Problem 1 – statistics (30%)

In a first step, you are to compute various statistics on the data (univariate, multivariate), and interpret these statistics results, e.g., the average movie rating indicates that [write your interpretation], the correlation between variables X and Y suggests that [write your interpretation], etc.

You can use any of the measures of central tendency, dispersion and correlation presented in the lectures (and reading lists.)

Honorable Mentions:

For teaching me how to do exploratory data analysis and working with the Pandas and Seaborn Library: https://www.youtube.com/watch?v=xi0vhXFPegw&t=160s&ab_channel=RobMulla

For helping me out with issues regarding data parsing, when I was having difficulty with converting objects to numeric types: ChatGPT for debugging, and Pandas, Scikit, and Seaborne documentation.

Univariate Analysis

In order to generate the information that I will use for Univariate Analysis, I first read in the csv file, converted it to a data frame, dropped columns that I would not be using, and then changed the types of some columns to be able to use Pandas .describe() function on, which I used to provide descriptive statistics that summarized central tendency, dispersion, and shape of the dataset's distribution.

This is the dataset where we drop the columns Unnamed: 0 and Description and make the according conversions to Gross, Votes and Year of Release in order to have pandas describe() provide information on the columns.

We check dtypes to see if the columns have been correctly converted.

```
In [24]: df.dtypes
Out[24]: Movie Name            object
         Year of Release        int32
         Watch Time             int64
         Movie Rating         float64
         Metascore of movie   float64
         Gross                float64
         Votes                  int32
         dtype: object
```

```
In [69]: df.describe()
```

Out[69]:

	Year of Release	Watch Time	Movie Rating	Metascore of movie	Gross	Votes
count	1000.000000	1000.000000	1000.000000	845.000000	838.000000	1.000000e+03
mean	1991.666000	124.253000	7.970200	79.011834	73.016014	3.190469e+05
std	24.188045	28.800355	0.275732	11.973800	115.068316	3.871814e+05
min	1920.000000	45.000000	7.600000	28.000000	0.000000	2.581300e+04
25%	1975.000000	103.000000	7.800000	71.000000	3.367500	6.238100e+04
50%	1999.000000	120.000000	7.900000	80.000000	25.475000	1.581340e+05
75%	2011.000000	139.000000	8.100000	88.000000	96.945000	4.436372e+05
max	2023.000000	321.000000	9.300000	100.000000	936.660000	2.777378e+06

Since there are NaN values in Metascores of movies and Gross, it makes sense that count does not equal to 1000. Everything looks to be in order.

Now, after computing the various statistics on the data, I will interpret those statistic results using the first dataset:

1) The average movie rating in the dataset is 7.97, meaning that on average, the movies in this dataset are positively received by the audience.
2) The range of the data for the year of release tells us that the movies in this dataset range from movies released in 1920 to 2023. This also gives us insight into the top-rated movies of IMDB, as the range shows that there are movies that are over a century old that are in critical acclaim while simultaneously having high ratings for more recent

movies. There is a wide range of data in this set in terms of the year of release for a movie.

3) The standard deviation for movie ratings is 0.276, which tells us that in terms of movie ratings, the data is clustered generally close towards the mean. In more practical terms, it appears that there is small variability in the movie ratings. This makes sense, as this dataset is the top 1000 movies on IMBD. It is reasonable to say that all of them should be clustered around the average movie rating.
4) The average watch time of movies are 124 minutes. This indicates that movies that are in IMDB's top 1000 movies are generally around the length of 124 minutes, or 2 hours and 4 minutes. This sounds correct because movies are generally around that length.

Multivariate Analysis

Displaying the Correlation values of the data.

```
In [37]: df_corr = df[['Year of Release', 'Watch Time','Movie Rating',
                       'Metascore of movie','Gross', 'Votes']].corr()
         df_corr
```

Out[37]:

	Year of Release	Watch Time	Movie Rating	Metascore of movie	Gross	Votes
Year of Release	1.000000	0.208913	-0.070580	-0.367392	0.216572	0.238391
Watch Time	0.208913	1.000000	0.272353	-0.038731	0.120537	0.154984
Movie Rating	-0.070580	0.272353	1.000000	0.256970	0.134560	0.483813
Metascore of movie	-0.367392	-0.038731	0.256970	1.000000	-0.061870	-0.066044
Gross	0.216572	0.120537	0.134560	-0.061870	1.000000	0.555361
Votes	0.238391	0.154984	0.483813	-0.066044	0.555361	1.000000

W

Now we proceed with multivariate interpretation of the data.

1) The correlation between the variables Movie Rating and Votes was 0.48. This means that there is a moderate positive relationship between these two variables. This suggests that movies with higher movie ratings tend to receive more votes. To me this indicates that more votes could be a potential indicator of a high movie rating and suggests that

movies with higher movie ratings tend to receive more votes. A potential (and untested) reason as to why this occurs is that maybe higher movie ratings draw in more attention and views from the audience, resulting in higher votes.

2) There also appears to be moderate positive correlation between the variables Gross and Votes, as the correlation coefficient is 0.53. This suggests that movies with high Gross tend to receive more votes as well, similarly to that of movie rating. A potential reason as to why this might occur is that movies that bring in more Gross tend to draw the attention of potential viewers, resulting in a larger audience and more votes.
3) The correlation between variables Year of Release and Metascore of Movie is -0.32, which indicates that there is weak, negative relationship between the two variables. This indicates that there is an inverse relationship between the two variables; when the Year of Release increases, the Metascore of a movie goes down. This suggests that newer movies that are released generally have lower Metascores; one of the reasons as to why this may be the case might be because older movies have more time to accrue more votes, and viewers to fully develop a stable Metascore. Newer movie ratings are more volatile, and are often subject to critical acclaim; thus, sometimes, the movie ratings of newer movies are sometimes lower than that of older movies.