**THIS BIT WAS A BRIEF OUTLINE OF THE DB ASPECT OF THE QUANT TEAM I PREPARED FOR ANDY SO THAT HE COULD THINK ABOUT WHAT TO INCLUDE ABOUT THE DB, IF ANYTHING, FOR THE PRESENTATIONS WHICH NEVER HAPPENED**

**Overview of SMF DB:**

* We used Amazon’s Relational Database Service (Amazon RDS) provided by Amazon AWS to set up and host our cloud-based (relational) database.
* We communicated with the DB using SQL (the particular extension used this year was MySQL, pronounced “my sequel or my S-Q-L”).

**Last year (2018/19):**

* We used the DB to store:
  + Historical closing prices from Yahoo Finance – for the optimiser
  + Historical PE values from Thomson Reuters Eikon – for the ranker
  + The portfolios that we got from using the optimiser

**This year (2019/20):**

* We used the DB to store:
  + Vast amounts of stock data from SNP500 and RUSSELL2000: ~2400 tables corresponding to 2400 stocks, with the following financial time series data:
    - Closing prices
    - Value metrics (PE, Price-to-book, Price-to-FCF, EV-EBITDA)
  + Sector/industry data for each stock
* All the data was obtained via Thomson Reuters Eikon this time.
* The data was extracted from Eikon, processed (e.g. data imputation for missing values, and organised into tables), and uploaded to the DB. This was done through Python. More specifically, the data was obtained through the Eikon Data API, processed mainly with pandas, and stored in the DB with sqlalchemy.

####################################################################################################################################################################

**BELOW IS AN OVERVIEW AND DESCRIPTION OF EACH FILE AND ITS USE, IF ANY**

NOTE: The actual DB isn’t there anymore because I created it using AWS Free Tier which expires after 12 months. Ain’t no one gonna make me pay to keep it running…

**FILES:**

tests/db\_test.py

* Not much here. Just basic usage of eikon and opening/closing DB connection.

credentials.json

* This file isn’t on GitHub, but it’s what I used to access the DB (via the DB credentials).
* The credentials should be available on the DB dashboard. Just fill them in within the file and change the filepath accordingly for the json.load() argument.

smf/db\_engine.py

* All DB-related methods and processes are defined here.

smf/db\_functions.py

* Never used it. This is actually a carry-over from last year’s SMF

db\_set\_up\_tables.py

* The #DONE comments were used to keep track of code blocks that were already run. I think I did this because the whole script didn’t work, and instead of deleting the DB tables that were created and then running the whole script again, it was easier to just comment out the blocks that ran without any issue. I also did this because we had to keep adding historical volatility tables, so there was no point in re-running the old code (which would actually cause an error).
* The volatilities\_days array values were the list of available volatilities.

db\_populate\_and\_update.ipynb

* Same as db\_set\_up\_tables.py. It was just easier to run code blocks/segments via Jupyter Notebook.