Project Overview:

This python code takes the reviews from the website IMDB on the movie The Dark Knight and performs a sentiment analysis, picking out the sentiments of each review. We hoped to learn what people's feelings were towards The Dark Knight. The next step of our code would be to create a word cloud by training the words in the reviews with the sentiment. Please look at the CSV file, 'dark_knight_movie_review.csv'

Implementation:

Our initial code was intended to scrape the website and create a CSV file, where one column contains the reviews, another column contains the ratings for each review, and the final column contains the sentiment of each review. We used the following packages to run this code: csv, urllib.request, BeautifulSoup, and nltk. The original data for reviews and ratings was in the form of HTML. The data we gathered for the sentiment utilized the data we scraped from the reviews, while the sentiment column contains reviews processed using sentiment analysis.

Our second code utilizes nltk and textblob to show the sentiment of each review. The nltk allows us to view each review's positive, negative, and neutral scores, in addition textblob allows us to infer the sentiment of the review overall as well as how subjective versus objective each review is. The positive words are then plotted on a word cloud (since the data gathered lacks any negative reviews).

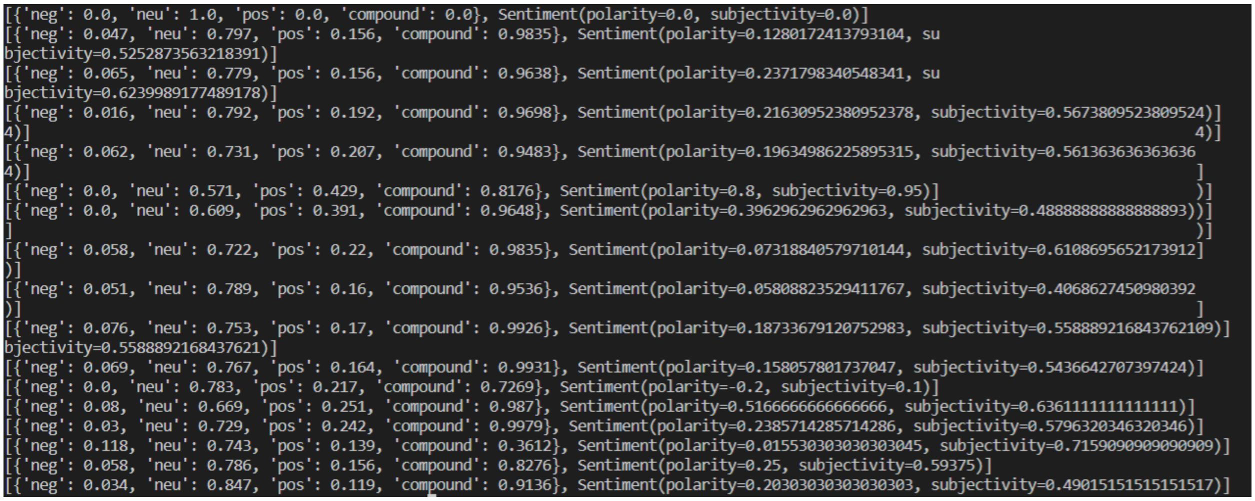
We were debating between a word cloud, histogram, or Markov analysis. In the end we decided to do a word cloud because we thought this would better display our analysis. Also, it was more visually appealing to look at.

Results:

We found that all of the reviews can be classified as positive even though the neutral score was higher than the negative and positive scores, this can be supported by the polarity scores. This can be seen in Figure 1. The classified reviews are then used to create the column "sentiment" seen in Figure 2. The purpose for doing this is to allow the words in each review to be trained against the overall sentiments of each review.

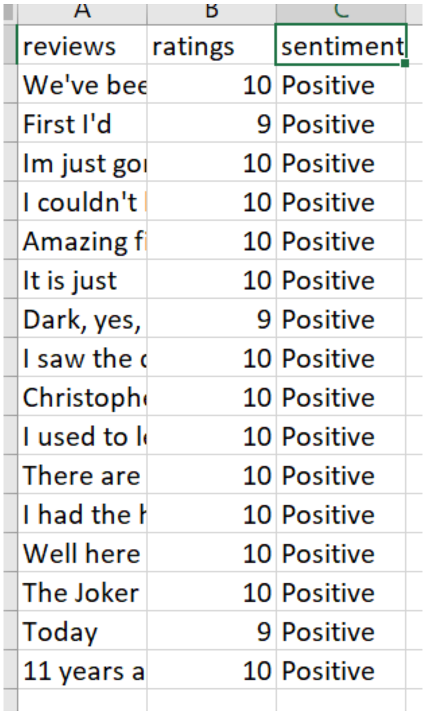
We created a word cloud (Figure 3) that represents the most used positive words by training the data against positive reviews. This, however, still included words that should have been excluded from the training data set, in addition to the stopwords. The data gathered also lacks any negative reviews to train against, so the word cloud generated might not be accurate.

Figure 1:



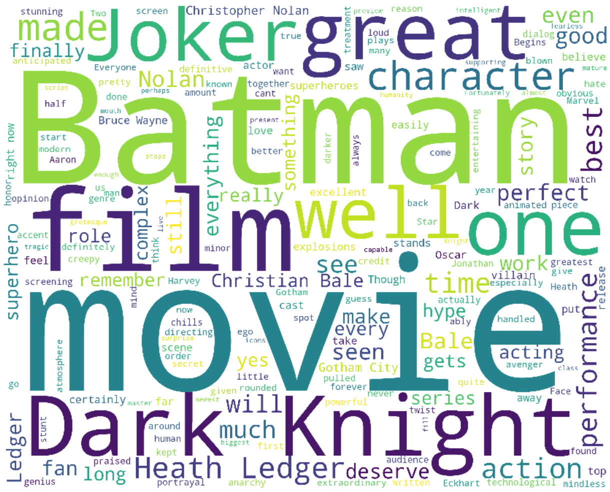
```
[{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}, Sentiment(polarity=0.0, subjectivity=0.0)]
[{'neg': 0.047, 'neu': 0.797, 'pos': 0.156, 'compound': 0.9835}, Sentiment(polarity=0.1280172413793104, su
bjectivity=0.5252873563218391)]
[{'neg': 0.065, 'neu': 0.779, 'pos': 0.156, 'compound': 0.9638}, Sentiment(polarity=0.2371798340548341, su
bjectivity=0.6239989177489178)]
[{'neg': 0.016, 'neu': 0.792, 'pos': 0.192, 'compound': 0.9698}, Sentiment(polarity=0.21630952380952378, subjectivity=0.5673809523809524)]
4)]                                                                                                                                  4)]
[{'neg': 0.062, 'neu': 0.731, 'pos': 0.207, 'compound': 0.9483}, Sentiment(polarity=0.19634986225895315, subjectivity=0.561363636363636
4)]                                                                                                                                   ]
[{'neg': 0.0, 'neu': 0.571, 'pos': 0.429, 'compound': 0.8176}, Sentiment(polarity=0.8, subjectivity=0.95)]                           )]
[{'neg': 0.0, 'neu': 0.609, 'pos': 0.391, 'compound': 0.9648}, Sentiment(polarity=0.3962962962962963, subjectivity=0.48888888888888893))]
]                                                                                                                                    )]
[{'neg': 0.058, 'neu': 0.722, 'pos': 0.22, 'compound': 0.9835}, Sentiment(polarity=0.07318840579710144, subjectivity=0.6108695652173912]
)]
[{'neg': 0.051, 'neu': 0.789, 'pos': 0.16, 'compound': 0.9536}, Sentiment(polarity=0.05808823529411767, subjectivity=0.406862745098392
)]                                                                                                                                    ]
[{'neg': 0.076, 'neu': 0.753, 'pos': 0.17, 'compound': 0.9926}, Sentiment(polarity=0.18733679120752983, subjectivity=0.558889216843762109)]
bjectivity=0.5588892168437621)]
[{'neg': 0.069, 'neu': 0.767, 'pos': 0.164, 'compound': 0.9931}, Sentiment(polarity=0.158057801737047, subjectivity=0.5436642707397424)]
[{'neg': 0.0, 'neu': 0.783, 'pos': 0.217, 'compound': 0.7269}, Sentiment(polarity=-0.2, subjectivity=0.1)]
[{'neg': 0.08, 'neu': 0.669, 'pos': 0.251, 'compound': 0.987}, Sentiment(polarity=0.5166666666666666, subjectivity=0.6361111111111111)]
[{'neg': 0.03, 'neu': 0.729, 'pos': 0.242, 'compound': 0.9979}, Sentiment(polarity=0.2385714285714286, subjectivity=0.5796320346320346)]
[{'neg': 0.118, 'neu': 0.743, 'pos': 0.139, 'compound': 0.3612}, Sentiment(polarity=0.015530303030303045, subjectivity=0.7159090909090909)]
[{'neg': 0.058, 'neu': 0.786, 'pos': 0.156, 'compound': 0.8276}, Sentiment(polarity=0.25, subjectivity=0.59375)]
[{'neg': 0.034, 'neu': 0.847, 'pos': 0.119, 'compound': 0.9136}, Sentiment(polarity=0.20303030303030303, subjectivity=0.49015151515151517)]
```

Figure 2:



A	B	C
reviews	ratings	sentiment
We've bee	10	Positive
First I'd	9	Positive
Im just goi	10	Positive
I couldn't	10	Positive
Amazing f	10	Positive
It is just	10	Positive
Dark, yes,	9	Positive
I saw the	10	Positive
Christoph	10	Positive
I used to l	10	Positive
There are	10	Positive
I had the h	10	Positive
Well here	10	Positive
The Joker	10	Positive
Today	9	Positive
11 years a	10	Positive

Figure 3:



Reflection:

As a team, we decided to pair program together in the beginning, however in actuality, we split the project up by tasks. We initially wanted to create a histogram with the data we analyzed, but we thought it would not visualize the data well; therefore we ended up using a word cloud. Also, we struggled to figure out the code for sentiment analysis, but after some extensive Googling and consulting the Professor we were able to figure out the code. For the word cloud we were unable to figure out the stopwords, so we couldn't delete words like "movie," "dark knight," or "Batman" from our word cloud. Also, there were no negative reviews, so we could not train the positive words against anything, resulting in the word cloud not being as accurate as it could be.