Python for Machine Learning and Data Analysis

Handan Liu, PhD

Northeastern University

Spring 2019

Outline

- Brief introduction
- Introduction to the course
- Syllabus
- Introduction to Python Infrastructure and Development Tools
  - Python
  - Anaconda
  - Jupyter
- Lab 1: install anaconda in your computer

Brief Introduction

Who am I?

- Dr. Liu, Handan
  - Professional expert on High Performance Computing (HPC) and machine learning on multithreading, multiprocessing and co-processing systems for many years.
  - Working at Research Computing @ Northeastern as a core leading on Discovery HPC Cluster.
  - h.liu@northeastern.edu

Discovery Cluster @ Research Computing

- The Research Computing group @NEU provides high-end research computing resources to Faculty, Students, Staff, Researchers, Visiting Scholars, and Partners at Northeastern University. These resources include high performance computing (HPC) clusters, storage, visualization tools, and software as well as high-level technical and scientific user support, consulting, education, and training.
- Supercomputer vs. Cluster and
- High Performance Computing vs. Parallel Computing

What the class is about?

- The main idea
  - This course we will cover is broadly applicable, and has led to significant advances in many fields. Once you understand the basics of machine learning technology, and the close connection between theory and practice, it's a very open field, where lots of progress can be made quickly.
- More important, you will transform your theoretical knowledge into practical skills using many hands-on labs in this course:
  - Especially, create accounts for the students in this course on HPC Cluster in the middle of semester; and
  - Chances to practice machine learning and data analysis on HPC cluster for CPU and GPU parallel computing.
- Prerequisites:
  - You should be comfortable programming in Python.
  - Basic knowledge of Unix/Linux will be helpful (CentOS 7.5 on Discovery)

Syllabus: Schedule

- Spring 2019: January 7 – April 26
- Time: Thursday 6:05pm – 9:35pm
- Location: Behrakis Health Sciences Center 220

Note: This schedule and contents will be adjusted as needed throughout the semester.

Course Schedule

Note: This schedule and contents will be adjusted as needed throughout the semester.

Week	Topics	Assignments
1 Jan 10	Introduction *Lab1:* Install Anaconda	Complete the installation of anaconda in your computer
2 Jan 17	Python Language Essentials: Introduction to Python's core language features and packages *Lab 2*: samples - writing python code and run it	
3 Jan 24	Numerical Analysis and Data Exploration with **NumPy** Arrays *Lab 3*: run samples	
4 Jan 31	Data Visualization with Matplotlib *Lab 4*: run samples	HW1
5 Feb 7	Pandas, the Python Data Analysis Library is a powerful package for working with tabular data for data aggregation and reorganization *Lab 5*: run samples	
6 Feb 14	***Lab 6***: Hands-on Lab: • Setup computers to be able to access Discovery Cluster • Learn how to submit and manage jobs on the HPC cluster • Write our first program and run on HPC cluster	

Week	Topics	Assignments
7 Feb 21	Parallel programming in Python: Python multiple threads: **multiprocessing** Python MPI: **mpi4py** *Lab 7*: run python parallel code on HPC cluster	HW2
8 Feb 28	Introduction to Machine Learning covers the frameworks and tools provided by **scikit-learn**, a widely used library for machine learning *Lab 8*: run samples	
9 Mar 14		
10 Mar 21	Machine Learning: Numeric Data Machine Learning: Categorical Data Machine Learning: Image Data *Lab 9*: run samples	HW3
11 Mar 28		Project
12 Apr 4	Brief Introduction to the Python Deep Learning Library **TensorFlow** and **Keras**; *Lab 10*: run python code of Deep Learning on HPC cluster for parallel on CPU and GPU	
13 Apr 11	Final exam; debrief	
14 -15 Apr 18, 25	Presentation of the project by every student	

Grading

- Homework: 40% (HW1 10%, HW2 15%, HW3 15%)
- Attendance: 10%
- Project: 30%
- Final exam: 20%

Assignments

- Lectures are complemented by homeworks (programming assignments) to bridge the theory with the practice. The homeworks are associated with the three main parts of the course that mostly consist of programming assignments to exercise a technology or programming model.
- Homework is due electronically. Assignments have a specific due date and time. Submissions will be accepted up to one day after the deadline with a 50% penalty. For example, an on-time submission might receive a grade of 80 points. The same assignment submitted after the deadline would receive 40 points (80 x 0.5).
- Submission will be done through **Blackboard**.

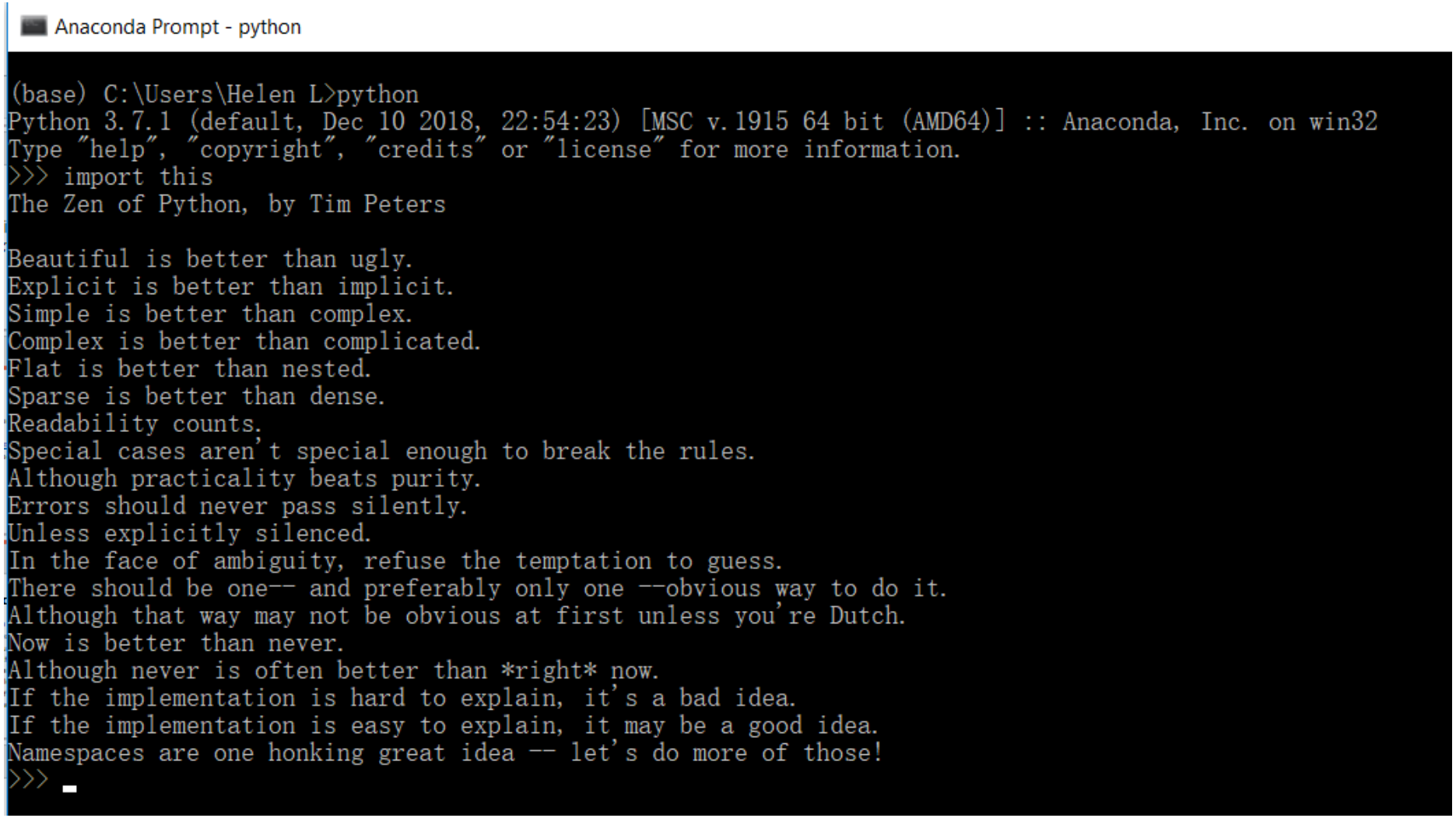
Make-up Policy

- Students who miss the presentation and the final exam will not, as a matter of course, be able to make up it. If there is a legitimate reason why a student will not be able to complete an assignment on time or not be present for the presentation or the exam, then they should contact the instructor beforehand. Under extreme circumstances, as decided on a case-by-case basis by the instructor, students may be allowed to make up assignments or exam without first informing the instructor.

Introduction to Python Infrastructure and Development Tools

Python

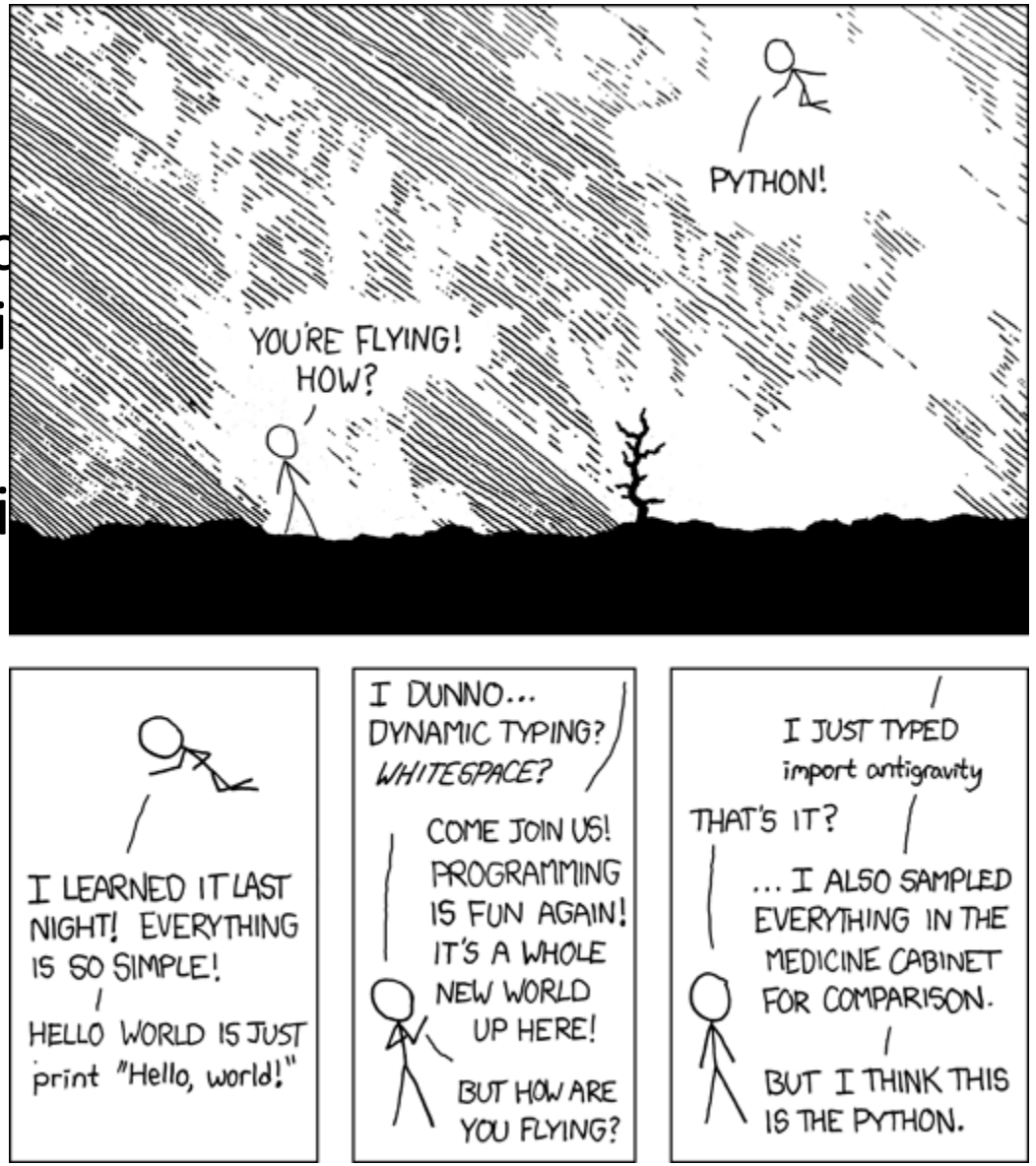
- Python is an interpreted, high-level, general-purpose programming language. -- wiki
- https://www.python.org/
- Why is Python so popular for Machine Learning and Data Analysis?
  - it’s easy to understand and learn.


Anaconda Prompt - python
(base) C:\Users\Helen L>python
Python 3.7.1 (default, Dec 10 2018, 22:54:23) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import this
The Zen of Python, by Tim Peters
Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>>

Packages, Packages everywhere!

- Want to work with images — numpy, opencv, scikit-learn
- Want to work in text — nltk, numpy, scikit-learn
- Want to work in audio — librosa
- Want to solve machine learning problem — pandas, scikit-learn
- Want to see the data clearly — matplotlib, seaborn, scikit-learn
- Want to use deep learning — tensorflow, pytorch, keras
- Want to do scientific computing — scipy
- Want to integrate web applications — Django
- …… and so on

- The best thing about using these pac
  learning curve. Once you have a basi
  can just implement it.
- Free to use under GNU license. Just i



What about others?

- Excel and SAS: data scientist
  - Can’t handle large datasets and less community support
  - Not free
- MATLAB:
  - Great packages for image analysis
  - Very slow. It can’t be used in deployment, but only for prototyping.
  - Not free
- R:
  - Open source, free and made for statistical analysis
  - A learning curve

Cons of Python

- The main reason or the only reason why Python will never be used very widely is because of the overhead it brings in.
- Small processors or low memory hardware won't accommodate Python codebase today, but for such cases we can have C and C++ as the development tools.
- And, when implementing an algorithm (e.g. Neural Network) for a particular task, we use python (tensorflow) on HPC cluster with parallel implementation.

Anaconda

The Most Popular Python Data Science Platform

- What is Anaconda?
  - https://www.anaconda.com/what-is-anaconda/
  - Open source Anaconda Distribution is the fastest and easiest way to do Python and R data science and machine learning on Linux, Windows, and Mac OS X.
  - It's the industry standard for developing, testing, and training on a single machine.
- Why use Anaconda?
  - Support Python and R
  - Pre-installed most popular Python packages: scikit-learn, numpy, pandas, scipy, conda, etc.
  - Comes with the Jupyter Notebook and Ipython distribution.

Anaconda vs. miniconda vs. conda vs. pip

- Anaconda: repository, built-in most packages;
- Miniconda: repository management, no package built in; smaller
- Conda: a cross platform package and environment manager that installs and manages conda packages from the Anaconda repository.
  - Conda packages are binaries: no need to have compilers available
  - not limited to Python software, also contain R, Ruby, Lua, Scala, Java, JavaScript, C/C++, FORTRAN.

    conda update conda; conda update anaconda
- Pip: Python Packaging Authority's recommended tool for installing packages from the Python Package Index, PyPI.
  - Pip installs: wheels or source distributions. The latter may require that the system have compatible compilers, and possibly libraries, installed before invoking pip to succeed.
    - Pip installs python; pip update pip

Jupyter or Ipython Notebook

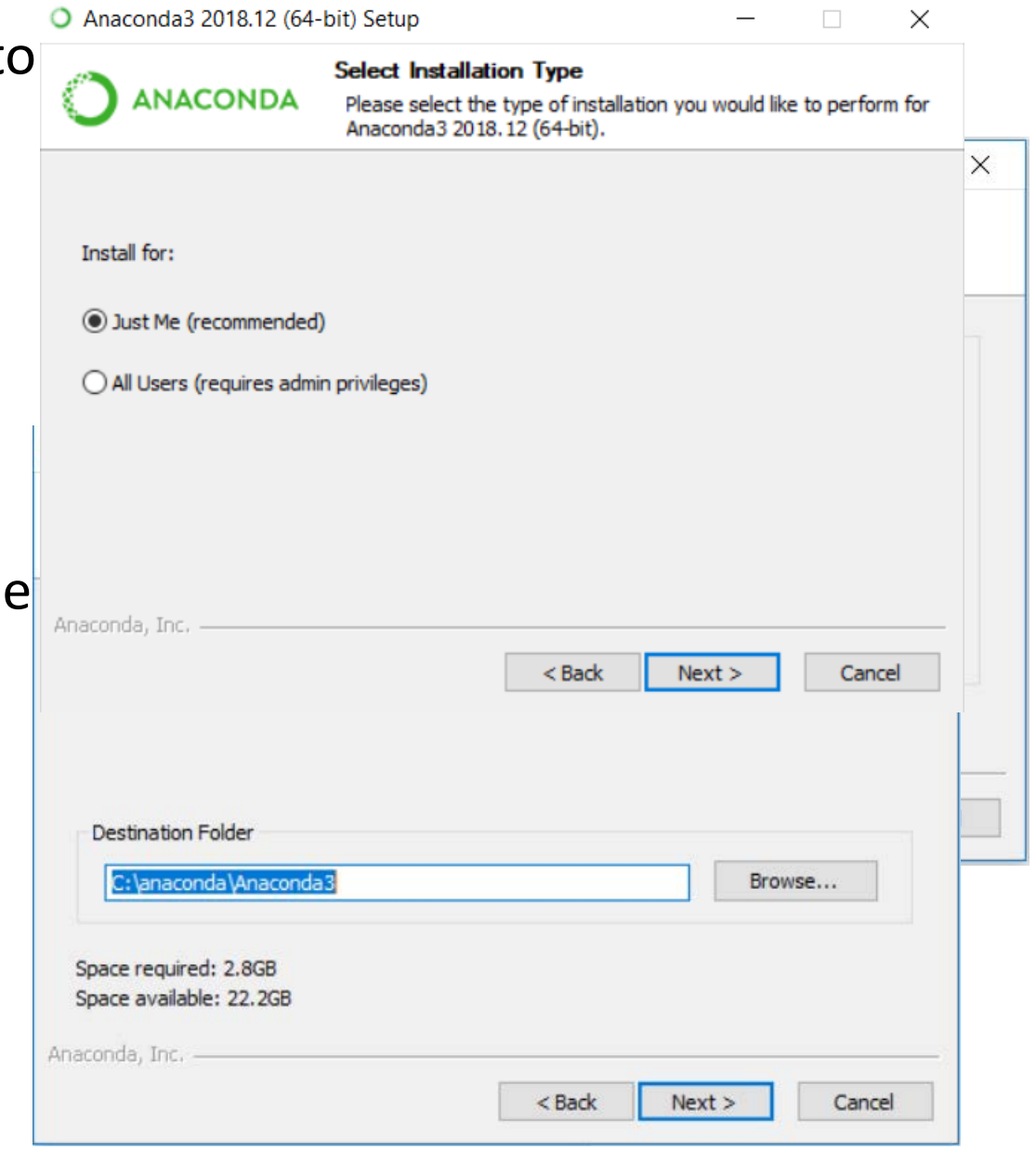
- A web application that allows you to run live code, embed visualizations and explanatory text all in one place.
- Enable users to create and share documents that combine
  - live code with narrative text,
  - mathematical equations,
  - visualizations,
  - interactive controls, and other rich output.
  - It also provides building blocks for interactive computing with data: a file browser, terminals, and a text editor.
- Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

Lab 1: Install Anaconda, Python and Jupyter

- Strongly recommend installing Python and Jupyter using the Anaconda Distribution, which includes Python, the Jupyter Notebook, and other commonly used packages for scientific computing and data science.
- First, download Anaconda https://www.anaconda.com/download/
  - Recommend downloading Anaconda's latest Python 3 version.
- Second, install the version of Anaconda which you downloaded, following the instructions on the download page.

When installing, some points to pay attention to

1. Temporarily turn off antivirus software if necessary.
2. Check “Just me”.
3. Do not use the admin for the installation.
4. In “Advanced Installation Options”, don’t che “Add Anaconda to my PATH environment variable”; otherwise it will affect other applications.
5. Do not use unicode characters.



Congratulations!

- Congratulations, you have installed Anaconda!
- Validate: at the Terminal (Mac/Linux) or Command Prompt (Windows)
  - Anaconda: Anaconda Navigator; or Anaconda Prompt
  - Conda:

    conda list; conda --help
  - Python:

    python
  - Jupyter Notebook:

    jupyter notebook

Installing Jupyter with pip

- As an existing or experienced Python user, you may wish to install Jupyter using Python's package manager, pip, instead of Anaconda.

- If you have Python 3 installed (which is recommended):

```
python3 -m pip install --upgrade pip
python3 -m pip install jupyter
```

- If you have Python 2 installed:

```
python -m pip install --upgrade pip
python -m pip install jupyter
```