CME 193: Introduction to Scientific Python

Winter 2017

Lecture 6: Pandas and sci-kit-learn

Blake Jennings

`stanford.edu/~bmj/cme193`

Contents

Today

- Quick word on Python environments
- Introduce pandas
- Introduce scikit-learn
- Show some examples in Jupyter notebook

HW2/Project

- Option: Homework 2 or project – **Due Thursday February 16th**
- Project proposals due today!
- Great ideas from many of you so far

Contents

Python environments

Pip is a package manager, and Virtualenv is an environment manager.

Conda is both.

`https://conda.io/docs/_downloads/`

`conda-pip-virtualenv-translator.html`

If you are using virtualenv, I recommend also using virtualenvwrapper:

You can install virtualenv with brew and virtualenvwrapper with pip.

Why do we want to use Python environments?

Tool to create isolated Python environments. virtualenv/conda creates a folder which contains all the necessary executables to use the packages that a Python project would need.

Download an open source project and easily install the requirements in a self-contained environment.

You can manage environments of Python 2 and Python 3, ensure dependencies don't clash.

Git

When working on a task like your project, it is often the case that we want to *version* our code.

git: version control system.

Tutorial: `https://medium.com/@abhishekj/an-intro-to-git-and-github-1a0e2c7e3a2f`

Contents

Pandas

`http://pandas.pydata.org/pandas-docs/stable/`

Introduced in 2011, Pandas is a Python library providing high-performance, easy-to-use data structures and data analysis tools

Motivated by R, pandas came out of the Finance industry. Now a key component to SciPy.

Provides fast, flexible, and expressive data structures designed to make working with relational or labeled data both easy and intuitive

Pandas - strengths

- Easy handling of missing data (represented as NaN) in floating point as well as non-floating point data
- Size mutability: columns can be inserted and deleted from DataFrame and higher dimensional objects
- Powerful, flexible group by functionality to perform split-apply-combine operations on data sets, for both aggregating and transforming data
- Make it easy to convert ragged, differently-indexed data in other Python and NumPy data structures into DataFrame objects
  Intelligent label-based slicing, fancy indexing, and subsetting of large data sets

Pandas - strengths

- Intuitive merging and joining data sets
- Flexible reshaping and pivoting of data sets
- Robust IO tools for loading data from flat files (CSV and delimited), Excel files, databases etc.
- Time series-specific functionality: date range generation and frequency conversion, moving window statistics, moving window linear regressions, date shifting and lagging, etc.

Main data structures

Table: Main data structures

Dimensions	Name	Description
1	Series	1D labeled homogeneously-typed array
2	DataFrame	General 2D labeled, size-mutable tabular structure
3	Panel	General 3D labeled, also size-mutable array

Pandas - introduction

Let's switch over to a Jupyter notebook and see how work with DataFrames and Series objects

Contents

scikit-learn

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib

`http://scikit-learn.org/stable/`

Classification, Regression, Clustering, Dimensionality reduction, model selection, preprocessing...

scikit-learn

Let's see a simple example to see an intro to the capabilities of sklearn.

Exercises

No new exercises today! Light exercises next week.

Please use the remaining time to construct a notebook to start exploring your data!

Feel free to email me with questions about project