Python: Scientific Computing Ecosystem

North American Econometric Society Meeting Workshop

June 15, 2016

Agenda

1. Scientific Programming with Python
2. Brief Introduction to Pandas
   - pd.Series
   - pd.DataFrame
   - Time Series Data
3. Demo
4. Resources

Pandas Examples

Notebook:

`https://github.com/QuantEcon/emet_summer_workshop`

1. Chicago Federal Reserve Bank Data (Excel)
    - Plotting Data
    - Jupyter Interactives
2. Working with **medium** sized datasets
    - International Trade Data - SITC Rev 2.
    - Compute RCA for 250 countries and 986 products across 52 years.
3. Extracting Tables from Web Data

Scientific Programming with Python

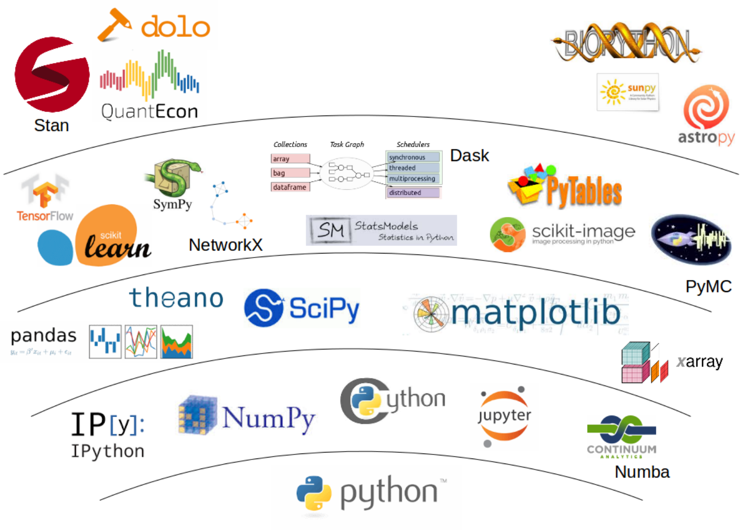
Rapid adoption by the scientific community

- engineering
- computational biology
- chemistry
- physics, etc., etc.

More recently

- AI, machine learning, “data science”

The Python Ecosystem of Packages



[1] Diagram inspired from a presentation given by Jake VanderPlas (Thanks @jakevdp)

Key Scientific Libraries

NumPy

- basic data types
- simple array processing operations

SciPy

- built on top of NumPy
- provides additional functionality

Matplotlib

- 2D and 3D figures

NumPy

NumPy Example: Mean and standard dev of an array

```
In [1]: import numpy as np

In [2]: a = np.random.randn(100)

In [3]: a.mean()
Out[3]: -0.0914807879869576O7

In [4]: a.std()
Out[4]: 1.093037615548889
```

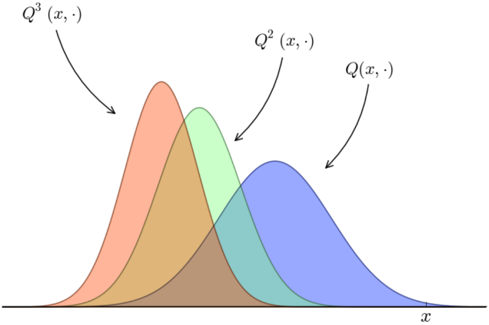
SciPy

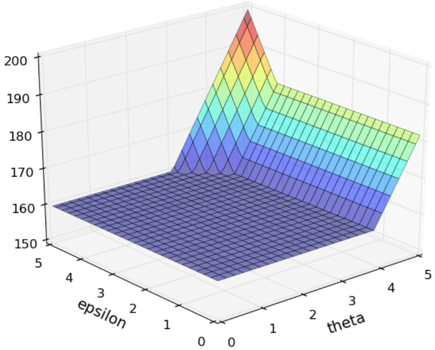
SciPy Example: Calculate

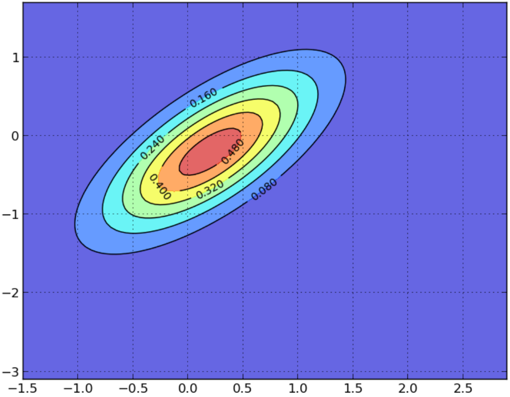
$$\int_{-2}^{2} \phi(z)dz \quad \text{where} \quad \phi \sim N(0,1)$$

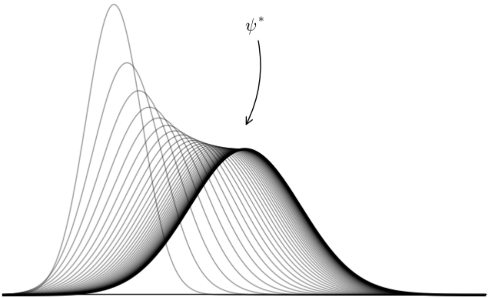
```
In [1]: from scipy.stats import norm
In [2]: from scipy.integrate import quad

In [3]: phi = norm(0, 1)

In [4]: value, error = quad(phi.pdf, -2, 2)

In [5]: value
Out[5]: 0.9544997361036417
```

Matplotlib examples




200
190
180
170
160
150
5
4
3
2
1
0
epsilon
0
1
2
3
4
5
theta


0.160
0.240
0.480
0.400
0.320
0.080
1
0
−1
−2
−3
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5


ψ^*

Other Scientific Libraries

Pandas

- array and tabular data objects
- statistics and data analysis

SymPy

- symbolic manipulations à la Mathematica

Still more:

- **statsmodels** — statistics / econometrics
- **scikit-learn** — machine learning in Python

Python Libraries for Economics

QuantEcon (`http://quantecon.org/`) provides

- Markov chains
- Dynamic programming
- LQ control
- etc

Dolo for quantitative macro

- A modeling language
- Many solution methods

Other Scientific Tools

Also tools for

- working with graphs (as in networks)
- parallel processing, GPUs
- manipulating large data sets
- interfacing with C / C++ / Fortran
- cloud computing
- database interaction
- bindings to high level languages like R and Julia
- etc.

See **Resources** slides at the end of the presentation for more info.

Pandas

Pandas is the key library for data work in Python and it is built on top of **NumPy**

Some things that Pandas is very good at:

1. Easy handling of missing data (represented as NaN)
2. Automatic and explicit data alignment
3. Hierarchical labeling of axes

Reference: `http://pandas.pydata.org/`[1]

[1] Current docs are 2,017 pages long!

Pandas

Pandas is focused on two primary abstractions:

1. pd.Series() - Array Like Data
2. pd.DataFrame() - Tabular Data

Pandas - Continued

Operations:

1. Powerful, flexible group-by functionality to perform split-apply-combine operations on data sets, for both aggregating and transforming data
2. Intelligent label-based slicing, fancy indexing, and sub-setting of large data sets
3. Intuitive merging and joining of data sets
4. Flexible reshaping and pivoting of data sets

Reference:

`http://pandas.pydata.org/pandas-docs/version/0.18.1/index.html`

Pandas - Continued

IO:

1. Robust IO tools for loading data from
   - flat files (CSV and delimited),
   - Excel files,
   - databases,
   - and saving / loading data from the fast HDF5 format

Reference:

`http://pandas.pydata.org/pandas-docs/version/0.18.1/io.html`

Pandas - Continued

Specialized Data Types: TimeSeries

1. Time series specific functionality:
    - date range generation and frequency conversion,
    - moving window statistics,
    - moving window linear regressions,
    - date shifting and lagging, etc.
    - time zone handling

Reference:

`http://pandas.pydata.org/pandas-docs/version/0.18.1/timeseries.html`

pd.Series Object

A **Pandas** Series is a one-dimensional labeled array capable of holding any data type (integers, strings, floating point numbers, Python objects, etc.).

```
import pandas as pd
s = pd.Series([5,4,3,2,1], index=[’a’, ’b’, ’c’, ’d’, ’e’])
```

Produces the following object:

```
a    5
b    4
c    3
d    2
e    1
dtype: int64
```

pd.Series Object

```
s.sort_values()
```

```
e    1
d    2
c    3
b    4
a    5
dtype: int64
```

pd.Series Object

```
    s[s > 2]
```

```
a     5
b     4
c     3
dtype: int64
```

pd.DataFrame Object

```
d = {’one’ : pd.Series([1., 2., 3.],
                        index=[’a’, ’b’, ’c’]),
     ’two’ : pd.Series([1., 2., 3., 4.],
                        index=[’a’, ’b’, ’c’, ’d’])}
df = pd.DataFrame(d)
```

Produces the DataFrame:

	one	two
a	1.0	1.0
b	2.0	2.0
c	3.0	3.0
d	NaN	4.0

Quick Pandas Demo

1. Time Series Data
2. Chicago Federal Reserve - CFNAI Data (Plotting)
3. FRED Data (Quick Access)

See: **intro-python-data-analysis.ipynb**

Resources

For a collection of scientific computing packages

`https://wiki.python.org/moin/NumericAndScientific`

Good starting points:

Working with Data and Analysis

1. pandas
2. Numba - Fast Loops in Python
3. statsmodels - Regression and Statistics
4. scikit-learn - Machine Learning

Packages ...

Working with Data and Analysis

1. dask
   - flexible parallel computing library for analytics
   - dask.DataFrame
2. odo - Data Conversions
3. NetworkX - Networks
4. xarray - N-dimensional Pandas (New)

Packages ...

Plotting

1. matplotlib
2. Plotly
3. Bokeh
4. ... many others

Packages ...

Databases and Files

1. pandas - Provides readers and writers
2. H5Py - Working with HDF Files
3. SQLAlchemy
4. ... many others

Packages ...

Web Scraping

1. lxml
2. Requests
3. Beautiful Soup
4. Scrapy
5. ... many others

Packages ...

Language Interfaces

1. Rpy2 - Interface to R
2. PyJulia - Interface to Julia

+++ many more