I decided to go with my dataset being a bunch of books from Project Gutenberg. I compiled the text documents of many classic books from the English, German, Portuguese, and Spanish languages. I wanted to see if I could use a specific metric, in this case first letter of word frequency distributions, to determine what language the book is. I define first letter of word frequency distribution as the occurrence rate of every letter as the first in a word per book, sort it by frequency, and generate a discrete distribution from it. I made a few assumptions for this. I assumed a 26 letter alphabet for every language (this is not true in real life, but R can only handle the traditional Latin alphabet). I also assumed that I could create a discrete distribution for every book’s first letter of words.

Before getting into the results (which were a little informal; sorry I’m a little time crunched), I chose to use first letter of word frequency distributions because I saw this in a research paper. It said that for every given language, the first letter of word frequency follows a specific distribution defined by an entropic transformation of Benford’s Law. I have the mappings for expected probability of occurrence on the side of every letter I graph as well. I thought that there would be enough variance of the actual distribution from the expected distribution that I could use the variance to predict what language the book is.

The graphs of all of the distributions are in the graphs folder. Check those out to see what the distributions look like for each book.

I used a hierarchical clustering algorithm to try to cluster the books into their respective languages. I can create a dendrogram from this (also in the graphs folder). It seems like the variance of the distributions is great enough that the hierarchical clustering function in R can cluster the books into the proper languages. It was essentially 100% accurate in its clustering power. A big thing with this is that I can’t predict what language a book is, but I can cluster it with other books to see if it is related other ones. With the image of the dendrogram, it looks like all of the books of the same language were related to each other than books of a different language. This is a good sign as there are also probably no outliers either.

Also, I ran a k means algorithm to try to cluster like that, but it doesn’t seem to work as well. I don’t know why k means isn’t working as well. It is probably due to k means looking at creating a certain amount of cluster. Hierarchical clustering will separate based on distance, and clusters would have to be visually identified.

One thing I would like to look at in the future is using a non-clustering algorithm to generate predictions. This would be a bit more powerful because I could use it to generate an actual prediction based on data (so the test data wouldn’t impact the actual model). The only issue with that at this current moment is that I have R on my computer and don’t know how to do this in R. Python would take me a bit of time to set up Jupyter Notebooks and brush up on the syntax.