(),
indiana.iloc[:, 1].dropna(),
indiana.iloc[:, 2].dropna(),
indiana.iloc[:, 3].dropna())
p
```

[ ]: 6.27277563979608e-10

[ ]:
```
# Jurassic Park
jurassic = df1.loc[:,list[4]]
h,p = stats.kruskal(jurassic.iloc[:,0].dropna(),
jurassic.iloc[:, 1].dropna(),
jurassic.iloc[:, 2].dropna())
p
```

[ ]: 7.636930084362221e-11

[ ]:
```
# Pirates of the Caribbean
pirates = df1.loc[:,list[5]]
h,p = stats.kruskal(pirates.iloc[:,0].dropna(),
pirates.iloc[:, 1].dropna(),
pirates.iloc[:, 2].dropna())
p
```

[ ]: 3.2901287079094474e-05

[ ]:
```
# Toy Story
toy = df1.loc[:,list[6]]
h,p = stats.kruskal(toy.iloc[:, 0].dropna(),
toy.iloc[:, 1].dropna(),
toy.iloc[:, 2].dropna())
p
```

[ ]: 5.065805156537524e-06

[ ]:
```
# Batman
batman = df1.loc[:,list[7]].dropna()
h,p = stats.kruskal(batman.iloc[:, 0].dropna(),
batman.iloc[:, 1].dropna(),
batman.iloc[:, 2].dropna())
p
```

[ ]: 4.1380499020034183e-19

[ ]:
```
# extra points
rows = newer_movies.shape[0]
newer_movie_rating_counts = []
older_movie_rating_counts = []
for i in range(newer_movies.shape[1]):
    newer_movie_rating_counts.append(rows - newer_movies.iloc[:,i].isna().sum())
for i in range(older_movies.shape[1]):
    older_movie_rating_counts.append(rows - older_movies.iloc[:,i].isna().sum())
t1,p1 = stats.ttest_ind(newer_movie_rating_counts, older_movie_rating_counts,␣
 ↪alternative="greater")
p1
```

[ ]:
```
4.2443663905911793e-05
```

[ ]:
```
row_wise = df1.dropna()
row_wise
```

[ ]:
```
     The Life of David Gale (2003)  Wing Commander (1999)  \
430                            0.0                    0.0
999                            3.0                    3.0

     Django Unchained (2012)  Alien (1979)  \
430                      4.0           0.5
999                      2.5           3.0

     Indiana Jones and the Last Crusade (1989)  Snatch (2000)  \
430                                        0.0            0.0
999                                        2.5            3.0

     Rambo: First Blood Part II (1985)  Fargo (1996)  \
430                                0.5           0.5
999                                2.5           2.5

     Let the Right One In (2008)  Black Swan (2010)  …  X-Men 2 (2003)  \
430                          0.0                3.0  …             0.0
999                          2.5                2.5  …             3.0

     The Usual Suspects (1995)  The Mask (1994)  Jaws (1975)  \
430                        0.0             0.0          0.5
999                        3.5             3.5          3.5

     Harry Potter and the Chamber of Secrets (2002)  Patton (1970)  \
430                                             0.0            0.0
999                                             3.5            3.0

     Anaconda (1997)  Twister (1996)  MacArthur (1977)  \
430              0.5             0.0               0.0
```

```
999              3.5             3.5               3.5

     Look Who's Talking (1989)
430                          0.0
999                          3.5

[2 rows x 400 columns]
```