Prepared for submission to JHEP

Python Cheatsheet

Steve Young

Abstract: Everything I know about Python.

Contents

TODO: make sidebar marker telling what python library command depends on (and needs to be loaded beforehand)

1 Data Processing

Read data from json file: `with open('data.json', 'r') as f: data = json.load(f)`

Write data to json file: `with open('data.json', 'w') as f: data = json.dump(f)`

Regex processing of text, `re` package: look at https://docs.python.org/3/library/re.html

2 Misc

Get indexed elements from list or NDArray: `enumerate(list)`

3 Numpy

Using `import numpy as np`

3.1 Data Processing

Import data from csv file: `np.genfromtxt('filename', delimiter='','')`

3.2 Number generation

Constant matrix: `np.full(shape, val)`

Matrix of ones/zeros: `np.ones(shape)`, `np.zeros(shape)`

Id matrix: `np.eye(dim)`

Uniform dist on (low,high): `np.random.uniform(low, high, numsamps or shape)`

Uniform dist on (0,1) with given dims: `np.random.rand(d1, d2, ..)`

Normal dist: `np.random.normal(mean, stddev, numsamps)`

Normal dist on with given dims: `np.random.randn(d1, d2, ..)`

Multivariate normal: `np.random.multivariate_normal(..args)`

Random permutation of elements in ndarray: `np.random.permutation(NDArray)`

Permute elements of (range or ndarray) *in place*: `np.random.shuffle(int or NDArray)`

Integers over specified range: `np.arange(start, stop)`

Even spaced numbers over specified range: `np.linspace(start, stop, numvals)`

3.3 NDArray handling

- NDArrays are naturally *row vectors*, and of shape $(m,)$.

Reshape array: `np.reshape(NDArray, tuple of shape)`

3.4 NDArray ops

max/min element of array: `np.max(NDArray)`, `np.min(NDArray)`

index of max/min element of array: `np.argmax(NDArray)`, `np.argmin(NDArray)`

fill diagonal of sq matrix: `np.fill_diagonal(NDArray, val)`

round elements to nearest int: `np.rint(NDArray)`

return bin counts in histogram: `np.histogram(NDArray, binboundaries)`

nth difference of array: [1] `np.diff(NDArray, n)`

3.5 Linear Algebra

Inverse matrix: `np.linalg.inv(square NDArray)`

[1] Think transforming array of tick prices into array of tick prices *changes*

Transpose matrix: `np.linalg.transpose(NDArray)`

evals and right evects: `np.linalg.eig(square NDArray)`

4 Matplotlib

Using `import matplotlib as mpl`, `import matplotlib.pyplot as plt`

show image (if not in inline mode): `plt.show()`

plot image: `plt.imshow(NDArray)`

set axis bounds: `plt.axis([xmin, xmax, ymin, ymax])`

set x,y axis label: `plt.xlabel(name)`, `plt.ylabel(name)`

set plot title: `plt.title(name)`

show plot legend: `plt.legend()`

visualize matrix vals as heat map: `plt.matshow(NDArray)`

pan/zoomable plots in PyCharm: insert `mpl.use('Qt5agg')` before `import matplotlib.pyplot as plt`

5 Pandas

6 sklearn

Cross Validation: `sklearn.model_selection.cross_val_score`

7 re

8 packages to try

text from PDFs: `PyPDF2`