# Week 6 – Assignment 2 – Python Text Analysis

By: Zach Adair

Regis University

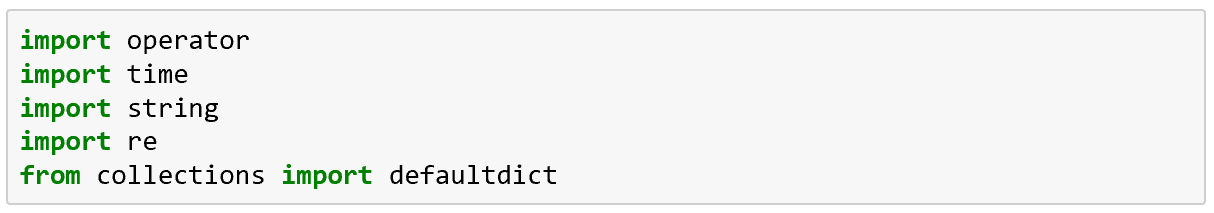
MSDS 650 – Data Analytics

## Introduction

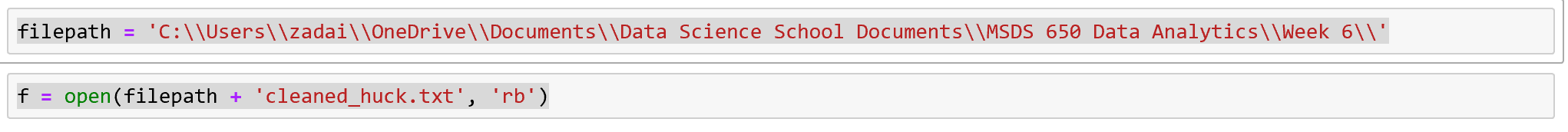
For the second assignment for week six I will be comparing two texts that I believe will give similar results. Before I did that analysis, I started by doing the exercise before which went over text analysis in python so I thought I would start by breaking down my walk through and getting a good grasp on what is going on there, then following it up with comparing two types of text.

## Python Text Analysis Exercise

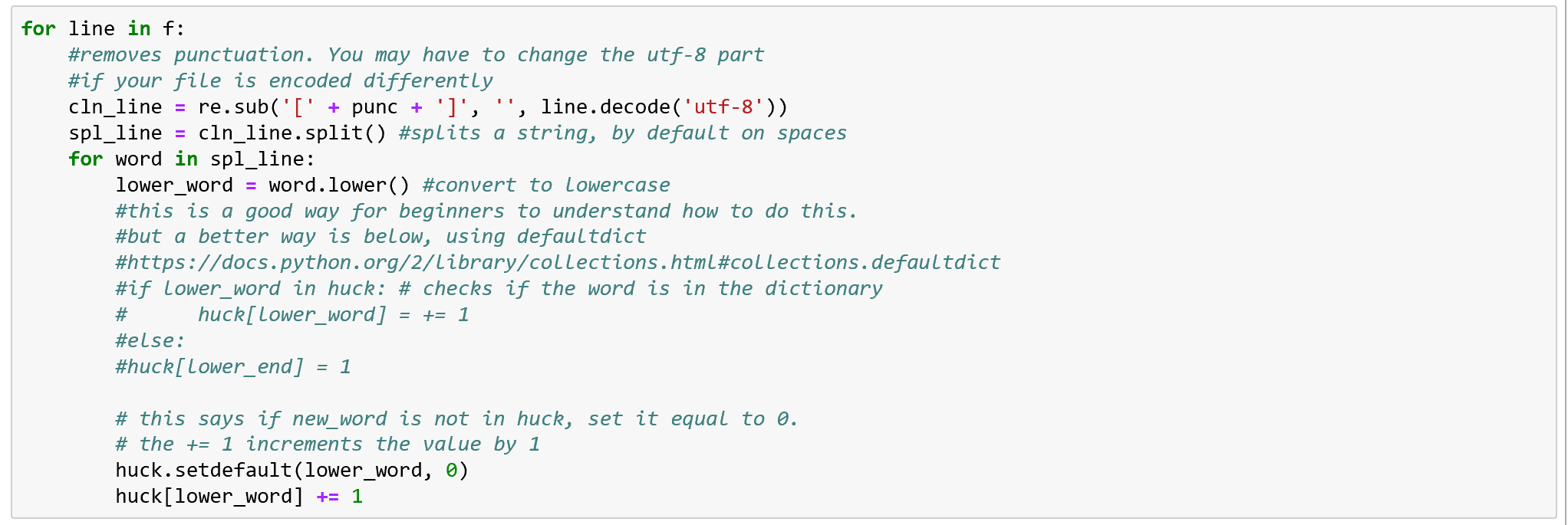
The first step in our analysis was to load the proper packages within python. Those were operator, time, string, re and defaultdict.



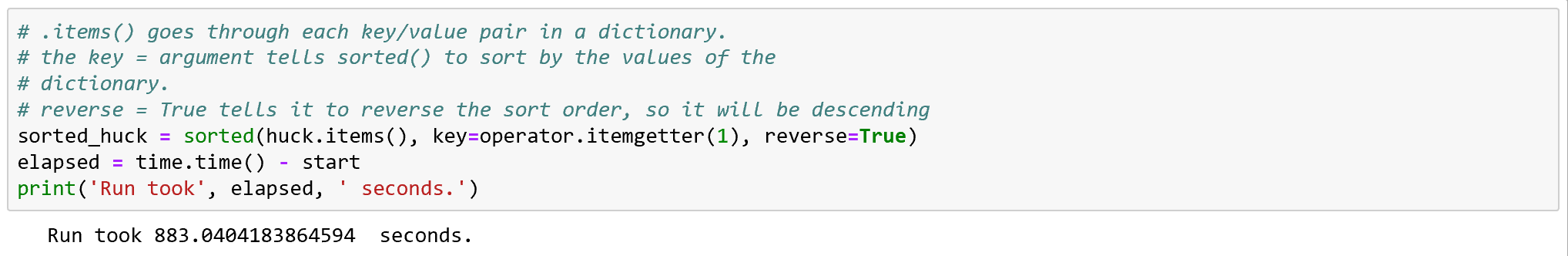
The next step is to set the file path and select the file that needs to be selected from the end of that path, the file that needs to be selected is the cleaned up text of Huckleberry Finn by Mark Twain.





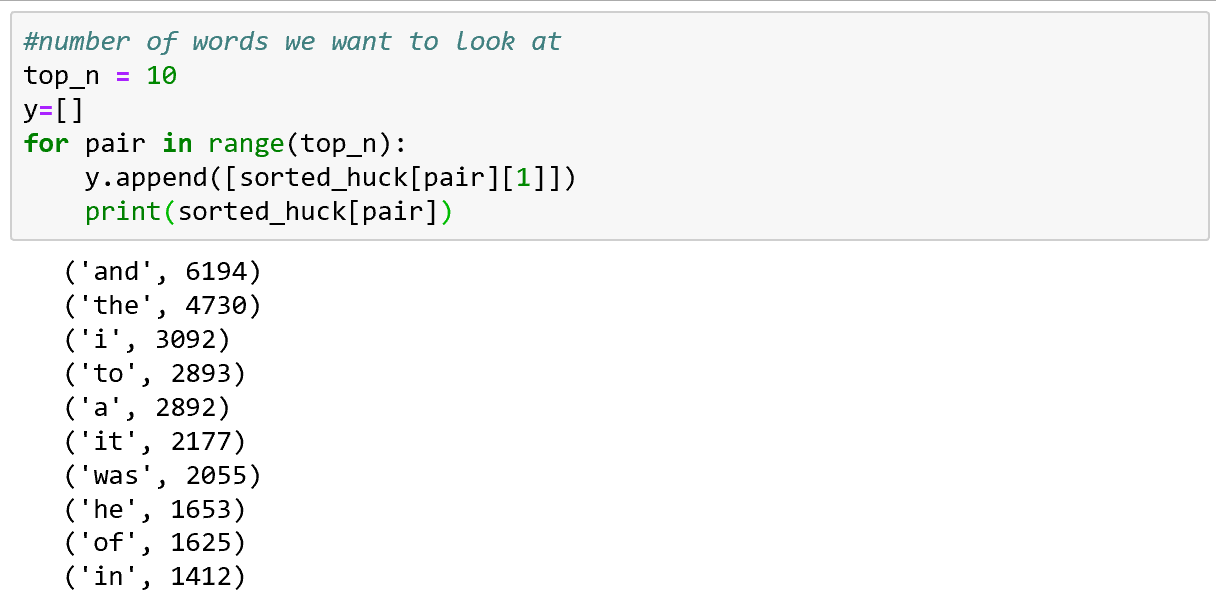


What was done above was we created a for loop which cleaned up our text a bit. It removed punctuation, split up strings by default spaces, and convert all the text to lowercase.



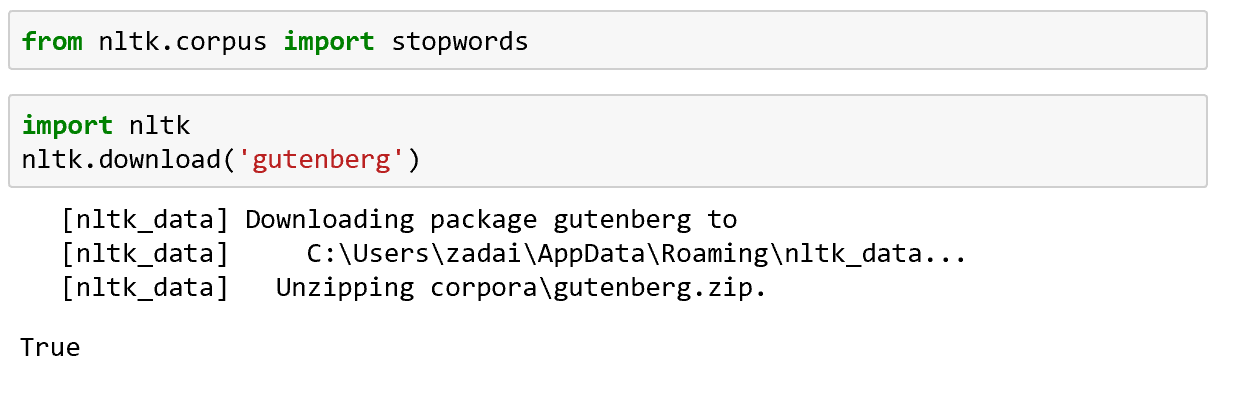


Now what was done was the code distinguished which words it wanted in a dictionary. And also what happened was created a variable called sorted\_huck which sorted the dictionary. Then another variable called elapsed which measures the time it took to run the process.

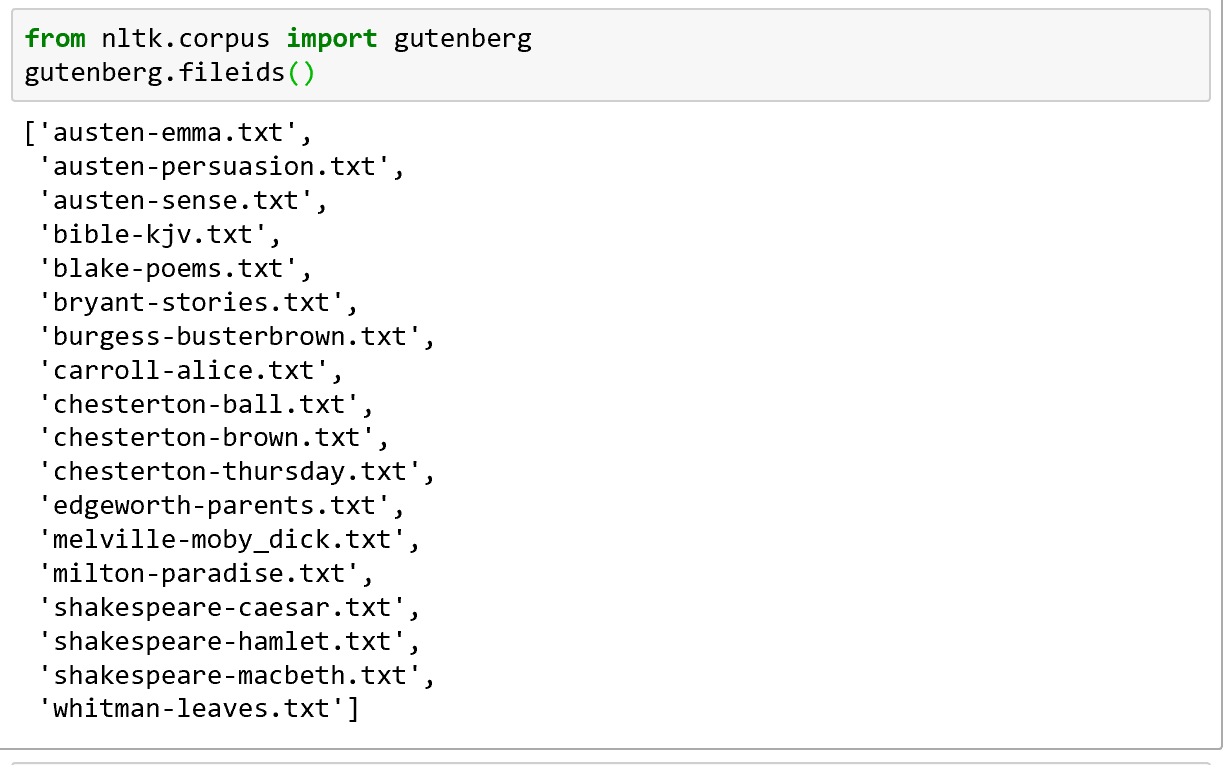


Finally here we are taking the dictionary created and listing the words within it from highest counted to lowest counted.

The next part of the exercise dealt with importing a package called stopwords from the Python Natural Language Tool Kit. So coming up, the nltk and stopwords packages will be uploaded and then we will do a similar analysis but this time with those packages involved.

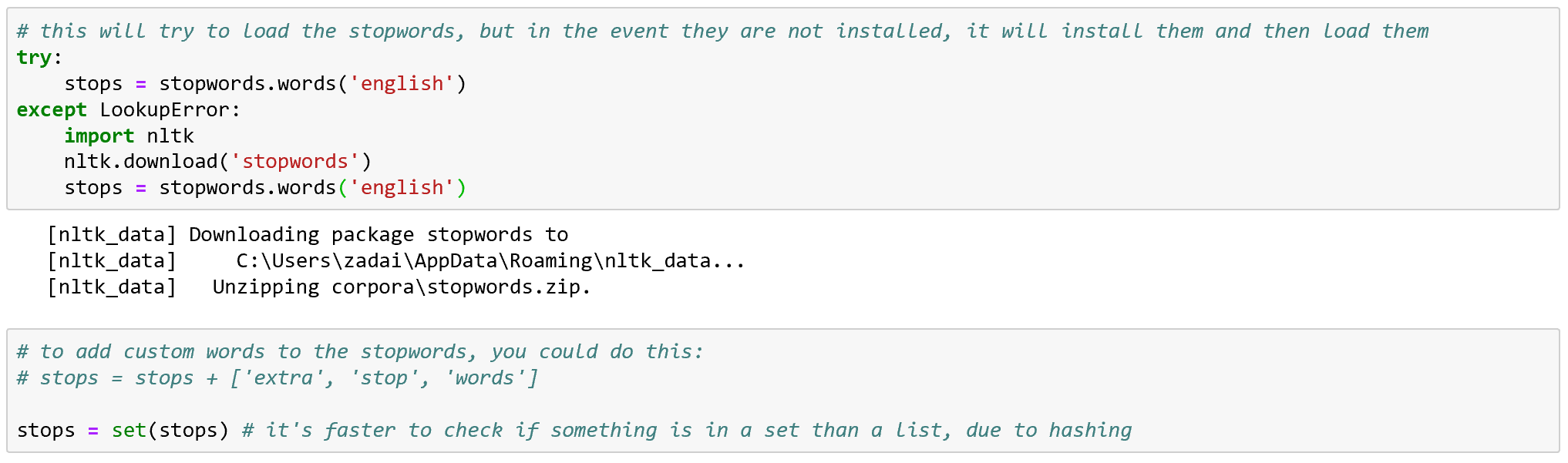


Those two packages are now imported into Python and I can proceed with the new version of text analysis.

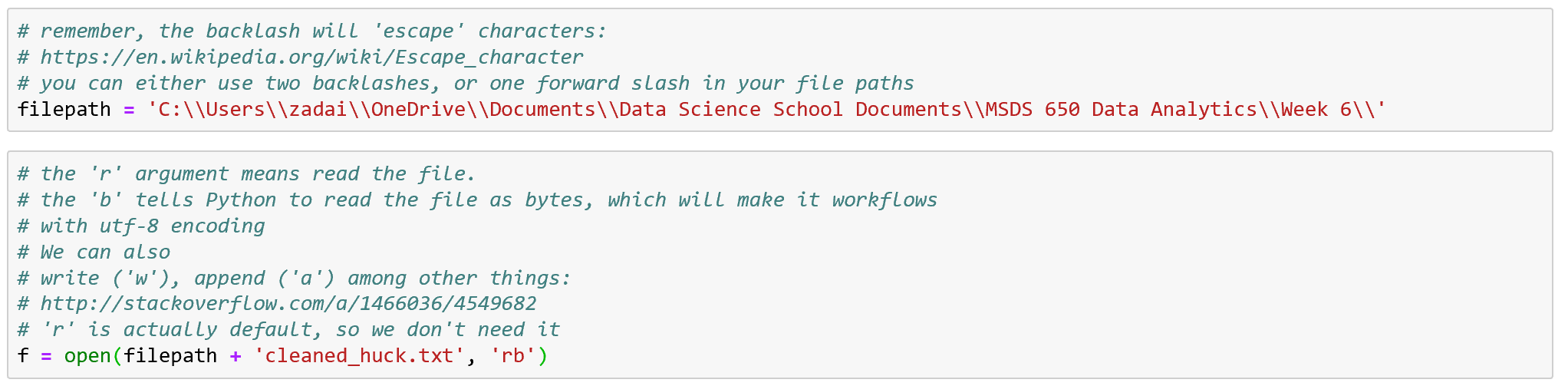


Another import into Python is Gutenberg which is a website with free texts and the list below is a list of some of the Gutenberg texts file IDs.

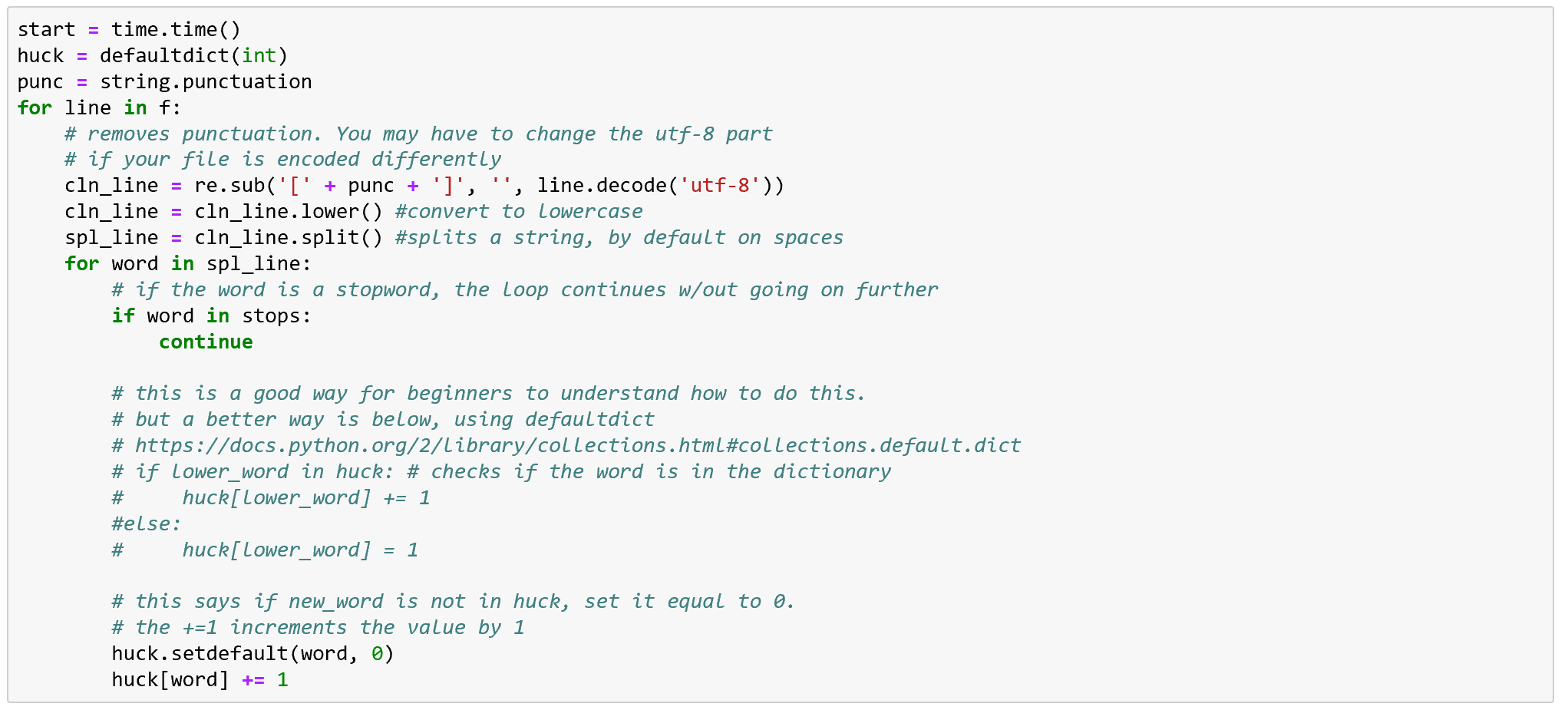
Here is where we get to the part where the python code is similar to what was just done before but now we are analyzing the data with our new imported commands.



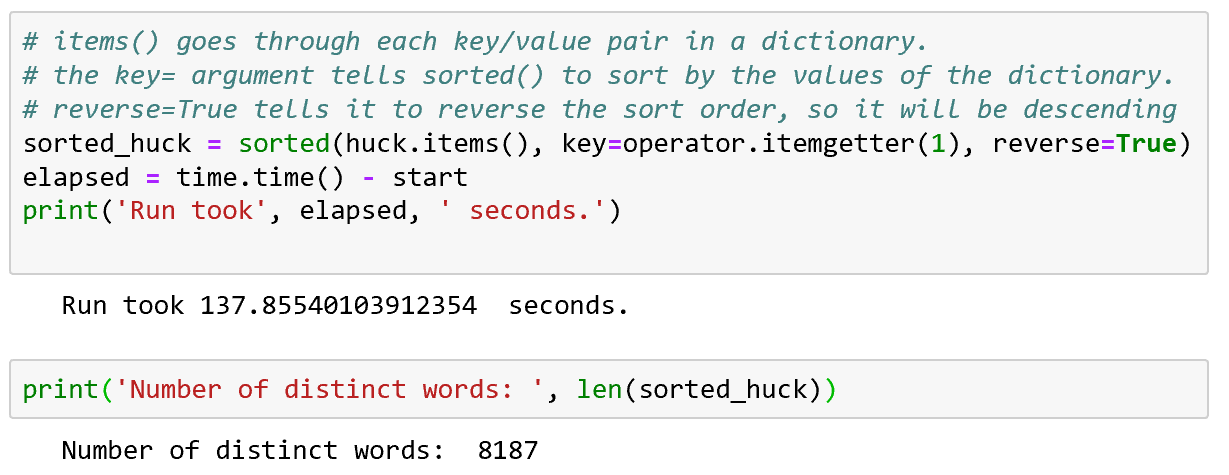
First we use a try command to try and load all the stopwords. Next we are making stops into a set because it will be quicker to check over making it into a list.



All we just did here was recommit the filepath and f to the same locations I had them pointed to before.



Next is the for loop, which does a lot of the same things which were done during the first time around.



We are now able to grab the run time and look at the distinct count of words within the text. Finally what is done next is sorting out the list of stop words and giving a count to how many is in the text as well as sorting them from the greatest to the least.



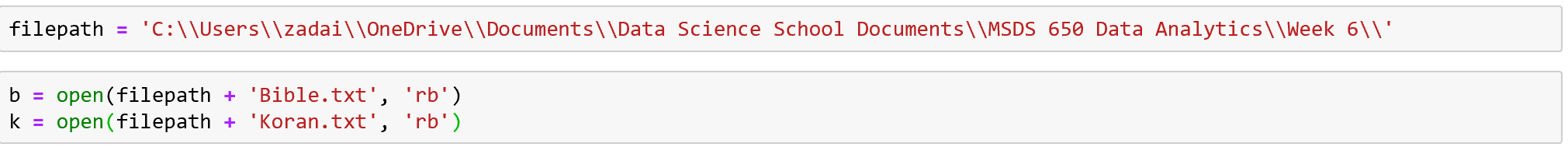
That is the assignment walk through from the PDF within the class. I found this a neat introduction to Text Analysis in Python. I found it really interesting to see the construction of the two times through the process, first time grabbing those filler words and the second time getting seeing the stop words. What I thought was interesting within them is the number of discrepancy between the two, it was expected because filler words are used much more than stop words but I wanted to see if the Python code would back up my assumption. Now that I’ve gone through the assignment, my next part of the entire assignment is to take two text documents and compare and/or contrast them based on what I see from my Python text analysis.

## Comparing Two Different Texts

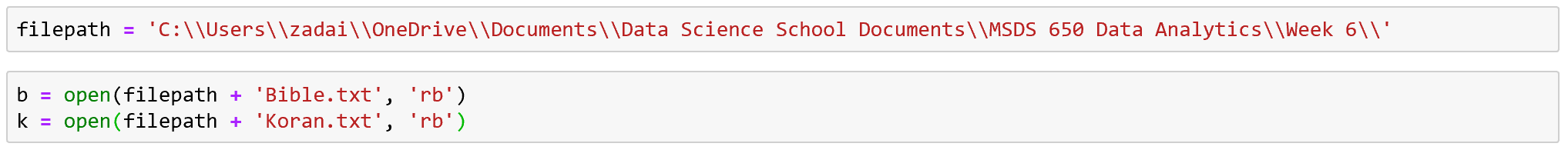
The two texts I’m going to compare for this analysis are two of the most religious texts in human history, the New Testament Bible and the Koran. I’m going to try and play around with this analysis and try to find out how similar they are and how different they are based on the text within them.

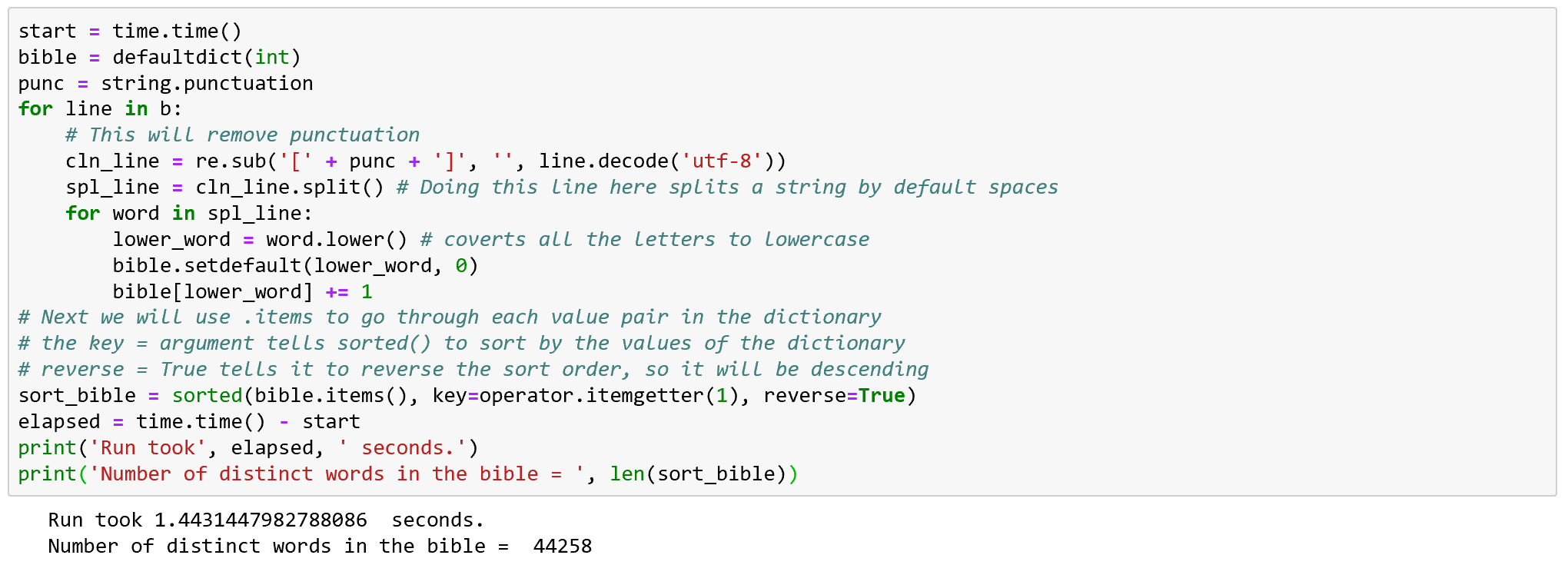
The first thing I’m going to do is I am going to compare the two texts based on the tests run within the exercise from above.

Start by setting the file path and opening up the text files.



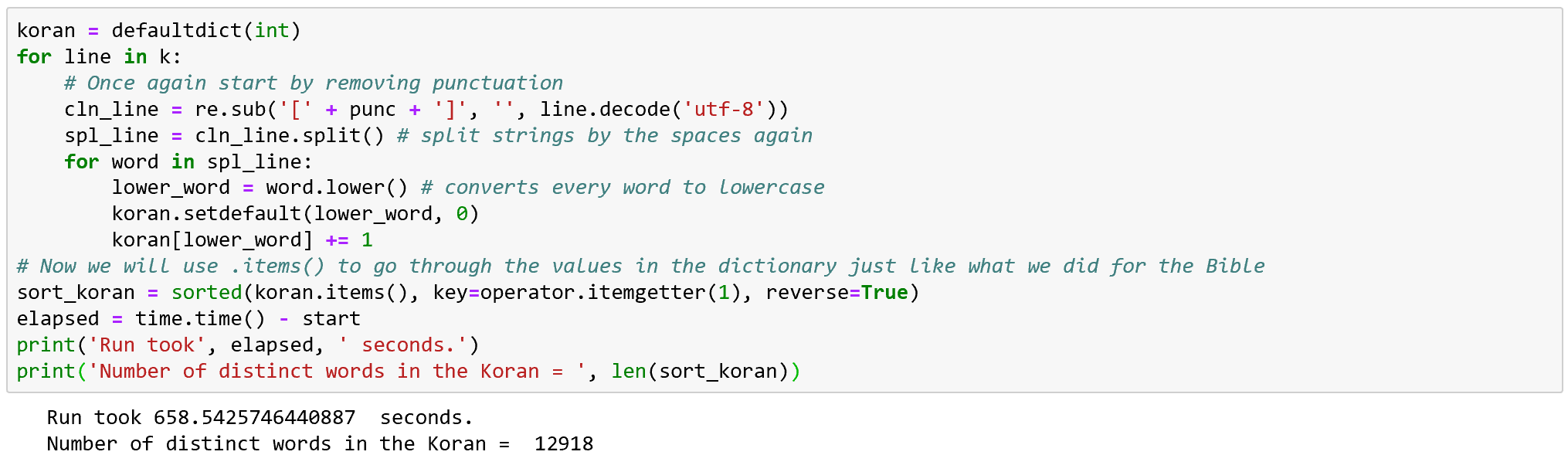
I have set b = Bible and k = Koran to distinguish between the two texts being analyzed. The text I started with first was the Bible.

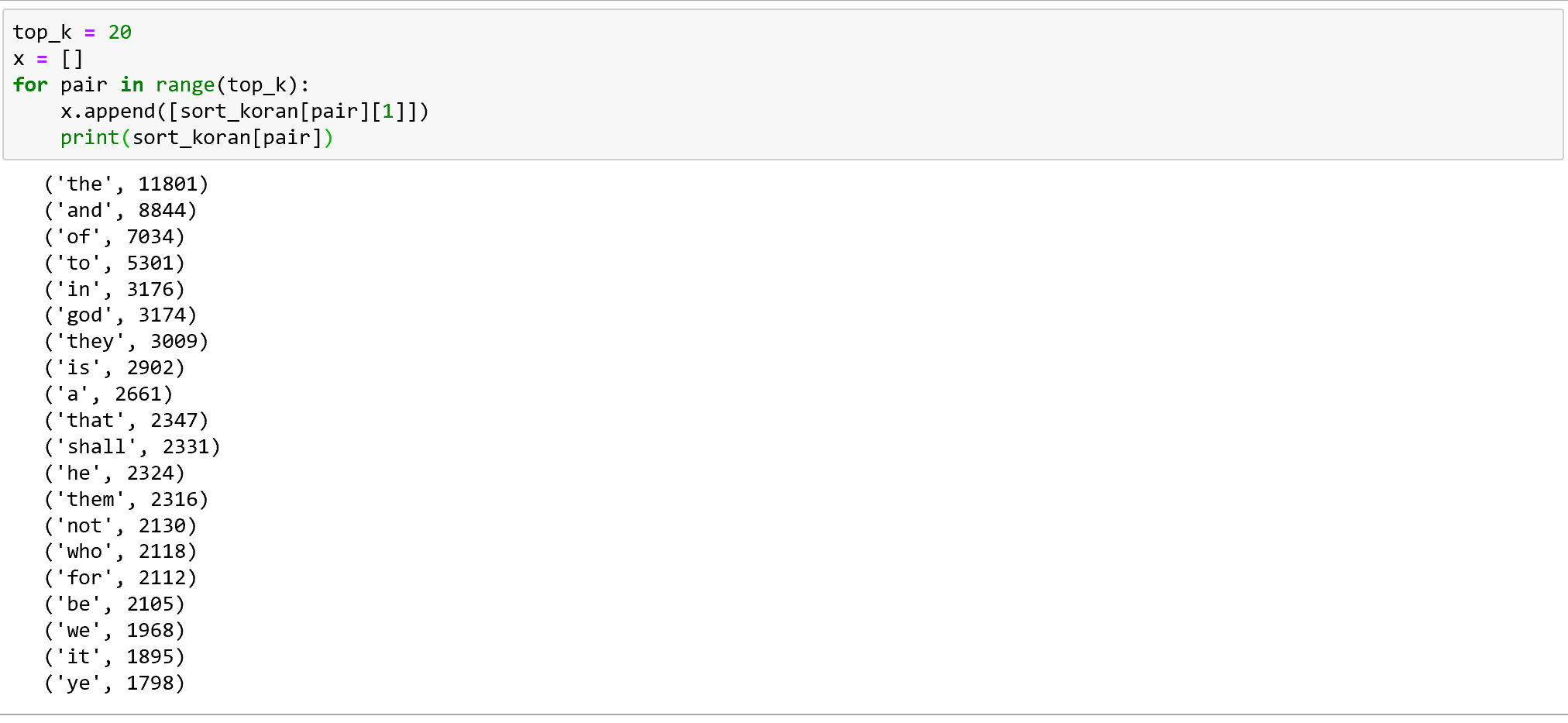






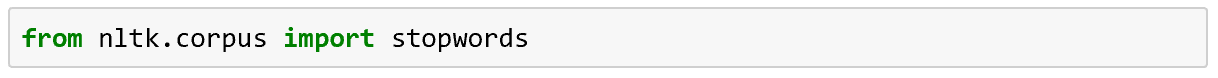
From our code we can see that there are a total of 44,258 distinct words in the Bible, and that the top word is ‘the’ which is used 64,115. In my list I have the top 20 words from the Bible, in this list is a lot of filler words though and the only word that I take as a religious word is ‘lord’ and it is in the Bible 7,830 times. Next I want to do the same thing for the Koran.





From looking at the Koran in the same way I previously just looked at the Bible I saw some similarities and differences within my bit of analysis and some differences. The similarities come within the list with many of the words being filler words. None of that is surprising but still interesting to see that only one of those words for each texts is something pertaining to the topic. In the Koran the word ‘god’ is used 3,174 times and is sixth on the list, while in the Bible the word ‘lord’ is used 7,830 times and is 14th on the list. The differences that I never really took note of from what I’ve briefly analyzed is the difference in the number of words. The Bible has over 64,000 words while the Koran has just under 13,000 which makes the bible almost 5 times as long which I really didn’t expect to be the case when beginning this analysis.

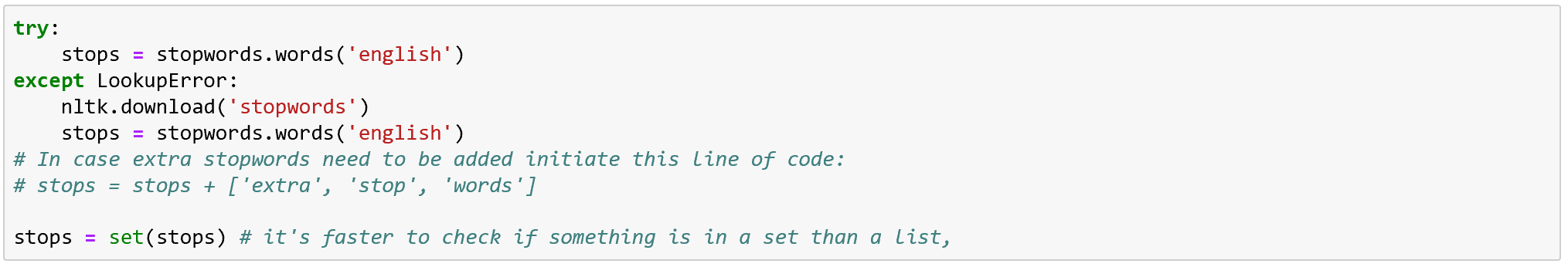
The next step in my process will be to try and run a similar process, but this time I will be removing the filler words from the result and see which words are used more from a list of non-filler words.



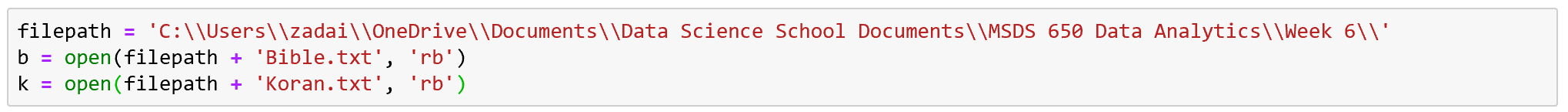
Started by bringing in the python packages one new one which will help get rid of stopwords.



Next we bring in the try function.

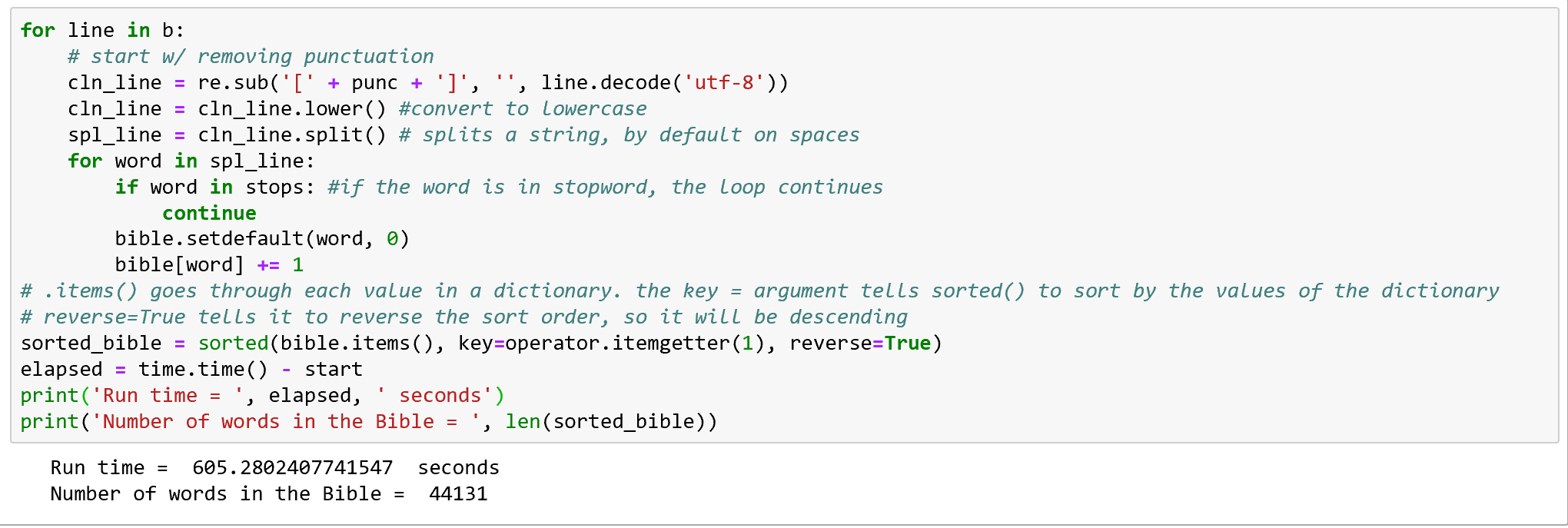


Set the file paths to make sure they are hitting our texts,

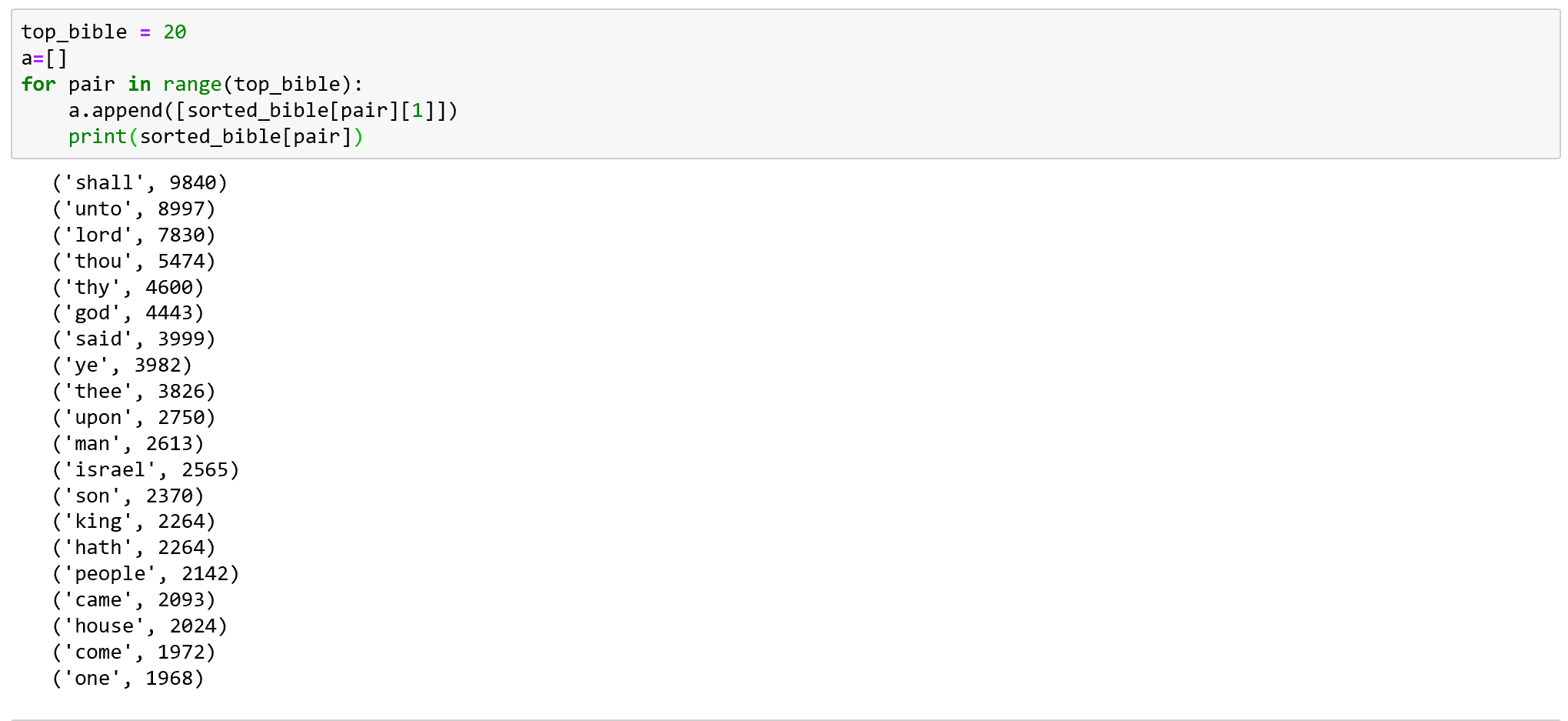
Set some of the necessary functions.



Now I will put together the for loop needed to analyze the text in the Bible.

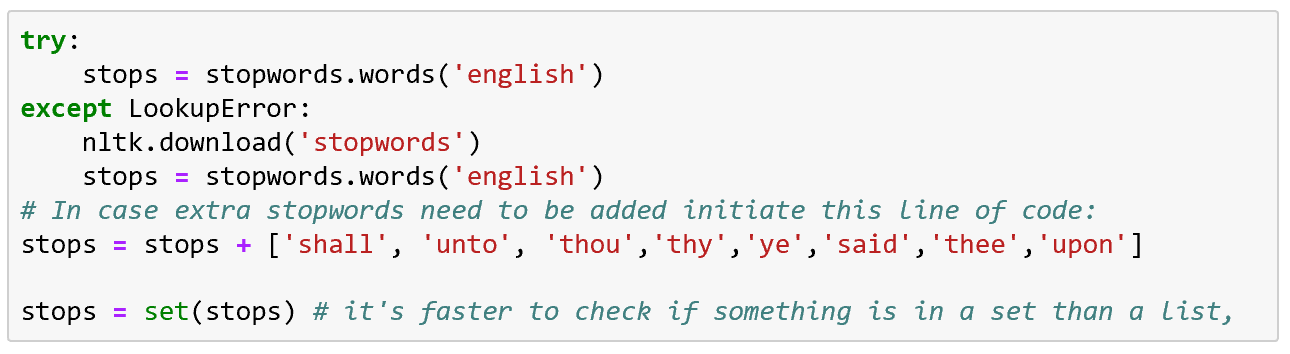


Taking out the filler words, I was able to find 44,131 distinct words in the Bible, now lets grab the top 20 used words within the text.

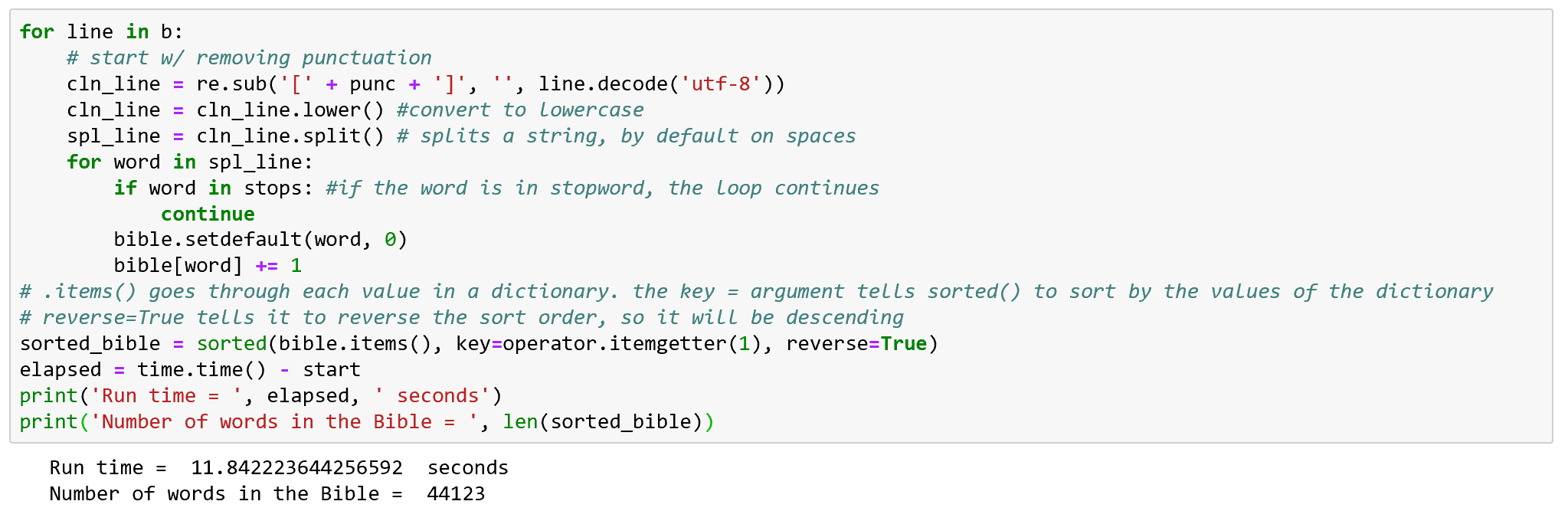


From taking out our standard filler words we have found that shall is the most used word in the Bible with 9,840 times it was used. Looking through this list I see a few words I would like to get rid of because they are older uses of filler words, so words like shall, unto, thou, thy and several others will be taken out and then I will run the for loop and compile a new list.

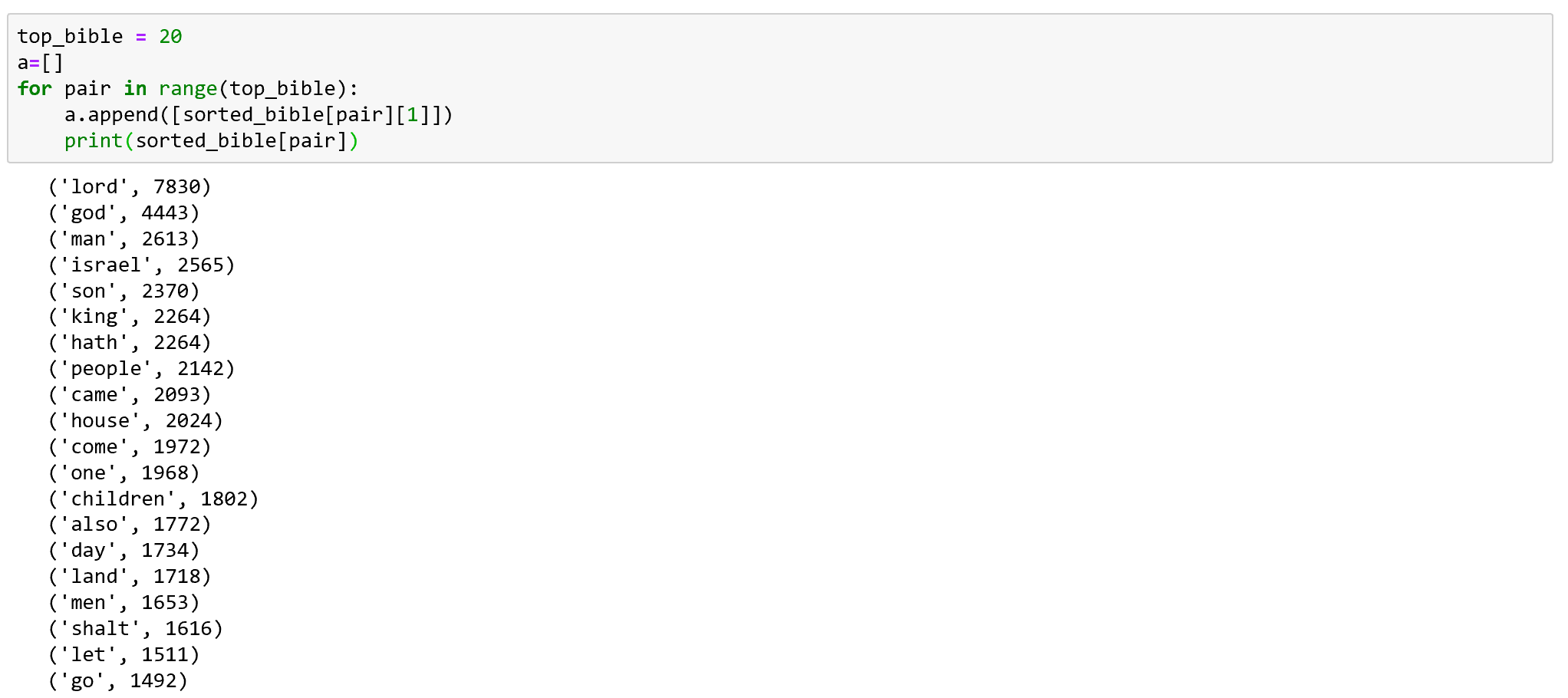
Here is the new list of filler words with the additions made.



Now I will run through the proceeding code and compile my new list count and total of distinct words.



The new distinct count is what is expected at 44,123 because I took out 8 words that I claimed to be fillers.

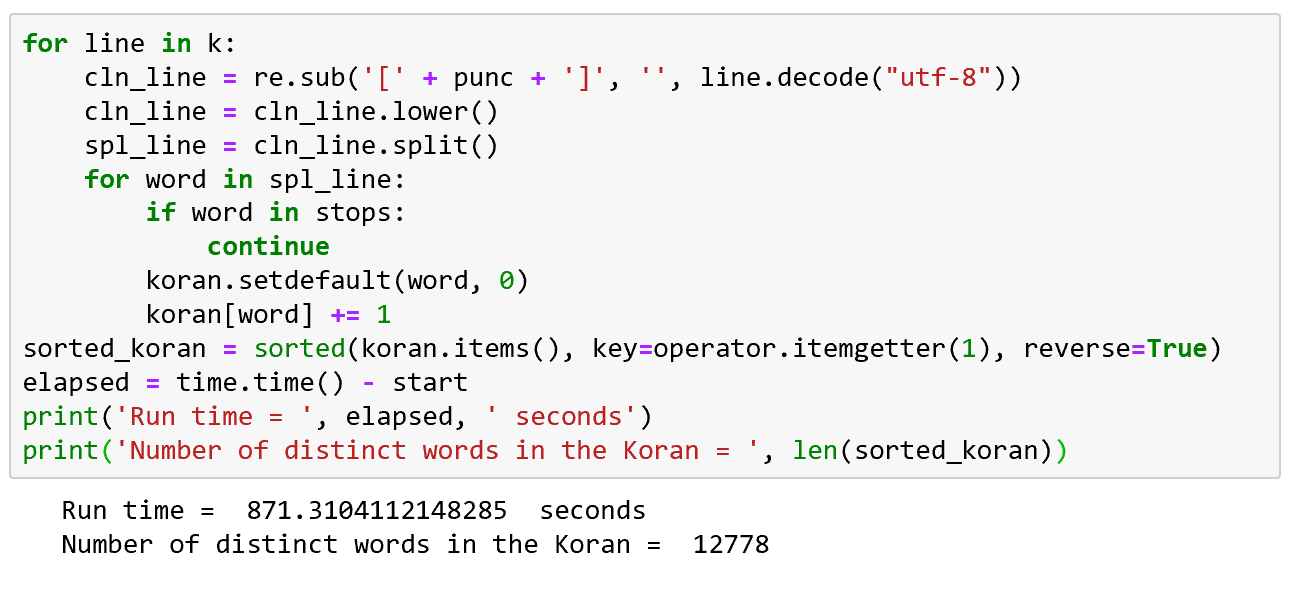


Finally, I have a list of words that are not fillers and seem to be in line with the context the Bible is trying to pass on to its readers. The top word is ‘lord’ with 7830, but the next several are ‘god’, ‘man’, ‘israel’ and ‘son’ which if you’ve read the bible are prominent in the rhetoric of the religious text.

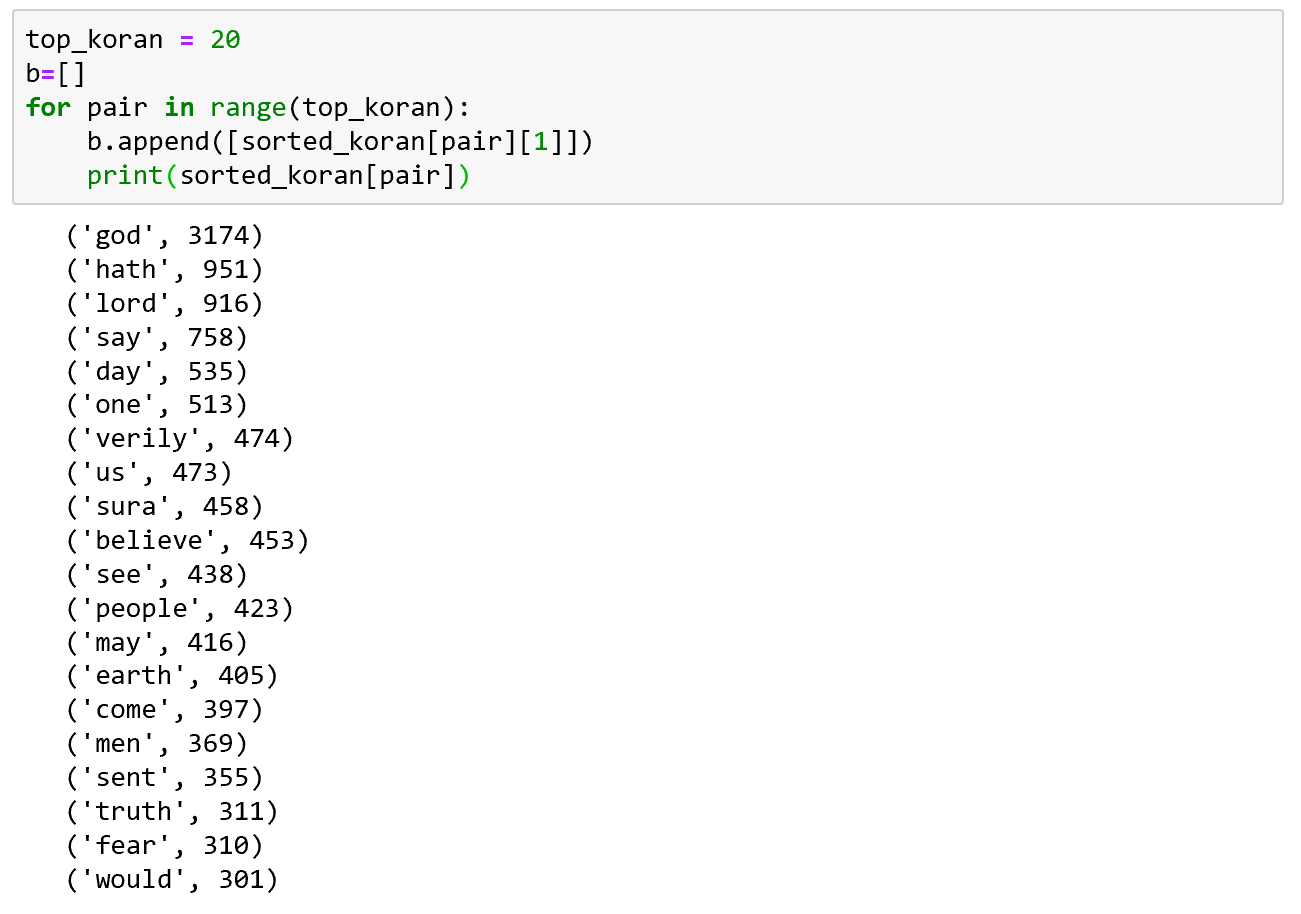
Now I will do the same analysis, this time on the Koran.

Before getting started writing the for loop, I needed to make sure my code adding filler words was commented out of my code, that way I can see the data run originally and then deceiver if additional filler words need to be added.





With the filler words taken out of the Koran, the Koran has only 12778 distinct words. Let’s see how that effected the list of most used words.



Taking out the filler words did a lot of good for seeing the meaningful words within the Koran. In the list god is overwhelmingly the number one word with over 3000 times it was used in the text but the words after are words I would expect to be in a religious text as well like the Koran. Comparing the two lists there is some obvious overlap between the two and then some omissions from each list that surprised me they weren’t in the other list. The overlapping words which I expected was ‘god’, because of religious context; hath, because back when these were originally written that was a popular filler word; and men, because each text is talking to people and referring to man or men was very common. Words that I was surprised to not see in the Koran that I saw in the Bible were words like words that were more encompassing and included everyone in the population like the words son, people, and children are all missing from the list in the Koran. While in the Koran the words ‘believe’, ‘earth’ and ‘truth’ are omitted from the Bible. From looking at these two lists and comparing and contrasting them there are some things that appear evident. One, they both are obviously religious text; two, the Bible seems to be more inclusive for every type of person; and three, the Koran is rooted more into deep faith and the importance of the Earth.

## Summary

This week was a very informative foray into text analysis using Python. I learned a lot about the process of text analysis using Python and from going into through the exercise and what I researched this week there is a lot to build from with text analysis. But for what I did in this assignment I found it really fun to compare two religious texts and see how they are similar as well as how they are different. I personally find those differences really interesting so being able to put that to use in class led to a lot of fun and interesting new thoughts about how they compare and contrast as well as what steps I would want to do next to see if I can pull more similarities and differences from the two texts. Overall though I found the exercises in general a great addition to the foundational knowledge that I need to build more upon when it comes to text analysis and the techniques and applications I need to apply to it.