For my dataset, I used a sampling of Amazon toy reviews found on Kaggle – the dataset can be viewed at <https://www.kaggle.com/PromptCloudHQ/toy-products-on-amazon>. It consisted of 17 columns and 10000 records, and was pared down from 115000 records previously scrapped from the web.

To begin with, I imported the data into a pandas dataframe using the pandas read\_csv function. This got us into a usable dataframe with minimal trouble – it was a very clean dataset. While the current headers were fairly readable, I switched them out for a slightly more readable English header group I called cleanheaders. I read that file in and replaced the original headers with the new ones.

We defined a new function, printRow, that took as the argument an integer referring to a specific row. This let us call the function and print out the answers in a machine readable format of a given row. The prices were given in pounds, and were actually a string – our function strips out the pound symbol, strips out any commas (some prices were listed as 4,560, for example) and converts to US dollars.

We also did some transformation on the in-stock column. We split it up, since our answers were a numeric quantity followed by NEW or USED or condition, etc. We split this and printed them separately.

We identified columns with NAN values for price. This was roughly 15% of our dataset. In addition, 18 of our rows were just strings – they would, for example, be “$596 - $1200”. The range wasn’t as useful for us, and because there were so few rows we would probably drop said rows from the data were we to move forward.

Each product had a Unique ID – we tested this as well and found that there were no duplicates based on the Unique ID. Finally, we used fuzzy matching to clean up the Manufacturer columns. Since we had a number of duplicate manufacturers that weren’t necessarily spelled correctly, we used fuzzy matching to replace each manufacturer with the closest match. This would have worked better with a set of specific manufacturers but we did not have that. We did find 7 blank Manufacturer columns and replaced them with ‘None’ for consistency.

The main challenges I had was translating the Python 2 code from the book to Python 3 code. Many of the functions worked completely differently, so it was difficult to figure out how to approach each step to approximate what the textbook was doing. In addition, the dataset was fairly clean – there weren’t as many opportunities I found to clean up the data. I would choose a more raw dataset next time.