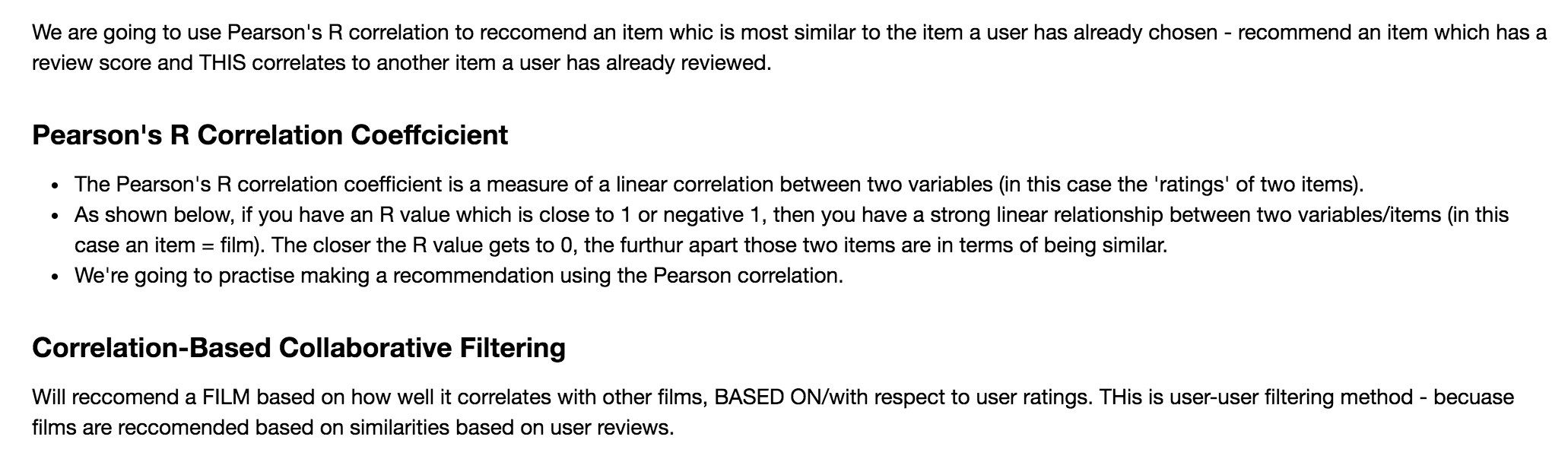
****

**Getting Data**

Import the two data sets

Ratings – which has userID, movieID, rating 0/5, and timestamp

Films – movieID, title and genre

- The 'ratings' dataframe has a rating for every unique film (found in 'films' dataframe) from a customer.

- A rating is given out of 5.

- A rating of 0 means the customer did not like the film very much and a 5 means they loved it.

- Both datasets have a similar column, which is moviesId.

- In the 'ratings' dataframe userId is in duplicate, this means the customer has reviewed more than one film.

![title](../img/pearson.png)

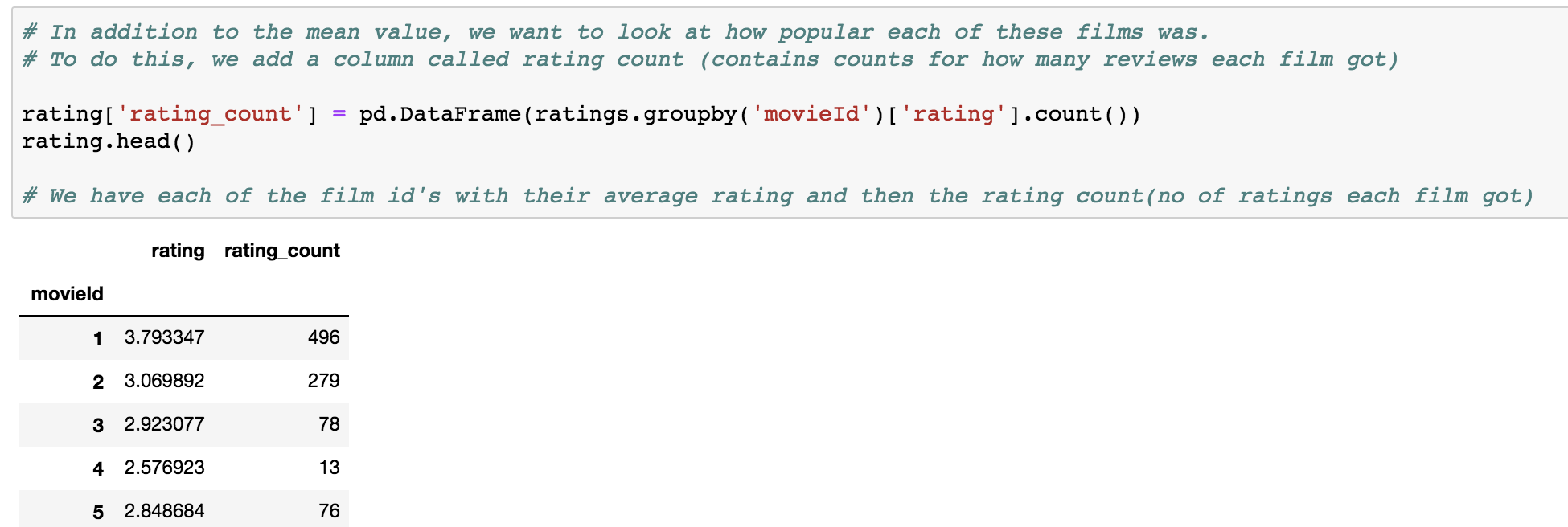
**Grouping and Ranking Data**

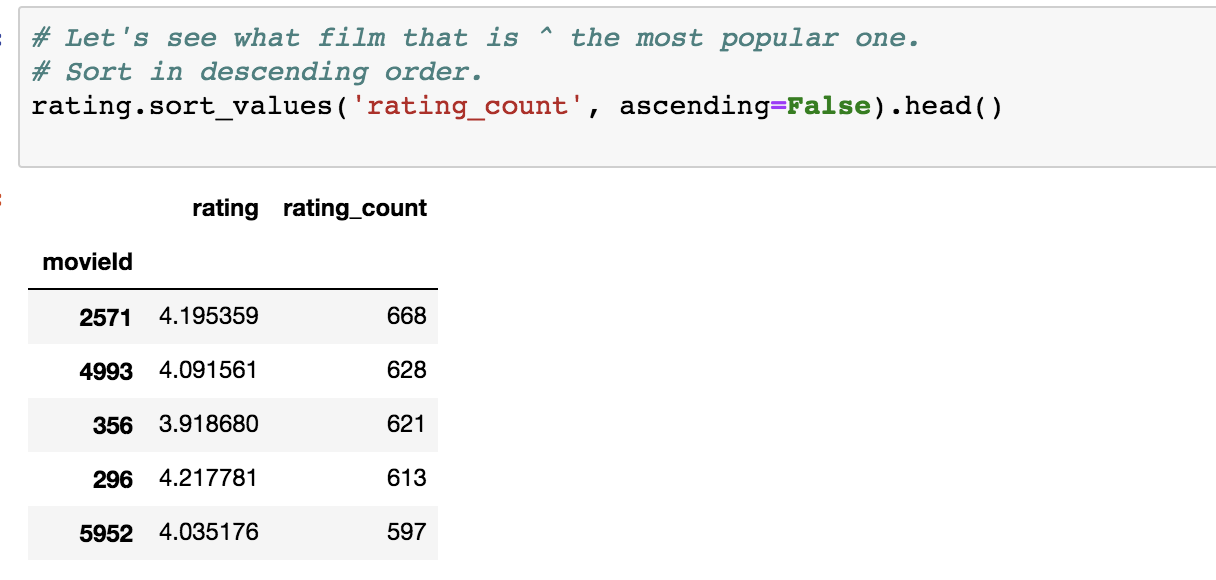
We're going to look at the ratings, these films are getting. To do this: we're going to look at the mean value of all the ratings which have been given to each film.

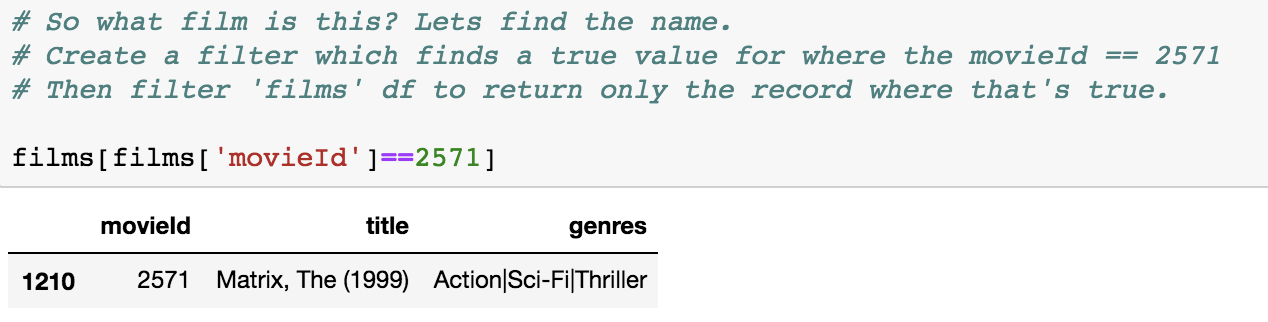
* New data frame 'rating' will be generated from the 'ratings' data frame.
* Take ratings data frame and group by movieId - then for each movieId we want to look at the rating column and want to generate the mean value for each rating that was given to each film.
* Lets look at the rating which each place are getting do this by looking at the MEAN value of ALL the ratings given to EACH place
* New data frame, generated from frame data frame - but group frame by placeID,
* Then for each PlaceID look at rating column and generate mean value for each of the rating which was given to each place

Film popularity final data frame:

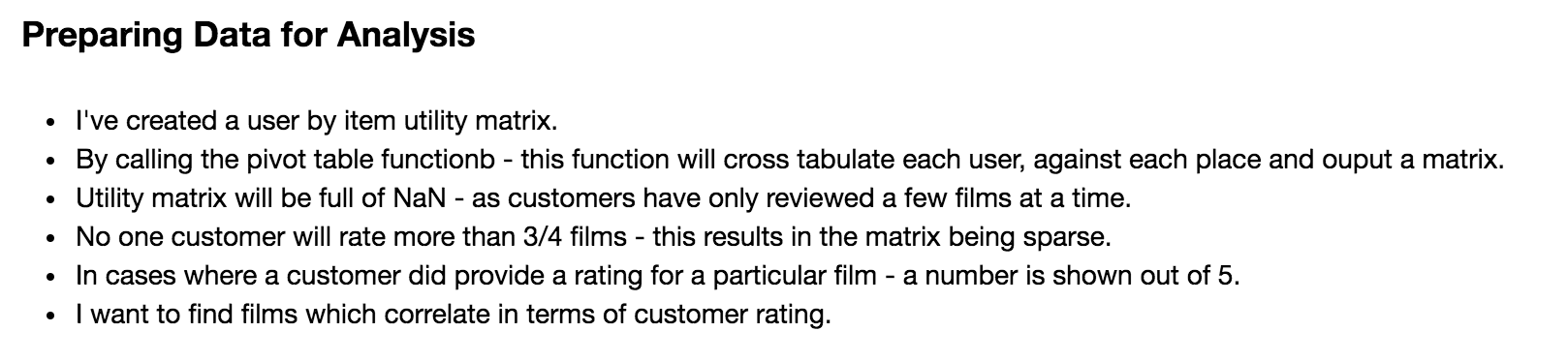
Now we have each of the films - their average rating and then the no. of ratings each film got .





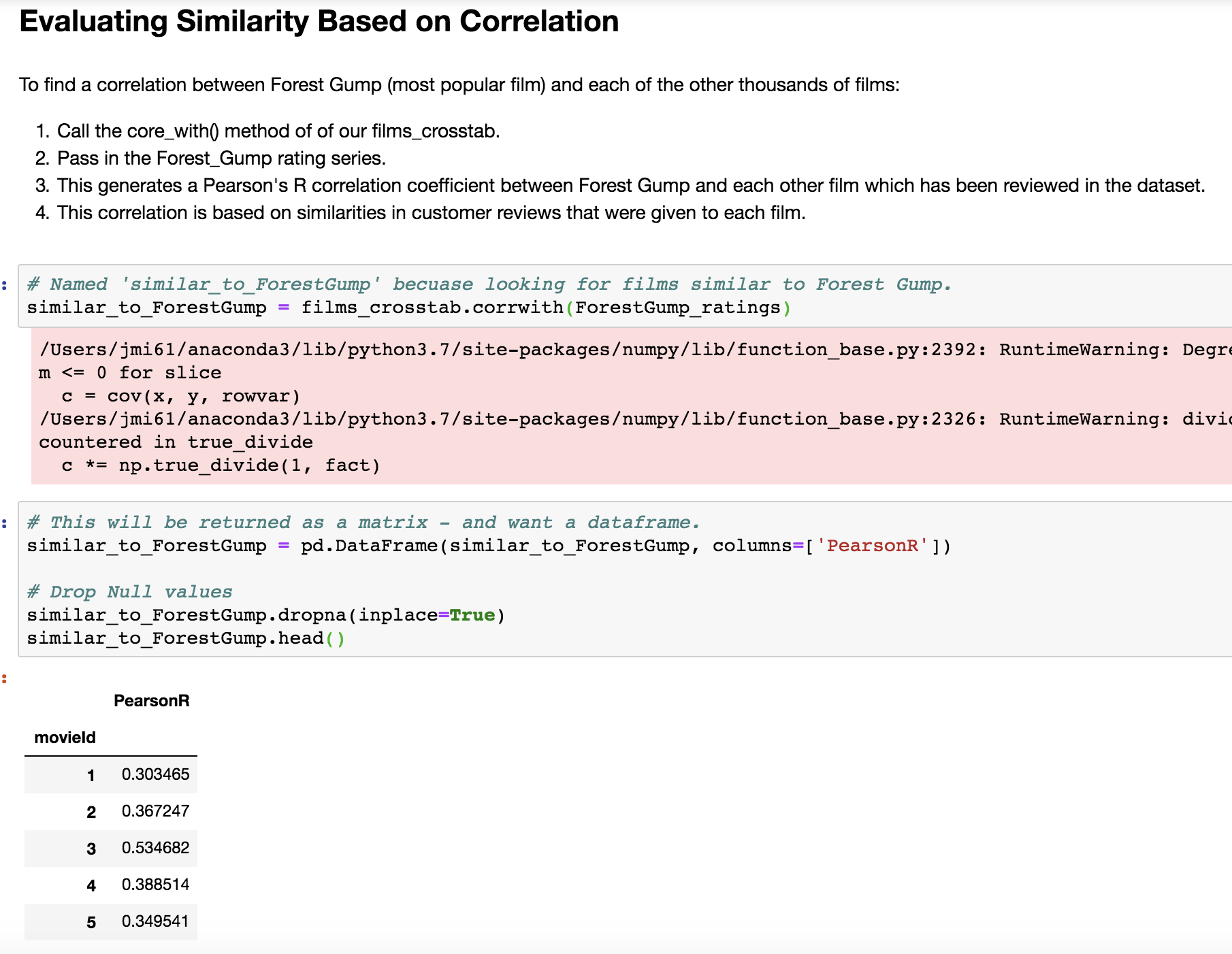


**Preparing Data for Analysis**

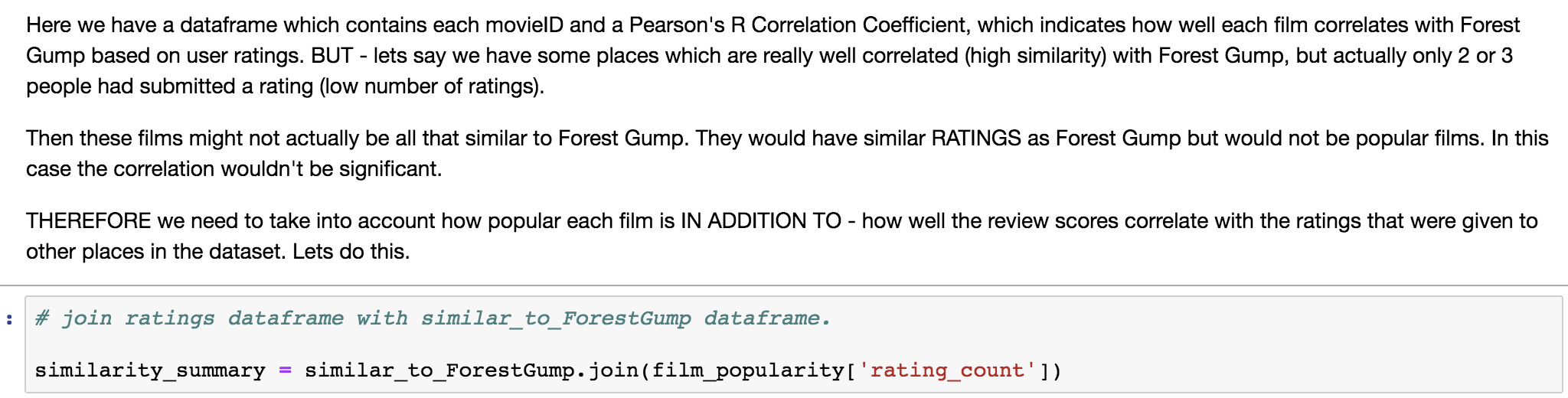




**Evaluating Similarity Based on Correlation**



Get rid of the issue of films with low rating count



# We've sorted the dataframe in descending order by correlation.

# We now have a list of TOP reviewed places similar to Forest Gump.

# Lets find out what the names of the top 5 similar films to Forest Gump are

