New Windows 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 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
```

1.1 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

See Anaconda Installation Directory, in anaconda prompt:

```
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

1.2 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 --add channels conda-forge
# normal update
conda update --all
```

1.3 Conda Test Python

```
# 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
```

1.4 Additional Packages

```
conda install statsmodels datashape seaborn
# conda install statsmodels
# conda install datashape
# conda install seaborn

conda install -c conda-forge interpolation awscli
# conda install -c conda-forge interpolation
# conda install -c conda-forge awscli

conda install -c anaconda boto3

conda install -c r r-irkernel
```

2 R Installation

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

2.1 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*

2.2 Install R (outside of Conda)

1. download R
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.

```
# Install R-tools
install.packages("installr")
library("installr")
install.Rtools()
# c:\Rtools\bin;
# c:\Rtools\mingw_32\bin;
# c:\Rtools\mingw_64\bin;

# main packagevps
install.packages(c("tidyr", "tidyverse", "tidymodels"))
# development packages
install.packages(c("devtools", "irkernel", "pkgdown", "roxygen2"))
# other packages
install.packages(c("AER", "minpack.lm", "knitr", "kableExtra", "matlab"))

# Install my package
devtools::install_github("fanwangecon/R4Econ")
```

2.3 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.

```
# Set up Channel
conda config --add channels conda-forge
# install key r packages within a r-environment
conda env remove -n r_env
conda create -n r_env r-essentials r-base r-tidyr r-tidyverse r-devtools r-irkernel r-pkgdown r-roxygen
# 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

2.4 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.

This did not work in windows, *tidymodels* failed to install. Perhaps because of directory issues. *tidymodels* required *rstanarm*, and *rstanarm* failed to install.

```
### One line
conda install -c conda-forge r-aer r-minpack.lm r-knitr r-kableExtra r-matlab
# conda install -c conda-forge r-rstanarm

########################################
### Estimation
########################################
# Estimation Tools
# tidymodels does not work fron conda install
# Linear Regression Package
# https://anaconda.org/conda-forge/r-aer
# Nonlinear Regression Package
# https://anaconda.org/conda-forge/r-minpack.lm
conda install -c conda-forge r-aer r-minpack.lm
# enter r and use devtools
# install.packages("tidymodels")

########################################
### Visualiation
########################################
conda install -c conda-forge r-knitr r-kableExtra

########################################
### Utilities
########################################
# Allowing for Invoking Matlab Files from Inside R
conda install -c conda-forge r-matlab
```

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")
```

2.5 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"
sudo apt install ./rstudio-1.2.5033-amd64.deb
<!-- for debian -->
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"
```

2.6 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"))

# Install my package
devtools::install_github("fanwangecon/R4Econ")
```

2.6.1 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
```

3 Other Programs to Install

3.1 Development Programs

- pycharm

3.2 Key Programs

- 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

3.3 Utilities

- 7-zip