New Computer Fan Data-Science Set-up
Python (Anaconda), R (Rstudio), Matlab and Latex (Texlive)
Atom, VScode, Pycharm

Go back to fan's Tex4Econ and Miscellaneous Repository.

1 Objective

Install various compilers and IDE:

1. Python (Anaconda, Pycharm + VSCode + Atom)
2. R (VSCode + Atom)
3. Matlab
4. Latex and PDF (texlive, Okular, VSCode + Atom + Texstudio)

Test at the end:

1. Do sample *py* files work?
   - work from command line
   - work from atom (hydrogen), Pycharm, VSCode
2. Do sample *rmd* files work?
   - work from r-studio
   - work from atom (hydrogen), VSCode
3. Do tex templates compile?
   - tikz files, template files with bib
   - work from texstudio, VSCode live compile, Atom

Installation Plan outline:

1. Conda install (python)
2. Install Atom along with Hydrogen and other packages.

2 Python and Anaconda Installation

Use conda across platforms, so that locally on windows and ubuntu and remotely on aws, can have the same software setup environment.

- Search for Anaconda Prompt, right click, choose run as administrator.
- Check software versions.

```
conda list anaconda
python -V
```

2.1 First Time Conda Install

Download Anaconda Python 3 and install for all users. Afterwards, Anaconda does not automatically get added to Windows Path. Need to use Anaconda Prompt to access programs. To access Anaconda packages from windows prompt, from git bash, from R, etc, need to Add Anaconda to Windows Path.

To install conda in Linux/Debian, follow these instructions. Overall, installations are very similar.

In command/anaconda prompt:

```
# To remove conda Fully
rm -rf ~/anaconda3

where anaconda
# C:/ProgramData/Anaconda3/Scripts/anaconda.exe
where python
# C:/ProgramData/Anaconda3/python.exe
where jupyter
# C:/ProgramData/Anaconda3/Scripts/jupyter.exe
where jupyter-kernelspec
# if R installed already
where r

###############################
# Add these to Windows PATH:
###############################
# C:\ProgramData\Anaconda3\Scripts
# C:\ProgramData\Anaconda3
```

To Add Anaconda to Path, In Windows 1. Search for: Environment Variables 2. Edit Environment Variables 3. Add new to Path (lower half): - C:/ProgramData/Anaconda3/Scripts/ - C:/ProgramData/Anaconda3/ 4. Now open up regular windows command Prompt, Type in: - conda –version - also Close and Open up Git Bash: conda –version

2.2 Conda Update

Open up Anaconda Navigator, it will update navigator automatically. If there are errors, might have to clean first.

There are different channels/repositories from which packages could be installed: *conda*, *conda-forge*, for example. Can change where to look for packages first. Many packages are in multiple channels. Conda-forge often have mroe recent packages.

```
# if there are bugs
# conda clean --packages
# use conda-forge as main channel, more updated packages
conda config --get channels
conda config --add channels conda-forge
conda config --get channels
# remove conda-forge from channels (do so for main env install)
conda config --remove channels conda-forge
# normal update
conda update --all

# install additional packages
conda install -y statsmodels datashape seaborn
conda install -c conda-forge -y interpolation awscli
conda install -c anaconda -y boto3
```

2.3 Python Command Line Test

Fist test python under command line:

```
# windows
python "C:/Users/fan/PyFan/ProjectSupport/Testing/Numpy/Functions.py"
python "C:/Users/fan/PyFan/ProjectSupport/graph/subplot.py"
# linux
python ~/PyFan/ProjectSupport/Testing/Numpy/Functions.py
python ~/PyFan/ProjectSupport/graph/subplot.py
```

3 R Installation

R could be installed from Anaconda or directly on windows outside of the Anaconda directories. When installing from within Anaconda, could create R environment that are isolated from the rest of the computer. The R environment could be updated and deleted without disturbing dependencies in conda main or the rest of the computer.

The computer could have multiple R installations. Several in different conda R environment, and also several under windows/linux primary main user R directories. Type *which R/where R* to see if you open up R from command line at the moment, which directory's R will be used. Inside Conda R environment, it will be a different R if there is a different R version there.

3.1 Fully Uninstall r

If R was installed in R enviornments, just delete the environment. Otherwise, use system's uninstaller. Before that, from terminal/command-prompt, type *R*, and *.libPaths()* to find paths. Do so inside conda main, outside of conda main, inside different environments, find all paths. After uninstaller finishes, check in the libPath folders to see if there are still stuff there, delete all, delete the folders.

```
.libPath()
# C:/Users/fan/Documents/R/win-library/3.6
# C:/Program Files/R/R-3.6.1/library
```

For Linux and for unsintalling inside conda:

```
# Exit Conda
conda deactivate
# where is R installed outside of Conda
which R
# /usr/bin/R
# To remove all
sudo apt-get remove r-base
sudo apt-get remove r-base-core

# Inside Conda base
conda activate
# Conda r_env
conda activate r_env
# Where is it installed?
which R
# /home/wangfanbsg75/anaconda3/bin/R
conda uninstall r-base
```

3.2 Install R (outside of Conda)

Need to install R. Then need to install also an editor/IDE for R. Rstudio is the dominant R IDE. VSCode and Atom also works well, especially VSCode as editors. Dramatically better experience for writing scripts outside of Rstudio which feels unwieldy and slow especially compared to VSCode.

Overall plan:

1. download R
    - for debian: Johannes Ranke. For Linux/Debian installation, crucial to update the *source.list* to include sources that have more recent versions of R. If not, will get very old R versions that is not compatible with many packages.
    - add R to path for Windows. In Windows Path, add for example: *C:/Program Files/R/R-3.6.2/bin/x64/* and *C:/Rtools/bin*
2. download R-studio
3. Open R-studio and auto-detect R
4. Install additional packages (from commandline)
    - various R data-science packages have dependencies, and additional not-R programs might need to be installed. Install them as needed, for example: *libcurl4-openssl-dev* and *libssl-dev* from command prompt first inside Debian.

3.2.1 Linux R Install

For linux/Debian, to Install latest R:

```
# Go to get latesdebian latest r sources.list
cat /etc/apt/sources.list
# Install this First (should already be installed)
sudo apt install dirmngr

# Debian R is maintained by Johannes Ranke, copied from https://cran.r-project.org/bin/linux/debian/:
apt-key adv --keyserver keys.gnupg.net --recv-key 'E19F5F87128899B192B1A2C2AD5F960A256A04AF'
# Add to source.list, for debian stretch (9)
# sudo su added for security issue as super-user
sudo su -c "sudo echo 'deb http://cloud.r-project.org/bin/linux/debian stretch-cran35/' >> /etc/apt/sou
# if added wrong lines, delete 3rd line
sudo sed '3d' /etc/apt/sources.list

# Update and Install R, should say updated from cloud.r
sudo apt-get update
sudo apt-get install r-base r-base-dev

# Also install these, otherwise r-packages do not install
# libxml2 seems need for tidymodels
sudo apt-get install libcurl4-openssl-dev
sudo apt-get install libssl-dev
sudo apt-get install libxml2-dev
```

3.2.2 Install Packages for R

From command prompt, enter R by typeing *R*, and install packages:

```
# Can enter R from Command Prompt Conda ENV, much faster than from r-studio
# Install R-tools, RTOOLS is for WINDOWS
install.packages("installr")
library("installr")
install.Rtools()
# c:/Rtools/bin;
# c:/Rtools/mingw_32/bin;
# c:/Rtools/mingw_64/bin;

# main packagevps
# tidymodel installation on linux difficult
install.packages(c("tidyverse", "tidymodels", "tidyr"))
# development packages
install.packages(c("devtools", "IRkernel", "pkgdown", "roxygen2", "kableExtra"))
# other packages
install.packages(c("AER", "minpack.lm", "knitr", "matlab"))

# Install my package, need to prepare package to work for ubuntu
# if install does not work, load locally
devtools::install_github("fanwangecon/R4Econ")

# Kernel
install.packages(c('repr', 'IRdisplay', 'evaluate', 'crayon', 'pbdZMQ', 'devtools', 'uuid', 'digest'))
IRkernel::installspec()
```

3.3 Install R (inside Conda)

Key packages are generally available in conda's default channel (official distribution) and also the more frequently updated conda-forge channel. Prioritize conda-forge to get the latest packages. After adding *conda-forge* to channels, that will be prioritized, so when creating a new environment, conda will install first from conda-forge if package exists there. Try the *r_env* generation line with and without first adding *conda-forge* to channel, and see that the packages installed will have different versions, reflecting versions in the default channel and in the *conda-forge* channels.

3.3.1 Set up Conda R environment

Create a Conda Environment called *r_env*, and update its channels.

- **Location**: Installed environments are shown appear in the *envs* folder of *C:/ProgramData/Anaconda3/envs/*. Can be easily deleted without disrupting the main base installation for conda. This is very important. During installation, could easily run into problems, and need to re-install. Much better to only re-install a folder. Note that when R is installed in *envs*, it will not show up under add or remove programs for uninstallation from there, has to be uninstalled, deleted directly here.
- **Use only in Env**: Note that if we installed r inside r_env, if we type *r* under the base environment, we can not enter r. We can only enter r within r_env. With PATH properly set-up, this happens under Conda Prompt, Windows Prompt, etc.
- **Multiple Pythons**: With the above installation for R, one benefit is that if R requires a different version of Python than what the base environment uses, the r_env could have a different python version. So type *python* in the base environment as well as inside the *r_env*. This also means potentially there could be duplicated installations I think. Look at the python versions under *conda list* in the base and in the r-environment.
- **File Size**: Note that this creates a large installation folder, *C:/ProgramData/Anaconda3/envs/r_env/*, without additional packages, is 1.3 GB in size

```
# outside of conda, install some depencies
sudo apt-get install libcurl4-openssl-dev
sudo apt-get install libssl-dev
sudo apt-get install libxml2-dev

# install key r packages within a r-environment
conda env remove -n r_env
# start empty environment
conda create -n r_env
# activate env and check, nothing yet
conda activate r_env
conda list

# check channel and add env specific channel
conda config --get channels
# to get latest r, use conda-forge, which has more recent Rs
# conda config --env --add channels conda-forge
# channels for base
conda config --get channels
# channels for cur env
conda config --get channels --env

# see all installed environments
conda env list
# activate an environment to use it, if in base, r does not exist, isolated in r_env
conda activate r_env
# see packages in environment
conda list
# to quit
conda deactivate
```

3.3.2 Install R and Some Packages in Conda

Either inside a R specific environment, or outside of it, now do additional installations for R and also R packages.

During installation, the R programs might require downgrading other programs in the Conda environment that Python for example also uses. That is why potentially it is safer to install in R environment. However, when installing in R enviornment, certain packages could have issues if installer does not realize it is in an environment with special folder structure, so installation could fail.

```
# install files
conda install -n r_env -c r r r-essentials r-tidyverse r-tidymodels -y

# trouble with igraph installation, install here from conda
# conda install -n r_env r-essentials r-base r-tidyr r-tidyverse r-devtools r-irkernel r-pkgdown r-roxy
conda install -n r_env -c r r-igraph
conda install -n r_env -c conda-forge igraph
conda install -n r_env -c r r-rstantools r-rstan r-rstanarm -y
```

3.4 Install Additional Packages inside the R

These are packages that I use, these should be installed inside the environment. Or they could be installed later from inside r-studio. Installing inside r-studio is a lot better. *tidymodel* has installation issues sometimes.

In *r_env*, type in *R*, and then install from inside terminal's R. For conda env, devtools will be installed and will install packages inside *envs/r_env/Lib/R*.

```
# enter into env
# conda activate r_env
# enter R or type in rstudio
# sicne R is only install inside r_env, has to enter rstudio from r_env
# R
# rstudio

# main packagevps
# tidymodel installation on linux difficult
```

```
install.packages(c("rstantools", "rstan", "rstanarm"))
install.packages(c("tidyr", "tidyverse", "tidymodels"))

# tidymodels
# development packages
install.packages(c("devtools", "IRkernel", "pkgdown", "roxygen2", "kableExtra"))
# other packages
install.packages(c("AER", "minpack.lm", "knitr", "matlab"))

# Install my package, need to prepare package to work for ubuntu
# if install does not work, load locally
devtools::install_github("fanwangecon/R4Econ")

# Kernel
install.packages(c('repr', 'IRdisplay', 'evaluate', 'crayon', 'pbdZMQ', 'devtools', 'uuid', 'digest'))
IRkernel::installspec()
```

3.5 R-Studio Set-up Download from Rstudio

1. Download R-studio, and install, as normal (not from conda, directly from rstudio website latest windows version)
2. Open up anaconda prompt, enter *r_env* environment: *activate r_env*
3. cd into rstudio exe file folder: cd "C:/Program Files/RStudio/bin"
4. start r-studio from inside *r_env*: rstudio.exe
5. inside r-studio, check: *.libPaths()*
    - "C:/ProgramData/Anaconda3/envs/r_env/Lib/R/library"

```
activate r_env
cd "C:/Program Files/RStudio/bin"
rstudio.exe

cd "C:/ProgramData/Anaconda3/envs/r_env/Lib/R/library"
"C:/Program Files/RStudio/bin/rstudio.exe"
```

For linux, note that rstudio offers Ubuntu as well as Debian versions that can be installed with wget:

```
<!-- for ubuntu -->
wget "https://download1.rstudio.org/desktop/bionic/amd64/rstudio-1.2.5033-amd64.deb"
<!-- for debian -->
sudo apt install ./rstudio-1.2.5033-amd64.deb
wget "https://download1.rstudio.org/desktop/debian9/x86_64/rstudio-1.2.5033-amd64.deb"
sudo apt install ./rstudio-1.2.5033-amd64.deb
conda install -c r rstudio
cd "C:/Program Files/RStudio/bin"
rstudio.exe

cd "C:/ProgramData/Anaconda3/envs/r_env/Lib/R/library"
"C:/Program Files/RStudio/bin/rstudio.exe"
```

Is RSTUDIO pointing to the R installation you want? Note you can open Rstudio from differen conda env command line, that will use different R installations. Or also set:

- *export RSTUDIO_WHICH_R=/usr/bin/R*
- *export RSTUDIO_WHICH_R=/home/wangfanbsg75/anaconda3/bin/R*

3.6 R Tests

Test the following file to see if we can execute a R file. Do it inside *r_env* and inside a *r* session.

```r
# A simple file with summary statistics using tidyverse
source('C:/Users/fan/R4Econ/summarize/dist/fst_hist_onevar.R')
# Another simple file with summary statistics using tidyverse
source('C:/Users/fan/R4Econ/support/tibble/fs_tib_basics.R')
# A file involving estimation
source('C:/Users/fan/R4Econ/optimization/cesloglin/fst_ces_plan_linlog.R')

# C:/Users/fan/R4Econ/summarize/dist/fst_hist_onevar.Rmd
# C:/Users/fan/R4Econ/support/tibble/fs_tib_basics.Rmd
# C:/Users/fan/R4Econ/optimization/cesloglin/fst_ces_plan_linlog.Rmd
```

4 Install Latex and PDF

Use a combination of overleaf and local compile. The goal is to have the same installation set-up for windows as well as linux. And hopefully, locally compilable file also can be compiled on overleaf (only use packages also available on overleaf).

There are several pieces of things to install:

1. latex compiler: pdflatex, etc
2. tex distributions: MikTex, texlive
   - these folders, once various packages are installed, could be very large, many GB
3. latex packages from distributions
4. editor (gui): Atom, Sublime text, texStudio, etc

4.1 Uninstall

- uninstall *MiKTeX*: delete from programs, then delete remaining folder in Program Files.
- uninstall *texlive*: go to *c:/texlive*, delete the folder

4.2 Install

Install on Windows

- Install Texstudio
- Install texlive

Install on Ubuntu

install texlive and texstudio ubuntu/debian.

```
sudo apt-get update
sudo apt-get install texlive-full
sudo apt-get install texstudio
```

Test file compilations in TexStudio.

4.3 SumatraPDF

For windows, install SumatraPDF. Can easily modify colors.

Settings

Kindle Sepia modifications:

- Background color:
  - Kindle Sepia: *fbf0d9*
  - Kindle Sepia Light: *fff8e6*
- Text color:
  - Kindle Sepia: *5f4b32*
  - Kindle Sepia Darker: *503f2a*
  - Kindle Sepia Darkerer: *1e170f*

```
FixedPageUI [
    TextColor = #1e170f
    BackgroundColor = #fff8e6
    SelectionColor = #f5fc0c
    WindowMargin = 2 4 2 4
    PageSpacing = 4 4
]
```

4.4 Okular PDF

Linux: *sudo apt-get install okular* Windows: install okular from windows app store

4.4.1 Okular Shortcuts and Features

Settings:

- Background color:
  - background behind pdf: configure okular, general, use custom backgroun color, Gainsboro gray: *DCDCDC*, kindle sepia: *FBF0D9*
  - pdf background itself: configure okular, accessbility, change color, change paper color, Light Kindle Sepia: *fff8e6*

Features:

- Okular 3 pages per row view: view, overview
- Automatic Update when PDF updated elsewhere: this is one of the key reasons to use okular, when I knit a Rmd file to PDF, the currently open Okular PDF file automatically updates and does not give file is open error like Acrobat.
- Background color:
  - background behind pdf: configure okular, general, use custom backgroun color, Gainsboro gray: *DCDCDC*, kindle sepia: *FBF0D9*
  - pdf background itself: configure okular, accessbility, change color, change paper color, Light Kindle Sepia: *fff8e6*

Shortcuts:

- Okular configure shortcuts: setting, configure shortcuts
- Okular if menu bar lost: *Ctrl + m*
- Okular comment pane: *F6*
- Okular close left pane: *F7*
- Okular full scrren: *Ctrl + Shift + F*
  - unlike acrobat, pdf allows for multiple scrrens

4.5 Adobe PDF

PDF Adobe Acrobat Installation + go to adobe website, log in using fwang26@uh.edu + go to my account, choose view and download my apps, choose acrobat & PDF, download Acrobat DC + UH account can only be activated on two accounts at once, so need to kick other computers out temporarily potentially

5 Install Various Editors

Atom, VSCode, Notepad++, Sublime, etc. These are essential editors that allows for pleasant programming and writing experiences. See here for fan instructions on vim installations.

All settings here JSON files here:

- windows: *C:/Users/fan/AppData/Roaming/Code/User*, can be modified here directly

5.1 VSCode Installation

VSCode is much more light-weight than Atom, much faster. On Windows, download here. VScode settings easy to change for various extensions. Very very good Latex Editor, probably the best I have used.

For Linux:

```
# Debian Install
curl https://packages.microsoft.com/keys/microsoft.asc | gpg --dearmor > microsoft.gpg
sudo install -o root -g root -m 644 microsoft.gpg /usr/share/keyrings/microsoft-archive-keyring.gpg
sudo sh -c 'echo "deb [arch=amd64 signed-by=/usr/share/keyrings/microsoft-archive-keyring.gpg] https://
sudo apt-get install apt-transport-https
sudo apt-get update
sudo apt-get install code # or code-insiders

# Change Font (https://wwFw.reddit.com/r/Crostini/comments/cesxr0/vscode_ui_and_fonts_too_small_heres_h
# modify: /usr/share/applications/code.desktop
nvim /usr/share/applications/code.desktop
#replace this line: Exec=/usr/share/code/code --unity-launch %F
Exec=sommelier -X --scale=0.8 --dpi=160 /usr/share/code/code "--unity-launch %F"
#replace this line: Exec=/usr/share/code/code --new-window %F
Exec=sommelier -X --scale=0.8 --dpi=160 /usr/share/code/code "--new-window %F"

# Open File
# low density mode to see cleary
code
```

5.1.1 VSCode Extensions

Press *Ctrl + P*, paste the *ext install* commands below to install extensions. Extensions are in: *%USERPROFILE%/.vscode/extensions* for windows. Can change code for extension there.

```
# [Microsoft Python Extension](https://marketplace.visualstudio.com/items?itemName=ms-python.python)
ext install ms-python.python
# [Latex Workshop](https://marketplace.visualstudio.com/items?itemName=James-Yu.latex-workshop)
ext install latex-workshop
# [R](https://marketplace.visualstudio.com/items?itemName=Ikuyadeu.r)
ext install ikuyadeu.r
```

5.1.2 VSCode Shortcuts and Features

Features:

1. open a file, if have workspace where the file is in, automatically expand tree and show where file is on left pane.

Shortcuts :

- Zen model: *Ctrl + k, z*, this means press control k, then press z
- Open up Settings: *Ctrl + ,*

- Preview Markdown in VSCode: *Ctrl + Shift + V*
- Open up left pane: *Ctrl + B*
- Wrap Text: *alt + Z*, switch between wrap or not to save screen space for example.

5.1.3 VSCode and R

VSCode works pretty well it seems with Rmd files. Rmd files are useful to work through algorithm, examples in preparation for writing usable functions. Rstudio has been often very slow for me, and I generally feel like can not stretch around and work in RStudio IDE.

In Atom, Hydrogen works, but really only works with very basic line by line testing.

Using VSCode with RMD files seems much quicker, and there is room to breath.

1. Install R following R installation instructions.
2. Then install extension R. Microsoft stopped supporting rmd in VSCode: R Tools for Visual Studio
3. properly set path (might have to call vscode from r env)
   - *r.rterm.linux*: */home/wangfanbsg75/anaconda3/envs/r_env/bin/R*
   - *r.rterm.windows*: *C:/Program Files/R/R-3.6.2/bin/x64/R.exe*
     - note *R.exe* at the end
4. Run:
   - ctrl + enter to activate R terminal
   - VSCode 2019 allows for Knit Rmd: *Ctrl + Shift + K*
   - also can run code segment: select lines and *ctrl + enter*

5.1.4 VSCode and Latex

Download Latex Workshop.

Settings

change background border pdf color (not pdf but border): *ctrl + shift + x*, LatexWorkshop, Settings, Configure Extension Settings, Search for background, change color name. or enter: *latex-workshop.view.pdf.backgroundColor*. use gainsboro gray: *#DCDCDC*, or lightgray *#D3D3D3*.

To take advantage of Okular (where background color can be none-white for example). Go to settings, and: 1. Change: *latex-workshop.view.pdf.viewer* to external 2. Change: *latex-workshop.view.pdf.external.viewer.command* to *C:/Program Files/SumatraPDF/SumatraPDF.exe* open up *C:/Users/fan/AppData/Roaming/Code/User/settings.json*, and copy these in:

```
"latex-workshop.view.pdf.external.synctex.command": "C:/Program Files/SumatraPDF/SumatraPDF.exe",
"latex-workshop.view.pdf.external.synctex.args": [
  "-forward-search",
  "%TEX%",
  "%LINE%",
  "-reuse-instance",
  "-inverse-search",
  "code \"C:\\Users\\fan\\AppData\\Local\\Programs\\Microsoft VS Code\\resources\\app\\out\\cli.js\" -r
  "%PDF%",
]
```

Features:

1. upon savings, rebuid file and update pdf (similar to overleaf)
2. hover offers equation preview

Shortcuts:

1. preview: *ctrl + alt + v*
2. synctex location: *ctrl + alt + j*

5.1.5 VSCode settings.json

The settings.json file, *%APPDATA%/Code/User/settings.json*:

```
{
  "r.rterm.windows": "C:/Program Files/R/R-3.6.2/bin/x64/R.exe",
  "window.zoomLevel": 0,
  "latex-workshop.view.pdf.viewer": "external",
  "latex-workshop.view.pdf.external.synctex.command": "C:/Program Files/SumatraPDF/SumatraPDF.exe",
  "latex-workshop.view.pdf.external.synctex.args": [
        "-forward-search",
        "%TEX%",
        "%LINE%",
        "-reuse-instance",
        "-inverse-search",
        "code \"C:\\Users\\fan\\AppData\\Local\\Programs\\Microsoft VS Code\\resources\\app\\out\\cli.j
        "%PDF%",
    ],
    "latex-workshop.view.pdf.external.viewer.command": "C:/Program Files/SumatraPDF/SumatraPDF.exe"
}
```

5.2 Atom Installation

Atom is slow but has clean look and nice git/github integration.

5.2.1 Atom Extensions

Terminal in windows or linux after installation: *apm install hydrogen.* apm is the atom package manager:

```
# Once atom is installed, can use apm to install packages
apm install hydrogen
apm install project-manager
apm install sublime-style-column-selection
apm install minimap

# vim distraction free editing
apm install vim-mode-plus
apm install zen

# Inside Conda open up atom
atom
```

5.2.2 Atom Shortcuts and Features

Features:

```
# open multiple projects from atom at the same time, after repos synced
atom ~/fanwangecon.github.io ~/Pyfan ~/Teaching ~/Tex4Econ ~/R4Econ ~/M4Econ ~/Py4Econ
# open up any py from from within, try 1+1, does it work?
# also use script, try ctrl + shift + b, run whole file (not hydrogen)
```

Shortcuts :

- Git Pane: *Ctrl + 9*, *Ctrl + 8*

5.2.3 Atom and Python

Having installed Anaconda, now install Atom and Hydrogen to Test with Python. For both Jupyter as well as Atom, always open from command prompt, open from *Anaconda Prompt* with admin rights.

Open up *Jupyter Notebook*, does the python kernel work? If does not, uninstall and re-install Anaconda.

```
# start jupyter lab, from prompt
jupyter lab
```

Try *1+1*.

5.2.4 Atom and Latex

Install the following two packages in atom:

1. latex package on atom.
2. language-latex package on atom

In Atom Latex setting, note that on different platforms, paths to texlive are different:

- Path:
  - Windows: *C:/texlive/2019/bin/win32*
  - Chromebook Linux: */usr/bin/latex*
- Which PDF viewer to use:
  - download Okular and use Okular

```
# set TeX path
# C:/texlive/2019/bin/win32

# Latex, after installation
# see where texlive appears under root folder directories
sudo find / -name "texlive"
# see latex version, if texlive is used
latex -v
# see main exe directory
which latex
# /usr/bin/latex
```

5.3 Other Editors

5.3.1 Notepad++

Notepad++. Many old projects have xml folder structures for Notepad++.

- Silently update file changes made in other editors: Settings, Preferences, MISC., File Status Auto Detection, Update Silently.

5.3.2 Sublime

Sublime

there are some issues with atom. When a file is edited inside Vim, when changes are saved, a temp file is created in atom. Sublime would, however, still show the file with the changes. This makes dual screen editing very difficult. Sublime text is used for this situation.

```
# Sublime Linux
wget -qO - https://download.sublimetext.com/sublimehq-pub.gpg | sudo apt-key add -
sudo apt-get install apt-transport-https
echo "deb https://download.sublimetext.com/ apt/stable/" | sudo tee /etc/apt/sources.list.d/sublime-tex
```

```
sudo apt-get update
sudo apt-get install sublime-text
# launch from terminal
subl
```

Sublime Setup:

- r and rmd setup

5.3.3 PyCharm

pycharmjetbrains.com/pycharm/)

6 Other Programs to Install

6.1 Package Managers

To install packages in windows more easily:

- scoop

```
# in powershell
# enter cmd
# type powershell in cmd to enter powershell from cmd
powershell
# permission
set-executionpolicy remotesigned -scope currentuser
# install
iwr -useb get.scoop.sh | iex
```

6.2 Utilities

- 7-zip
- evernote

6.3 Security

- keepass
- Proton VPN: unlimited usage free version available.
- Express VPN