**DynamicTech Mini Project**

1. **Framing the problem and objectives with assumptions**

The objective is to download daily files from the SGX website: <https://www.sgx.com/research-education/derivatives>

1) WEBPXTICK\_DT-\*.zip

2) TickData\_structure.dat

3) TC\_\*.txt

4) TC\_structure.dat

Exploring the website, SGX publishes the files for the past 5 market days. The assumption we will make is that running our script/job daily by default will give us the previous day files since it is the most updated eg. Running on the 11th August will gives us 10th August files. For any other user input it will give the exact date files that the user inputs.

1. **Exploratory analysis**

Inspecting the HTML page on SGX website, we can see that certain elements are hidden unless the drop down list is clicked. It also seems that the website is AJAX coded and given the options from the drop down list it will query from their own database on the backend when the download button is clicked. We can see that the download link generated is in the structure where id is queried from database and filewould be determined from drop down list:

**https://links.sgx.com/1.0.0/derivatives-historical/ + id + file**

One example would be: **https://links.sgx.com/1.0.0/derivatives-historical/4700/WEBPXTICK\_DT.zip**

1. **Proposed ideas & solutions**

The proposed solution would be to use selenium to interact with the webpage and to click the download button to obtain the files for dates that are available on the SGX website (past 5 market days). For input dates that are not available on the website, we would use a get request using the download URL to obtain the files by determining the date idusing a function and algorithm which will be further explained.

We create a workflow to determine the possibilities and outcomes that will occur:

1. Input date(< 5 days) → Available on website → Selenium → Download complete
2. Input date(> 5 days) → Get request → Download complete
3. Input date(> last updated date on website) → Not updated on website → Download failed
4. Input date(weekend) → Download failed

We will restrict user inputs to only market days hence we will need to properly define what is a market day. The initial thought process is that market days will exclude public holidays and weekends, however upon further exploration, derivatives are traded even on public holidays and there were files generated. The assumption that files would not be generated on weekends seems consistent with SGX however there was a particular instance whereby files were generated on a Saturday 1st February 2020, exploring the website for their trading schedule did not produce any results and we are unable to determine if this was abnormal. Also, we are unable to determine if this date was available on their website as an option to download from. Hence, for this project we will define market days as weekdays including public holidays and thus by default any input dates which are weekend would lead to a download failed based on our workflow.

1. **Algorithm to determine date ID**

To determine the date ID, we would first need a reference date and its corresponding ID. Then, the function would take in an input date and the reference date and calculate the number of days between them excluding the weekends using numpy function busday\_count. Using the difference we would either add or subtract from the reference date ID and obtain the input date ID. To account for irregularities such as files being available on the weekend, we would use an offset to be added to the ID as it is a weekend and the new reference date and ID would be updated in the config file. Moving forward, this would ensure that our algorithm would be robust enough to handle such exceptions and the get request method to download the files would be accurate.

1. **Recovery plan and Caveats**

Since we are unable to determine which weekends had files generated besides iteratively downloading all files starting from ID:1 and checking the dates, this get request method would only work starting from 3rd Feb 2020 since the first abnormality was on the 1st Feb 2020. However, moving forward assuming that we run the script daily to check if a file was generated on the weekend based on SGX website, we would be able to update the reference ID and offset accordingly.

If files fail to download based on a correct weekday date input format, the most likely reason is that the date given was a future date or on the rare occasion that SGX website has not been updated and the file is not available to be downloaded. In this case, the script should be run manually at a later time to redownload the files. In all other cases that files fail to download, it could be a more serious issue such as URL change or website design has changed. This would require source code changes.

1. **Possible improvements**

To further enhance user experience, a simple GUI could have been created using tkinter. To automate and deploy script to run daily, we could use windows task scheduler or cron(linux) to schedule a job writing a bat script.