Lecture 1 (part 2): Jupyter and IPython

Advanced Business Analytics (CIS442D/85)

Simon Business School

1/4/2017

Python for scientific computing

- Advantages: Cross-platform, Interactive (experiment while developing code), intuitive syntax, extensive, ...
- Limitations
  - Slower than compiled languages (e.g., C, Pascal)
  - Exploration code disappears with shutdown
  - Sharing results – must complement with word/excel/power point to present images and tables

Python for scientific computing

- Advantages: Cross-platform, Interactive (experiment while developing code), intuitive syntax, extensive, ...
- Limitations
  - Slower than compiled languages (e.g., C, Pascal)
  - Exploration code disappears with shutdown
  - Sharing results – must complement with word/excel/power point to present images and tables
- IPython - enhanced Python
- Jupyter - notebooks that contain code, images, text (equations), video

- This lecture: Jupyter and IPython

Jupyter Notebooks

- Web-based interface to IPython (also Julia, R, Ruby, ...)
- Create documents that run code and visualize data (similar to Maple and Mathematica)
- Share: online (nbviewer) or export to HTML, pdf, Python, ...
- Work remotely (Notebook server)

Jupyter Notebooks

- Lunch
  - Start menu
  - Anaconda navigator
  - From command prompt/terminal: jupyter notebook
- Runs a local web-server
  - `http://localhost:8888/tree`
  - close terminal to shutdown server
- Dashboard
  - Files, folders, .ipynb files
  - Create, delete, and upload
  - Menu bar: Running (manage notebooks), Clusters (parallel computing), Conda (package management)

(1) Notebooks

- Notebook's title is the same as file's name
- Three main areas: menu, toolbar and cells area
- Cells area
  - Code and markdown (formatted) cells
  - Execute cells individually
  - Notebooks are independent (different kernels)
  - Kernel can be stopped and restarted
  - * denotes kernel is running
  - Output area
    - ★ Displays output asynchronously
    - ★ Large output can be collapsed
  - Markdown - language for text formatting (superset of HTML)
    - ★ Latex equations, HTML code, images, video, etc.
    - ★ Resources: Wikipedia, Jupyter documentation, GitHub markdown guide, and a Markdown Cheatsheet
- All homeworks should be submitted as jupyter Notebooks

(2) IPython

- Enhanced Python shell
- More readable (colors, line numbers, printing objects)
- Auto-completion (variable names, functions, attributes, packages, filenames)
- Object introspection (?x)
- Command history
- Extended functionality through "Magic" commands
- Direct access to the operating system (!pwd)
- Other nice features: debugging, profiling, and parallel computing

(3) Operating System Shell Commands

- OS shell
  - Interface to the OS
  - Include a variety of useful programs
- Examples
  - Browsing: ls, pwd, cd
  - Files: cp, rm, mv, mkdir/md,
  - Text: cat, cut, head, sort, uniq, wc, grep,
  - But also: plotting, image conversion, encryption, compression, networking, etc.
- Can be called from IPython using "!"
- Save output to python variables
- Use $ to pass Python variable as shell commands arguments

Exercise 1

For each of the text files in the code directory, print its first row and the total number of lines and words

Magic Commands

- Facilitate common programming tasks
- Start with "%" (for lines) and "%%" (for cells)
- %quickref - quick reference for IPython commands
- %lsmagic (list magic commands)
- %run - executes python files
  - loads the created objects to memory
  - by default runs the files on an empty environment (-i to use IPython namespace)

Magic Commands - cont.

- %time - time code execution of a single command
- %timeit - average running time over multiple runs
- %who, %who_ls, %whos - print interactive variables
- %debug - enters debug mode just before exception raised (exit debug mode by typing "exit")
- %prun - run the code through a profiler (useful for optimizing code)

Magic Commands - cont.

- Used at the beginning of a cell (at most one per cell)
- Examples:
  - %%time/%%timeit - measures the running time of the entire cell
  - %%writefile file.py - exports the cell to a python file (-a for appending)
  - %%capture c - captures the output from a cell (standard and errors) to the variable c. Access using c.show(), or c.stdout and c.stderr
- Magic commands may be used without % (unless variables with similar names exist)
- Do not support commands that require input
- More on magic commands: IPython documentation

(5) Directory history

- %dhist - directory history
- use %cd -n to enter the n-th entry in the directory history
- %bookmark mydir - bookmarks the current directory
- cd mydir - change the current directory to mydir
- %store - store variables or functions for future usage

(6) Alternative Interface: IPython Terminal

- Lunch by typing "ipython" in the shell/command prompt
- No inline graphics (open in separate windows, not documents)

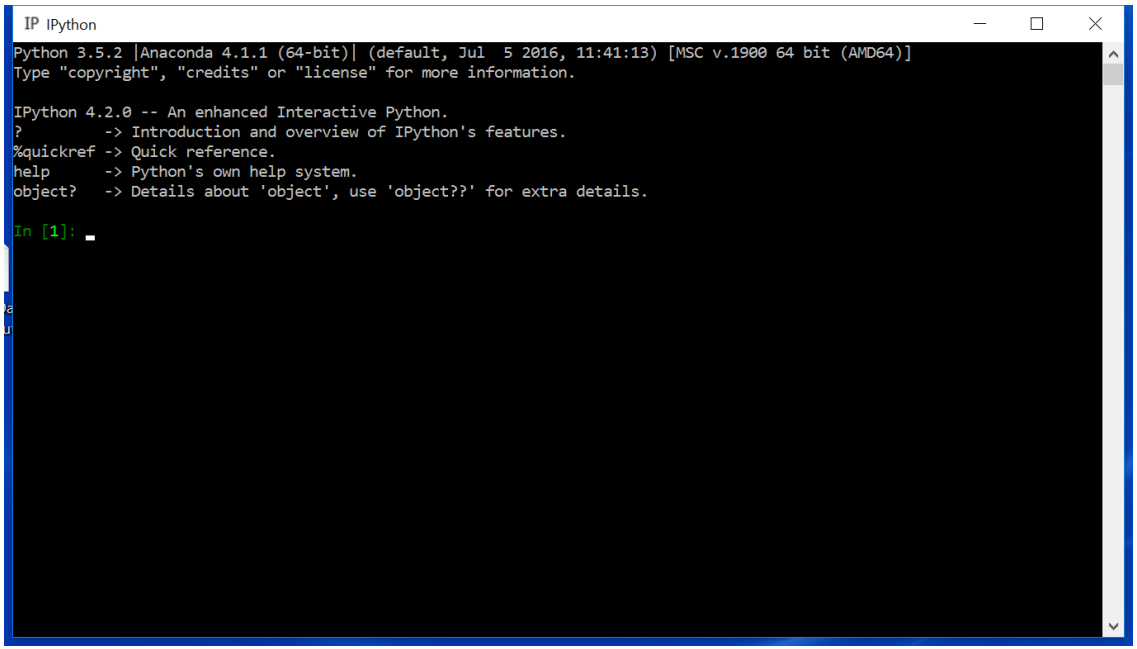

Figure: IPython Terminal

Alternative Interface: Jupyter QTConsole

- Lunch by typing "jupyter qtconsole" in the shell/command prompt (or "Jupyter QTConsole Desktop app" on Windows 10)
- Supports multi-line editing and inline graphics

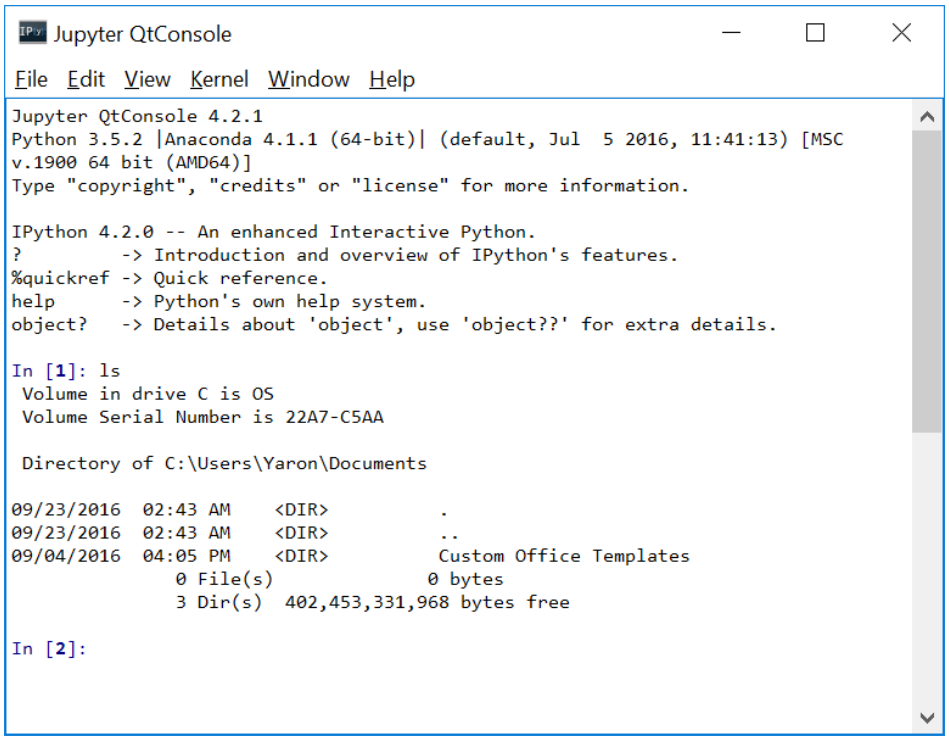

Figure: QT Console

Further Reading

- Shell commands: grep, sort, awk, uniq, cut, ... (tutorials: [1], [2], [3])
- Debugging and Profiling with IPython: Chapter 3 in Python for Data Analysis and the IPython tutorial
- Parallel computing
- Remote Server
- IPython IDEs: Spyder, Enthought, Python(x,y), WinPython, Pyzo
- Version control and sharing notebooks
  - GitHub gist
  - GitHub and Jupyter nbviewer
  - Gallery of IPython notebooks with many tutorials

Summary

- IPython - an enhanced Python shell (auto-completion, object introspection, magic commands, OS integration, ...)
- Jupyter
  - web-interface for creating formatted documents that run python code
  - our main work environment throughout the course