Homework Assignment #3

(Data Wrangling with Python and Pandas)

Congratulations are in order yet again, as your new Data Scientist career at Netflicks.com is flourishing! Your boss just heard a rumor that you're an expert on Python and Pandas, and since his friends at Friday cocktail parties have been raving about how great those tools are for Data Science work, you're back on his radar screen again. He's still interested in using the Internet Movie Database data from IMDB, so you've just been tasked with continuing down that path for the coming week. His exact thought is, since you've already managed to load all the IMDB data into PostgreSQL, "Hey, how hard can that be…?"

Your next assignment is to show your boss how one might explore the PostgreSQL-resident IMDB data using Pandas.

Get Ready...

In addition to PostgreSQL, this week's tasks will require you to use a Jupyter notebook and Pandas dataframes. It's time to install Anaconda if you haven't already done so! You will also need to find and install SQLAlchemy in order to move data between the SQL data world and the world of Python/Pandas tools. Get them up and running! You might want to start by making your own copy of the materials from Monday's lecture and running the associated Jupyter notebook to ensure that "all systems are go" with respect to your installations of everything.

Get (Data) Set...

Your next step is to start grabbing the IMDB data from PostgreSQL and loading it into Pandas dataframes using SQLAlchemy. (Again, see the Jupyter notebook from Monday's lecture for an illustrative example.) Start by doing this for the titleBasics data set -- i.e., use SQLAlchemy to do a SELECT * query from your HW#1 solution's title.basics dataset's SQL table that targets a Pandas dataframe. Since the focus here is going to strictly be about movies, your query should ensure (via its WHERE clause) that only "movie" data is loaded into your dataframe. Now do the same for the titleRatings data set, writing your query in such a way as to ensure that you only grab ratings for movies.

Go...!

It's time to get to work. Tackle the following questions/problems -- this time using Pandas in Jupyter instead of SQL in psql as your data exploration tool:

1. Print the titles dataframe. (Notice how Jupyter automatically elides most of the data to show you just a head/tail-oriented sample of its content.)
2. Use the describe method that dataframes have to get an initial sense of the numeric data distributions in the titles dataframe.

3. List the movie titles, start years, and genres for which the primary title contains ‘Star Wars’ but the primary title is not the movie’s original title.
4. List the movie titles and number of votes for movies with the lowest possible rating (1).
5. Examine the distribution of ratings by printing, for each distinct averagerating value, the number of movies with that rating and the min, max, average, and median vote counts for movies with that rating.
6. Analyze the movies from a genre perspective by using Panda’s get_dummies function to create a dataframe with a row for each movie and a category indicator column for each of the possible movie genres (set to 1 for each genre that a given movie belongs to). This dataframe should be indexed by movie identifier (tconst) rather than by position.
7. Create a new movie titles dataframe by merging the original titles dataframe with your new indicators dataframe. Again, use movie identifier indexing for this dataframe.
8. Last but not least, using your newest dataframe, list the movie titles, start years, and genres for all of the pre-1950 biographical crime dramas. (:-))

What To Turn In

When you have finished your assignment you should use EEE to turn in a PDF export of the Jupyter notebook showing your work.