This tutorial assumes that you already have Python installed and have loaded the Python Package Index (specifically the xlrd module). For a how-to on installs please refer to some of the great sites on Python basics or look for a future post on this site.

So, let us discuss the business case for utilizing pandas’ merge operation. I like to use it when I want to combine two or more data sources and I rather not use a BI tool or manual VLOOKUP’s in Excel. Two main advantages of doing this in Python, one of which is the performance. Second, I like to be able to “visualize” my data architecture and explore the data frame(s) as well and Python allows me to do just that!

For this example, I am going to be using Jupyter Notebook. I find myself using the Tableau Superstore data set quite often, and I thought it would be cool to join some of the sheets of that xlsx prior to connecting to it in Tableau (I usually perform my joins in Tableau).

So, first thing, open a new prompt (I like the Anaconda Prompt or you can use Git) and just type $ Jupyter Notebook as such:



Enter

Jupyter should now launch in your browser.

Make sure you have xlrd installed since we are going to be reading in Excel files. You can download the pip.py file here and then create a directory and point to it from the cmd prompt. You can find more information on installing pip.py **here**. Once that is complete, you can verify successful installation by going to the command prompt and typing pip -V at the prompt:

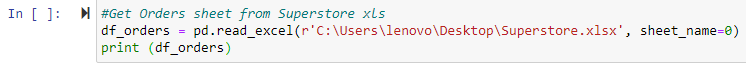


First, import Pandas and Numpy (if needed)



The Superstore data set has three sheets, for this exercise I only want to join or merge the Orders sheet and the Returns sheet. I want to read those sheets into my dataframes, now you can do this with individual kernels or in one (if you decide to copy and paste your code, make sure you change the name of your df for Returns).

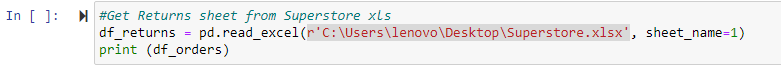
For this step we are going to use the read\_excel function in pd. I’ve named my df ‘orders’ for simplicity:



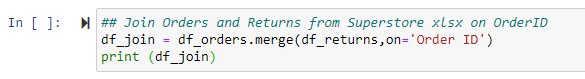
Make sure to note the directory where your file resides and give Python the correct path. You’ll notice that for sheet name, I’m using indices to identify the sheet I want to read. Python starts at 0, so since Orders is the first sheet in the file, the sheet name is set to 0.

And that is it! We can also print the df to get an idea of my data structure now that its read.

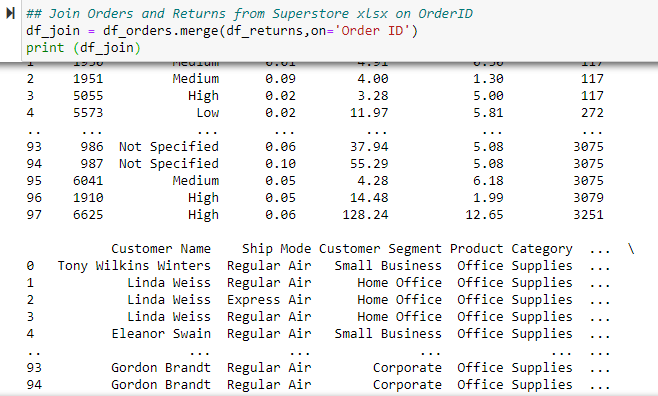
Now we will need to do the same for my second sheet, Returns:



Next, using the xlrd merge function, we will join Orders and Returns on a unique ID which in this case happens to be Order ID:



Again, we can print and see what our data set looks like after the merge.



Keep in mind that merge is equivalent to an inner join in SQL, so we only get relevant rows that exist in both dfs. In Excel terms, it is the same as a VLOOKUP where you would filter out the #NAs.

Lastly, I want to export our new data set as an xlsx file. We also do not care about the index generated column so we can drop that as well by setting index=False:



Now our files are combined and ready to connect all with a few lines of code!