Documentation for BankRisk GitHub repository:

10/18/2018 (revised)

Installing Anaconda:

* This is the software that I have been using. Recently, it has undergone major modifications in what was formerly called ipython notebook. It is now called jupyter notebook. (I think this switch must exist with Canopy Express, too.) The changes are informally referred to as The Big Switch because the developers are reconfiguring the logic of the software. As such, there are some changes that have affected how the software runs. In particular, there have been changes to the default working directories. This note shows how to return to the prior notebook working directory. (Sorry, I only ran into this issue a couple of weeks ago.)
* Choosing between Python 2 and Python 3 versions. This is actually a fairly big deal to many, many python users. I am agnostic. Choose either one. Believe it or not, it appears to me that all of the python veterans seem to stick with Python 2. There are two main differences. (1) The print command requires parentheses in Python 3. Before, parentheses would cause errors in Python 2, but I think that is fixed now. (2) Division: In Python 3, division defaults to floating point. In Python 2, division defaults to integer. (In Python 2, 5 divided by 2 is 2, not 2.5.) To fix this, simply add “from \_\_future\_\_ import division” (without quotes) at the top of the code. You will see this in several of my code modules.
* After installing Anaconda 2 or 3, go to a Windows command prompt window (click on “Command Prompt” after right-clicking the Windows icon in the lower left corner of the desktop) and type: “conda update conda”, “conda update --all”, and “conda install pandas-datareader”. (Don’t type the quotes, do these one at a time and follow the prompts.) This will update all of the software packages and add the pandas-datareader package used for downloading data from the FRB St. Louis FRED database.

Files:

* BankRisk\_Demos\_pt1.zip: Contains demo and exhibit excel files for chapters 1-6 (credit risk). These are the files I used to develop all of the exhibits for the book. Many use excel data tables (which I find to be indispensable.) Only one file is incomplete: chap05\_exhib\_2011allbanks\_pds\_template.xlsx. Two worksheets within that file were too big to upload to GitHub. The first worksheet needs to be populated with all of the data in 2011allbanks\_pds\_20111231\_roe0mm2.csv. (I left the first 20 rows or so to show how the data must be arranged.) There are roughly 7000 rows here. Next, there is a huge worksheet named 200903\_201412 that must be populated with all of the data in allbanks\_f2468\_roe0mm2\_2009\_2014.csv (74MB and 165,000+ rows). Again, I left 20 or so rows to show how the data must be arranged. (Note: Those files will be generated by running the ipynb code for the PD/LGD models.) Place these files into the “C://BankRisk” directory.
* BankRisk\_Demos\_pt2.zip: Contains demo and exhibit excel files for chapters 7-11 (interest rate risk). Place these files into the “C://BankRisk” directory.
* BankRisk\_ipynb\_20181018py3.zip: All of the ipython (Jupyter) notebook code files. These files should be placed into a subdirectory called: C://Users/yourname/bankrisk. The python software will look for the code files in the C://Users/yourname directory. When it opens you should see bankrisk listed as a subdirectory. Just click on it to work with the code files. (Do not attempt to run these files if you are not proficient in python and ipython notebook. This is not the material to use if you are just beginning to learn python.) (See appendix below for recent changes in these programming files caused by changes to python and pandas.)
* BankRisk\_processingtables.csv: These are all of the ancillary data tables that are referenced by the ipynb files. They should all be placed into the root directory called “C://BankRisk”.
* df2\_2001\_200912.7z and df2\_2010\_201806.7z: These are the Call Report raw data extracts for the indicated years for all banks in the country. (I also provided the python code that created these files from the FDIC bulk data files in case you are curious.) You must use 7-Zip open source software (which is superior in many ways to WinZip) to un-compress these files. Put them in the “C://BankRisk” directory.

How to get started:

1. Run br001b\_CallDataExtr\_concat\_201712.ipynb. This concatenates the two large df2… files into one even larger one which is used for everything.
2. Run br001c\_CallDataCalcs\_AllBanks\_201712.ipynb. This processes the raw Call Report data stored in the df2… files into calculated fields. For example, the Call Report shows year-to-date income and expense numbers. This processing will convert them into quarterly numbers to allow rolling 4q averages to be calculated. (This takes a long time… about 2 hours on my laptop. The output contains timestamps that you can examine before executing each block to get an idea of the time required at each stage.)
3. Once steps 1 and 2 are completed, you are ready to start the stress test code files. These contain either “stress” or “univstress” in the filenames.
4. Definitely study: br003ma, br003mb, and br003mc code files. They are the same code, but they are configured to illustrate how to set up specialized BHCs, large aggregates, or small aggregates. There is a large block of code near the top of the files where all of the selections are made. The output from these three files is discussed in detail in the book.
5. Generally, if you scan through any of the ipynb notebook files, you should recognize the outputs from the book. This will give you a good idea about what each module is used for. Please note that the “univstress” modules provide output for the full models, the 3-variable models, and the 4-variable models. You can select any of those you like.
6. Good luck! Let me know if you run into problems.

Update for 2016, 2017 and 2018 CCAR models:

* The latest set of ipynb files that run CCAR stress tests have generally been updated to run the 2016, 2017 and 2018 CCAR scenarios. Note that the start date and end date for the forecasts was shifted in 2016. Instead of starting with 2015-09-30 as earlier CCAR years did, it now starts with 2015-12-31. Also, instead of ending with 2018-12-31, it now ends with 2019-03-31. Certain code files run sets of four CCAR years (such as 2012, 2013, 2014, and 2015). These files contain “bigloop” in the filename. In each, there is a variable “ccyr” that defines which four years to run. If you want the latest four CCAR years, change this variable as follows: ccyr= [ 2015, 2016, 2017,2018].

SPECIAL APPENDIX (2018-10-16, revised 10/18/18):

There are now two zip files containing the ipynb programming files with the dates 201808 (Aug-2018) and 20181018 (10/18/18) embedded in the zip file names. Importantly, the August file also has “donotuse” in the zip file name. This appendix explains what happened.

The zip file designated 201808 was the file present with the Aug-2018 update for adding data through 2018-June. I always check that the code files will process without errors before posting this type of update. However, I did not bother to update all of the python and pandas language files before checking for errors. Unfortunately, periodically, some of the prior commands get deprecated… meaning that they cease to work and will generate errors instead. These are situations where the designers of the language decide that there is some problem or ambiguity with the old syntax and decide to change it. Hence, it is possible that code that works with older versions of the python/pandas language can stop working at an unpredictable time in the future.

To my horror, I discovered that this happened with the August update. It worked in August using older versions of python and pandas, but it stopped working in October after I updated the python and pandas languages. In the spirit of full disclosure, the changes related to three items:

1. Rolling\_mean(args) and rolling\_sum(args) stopped working and were replaced with rolling(args).mean() and rolling(args).sum().
2. Column name changes must to handled using a new “rename” command.
3. Slicing the lower limit of bin cuts cannot use the “index” attribute.
4. Data column duplication related to “group by” commands.

These are very technical changes. I have included both sets of ipynb programming files (August and October) in case any of you wish to check the detailed syntax changes and/or differences. As always, I will respond to any direct inquiries via my email address available on my GitHub site.

My apologies for any inconvenience these issues may have caused.