1 Data preparation

1.1 Data Retrieving

When load the csv file and save it to data frame starWar, I skipped the first two rows which are header and set the first column index. Column names were changed to succinct names. starWar.head().T was used to print the first 5 rows of the data vertically to preview. starWar.dtypes was used to check data types of each columns. There were objects and float64s in the data frame. starWar.shape was used to check how many rows and columns in the data frame. There were 1186 rows and 37 columns, which was the same with the original csv file.

1.2 Data Check

I defined a function checkValue() to check distinct values in each column to have an overall preview of the data set and understand how this survey and answers designed. By checking the values, we can see some typos and impossible values in some columns. Then a function checkNa() was defined to check total number of missing values in each columns. I found that in most movie related columns there were around 350 missing values, so I presumed that there were around 350 people did not answer most of these questions. Similarly, there were around 140 missing values in demographic columns. In csv file, I filtered by column “Star War Fan” and found that those who left it blank did not answer other movie related questions neither. Similarly, those who left “Gender” blank did not answer other demographic questions. These rows have very limited value and are removed. Although there are 16 rows with movie related columns filled and no demographic questions answered, they occupied less than 2% percent of the total data. I presume drop them will not affect the whole distribution in this research.

1.2.1 Find the Issues

Then I took a closer look at distinct values and missing values in each column and found following issues:

The “Seen Star War” column has extra-whitespaces to be trimmed. The “Star War Fan” column has typo “Yess” and “Noo” to be amended and missing values to be filled. From “Seen Episode 1” to “Seen Episode 6” columns, there are missing values to be filled. From “Episode 1 Rank” to “Episode 6 Rank” columns, data types are float64 with missing values to be filled. The rank values need to be converted to categorical values with order. Attitude to characters group has missing values and needs to be converted to categorical value with order. The answer “Neither favorably nor unfavorably (neutral)” is verbose and can be replaced by more succinct answer. The “Character Shot 1st” column has missing values to be filled. The answer "I don't understand this question" is verbose and can be replaced by more succinct answer. Column “Universe Familiar” has missing values. Column “Universe Fan” has missing values and typo “Yess”. Column “Star Trek Fan” has typo “Noo” and lower-case strings “yes” and “no”. Column “Gender” has lower case strings “male” and “female”, missing values and typo “F” for “Female”. Column “age” has impossible value “500” and missing values. It also needs to be converted to categorical data with order. Columns “Household Income” and “Education” have missing values and need to be converted to categorical data with order. Dollar signs are not showing in the data. Column “location” have missing values. To make string values case insensitive, all string values need to be converted to upper case. All other columns with “Yes” or “No” answers are object data type. They all need to be converted to categorical data with no order and create a new category “No Answer” to fill missing values.

1.3 Implementation

After understanding the data and locating the issues, I started to amend them in a systematic way. Rows with limited value were removed firstly by using masks.

1.3.1 Three group columns

I started from the three group questions including “Seen Episode 1” to “Seen Episode 6”, “Episode 1 Rank” to “Episode 2 Rank” and “Attitude to Han Solo” to “Attitude to Yoda”. In each group, they have common issues and they are quite different from other single columns. I created three lists to store these groups of columns and manipulate them by group.

In “Seen Episode” group of questions, the answers are names of episodes or missing value. Assume that those who seen the episode will have the episode name in the answer and missing value means have not seen. I created a list named “seenList” to store this group of columns. I filled missing values with “NO” and changed episode names to “YES” with for-loop iterations for the whole group. After that I converted columns in this list to categorical data without order.

Ranking questions` data type is float64, values from 1.0 to 6.0 and missing values. I filled missing values with 0 and converted data type to categorical integer data with order.

For attitude questions, I filled missing values with “No Answer” and replaced answer “Neither favorably nor unfavorably (neutral)” by more succinct answer “Neutral”. I converted all string values to upper case strings and converted data type to categorical data with order from “VERY FAVORABLY” to “VERY UNFAVORABLY” and created a new category “No Answer”.

1.3.2 Check and replace missing values

After addressing issues in above groups, I started to amend issues in other columns. I filled missing values in the rest columns with “NO ANSWER”. Then I checked missing values for each column. All missing values were replaced.

1.3.3 Convert string values to uppercase and trim extra-whitespaces

I created a list “upperList” to store columns with string values who need to be converted to upper case. Then I used a for-loop to convert string values to upper case. I did the same way for columns who need to be stripped by creating a list “stripList” and using iteration.

1.3.4 Correct typos

I corrected typos in “Star War Fan”, “Universe Fan”, “Star Trek Fan” and “Gender” by using masks and assigned correct values to the selected cells.

1.3.5 Sanity check

For column “Age”, my strategy for sanity check is as follow. I created a list to store all possible values in the column. If a value is not in the list, it will be treated as an impossible value. Then I created a mask to store the index of impossible values. There was only one impossible value “500”. I made a sensible guess that the answer should be “50”, so I replaced it to “45-60”.

1.3.6 Replace dollar signs

To prevent the pairs of bare dollar signs from being interpreted as math markup, replace the bare $ with \$ in column “Household Income”. (unutbu, 2016)

1.3.7 Convert to categorical data

I converted “Age”, “Household Income” and “Education” to categorical data with order. For the rest columns whose data types are objects, because they only have limited answers, I created a list “categoricalList” and converted them to non-order categorical data with a for-loop.

1.3.8 Final check

After all these steps finished, I checked distinct values in each column to make sure all typos, impossible values and extra-whitespaces have been fixed. All missing values are replaced. Data types of all columns have been converted to categorical data. The shape of the data frame is (820, 37) now, which means there are 820 valuable rows left for further exploration. All the implementations succeed, and the data is ready for the next step.

2 Data Exploration

In this task, I flited data from people who seen the movie and saved it to a data frame “star_seen” for further investigation. The following explorations are all based on data frame “star_seen”.

2.1 Explore a survey question

To have an overall ranking score of each episode, I calculated the average episode ranking score for each episode (Survey Monkey). I assigned different weights to each rank. Rank 1 has the highest weight 6. Rank 2 weights 5. Rank 3 weights 4. Rank 4 weights 3. Rank 5 weights 2. Rank 6 weights 1. “NO ANSWER” weights 0. Then I calculated each episode`s average score with the following equation and rounded to two decimal places:

$$\text{avgScore} = \sum_{i=1}^{6} (Percentage\ of\ \ Rank\ i * Weights)$$

As can be seen in Figure 1, Episode 5 has the highest average score 4.49. Episode 6 is the second favorable episode with average score 3.96, then followed by episode 4 which scored 3.74. Episode 3 is the least favorable episode with average score 2.65.

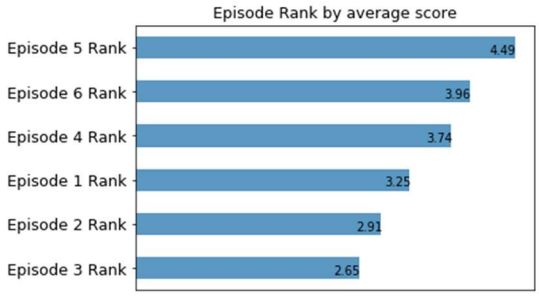

Figure 1

To have a more detailed analysis, I plotted bar charts with ranks and their percentage for each episode.

Regarding to the most popular episode 5, illustrated in Figure 2, there are 35% people ranked 1 and 28% people ranked 2. Only 5% of people ranked 6.

For the second favorable episode 6, illustrated in Figure 3, although only 18% ranked 1, most people ranked 2 and 3 which occupied 28% and 26% respectively.

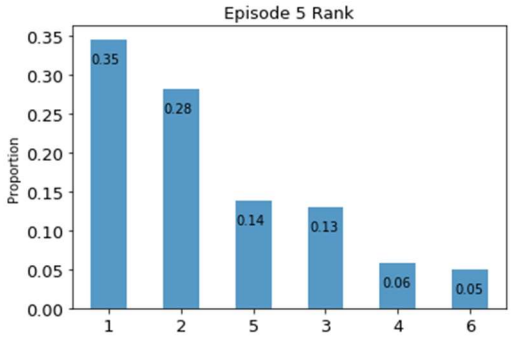

Figure 2

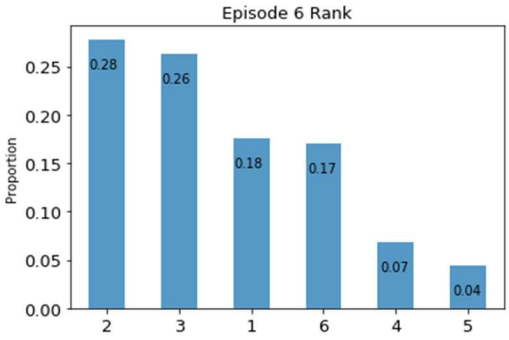

Figure 3

Episode 4 is the most controversy episode. As can be seen in Figure 4, 25% people ranked 1 while 19% people ranked 6.

For episode 1, illustrated in Figure 5, most people ranked 4 which occupied 28% followed by 20% people ranked 6.

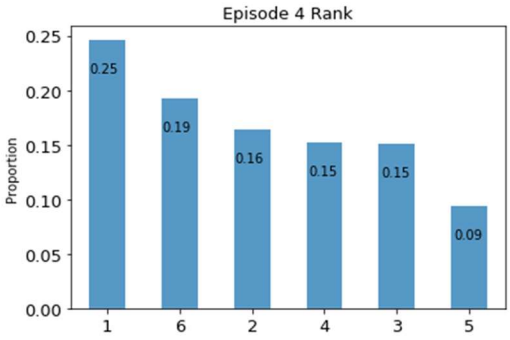

Figure 4

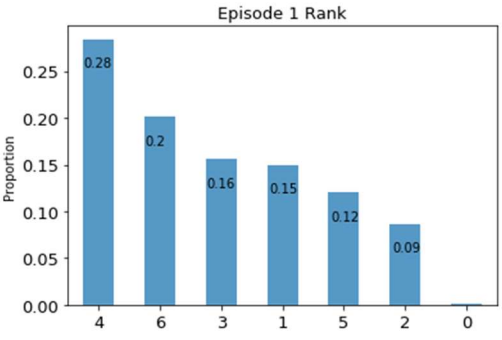

Figure 5

For episode 2, illustrated in Figure 6, 36% people ranked 5 while only 4% of people rated 1.

The least favorable is episode 3. As can be seen in Figure 7, most people ranked 6 and 5 which occupied 26% and 24% respectively. By contrast, only 4% ranked 1 and 5% ranked 2 regarding to episode 3.

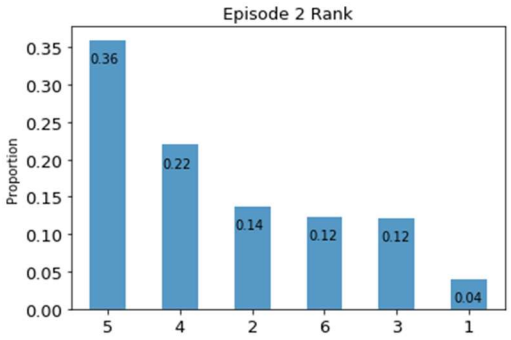

Figure 6

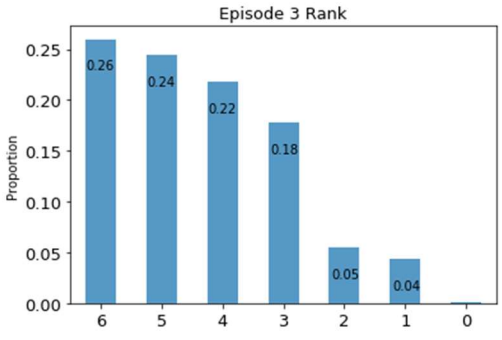

Figure 7

2.2 Relationships between columns

2.2.1 Hypothesis 1: Star Trek fans are probably also Star War fans

Star War and Star Trek share many similarities and commonalities. They both have their origins in the space western subgenre and both stories depict societies consisting of multiple planets and species (Wikipedia). As can be seen in figure 8, 89% of Star Trek fans are also Star War fans.

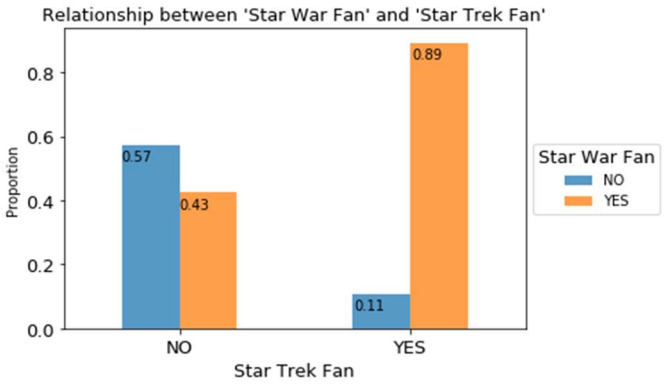

Figure 8

2.2.2 Hypothesis 2: Most Star War fans choose Han shot first while most non-fans don`t know the question

The controversy over who shot first refers to a change in the 1997 Special Edition of Star Wars. Han shoots Greedo dead in the original version while in later versions, Greedo attempts to fire at Han first. Many fans believe that Han shot first makes his later transition from anti-hero to hero more meaningful (Wikipedia). As can be seen in figure 9, 49% fans chose Han shot first while 28% fans chose Greedo shot first. 64% non-fans don`t understand the question.

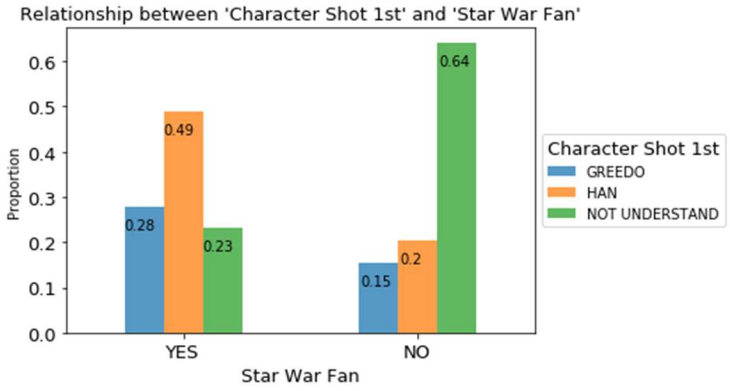

Figure 9

2.2.3 Hypothesis 3: Star War fans rank higher on Episode 4 as it is the first released episode

Star Wars: Episode IV – A New Hope is the first of the franchise released in 1977 (Wikipedia). As illustrated in figure 10, people have different opinions on Episode 4 ranking. 29% fans ranked 1 and 21% fans ranked 2. On the other hand, 25% non-fans ranked 6 and 24% non-fans ranked 4.

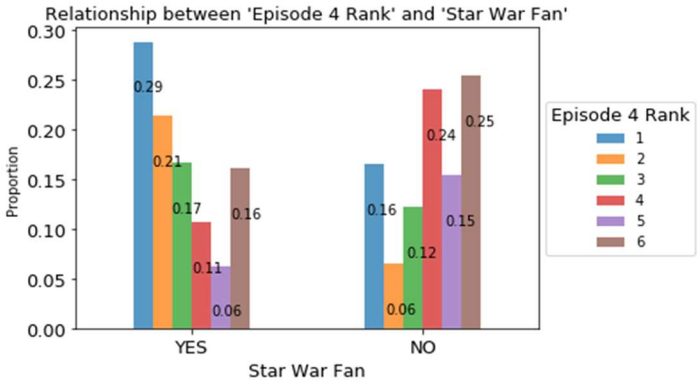

Figure 10

2.3: Explore a specific relationship

By checking the bar charts of demographic columns, the distribution of "Household Income", "Education" and "Location" are very uneven. There are much less data from some categories. For example, there are only 3 people who have "less than high school degree" and only 4% of people from "East South Central". To avoid biased analysis, I focused on exploring the relationships between "Gender", "Age" and their attitudes to Start War characters by plotting bar charts. I examined each bar chart to explore if there is any surprising relationship. Followings are what I found.

2.3.1 Gender and Attitude

Most of males (43%) chose very favorably regarding to Darth Vader while many females (23%) chose very unfavorably.

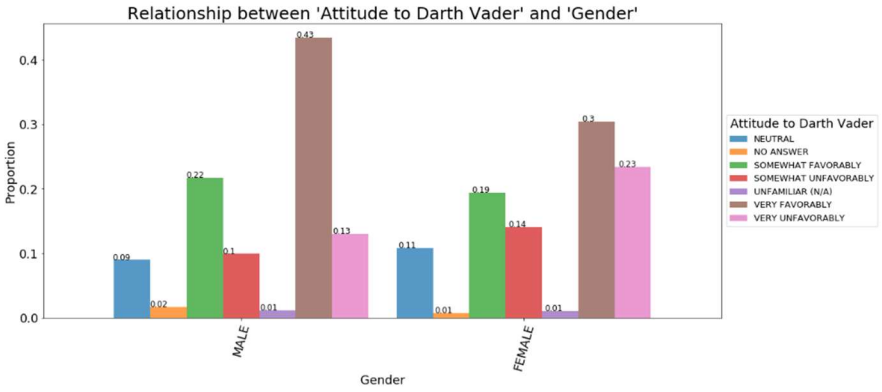

Figure 11

31% of males chose very unfavorable to Jar Jar Binks while much less females (18%) agreed.

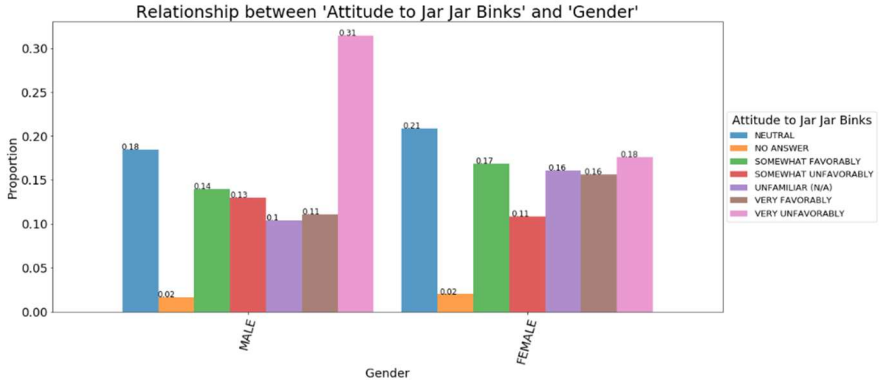

Figure 12

2.3.2 Age and Attitude

Younger people tend to like Boba Fett. There are 23% of people age 18 to 29 and 24% of people age 30 to 44 chose very favorably to Boba Fett while only 8% of people age 45 to 60 and 11% of people older than 60 have the same choice.

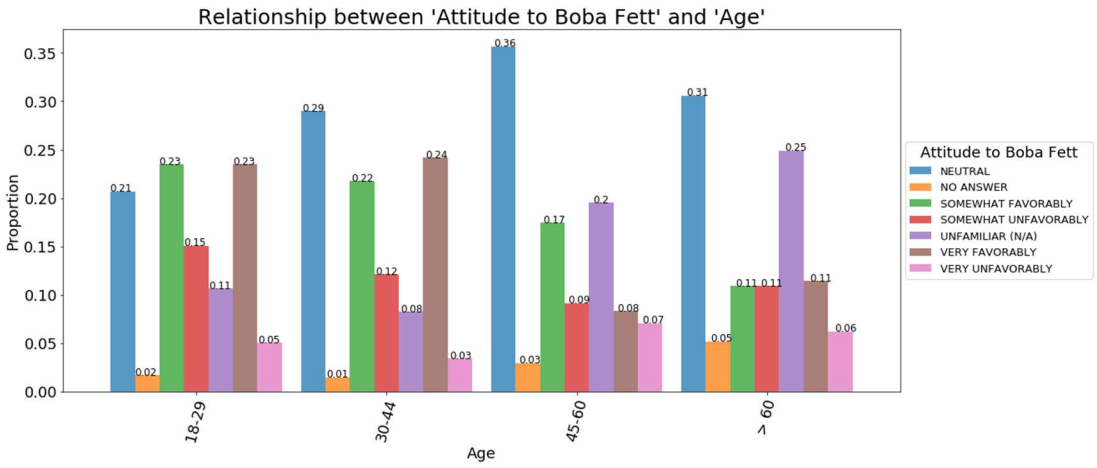

Figure 13

By contrast, younger people tend to dislike Jar Jar Binks. There are 31% of people age 18 to 29 and 36% of people age 30 to 44 chose very unfavorably to Jar Jar Binks while only 11% of people older than 60 chose the same answer.

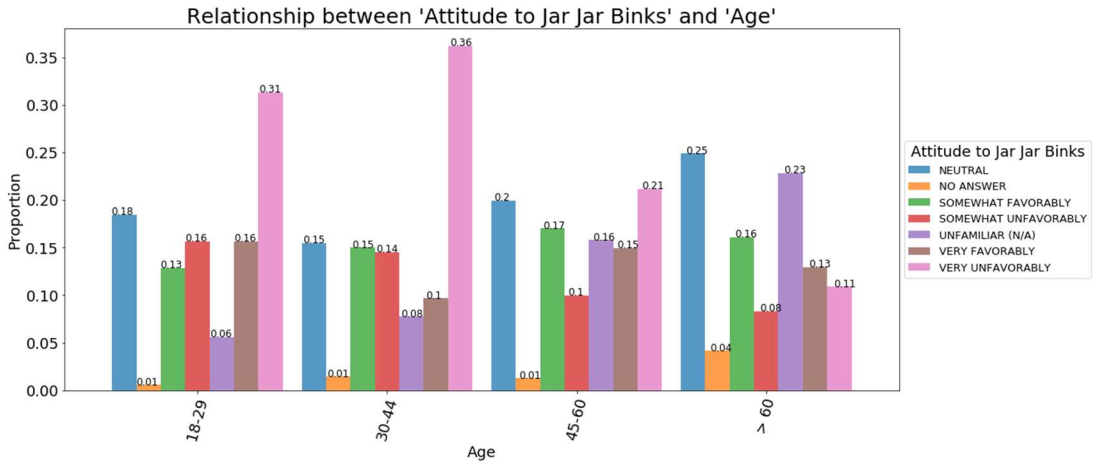

Figure 14

2.4 Further analysis

For further exploration, the function dataFrame2BarPlot () and compareListPlot () can be used for visualizing more column pairs. Machine learning algorithms can be used for classification or other analysis.

References:

Survey monkey, *Ranking Question*, viewed 10 April 2020, <https://help.surveymonkey.com/articles/en_US/kb/How-do-I-create-a-Ranking-type-question>.

Stack overflow, *Pandas to MatPlotLib with Dollar Signs*, 01 July 2016, viewed 13 April 2020, <https://stackoverflow.com/questions/38151646/pandas-to-matplotlib-with-dollar-signs?noredirect=1>.

Wikipedia, *Comparison of star trek and star wars*, viewed 10 April 2020, <https://en.wikipedia.org/wiki/Comparison_of_Star_Trek_and_Star_Wars>.

Wikipedia, *Han shot first*, viewed 10 April 2020, <https://en.wikipedia.org/wiki/Han_shot_first>.

Wikipedia, *Star Wars*, viewed 10 April 2020, <https://en.wikipedia.org/wiki/Star_Wars_(film)>.