Performance Loss Analysis Methods on UNSW TEBT PV System

Christopher H. Fowler
z3470052
Supervisor: Dr. Jose Bilbao

1. Reporting on Progress

To report on the progress I have made over the term, given the data focused nature of this project, I have decided it's easier to present iterations of code and results that formed parts of the methodology that I've developed so far. As a refresher, the goals of the projects are as follows:

Primary Goals

- Implement standard performance loss methodologies (LR, STL, YOY)
- Apply performance loss methodologies to UNSW TEBT data

Secondary Goals

- Determine effectiveness of data cleaning methods used in PL analysis
- Determine effectiveness of data fitting methods used in PL analysis
- Observe non-linear PL in analysis
- Explore alternative methodologies as time permits

The primary goals have had major hiccups in 'Applying performance loss methodologies to UNSW TEBT data' where processing the TEBT data into a useable form has taken a significant portion of time. The trials of completing this task make up the bulk of this report, along with reporting on the regular methodology test examples that have been used.

The Industry Methodology

To fulfil any of the standard performance loss methodologies we follow the 5 general steps outlined by the industry for this type of problem:

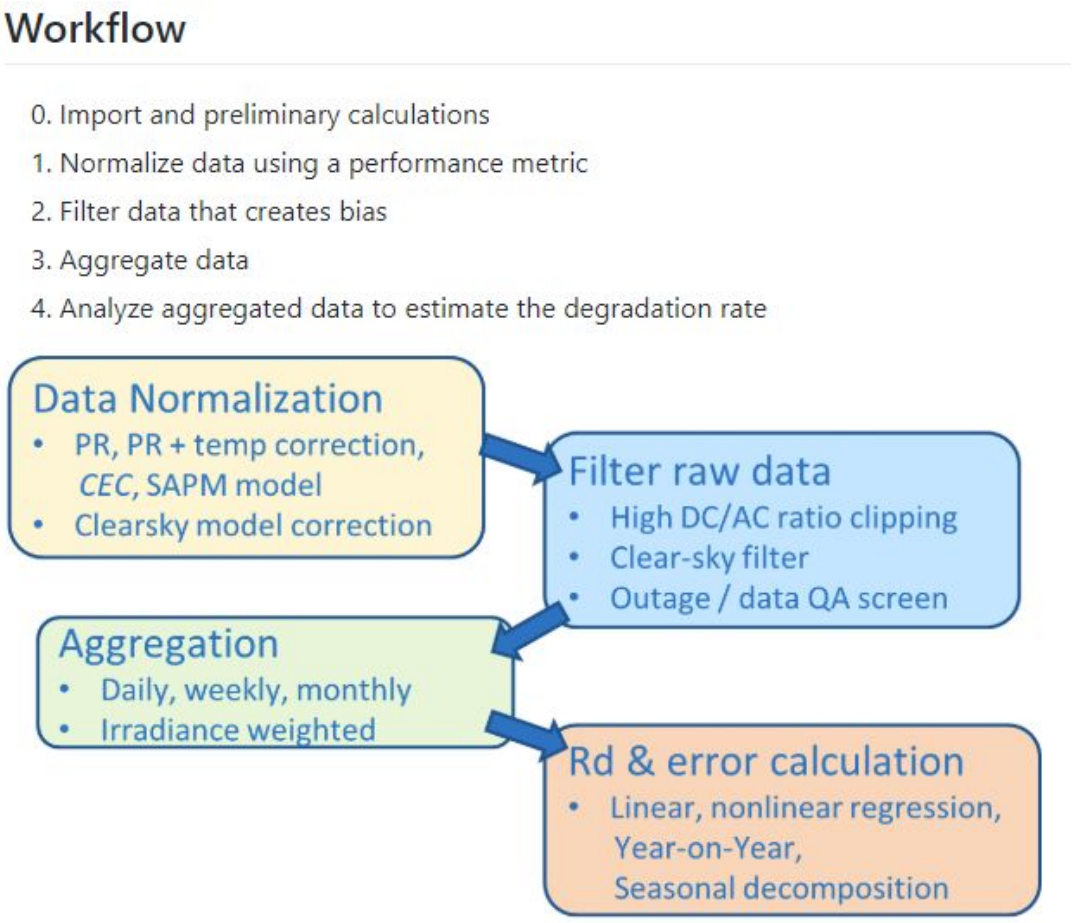

Fig 1: RDTools methodology

[RdTools, version 1.2.2, https://github.com/NREL/rdtools, DOI:10.5281/zenodo.1210316]

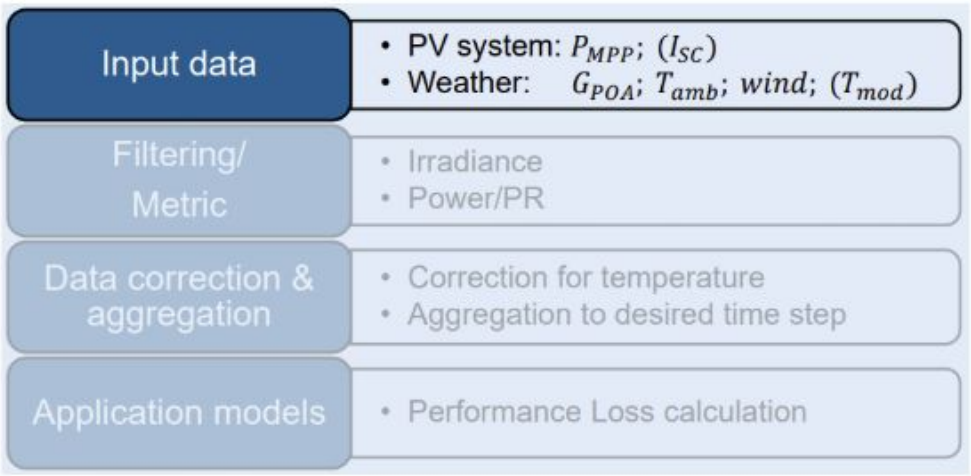

Fig 2: EURAC Methodology
[D Moser, 2019, “Performance, Operation and Reliability of Photovoltaic Systems”]

As part of correctly using these methodologies, the RDTools (giving YOY w/ stages 1-4) and Statsmodels (giving STL w/ stage 2-4) packages give working examples on test datasets that can then be modified to fit the dataset we provide. With respect to how have been and am going about implementing these methodologies, please refer to the ‘Revised Project Plan’ section.

Overview of Progress through Git commits

Setting up and using git to maintain a healthy codebase is essential as part of any coding project, and a tool that I am quite the novice at using. Below in fig 3 is the result of my amertuer attempts to use git properly - the list of commits is relatively short with very large old commits at the bottom and very small more recent commit at the top as I began to use git more appropriately. There were issues with some merge conflicts between my two workstations which were finally fixed in a large merge under the master branch. The ‘ThesisNight’ branch is the culmination of more recent attempts at importing data from TEBT. Some assistance has been given from a helpful computer science 4th year undergrad who was more than eager to groan at my many mistakes in using this tool, however progress has been made to use it properly.

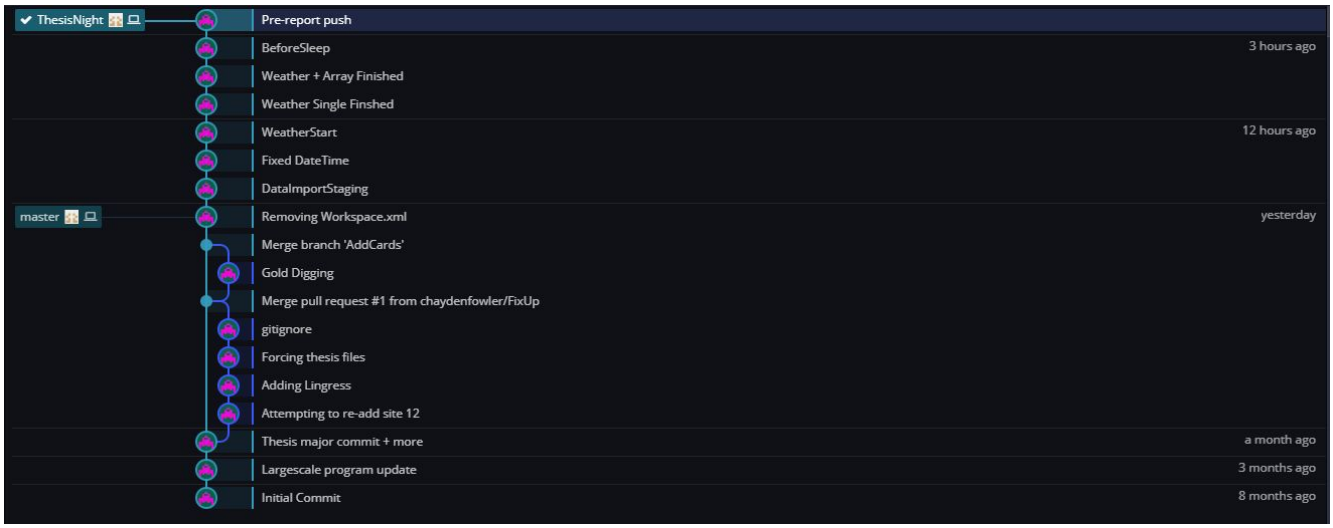

Fig 3: Flow of git commits

Test Examples on Methodologies

In order to test the methodologies, package devs tend to create sample datasets to test on. In order to implement the methodologies, the first step is to execute these test sets to ensure correct installation and use. Below in figures 4 and 5 are the (expected) results of the test examples. Installing the correct package version of statsmodels was more difficult than expected, as the dev branch requires compilation and several OS specific dependencies not linked to the pip package manager, as a result fig 6 is my proof of success in properly getting this package to work.

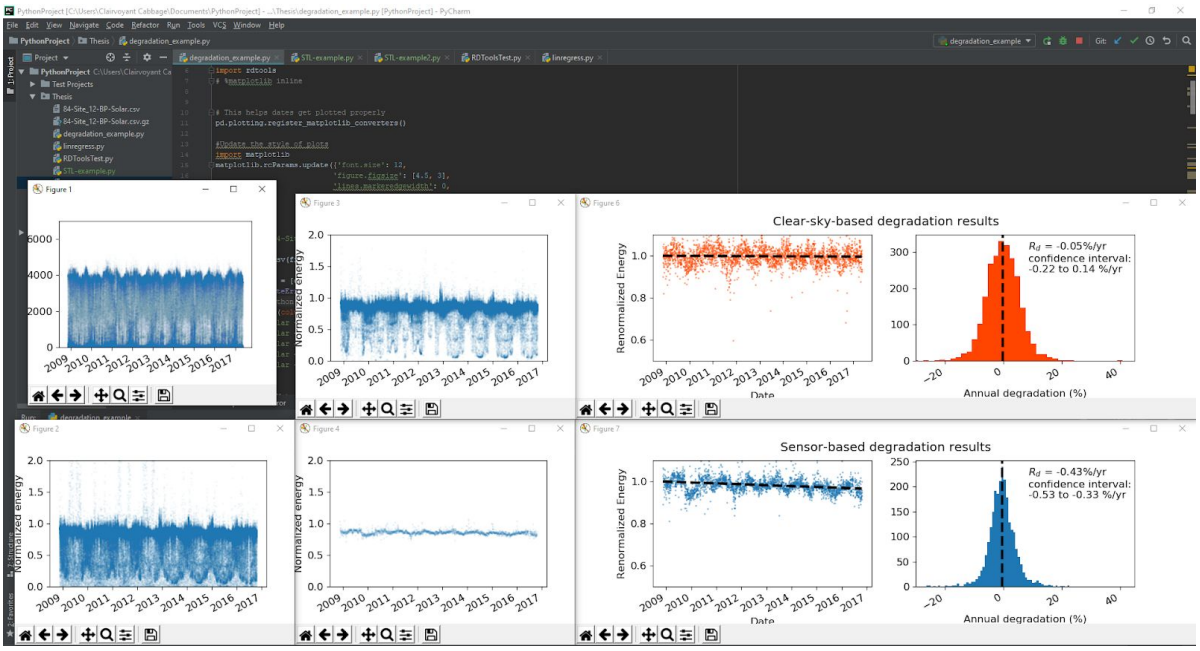

Fig 4: RDTools test example on Desert Knowledge unit 12

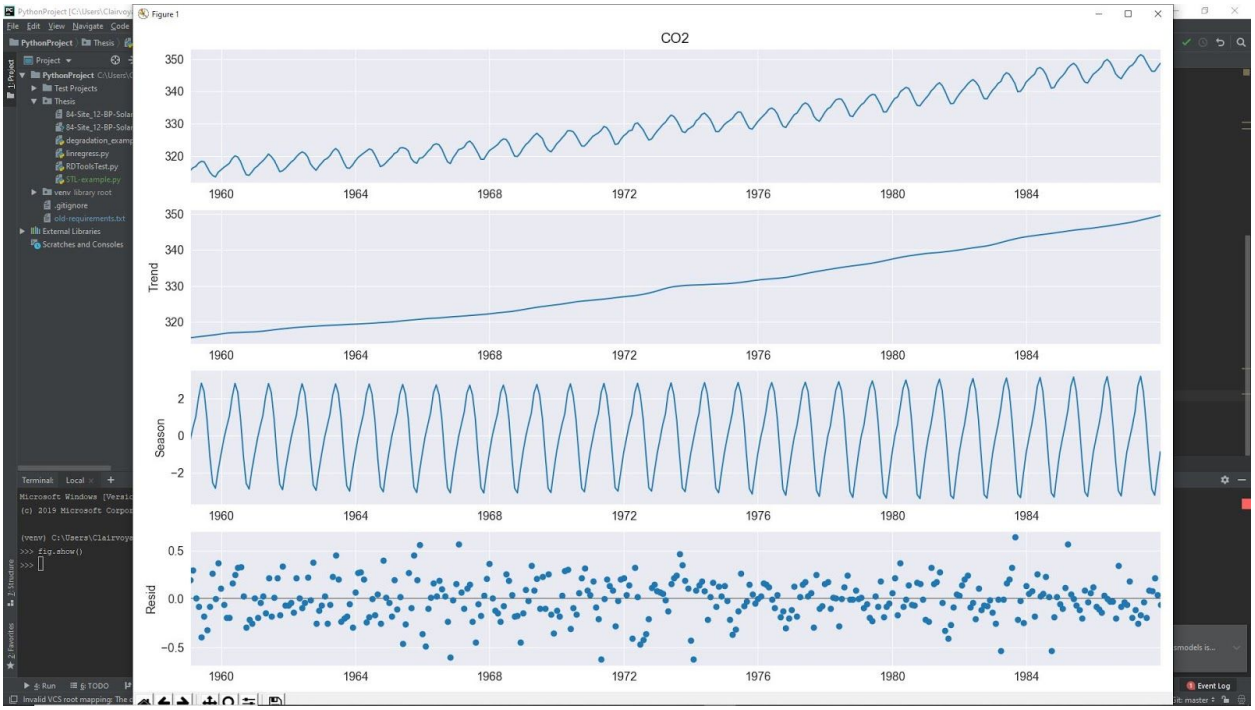

Fig 5: Statsmodels test example on CO2 emissions

Command Prompt

```
(venv) C:\Users\Clairvoyant Cabbage\Documents\PythonProject>pip list
Package         Version
--------------- -------------------------
certifi         2019.9.11
chardet         3.0.4
cycler          0.10.0
Cython          0.29.13
h5py            2.10.0
idna            2.8
kiwisolver      1.1.0
matplotlib      3.1.1
numpy           1.17.3
pandas          0.25.2
patsy           0.5.1
pip             19.3.1
pvlib           0.5.2
pyparsing       2.4.2
python-dateutil 2.8.0
pytz            2019.3
rdtools         1.2.2
requests        2.22.0
scipy           1.3.1
seaborn         0.9.0
setuptools      41.4.0
six             1.12.0
statsmodels     0.11.0.dev0+517.g0315fdd24
urllib3         1.25.6

(venv) C:\Users\Clairvoyant Cabbage\Documents\PythonProject>
```

Fig 6: Successfully installing dev branch of statsmodels

Jupyter Notebooks

In order to become more accustomed to the way the industry is reporting changes, test examples and explaining how their product works, I've taken to using Jupyter notebooks for the data import phase of my methodology. The notebooks are quite space inefficient when converting from ipynb to pdf, and so this section will exceed the '15 page soft limit' under the pretense that there is significantly less information to read through per page than usual.

Formatting space could be saved by instead screenshotting the results I have, but for the purposes of reproducibility and being thorough, the code and results should be fully available by the auto-formatted pdfs. As Jupyter can export .TeX files, in future LaTeX will be considered to improve report formatting.

Some reflections on progress are included in the Jupyter notebooks as being next to the issue or solutions is more relevant than in its own section, although reflections on progress will be summarised later also.

2. Reflection on Progress

Challenges

The reflections on issues faced over the process of building this project are primarily explained with the challenge or solution relevant in the Jupyter notebooks, however I will summarise some of the issues faced.

When it came to importing the data from UNSW TEBT, overcoming the awkward file composition, missing data and corrupted files was quite the learning experience. After choosing to include some tolerance of failure for the import as shown in fig 7, I see that trying to get absolutely everything to work would have been unfeasible, as somewhat expected. Of the 350 or so weather files (which we could only hope contained 365 days of data) around 20 or so failed on any attempt, sometimes changing depending on what data was being asked from it. Any 0KB files always failed, but depending on what columns were requested, or what dtypes pandas thought the column was, sometimes more or less files were accepted, but this was always consistent when using the same process.

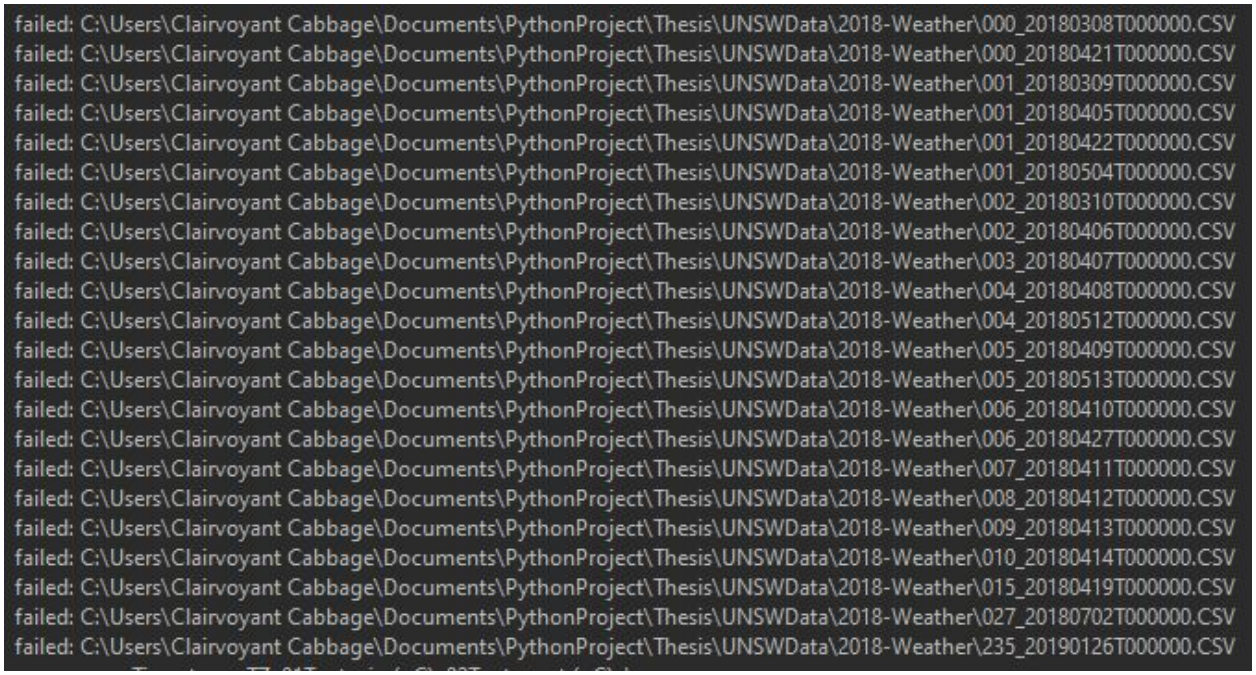

Fig 7: Importing weather data files, sometimes things just don't work

Failing at getting STL to work on my 1 year TEBT dataset was quite disheartening, but this is still in early works and hopefully can be remedied. Some of the other coding challenges were quite enjoyable, if time consuming, the more minor of which can be found in the Jupyter notebooks as there are too many more to list.

Differences in Research and Learning

This thesis project has definitely changed direction and momentum over the course of its life. Originally on the case of my literature review, a large section of my personal research was into degradation failure modes and more into the physics based models for identifying degradation sources. These topics were somewhat common when discussing with other students, postgraduates, and professors. Upon finding the analytics and statistics work being done by the international community, the project completely shifted and the skills required more suited to what I had in mind when I originally was deciding my thesis project.

Now that the project is less focused on physics and failure mechanisms I have more freedom to learn about the statistics and methodologies behind the analysis, rather than the science behind the degradation modes (although this was initially done at the lit review stage). In this way, I've completely changes the angle of approach for the thesis to something I find more satisfying.

The literature presented when doing both my lit review and project replanning are quite undiscussed during internal courses. Some talks and seminars have discussed these areas of knowledge however. When I was learning the information required, almost all information was completely new to me (although perhaps some bias towards ignoring the context I already have), and I am glad that this research has broadened my skill set.

3. Revised Project Plan

The project plan is still changing a little as goals are being reached, or not reached. After my literature review the 'Initial plan' and goals were mostly scrapped for the 'Recalibrated Plan' and goals already. As we are now reaching the end of thesis B some further planning is required, and updates to my recalibrated plan must be made to accomodate thesis C. The thesis C plan will include a significant portion of time attributed to writing in order to develop a cleaner final report.

Initial Goals and timeline

Primary Goals

- Determine whether Clear-sky YOY or RPCA is a more effective methodology
- Determine degradation rate of UNSW TETB array using Clear-sky YOY and RPCA

Secondary Goals

- Determine effectiveness of data cleaning methods used in degradation analysis
- Determine effectiveness of data fitting methods used in degradation analysis
- Observe non-linear degradation in analysis
- Determine soiling rates of UNSW PV arrays based on tilt

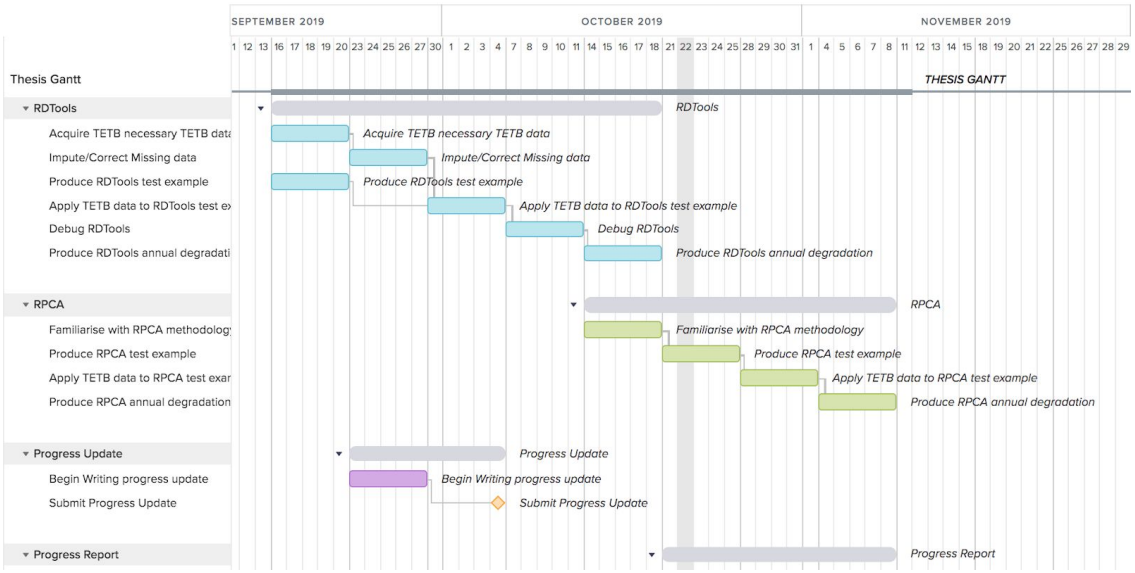

Fig 8: Initial Plan

Recalibrated Goals and timeline

Primary Goals

- Implement standard performance loss methodologies (LR, STL, YOY)
- Apply performance loss methodologies to UNSW TEBT data

Secondary Goals

- Determine effectiveness of data cleaning methods used in PL analysis
- Determine effectiveness of data fitting methods used in PL analysis
- Observe non-linear PL in analysis
- Explore alternative methodologies as time permits

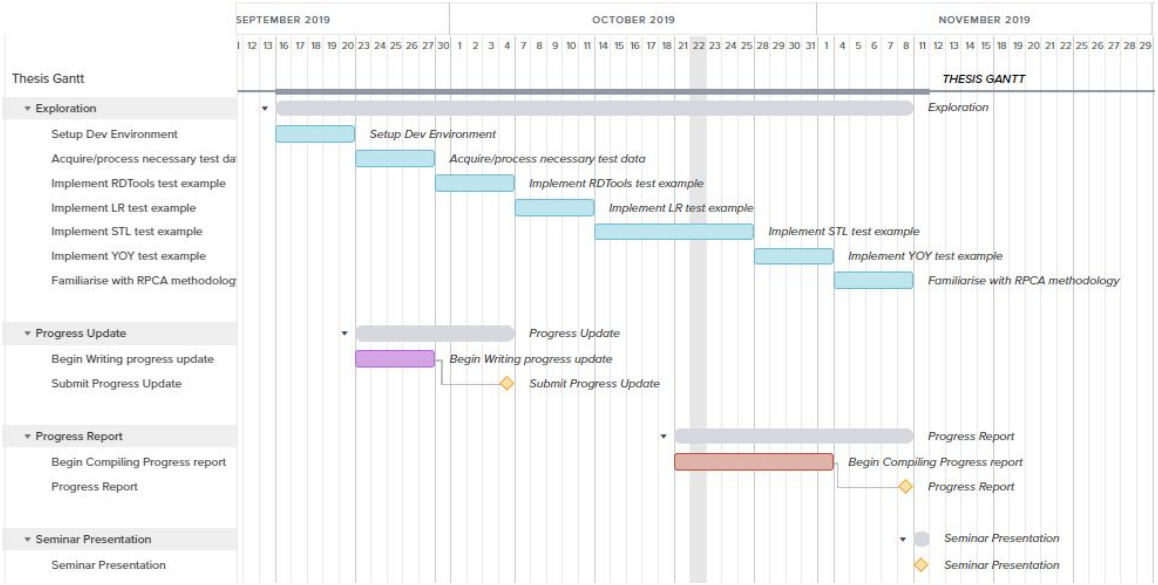

Fig 9: Recalibrated Plan

Thesis C plan

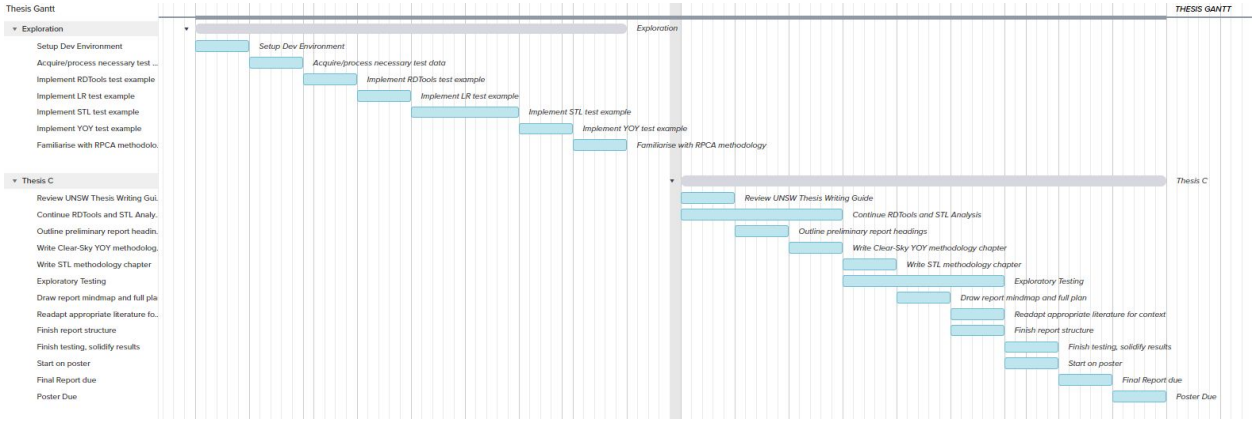

Fig 10: Recalibrated + Thesis C plan