Lecture 0
The Ecosystem

Econ 2713A: Computation

University of Pittsburgh, 2018

Purpose

We aim to compute things because they are useful (not because they're difficult or interesting)

- but interesting isn't a bad thing!

In the end we can make predictions using identified models

- positive: what's going to happen, or models predict what has happened
- normative: e.g. what's the best policy to adopt in a given setting

Writing Code

A good **editor** is important! You should probably use Atom or Sublime Text for serious coding.

Even more important is **version control**. Get a GitHub account today. I'll go over the basics of how to set up a repository.

- like Dropbox, but you explicitly tell it when to sync
- can revert changes and look at full history

If you can, run Linux or OSX If you can. But even Windows run Linux now

Language

Most of what we do will by in Python. There is almost no need for MATLAB these days

- Total time = writing code + debugging code + running code

Important words of wisdom:

> *Premature optimization is the root of all evil.*
> *— Donald Knuth*

Python

Python is scripted (as opposed to compiled) and weakly typed (as opposed to strongly typed). Also referred to as duck typing

If runtime performance becomes an issues, you can

- eliminate loops by vectorizing
- rewrite core functions in Cython (compiled)
- use Numba to automatically speed up core functions
- parallelize across CPU cores or with a cluster

MATLAB and Julia are just-in-time compiled alternatives, but are usually not worth the trouble

Distribution

Easiest way to install Python + packages across platforms is using Anaconda (https://www.anaconda.com/download)

Those on Linux and OSX can also install natively and use `pip` to install packages

On top of the core language there are many (~150K) packages that implement new functionality

- prevents excessive code reuse
- ensure you're using latest algorithms (properly implemented)

Python Packages

The numerical packages that you can't do without

- `numpy`: data arrays and linear algebra
- `scipy`: optimization, interpolation, distributions
- `pandas`: high level data processing and aggregation

More specialized packages we'll be using

- `statsmodels`: regression and estimation techniques
- `sklearn`: light machine learning and data processing
- `tensorflow`: heavy machine learning + GPU

Interactivity

Usually you'll want to develop things interactively. For this we the have `IPython` command prompt

For a more seamless visual experience, there is also the acclaimed `jupyter` notebook. This is a web based interactive suite

Plotting and visualization is usually handled through `matplotlib`

- additional fanciness can be attained using `seaborn`
- can do 3D and interactive plots with some effort

Jupyter

In simplest setting, Jupyter notebook runs locally on your machine and exposes Python interpreter with web interface

Because of this, notebooks can run on remote server and be accessed on internet

- Microsoft Azure hosted notebooks (https://notebooks.azure.com/)
- Google Colaboratory (https://research.google.com/colaboratory/)