New Computer Data-Science Set-up: Python (anaconda, pycharm), R (r-studio) and other Programs Installations

Go back to fan's Tex4Econ and Miscellaneous Repository.

1 Objective

For the following types of programming tasks, install relevant programs, that are cross platform compatible. Test at the end that example programs work.

1. Python
   - Anaconda Dist
   - Pycharm
   - atom hydrogen
2. R
   - R
   - Rstudio
   - atom hydrogen
3. Matlab
   - Matlab
4. Latex and PDF
   - atom latex
   - texstudio
   - texlive
   - okular pdf

Test at the end:

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

2 Conda

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 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 Debian, follow these instructions.

See Anaconda Installation Directory, in 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.

```
# # 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 Conda Test Python

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
```

2.4 Jupyter and Atom Hydrogen

For both Jupyter as well as Atom, always open from command prompt, open from *Anaconda Prompt* with admin rights.

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

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

Now test python within atom:

1. install hydrogen in atom. Follow instructions.
    - terminal in windows or linux after installation: *apm install hydrogen*
    - apm is the atom package manager.
2. go to the programs above and try to run.

```
# Once atom is installed, can use apm to install packages
apm install hydrogen
# Inside Conda open up atom
atom
# 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)
```

3 R Installation

Try to install R from within Conda. The procedure here is the same for windows as well as linux machines.

3.1 Fully Uninstall r

Open up RStudio, and see where paths are. also open up R from command prompt, see where paths are.

```
.libPath()
```

Open up install and uninstall programs, find R and Rstudio, and uninstall them. Then delete all files found path paths shown by *.libPath()*, looking at these: - *C:/Users/fan/Documents/R/win-library/3.6* - *C:/Program Files/R/R-3.6.1/library*

For Linux:

- if used apt-get to install, R installed under main user drive, deactivate conda, and check which 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)

1. download R
   - for debian: Johannes Ranke
2. download R-studio
3. Open R-studio and auto-detect R
4. Install packages

note that installation will not work if the machine's version of R is not up to date, this could be the case with default R versions for some linux installations. For Debian for example, need to update the *source.list* to include sources that have more recent versions of R.

Additionally, need to install *libcurl4-openssl-dev* and *libssl-dev* from command prompt first inside Debian.

```
# If inside Debian, due to some tidymodel install issues, do this`
conda install -c r r
conda install -c r r-essentials
conda install -c r r-igraph
conda install -c conda-forge igraph

# 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
```

Which R?

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

```
# 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 Set up R-Environment in Conda

Conda environments are language agnostic. pip is a package manager for Python. venv is an environment manager for Python. conda is both a package and environment manager and is language agnostic. Inside Anaconda Prompt:

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.

```
# 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

# 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

# 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
```

- **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

3.4 Install Additional Packages inside the R-environment

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:

```
# 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()
```

Install additional packages from inside R. The devtools install will install packages inside *envs/r_env/Lib/R*.

```
#######################################
### Own Package
#######################################
devtools::install_github("fanwangecon/R4Econ")
# install_dev("cli")
```

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"
```

3.6 Install Packages from R-Studio

Install addition files inside Rstudio. Easier to debug potentially. Install rtools as described here: rtools download. Alternatively:

```
install.packages("installr")
library("installr")
# Note that this will generate several pop-up windows
install.Rtools()
# choose to only install 64 bit toolchain
# this installs g++ which is needed for rstanarm
# Add these below to windows path
# c:/Rtools/bin;
# c:/Rtools/mingw_64/bin;

if (!require(devtools)) {
  install.packages("devtools")
  library(devtools)
}
library(devtools)
install_github("stan-dev/rstanarm", args = "--preclean")

setwd('C:/ProgramData/Anaconda3/envs/r_env/Lib/R/library')

install.packages("stan-dev/rstanarm", dependencies=TRUE, INSTALL_opts = c('--no-lock'))

# other packages
install.packages(c("tidymodels", "AER", "minpack.lm", "knitr", "kableExtra", "matlab"))

# for r-kernel to work in Atom
install.packages(c('repr', 'IRdisplay', 'evaluate', 'crayon', 'pbdZMQ', 'devtools', 'uuid', 'digest'))
devtools::install_github('IRkernel/IRkernel')
IRkernel::installspec()

# Install my package
```

```
devtools::install_github("fanwangecon/R4Econ")
```

3.6.1 Test Files

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

```
# 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

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, 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
```

4.3 Compile in Atom and Texstudio

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
```

Go to Tex4Econ and compile various template files and other types of files there to see if tex is working.

Compile the same files in texstudio.

5 Other Programs to Install

5.1 Development Programs

- pycharm

5.2 Key Programs

PDF

- PDF Okular
  - Linux: *sudo apt-get install okular*
  - Windows: install okular from windows app store
- PDF Adobe Acrobat
  - 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

VPN

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

5.3 Utilities

- 7-zip
- evernote
- keepass