Lecture 1 (part 1): Introduction to Python

Advanced Business Analytics (CIS442D/85)

Simon Business School

1/4/2017

Announcements

- Updates to Syllabus
- Homework 1 due Wednesday, 1/10 at 23:55
- Form teams of 3 students and update Solomon (TA) by email

Teaching Team

- Instructor: Yaron Shaposhnik
  - Email: yaron@simon.rochester.edu
  - Office hours: Thursdays, 4:30-5:30pm
- TA: Solomon Abiola
  - Email: solomon_abiola@urmc.rochester
  - Office hours: Mondays, 11am-12pm
- TA: Yu Wang
  - Email: ywang176@ur.rochester.edu
  - Office hours: Tuesdays, 11am-12pm

Organization

- Website: Blackboard
- Course material
  1. An Introduction to Statistical Learning with Applications in R / James, Witten, Hastie and Tibshirani (Download, Amazon)
  2. The analytics edge / Bertsimas, O'Hair, Pulleyblank (Amazon, OpenCourseWare)
  3. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython / McKinney (Amazon, somewhat outdated but a new edition is expected soon)
  4. Online documentation
- Grading
  - 10% Professional standards
  - 20% Midterm (week 5)
  - 30% Individual weekly homework assignments (lowest grade excluded)
  - 30% Team analytics project (week 8)
  - 10% Team Python tools project (week 9)

Advanced Business Analytics (CIS442D)

- Analytics?

Advanced Business Analytics (CIS442D)

- Analytics?
- From Bertsimas et al. 2016: "Analytics is the science of using **data** to build **models** that add **value** to **decisions** made by companies, institutions and individuals"

Advanced Business Analytics (CIS442D)

- Analytics?
- From Bertsimas et al. 2016: "Analytics is the science of using **data** to build **models** that add **value** to **decisions** made by companies, institutions and individuals"
- Learning objectives:
  1. Learn how to load, represent, clean, manipulate and visualize data in Python
  2. Study advanced descriptive and predictive statistical models

Python

- Open source (free), Cross-platform (runs on Windows, OS X, Linux), high-level programming language
- General-purpose (networking, databases, GUI, web-servers, ...)
- Supports multiple programming paradigms (objected oriented, functional, ...)
- Built-in tools and third-party utilities
- Running modes:
  1. Interactively
  2. Running module files
  3. Embedded

Python for data science

- Interactive (interpreted language)
- Simple and easy to learn (intuitive syntax, dynamically typed, automatic memory management, ...)
- Modules for data science
  - Numerical computations (NumPy,SciPy)
  - Manipulating data (pandas)
  - Visualization (matplotlib)
  - Machine learning/data mining (scikit-learn,statsmodels)
- Large and active scientific community (documentations, tutorials, blogs, forums, organizations, ...)

Python for data science

- Interactive (interpreted language)
- Simple and easy to learn (intuitive syntax, dynamically typed, automatic memory management, ...)
- Modules for data science
  - Numerical computations (NumPy,SciPy)
  - Manipulating data (pandas)
  - Visualization (matplotlib)
  - Machine learning/data mining (scikit-learn,statsmodels)
- Large and active scientific community (documentations, tutorials, blogs, forums, organizations, ...)

- This lecture:
  1. Python development environments
  2. Exercise Python programming

Python Development Environments

- Minimal: notepad (running files) or Python shell (interactive)
- Basic: IDLE (Syntax highlighting, auto-completion, smart indent, debugging)
- Software development: PyDev (Eclipse), Visual Studio
- Data analysis (IPython): Spyder, jupyter

Exercise: Basic file processing

Using Spyder, write a code that performs the following basic analysis of the file novel.txt (source)

1. Print the first row
2. Print the first 10 rows
3. What is the total number of rows?
4. How many rows are not empty?
5. How many rows containing the word "Christmas"?
6. Create a list of the words appearing in the document. What is the total number of words? print the first 20 words.
7. Repeat (6), this time removing any non alpha-numeric characters
8. Compute the frequency of each word.
9. Print the 20 most frequently used words
10. Export the frequencies to a file. Each row should contain the frequency of a word follows by comma, and the word. Words should appear from the most to least frequent.

Regular Expressions (RE)

- Language for specifying patterns in strings
- Typical usage:
  - Does string match pattern? (e.g., email address)
  - Is there a match inside a string?
- Building blocks
  - characters: abc123
  - meta-characters: . ^$ * + ? { } [ ] \ | ( )
  - classes: [ab2], [^ab2]
  - repetitions: [ab2]*, [ab2]+, [ab2]?, [ab2]{3,5}
  - Examples: "ab2aba", "a", "aba2", "ac"
  - predefined classes. digits: \d, \D, alpha-numeric: \w, \W, white-spaces: \s, \S
- Python documentation: [1], [2], [Online tutorial]

Summary

- Python development environment: files and console
- Basic data cleaning tasks
  - Accessing files
  - Control of flow (if, else, for, while)
  - Defining functions (identified and anonymous)
  - Built-in data structures (lists, dictionaries, tuples)
  - Built-in tools: filter, sort, map, list comprehension
  - Strings, string formatting, and regular expressions
- Onine Python tutorial (exceptions, classes, additional tools)
- "Automate the Boring Stuff with Python: Practical Programming for Total Beginners" by Al Sweigart [Online] [Amazon]