Software components. High level description of the software components such as: *data manager*, which provides a simplified interface to your data and provides application specific features (e.g., querying data subsets); and *visualization manager*, which displays data frames as a plot. Describe at least 3 components specifying: what it does, inputs it requires, and outputs it provides.

There are four main components, three of which quantify the *diversity score* and one of which quantifies the *sentiment score*.

Sentiment score components:

1. getYoutubeURL:
    a. input: IMDB movie information dataframe
    b. output: movie youtube trailer URLs
    c. what it does: retrieve the URL of the movie trailers in the IMDB dataset
2. getYoutubeComments:
    a. input: movie youtube trailer URLs
    b. output: all youtube comments (for each movie)
    c. what it does: extract youtube comments from the youtube trailer page
3. getSentimentScore:
    a. input: all youtube comments (for each movie)
    b. output: sentiment score (for each movie)
    c. what it does: evaluate the sentiment of the commenters on the youtube page (negative versus positive)

Diversity score components:

1. getDiversityScore:
    a. input: IMDB movie information dataframe, dataframe with movie title and sentiment score
    b. output: dataframe with movie title, sentiment score, and diversity score
    c. what it does: given cast information/data, compute how gender diverse each film is
2. computeCorrelation:
    a. Input: Dataframe containing movie title, sentiment score and gender diversity score.
    b. Output: results from correlation analysis including a plot.

c. What it does: computes the correlation between the sentiment score and gender diversity score.

Interactions to accomplish use cases. Describe how the above software components interact to accomplish at least one of your use cases.

When the user runs the program, getYouTubeUrl will iterate over every movie title in the imdb dataframe and pass the URL to getYoutubeComments which will then retrieve all of the comments from the YouTube movie trailer at the given URL. These comments will be stored in csv files in a directory called “comments”. getSentimentScore will read the csv files in the “comments” directory and train a naive bayes classifier on a provided labelled dataset. The classifier will then classify all the unlabelled comments into positive and negative labels. An average score will be computed based on all comments for a movie, which is the sentiment score. A new dataframe will be created with the movie title and sentiment score. This dataframe will then be passed to getDiversityScore, which will compute a score based on the gender diversity in the movie and add a column to the dataframe with the gender diversity score. getDiversityScore will pass the data frame to computeCorrelation which will compute the correlation between the gender and diversity scores. computeCorrelation will print the results from the correlation analysis to the console and present a visual representation of the analysis.

Preliminary plan. A list of tasks in priority order.

Eric: work on getYoutubeURL
Jamie: work on getYoutubeComments
Rachel: work on getSentimentScore component
Trent: work on getDiversityScore component
Kate: work on getDiversityScore AND getSentimentScore components (and eventually lead correlation analysis between diversity score and sentiment score)