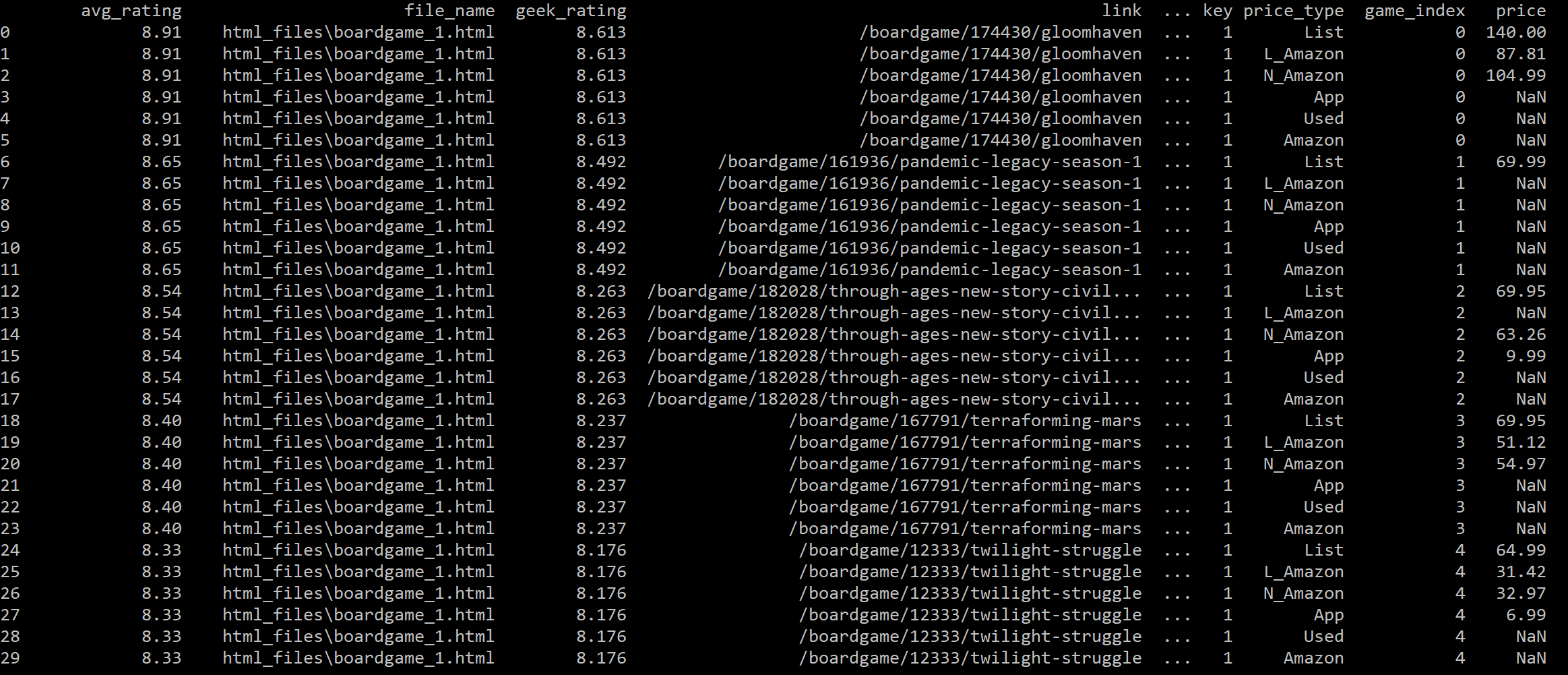
I chose to scrape the data on game names, ratings, number of votes, and prices on boardgamegeek.com. This website contains a table of the game ratings and rankings at the following URL: <https://boardgamegeek.com/browse/boardgame/page/1>

My scrapping program is **‘1\_boardgamegeek\_request.py’**. At the time of scraping this page, these data were contained on tables across 1,059 pages. Since the URL is denotes the page number of each part of the table, a simple loop could iterate through all the pages of the table. I chose to use a while loop to scrape these data that continued to iterate to the next page as long as data was contained on the table. If no data was contained on the next table, then the program would terminate because the end of the table had been reached. Each page was contained on 1059 html files.

These html files were then parsed using **‘2\_boardgamegeek\_parse.py’**. Data on game name, game rank, game year, geek rating, avg rating, number of votes, prices, and the URL extension for each games information page were extracted. The challenge here was parsing the price data. Since some games have different prices reported, such as “List”, “New Amazon”, “Lowest Amazon”, “Amazon”, “iOs App”, no price, or any combination of the above, I chose to pull all the price text as a string and extract the price data in a later step.

I wrote an additional data building program to clean my game data as well as extract the different prices for each game. **‘3\_boardgamegeek\_pricedata\_clean.py’** cleans the data and splits the price string, described above, into individual observations for each price. The resulting dataframe has rows for each game and price type, where price type is “List”, “New Amazon”, “Lowest Amazon”, “Amazon”, or “iOs App”. If a game did not have a particular price listed, the price column would be missing for that game and price type.

The first 30 rows of my dataframe are here:



The dataset is described below:

