Zachary Tankersley

Southern New Hampshire University

DAT-430-J7783

January 15, 2023

**Module 2 Blog Post**

Today we will quickly explore how to combine or “concatenate” two or more excel, .CSV, etc files, in a scalable way that can grow with added datasets over time. To do this we will use Python. The method I will demonstrate is simple enough for those with no prior Python experience to use:

1. First we will open our Python IDE. The first step in this code is to import the “packages” that python will use for this function. Those packages are Pandas and Glob (or global) which will allow us to use and manipulate data frames and used to return and use file paths.
2. Next we will set the “path” that our data sets will come from which should be set to your working directory which in my case is “E:\SNHU\DAT 430. Then we will specify the “files” we want to import from in the directory which will be all .CSV (in this example)
3. Now we will use a “for loop” to create a list of all the .CSV files in that directory and we will use another variable (df\_concat) to store the results of that list after using a concatenation function to combine them all.
4. Now with the two data sets used for this example, the data appears to be in proper format already, however this is when you would normally “clean” your data and do any transformations and edits needed to bring your data in line with what your processing requirements are.
5. Lastly we will export the concatenated data set by designating a file path and file name (CombinedDataset.csv in this case) and will print the dataset to verify that the function performed as intended.

In order to make things as easy as possible, both for beginners and experienced Python users, the PyCharm IDE is recommended over others and the Python Shell is very much discouraged for beginning users. PyCharm will show if you have made any errors (including simple spelling) and save you a lot of headache. The “for loop” and list technique demonstrated below may seem complicated on further exploration, however this is designed so that if additional data sets are added to your working directory later, the function will not “break” and therefore has scalability in mind at stage one. It will save a lot of effort over time. You can see from the screenshot of the codes results, that the new data set has 1400 rows, with accurate row data in each row, so the function works as described.

