Assignment 2
Gunnar Yonker

1.

tableA.csv

rank,title,rating,show_type,episodes,run_time,member_count

tableB.csv

rank_id,title,show_type,rating,votes,runtime_minutes,episodes

2.

Same schema(votes will be treated the same as member_count):

6 Total Attributes

rank,title,rating,show_type,votes,episodes

3.

I created a python program that took the tableA.csv file and then re-ordered the attributes into the agreed upon same schema above and saved the file as tableAcleaned.csv for this analysis.

tableAclean.py

Saved as:

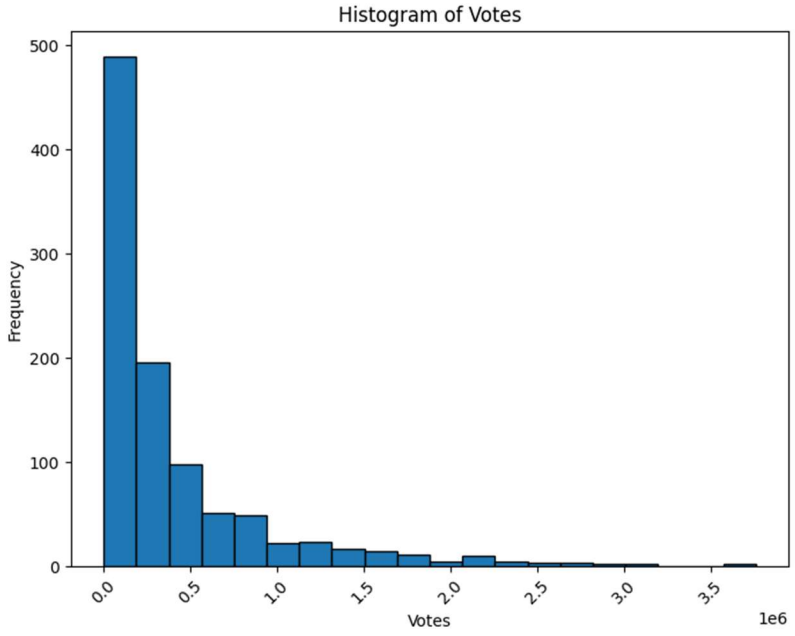
tableAcleaned.csv

rank,title,rating,show_type,votes,episodes

- Missing Values: I created a simple python program titled emptyvaluechecker.py that would report how many missing values were in each column so that they could be quickly analyzed since there are 1000 tuples. There are no missing values in the tableAclean.py for any of the attributes in X.
    - rank: 0%, 0
    - title: 0%, 0
    - rating: 0%, 0
    - show_type: 0%, 0
    - votes: 0%, 0
    - episodes: 0%, 0
- If there were missing values for subsequent steps, how they were filled in would depend on what value was missing. For example if the title was missing, that would be a difficult field to fill in with anything other than a filler such as “No data” because you would not be able to take the average. If there was missing data in votes, a relationship between the rating and votes attributes could be taken to then calculate the number of expected votes based on the rating. This would provide an estimated value of how many votes would be expected with the rating in that tuple. A couple of solutions that could be used to fill in missing data depending on what it is would be mean/mode/median imputation to fill in values with the same attribute, regression imputation which can potentially predict missing values, or using multiple imputation to create

multiple plausible imputations for any missing values based on the observed data and its distribution.

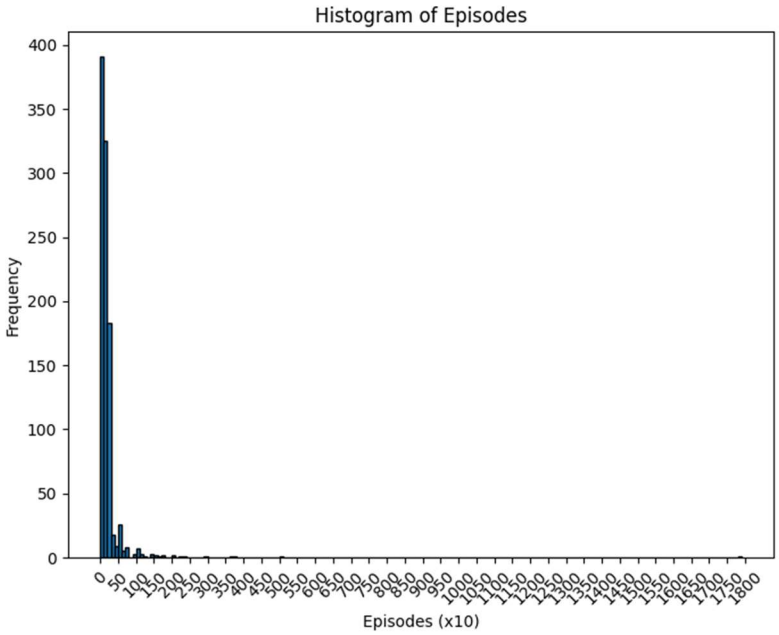
- Classification of each attribute:
  - rank: categorical (represents rankings)
  - title: textual (the values are titles of anime shows)
  - rating: numeric (the values represent ratings)
  - show_type: categorical (the values represent the type of show, such as TV, Movie, OVA, etc.)
  - votes: numeric (the values represent the number of votes)
  - episodes: numeric (the values represent the number of episodes)
- title attribute is textual, created a simple python program to count(titlevaluecounter.py):
  - Average length of 'title': 27.18 characters
  - Minimum length of 'title': 2 characters
  - Minimum title: 86
  - Maximum length of 'title': 87 characters
  - Maximum title: Dragon Ball Z Special 2: Zetsubou e no Hankou!! Nokosareta Chousenshi - Gohan to Trunks
- Votes and Episode Count Histogram for outliers
  - Votes:



    - Potential outliers are hard to determine because the graph is shifted to the left very heavily. However, there is a point way off to the right that may be an outlier.
    - Also, some of the very frequent number of votes such as the first column could potentially be an outlier because it is much higher than the rest.
      - 1a 3189241
      - 2a 2450766
      - 6a 2116345
      - 10a 2670226

Assignment 2
Gunnar Yonker

- 17a 2204335
- 28a 2609313
- 52a 2159531
- 56a 2183719
- 69a 2119878
- 71a 2273856
- 80a 2223912
- 81a 3728067
- 109a 3759598
- 126a 2578373
- 127a 2828706
- 128a 3071085
- 282a 2360148
- 308a 2100008
- 450a 2419754
- 485a 2313162
- 490a 2128186
- 512a 2015524
- 537a 2093727
- 605a 2728327
- 655a 2085725
- 747a 2893580
- 938a 2708660



- Episodes:
  - Potential outlier values(csv line value, episode count):
    - 282a 500.0

        - 394a 291.0
        - 461a 237.0
        - 605a 220.0
        - 691a 366.0
        - 919a 1787.0
        - 954a 373.0
    - This histogram is difficult to represent because the x axis needs to be scaled out so far for the one outlier of 1787 episodes which is vastly further off than the bulk of the data.
- As far as I can tell with this data from tableAcleaned.csv, there are no formats done on the values that will make any issues moving forward.
- I don't think with any of these data values that there are any synonyms among attribute values because the only attribute that could potentially have that is show_type but the values are TV, Movie, OVA, etc.
- I do not have any issues with attribute values being "sprinkled" all over an item with any of the attribute values in this schema, most, if not all, of the data was broken up in easy to gather ways.
- I do not see any other data quality problems with the data in tableAcleaned.csv. It is a complete data set with a good amount of attributes to use in further steps moving forward. The only aspect that could have been more beneficial is if there were more identical schema than the 6 that are being used.

4.

Python packages used for the scripts in this analysis all built into python such as the import csv package.