COSC2670

Practical Data Science with Python

Semester 1, 2020

Assignment 1 – Data Cleaning and Summarising

Name: Duncan Do

Student Number: s3718718

Contents Page

Data Preparation

Task 1.2 – Check Data Types

First step in rectifying any incorrect data types is to identify what are the data types of each column when the file was imported. Once all the datatypes have been identified, the incorrect ones (decided based on the csv file which the data was based on) are re-written with the command ***<column name>.astype(<desired data type>***. This was done to all columns that did not match the data type. Once completed, a check on all column data types was done again. Note: string data types still come up as object when checking with ***<table name>.dtypes***, this is simply the nature of how python refers to string variables; it can still be guaranteed that the column is of type string due to the 'astype' function performed.

Task 1.3 – Typos

To ensure all responses to the survey questions were left without spelling errors, we must check what the actual responses were. Using the ***<column name>.value_counts()*** command, we can identify every response given in that column as well as its frequency. Here we can examine the natural 'correct' answers and any outliers that exist beyond that correct subset. This can the be rectified by using the ***<column name>.replace(<incorrect value>, <correct value>, inplace = True)*** to replace the incorrect value with its most logical counterpart (E.g. replace 'Yess with Yes'). Other alternatives are replacing it with the mean value (For integer and integer related fields) or an empty/na kind of state.

Task 1.4 – Extra-whitespaces

A similar initial process to typos will be done to check for extra-whitespaces (This whole process only applies to string or string like typed fields. First, identify all the responses using ***value_counts().*** Secondly, use the ***<column name>.str.strip()*** to remove all white spaces from any response in that column; this broader use of the function should only be used for columns where the responses are limited to singular words (E.g. Answers are 'Yes' or 'No'). More complex extra-whitespaces will need more homed in applications of the function; thankfully, none such cases of extra-whitespaces exist in this data set. Note: Extra-whitespaces can be easily spotted if 2 seemingly identical responses are both listed from ***value_counts()***, resulting to the invisible nature of whitespaces (E.g. "Apple" and "Apple ").

Task 1.5 – Upper/Lower-case

To force a value to transform into upper-case or lower-case requires the ***<column name>.str.upper()*** (Or ***…str.lower()***). In this dataset, any case of needed to force a value to 'upper' or 'lower' were taken care of using other means and functions, such as using ***replace()***.

Task 1.6 – Sanity Checks

As previously stated, a combination of ***value_counts()*** and ***replace()*** can identify and rectify events of 'incorrect values. This covers impossible values that you would come across sanity checks. If ***value_counts()*** identifies a value that is deemed 'impossible', ***replace()*** can be used to rectify it, using the correction same methods as typos. (Replace with correct counterpart, mean value or empty/na state)

Task 1.7 – Missing Values

From the function ***value_counts()***, missing values will show up as NaN. Once identified, the function ***<column name>.fillna(<replacement for NaN>, inplace = True)*** can be used to fill in all NaNs with an appropriate substitute. Once again, the replacements being values such as the mean value or an empty/na state.

Data Exploration

Task 2.1 – Explore a Survey Question

- Process of subdividing the data set to isolate and analyse how people rank the Star Wars Movies.

To avoid working with unrelated data, a secondary table was created with just the 6 columns ranking the 6 movies. Then any rows with the previously entered "Empty state" (0) were removed to not compromise the results when calculating the overall ranking of the movies. For a concise view of the perception of each movie, the ranking values for each movie were compressed into a singular average value using the ***<column name>.mean()*** function. These values were then placed into their own table so they can be viewed together. Here, a clear ascending list of which Star Wars movie was overall ranked least to best can be seen. Further analysis was done by using ***value_counts()*** to view the frequency of each rank for each movie.

- Why these methods were implemented.

These methods were implemented in order to obtain an accurate perception of the movies without over complex comparisons between the 6 columns in their entirety. Thus they were compressed into easy to read average values. The further analysis of the frequency of each rank was to see which movie was most favoured to be the "best" and "worst".

- What was found.

It appears the popularity of Star Wars movies, from most popular to least is as followed:

1. Star Wars Episode 5: Revenge of the Sith
2. Star Wars Episode 6: Return of the Jedi
3. Star Wars Episode 4: A New Hope
4. Star Wars Episode 1: The Phantom Menace
5. Star Wars Episode 2: Attack of the Clones
6. Star Wars Episode 3: Revenge of the Sith

This is from an average statistic of all votes (1 point to 6) taken into consideration. In terms of which was the most frequent favourite movie, Episode 5 wins, with 34.6% of voters ranking it number 1. On the contrary of frequency of being the favourite movie, Episode 2 ranked last with 0.04% of voters ranking it number 1.

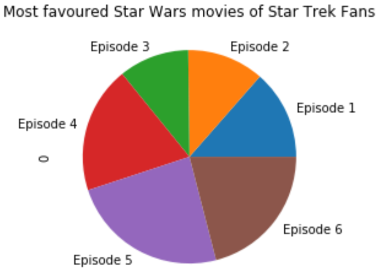
On the other hand, in terms of which was the most frequent LEAST favourite movie, Episode 2 had 35.9% of voters ranking it number 6. On the contrary of frequency of being the least favourite movie, Episode 6 had ZERO percent of voters ranking it number 6.

Task 2.2 – Relationships between Columns

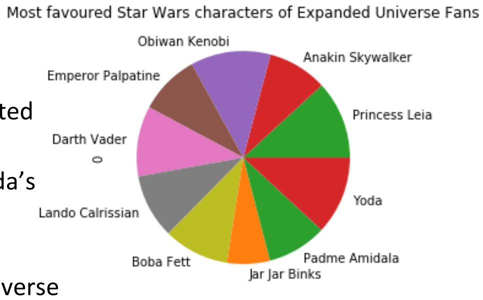
In this visualisation, the aim was to see what movies from the Star Wars saga do Star Trek Fans enjoy the most.

Interesting Relationships.

It appears much like the core Star Wars fandom, the Star Trek fandom tends to favour the original trilogy more than the prequal trilogy. Since this data is from people who are Star Trek AND Star Wars fans, the reasons and relations could be attributed to the same reasons as 'solo' Star Wars fans. However, a possible contribution to the skewed favour to the original trilogy movies may be due to the nature of the original trilogy being 'old sci-fi' which is the genre most Star Trek movies fall under; whereas the prequal trilogy modernised the fiction at the turn of the century.



In this visualisation, the aim was to see which Star Wars characters were most favoured by those who also are aware of the Expanded Universe content.



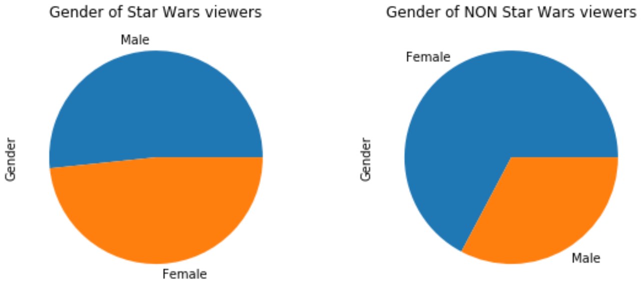
Interesting Relationships.

The larger wedges favouring some of the light side heroes could be attributed to them having more of their backstory and character fleshed out in expanded universe material such as the TV shows. Anakin, Obiwan and Yoda's edging over the others may be due to their focus on the Clone Wars animated TV series. However, this observation may be isolated to just the Clone Wars, as characters like Princess Leia were featured in Expanded Universe material like Star Wars Rebels, she was not on the show that much. Thus a lot of these results could still have major influence from movie prevalence.

In this visualisation, we aim to see the splitting between male and female Star Wars waters.

Interesting Relationships.

From both graphs, there is a clear male bias in the Star Wars watchers. This may be due to the marketing of 'old Star Wars' being more male oriented, as their promotional material and toys were more geared towards boys. Another reasoning for this relationship is the protagonists being male. A coming of age and hero story like the one in Star Wars is gender neutral, but the lack of strong female leads may have deterred females from trying the saga at all. Perhaps this data may be drastically different if we were to include the sequel trilogy which features a prominent female lead.



Task 2.3 – Explore a specific Relationship

- The process of subdividing the data to isolate character opinions based on factors

To obtain the data about which contributing factors affect people's opinion on each character, each character must be isolated from each other for proper analysis of their fanbase. Thus each character was given its own table populated with just the opinions on them and all the demographics of Star Wars viewers. From here, a single graph was produced for each option of each demographic factor. (E.g. a graph of opinions on Han Solo by Education level, specifically bachelor's degrees). Doing this for every option of every factor, we can view the popularity of each character from many audiences.

- Why this process was done

This method was chosen so that all the user data gathered about each Star Wars fan could be utilised to narrow down any possible factor influencing their character preference, from gender to location.

- What was found

  General findings can be mentioned such as good guys being rated higher than bad guys, despite a large fanbase for villains being present. As well as the universal hatred of Jar Jar Binks. It is clear there is a gender bias; female characters are more favoured by female watchers (Not to say female characters are not popular among the male audience). Income and location as well as education play very little role in swaying character preference. This speaks to the universal appeal of Star Wars as a franchise and its characters.

References

(All for study purposes: learning python 3 as a language)

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.Series.str.lower.html

https://www.geeksforgeeks.org/isupper-islower-lower-upper-python-applications/

http://www.datasciencemadesimple.com/convert-character-column-python-dataframe-lowercase/

https://www.geeksforgeeks.org/apply-uppercase-to-a-column-in-pandas-dataframe/

https://www.dataquest.io/blog/adding-axis-labels-to-plots-with-pandas-and-matplotlib/

https://chrisalbon.com/python/data_wrangling/pandas_dropping_column_and_rows/

https://www.shanelynn.ie/using-pandas-dataframe-creating-editing-viewing-data-in-python/

https://stackoverflow.com/questions/18172851/deleting-dataframe-row-in-pandas-based-on-column-value

https://thispointer.com/python-pandas-how-to-drop-rows-in-dataframe-by-conditions-on-column-values/

https://docs.python.org/3.4/extending/newtypes.html

https://www.pitt.edu/~naraehan/python3/data_types_conversion.html

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.astype.html

https://github.com/pandas-dev/pandas/issues/7758

https://stackoverflow.com/questions/37593550/pandas-replacing-elements-not-working

https://www.interviewqs.com/ddi_code_snippets/rows_cols_python

https://cmdlinetips.com/2018/02/how-to-subset-pandas-dataframe-based-on-values-of-a-column/

https://stackoverflow.com/questions/17071871/how-to-select-rows-from-a-dataframe-based-on-column-values

https://stackoverflow.com/questions/42128467/matplotlib-plot-multiple-columns-of-pandas-data-frame-on-the-bar-chart

https://stackoverflow.com/questions/17839973/constructing-pandas-dataframe-from-values-in-variables-gives-valueerror-if-usi

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.drop.html

https://kite.com/python/answers/how-to-sum-two-columns-in-a-pandas-dataframe-in-python

https://www.javatpoint.com/pandas-sum

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.plot.html

https://datascience.stackexchange.com/questions/25596/how-to-plot-two-columns-of-single-dataframe-on-y-axis

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html

https://mode.com/python-tutorial/pandas-dataframe/

https://www.geeksforgeeks.org/python-pandas-dataframe-mean/

https://medium.com/@kasiarachuta/choosing-columns-in-pandas-dataframe-d0677b34a6ca

https://stackoverflow.com/questions/56488402/how-to-replace-misspelled-words-in-a-pandas-dataframe

https://stackoverflow.com/questions/38134643/data-frame-object-has-no-attribute

https://www.geeksforgeeks.org/python-pandas-series-str-contains/

https://towardsdatascience.com/data-cleaning-with-python-and-pandas-detecting-missing-values-3e9c6ebcf78b

https://stackoverflow.com/questions/34165876/trying-to-create-grouped-variable-in-python

https://github.com/pandas-dev/pandas/issues/11179

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.fillna.html

https://datatofish.com/replace-nan-values-with-zeros/

https://stackoverflow.com/questions/13295735/how-can-i-replace-all-the-nan-values-with-zeros-in-a-column-of-a-pandas-datafram

https://stackoverflow.com/questions/20633506/how-to-solve-the-pandas-issue-related-to-series-fillna

https://datascience.stackexchange.com/questions/37435/i-got-the-following-error-dataframe-object-has-no-attribute-data

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html

https://honingds.com/blog/pandas-read_csv/