JÜLICH
FORSCHUNGSZENTRUM

# Statistics using Python

## Porting Code from Matlab to Python - 2017

10 October 2017 | Rajalekshmi Deepu

# Statistics in Matlab and python

- Matlab:
  - Proprietary software.
  - Need “Statistics” toolbox. (extra cost )
- Python:
  - Opensource
  - Extended with a fantastic ecosystem of data-centric packages like: **numpy**,**scipy**, **matplotlib**, **scikit-learn**,**pandas**, **…**

# Numpy and Statistics(Descriptive)

- Contains in-built statistical functions like Mean, Median, Standard Deviation and Variance.

Matlab

```
>> load wages.dat
% Mean
>> Mean_value = mean(wages)
% Median
>> med_value = median(wages)
% Standard deviation
>> std_value = std(wages)
% Variance
>> var_value = var(wages)
```

Python (using Numpy)

```
>>> import numpy as np
>>> X = [16.92, 96.10, 11.82, 44.32,
55.66, 10.75]
>>>  mean = np.mean(X)
>>>  median = np.median(X)
>>>  sd = np.std(X)
>>> variance = np.var(X)
```

# SciPy (SciPy)

- Scientific Computing Package for Python.

  >>> help(scipy)

- Built on top of Numpy and uses Numpy arrays and data types.
- Scipy package is organized into several sub-packages.
- Imports all functions in the Numpy package, and several commonly used functions from sub-packages, into the top level namespace.

  e.g: scipy.var and numpy.var

  both refers to function var in module numpy.core.fromnumeric

  scipy.array and numpy.array

  both refers to built-in function array in module numpy.core.multiarray

# SciPy and Statistics (Inferential)

- SciPy offers an extended collection of statistical tools such as distributions (continuous and discrete) and functions.
- Few sub packages for statistics are:

  scipy.cluster --- Vector Quantization / Kmeans

  scipy.stats --- Statistical Functions

  scipy.stats.t --- Student's T test

**Remember: Subpackages requires an explicit import**

e.g: >>> import scipy.cluster

>>> from scipy import stats

# scipy.stats

```
Help on package scipy.stats in scipy:

NAME
    scipy.stats

DESCRIPTION
    ==========================================
    Statistical functions (:mod:`scipy.stats`)
    ==========================================

    .. module:: scipy.stats

    This module contains a large number of probability distributions as
    well as a growing library of statistical functions.
```

```
Statistical functions
=====================

Several of these functions have a similar version in scipy.stats.mstats
which work for masked arrays.

.. autosummary::
   :toctree: generated/

   describe          -- Descriptive statistics
   gmean             -- Geometric mean
   hmean             -- Harmonic mean
   kurtosis          -- Fisher or Pearson kurtosis
   kurtosistest      --
   mode              -- Modal value
   moment            -- Central moment
   normaltest        --
   skew              -- Skewness
   skewtest          --
   kstat             --
   kstatvar          --
   tmean             -- Truncated arithmetic mean
   tvar              -- Truncated variance
   tmin              --
   tmax              --
   tstd              --
   tsem              --
   variation         -- Coefficient of variation
   find_repeats
   trim_mean

.. autosummary::
   :toctree: generated/
```

# Scipy and Matlab

- scipy.io.matlab - Utilities for dealing with MATLAB files.
- Included functions:
  - scipy.io.loadmat - Load MATLAB file. Returns dictionary with variable names as keys, and loaded matrices as values.
  - scipy.io.savemat - Save a dictionary of names and arrays into a MATLAB-style .mat file.
  - scipy.io.whosmat - List variables inside a MATLAB file.

Jupyter Notebook: Load_List_Save_MAT_files

# Matplotlib

## ➢ matplotlib.mlab

Numerical python functions written for compatibility with MATLAB commands with the same names.

MATLAB compatible functions
:func:`cohere` Coherence (normalized cross spectral density)
:func:`csd` Cross spectral density using Welch's average periodogram
:func:`detrend` Remove the mean or best fit line from an array
:func:`find` Return the indices where some condition is true; numpy.nonzero is similar but more general.
:func:`griddata` Interpolate irregularly distributed data to a regular grid.
:func:`prctile` Find the percentiles of a sequence
:func:`prepca` Principal Component Analysis
:func:`psd` Power spectral density using Welch's average periodogram
:func:`rk4` A 4th order runge kutta integrator for 1D or ND systems
:func:`specgram` Spectrogram (spectrum over segments of time)

# Principal Component Analysis (PCA)

- Way of identifying patterns and expressing the data to highlight their similarities and differences.
- Powerful tool for analyzing high dimensional data.
- Enables data compression without much loss of information by reducing the number of dimensions.

# Matlab code for PCA (An example)

```
rd = load_untouch_nii('edtd.nii');
rd = double(rd.img);
sz = size(rd)
nrows = sz(1)
ncols = sz(2)
nslcs = sz(4)
s = reshape(rd,nrows*ncols,nslcs);
[coeff,score] = pca(s);
s = reshape(score,nrows,ncols,nslcs);
n = make_nii(s);
save_nii(n,'results/pca.nii')
```

*Ref: https://de.mathworks.com/help/stats/pca.html*

*https://stackoverflow.com/questions/35651133/matlab-and-python-produces-different-results-for-pca*

# PCA using Python (matplotlib.mlab)

- Hint:
  - Use matplotlib.mlab.PCA
  - Imported as given below:

    from matplotlib.mlab import PCA
  - Dataset: *edtd.nii*
  - Ref:

http://matplotlib.org/api/mlab_api.html#matplotlib.mlab.PCA

http://nipy.org/nibabel/nibabel_images.html

Jupyter Notebook

# Scikit-learn or sklearn

- Meant for machine learning in Python
- sklearn.cluster.KMeans
- ‘sklearn.decomposition’ module includes matrix decomposition algorithms, including among others PCA, NMF or ICA.

  e.g. modules:

  - sklearn.decomposition.nmf - Non-negative matrix factorization
  - sklearn.decomposition.pca - Principal Component Analysis

- Most of the algorithms of this module can be regarded as dimensionality reduction techniques.

# PCA using Python (sklearn)

- Hint:
  - Use sklearn.decomposition.PCA
  - Imported as given below:

    from sklearn.decomposition import PCA
  - Dataset: *edtd.nii*
  - Ref:

    http://scikitlearn.org/stable/modules/generated/sklearn.decomposition.PCA.html

    http://nipy.org/nibabel/nibabel_images.html

Jupyter Notebook *Optional*

# Other Python modules for Statistics

- Seaborn : Statistical data visualization

  http://seaborn.pydata.org

- Statsmodels : Library for statistical and econometric analysis in Python.

  http://statsmodels.sourceforge.net/

Jupyter Notebook : seaborn_savefig

# References


The Python Language Reference: http://docs.python.org/2/reference/index.html

The Python Standard Library: http://docs.python.org/2/library/

https://docs.scipy.org/doc/scipy/reference/tutorial/stats.html

http://matplotlib.org/api/mlab_api.html#module-matplotlib.mlab

http://conference.scipy.org/proceedings/scipy2010/pdfs/seabold.pdf

http://seaborn.pydata.org

https://www.datacamp.com/community/data-science-cheatsheets

PEP 20 -- The Zen of Python :https://www.python.org/dev/peps/pep-0020/

https://docs.scipy.org/doc/numpy-dev/user/numpy-for-matlab-users.html

https://www.tiobe.com/tiobe-index/