Still a work in progress...

There are many ways to set-up LaTeX on your local machine. My favorite is to use VS Code with Latex Workshop. There are several guides on the internet for the basic set-up of VS Code with Latex Workshop. But here are a couple things I have learned and found useful.

0.1   Using LuaLatex without using latexmk

Most of the time, it is better to do a single compilation between minor changes as you edit and then a couple more recompiles to make sure everything is settled right before submission. I also like auxillary files to be in a separate folder and therefore provide instructions to that end.

To do this, we need to add extra recipes in order to have an option to just run LuaLatex or possibly LuaLatex + Biber + LuaLatex (for after bibliography changes). Go to Settings, search for Recipes, then choose the option to open Settings.json. Search for "latex-workshop.latex.tools" and **add** to the list there:

```
{
        "name": "lualatex",
        "command": "lualatex",
        "args": [
            "--aux-directory=%DIR%/LatexAux",
            "--interaction=nonstopmode",
            "--synctex=1",
            "--c-style-errors",
            "--output-directory=%DIR%",
            "%DOC%"
        ],
        "env": {}
},
{
        "name": "biber",
        "command": "biber",
        "args": [
            "%DIR%/LatexAux/%DOCFILE%",
            "--output-directory=%DIR%/LatexAux",
            "--input-directory=%DIR%"
        ],
        "env": {}
},
```

Then, search for "latex-workshop.latex.recipes" and add to the list there:

```
 {
        "name": "lualatex-plain",
        "tools": [
            "lualatex"
        ]
    },
    {
        "name": "lualatex with biber",
        "tools": [
            "lualatex",
            "biber",
            "lualatex"
        ]
    },
    {
        "name": "biber",
        "tools": [
            "biber"
        ]
    },
```

Now, in the TeX menu there will be the option to run just "lualatex-plain", "biber", or "lualatex with biber." You can change the default used for your file by adding to the top of thesis.tex

```
%!LW recipe=lualatex with biber
```

0.2   Using Version Control

Using version control is a great way to back-up your work and be able to see the history of the document. VS Code has built-in support for GitHub, which you can find by choosing the version control tab on the left. You will need a GitHub account (free; GitHub is owned for Microsoft).

For version control to be at its best, however, you don't any one line to be too long. One way to accomplish this is to have every sentence on its own line (as LaTeX ignores single new-lines). You can do this manually, or you can configure VS Code to automatically run a script called "latexindent.pl" (https://github.com/cmhughes/latexindent.pl) when you save the file to format your file for you. Occasionally, this adds line breaks where they should not be, but it is usually a minor problem.

If you want to use latexmk, you need to adjust the settings to allow it to modify the line breaks. You can do this e.g. with the settings in localSettings.yaml. Further, in VS Code, you need to modify the setting "Latexindent: Args" and add "-l" (use local settings) and "-m" (modify line breaks).