PYTHON FOR DATA GEEKS

Ouz Gencoglu

Tampere University of Technology, Finland

October 2015

Outline

1 Python Basics

2 Packages

3 Notes

Python Basics

How to start?

- Readability is part of the syntax!

Python Basics

How to start?

- Readability is part of the syntax!
- Indexing starts from 0

Python Basics

How to start?

- Readability is part of the syntax!
- Indexing starts from 0
- Cyclic

Most Useful Libraries

- numpy, scipy, statsmodel, pandas

Most Useful Libraries

- numpy, scipy, statsmodel, pandas
- Generic ML: scikit-learn

Most Useful Libraries

- numpy, scipy, statsmodel, pandas
- Generic ML: scikit-learn
- NLP: nltk, spaCy

Most Useful Libraries

- numpy, scipy, statsmodel, pandas
- Generic ML: scikit-learn
- NLP: nltk, spaCy
- Deep Learning: caffe, keras, lasagne

Most Useful Libraries

- numpy, scipy, statsmodel, pandas
- Generic ML: scikit-learn
- NLP: nltk, spaCy
- Deep Learning: caffe, keras, lasagne
- matplotlib, seaborn, bokeh (interactive)

Most Useful Libraries

- numpy, scipy, statsmodel, pandas
- Generic ML: scikit-learn
- NLP: nltk, spaCy
- Deep Learning: caffe, keras, lasagne
- matplotlib, seaborn, bokeh (interactive)
- PyPy, Cython

Most Useful Libraries

- numpy, scipy, statsmodel, pandas
- Generic ML: scikit-learn
- NLP: nltk, spaCy
- Deep Learning: caffe, keras, lasagne
- matplotlib, seaborn, bokeh (interactive)
- PyPy, Cython
- rPython, RPy2

Notes

- use map, zip and lambda

Notes

- use map, zip and lambda
- do not unzip just read from zip file

Notes

- use map, zip and lambda
- do not unzip just read from zip file
- sklearn pipeline

Notes

- use map, zip and lambda
- do not unzip just read from zip file
- sklearn pipeline
- never use gradient boosting from sklearn, instead xgboost

Notes

- use map, zip and lambda
- do not unzip just read from zip file
- sklearn pipeline
- never use gradient boosting from sklearn, instead xgboost
- set n_devices = -1 for RF

Notes

- use map, zip and lambda
- do not unzip just read from zip file
- sklearn pipeline
- never use gradient boosting from sklearn, instead xgboost
- set n_devices = -1 for RF
- matlab-like (c-like) structs - use dicts

Notes

- use map, zip and lambda
- do not unzip just read from zip file
- sklearn pipeline
- never use gradient boosting from sklearn, instead xgboost
- set n_devices = -1 for RF
- matlab-like (c-like) structs - use dicts
- use numpy memmap arrays for large files

Notes

- use map, zip and lambda
- do not unzip just read from zip file
- sklearn pipeline
- never use gradient boosting from sklearn, instead xgboost
- set n_devices = -1 for RF
- matlab-like (c-like) structs - use dicts
- use numpy memmap arrays for large files
- save the list of packages - version control - do not update all the time

Notes

- use map, zip and lambda
- do not unzip just read from zip file
- sklearn pipeline
- never use gradient boosting from sklearn, instead xgboost
- set n_devices = -1 for RF
- matlab-like (c-like) structs - use dicts
- use numpy memmap arrays for large files
- save the list of packages - version control - do not update all the time
- `https://github.com/ogencoglu/Templates/tree/master/Python`

R vs. Python

- R was developed by statisticians

R vs. Python

- R was developed by statisticians
- R has terrible error reporting

R vs. Python

- R was developed by statisticians
- R has terrible error reporting
- R syntax is bad: function names do.this, doThis, do_this

R vs. Python

- R was developed by statisticians
- R has terrible error reporting
- R syntax is bad: function names do.this, doThis, do_this
- Memory and speed issues in R

R vs. Python

- R was developed by statisticians
- R has terrible error reporting
- R syntax is bad: function names do.this, doThis, do_this
- Memory and speed issues in R

- Python has backward compatibility issues between 2.x and 3.x

R vs. Python

- R was developed by statisticians
- R has terrible error reporting
- R syntax is bad: function names do.this, doThis, do_this
- Memory and speed issues in R

- Python has backward compatibility issues between 2.x and 3.x
- In any case both are slow compared to C++ etc.

R vs. Python

- R was developed by statisticians
- R has terrible error reporting
- R syntax is bad: function names do.this, doThis, do_this
- Memory and speed issues in R

- Python has backward compatibility issues between 2.x and 3.x
- In any case both are slow compared to C++ etc.

For stats and statistical inference use R. For data cleaning and plotting it does not matter. Anything else: use python.

R vs. Python

- R was developed by statisticians
- R has terrible error reporting
- R syntax is bad: function names do.this, doThis, do_this
- Memory and speed issues in R

- Python has backward compatibility issues between 2.x and 3.x
- In any case both are slow compared to C++ etc.

For stats and statistical inference use R. For data cleaning and plotting it does not matter. Anything else: use python.

Start learning Go and Julia