Exercise: Movie Ratings

Table of Contents

Background

The goal of this exercise is to practice merging DataFrames and using split-apply-combine to analyze data. For this exercise, you will read in data about movies and ratings of those movies, and you will find the highest-rated movies in the dataset.

About the data

The data are derived from the MovieLens 100k dataset, available at https://www.kaggle.com/prajitdatta/movielens-100k-dataset. Two files are provided: `movies.csv` and `ratings.csv`. `movies.csv` contains information about a number of movies, and `ratings.csv` contains users' ratings of these movies. There are several ratings for each movie.

Template

A template script, `movies.py`, is provided. This script contains import statements, a `parse_args()` function to process command-line arguments, and an `if __name__ == "__main__()":` statement to run the program.

Instructions

`best_movies()` function

Write a function called `best_movies()` with two parameters: the path to a file of movie data (such as `movies.csv`) and the path to a file of rating data (such as `ratings.csv`).

Your function should read each CSV file into its own DataFrame. Do not hard-code the filenames into your function; use the parameters you defined instead. Merge the DataFrames using the `"item id"` column of the ratings DataFrame and the `"movie id"` column of the movie DataFrame. This should be an inner merge.

Group by the `"movie title"` column and find the average (mean) value of the `"rating"` column. This

should give you a Series where the index labels are movie titles and the values are average ratings. Store this Series in a variable.

Return a sorted version of your Series of average ratings by appending `.sort_values(ascending=False)` to the name of your variable. For example, if you used the variable name `average_ratings`, your `return` statement could be

```
    return average_ratings.sort_values(ascending=False)
```

This will sort the values in your Series from highest-rated movie to lowest-rated.

Running your script

Your script should require two command-line arguments. Assuming your script is called `movies.py` and is in the same directory as `movies.csv` and `ratings.csv`, and assuming this is the working directory of your terminal, here's how you could run your script (Windows users, replace "python3" with "python"):

```
python3 movies.py movies.csv ratings.csv
```

The top-most rated movie in this dataset is "Close Shave, A (1995)" with an average rating of 4.491071.