

Python libraries

Dr. Hans van der Kwast

Senior Lecturer in Ecohydrological Modelling

IHE Delft Institute for Water Education

h.vanderkwast@un-ihe.org

Schedule

- Monday

- Install software
- DOS and GDAL

- Tuesday

- Introduction to Python

- Wednesday

- Python libraries
- Map Algebra with PCRaster Python

- Thursday

- Spatial-dynamic modelling with PCRaster Python

- Friday

- PyQGIS

Learning objectives



After this lecture you will be able to

- Define what a Python library is
- Look for useful Python libraries for your research and analysis

- A Python library is a collection of functions and methods that allows you to perform lots of actions without writing your own code.
- You can import a library into your scripts

```
1 import os, glob
2 idrисиFiles = glob.glob("*.rst")
3 for idrисиFile in idrисиFiles:
4     print("Processing file", idrисиFile)
5     cmd = "gdal_translate -of GTiff " + idrисиFile + " " + \
6         os.path.splitext(idrисиFile)[0] + ".tif"
7     print(cmd)
8     os.system(cmd)
9 print("Done!")
```

- **Module**

- A module is a file containing Python definitions and statements. The file name is the module name with the suffix `.py` appended.

- **Package**

- Packages are a way of structuring Python's module namespace by using "dotted module names".

You can think of packages as the directories on a file system and modules as files within directories

- **Script**

- Refers to a module whose aim is to be executed. Similar to [program](#) or [application](#).

- **Library**

- generic term for a bunch of code that was designed with the aim of being usable by many applications.
- When a module/package/something else is "published" people often refer to it as a library. Often libraries contain a package or multiple related packages, but it could be even a single module.

Where to find libraries?

- Distributions come with lots of libraries
- Distributions have their own way to install new libraries
- E.g. Anaconda: `conda install scipy`
- Python Package Index (PyPI): repository of software for the Python programming language
 - <https://pypi.org/>
 - Contains 147,714 projects!

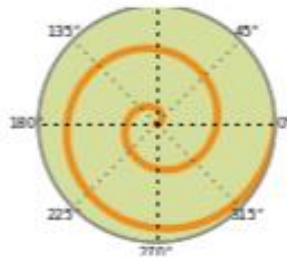
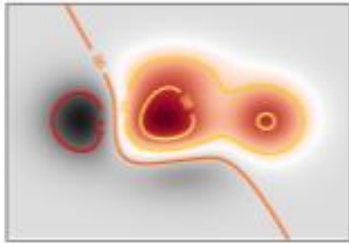
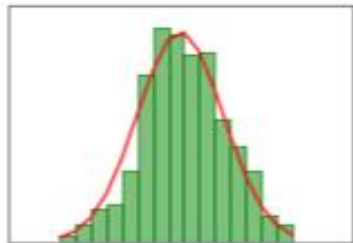
- **NUMPY**

- Defines the numerical array and matrix types and mathematical functions to operate on these arrays (cfr. Matlab)
- Accessing NumPy arrays is faster than accessing Python lists
- Matlab-style!
 - http://www.scipy.org/NumPy_for_Matlab_Users
 - <http://www.docstoc.com/docs/20492925/MATLAB-commands-in-numerical-Python>

- **Pandas**

- Uses NumPy to do advanced math, signal processing, optimization, statistics and more
-

- Easy to use 2D plotting library (in Matlab-lookalike syntax)
- Toolboxes for 3d-plotting (mplot3d) and maps (basemap)
- Interactive window or save-to-file
- Latex-styling possible
- Well-documented (<http://matplotlib.sourceforge.net/>) + Example gallery!



- Provides classes and functions for manipulating, reporting, and plotting time series of various frequencies
- Based on numpy arrays
- Interaction with Matplotlib
- Well documented
(<http://pytseries.sourceforge.net/contents.html>)

```
• Raint=np.loadtxt('rain.tss',skiprows=4)
• start_date = ts.Date(freq='D', year=1996, month=1, day=1)
• Rain=ts.time_series(Raint[:,1], start_date=start_date) #Create Timeserie
#Each timeserie got .date, .dates and .mask
• All_mean=Raineff.mean()
• Year_mean=Raineff.convert('Y',np.ma.mean) #Mean for each year individual
• Month_mean=Raineff.convert('M',np.ma.mean) #Mean for each month individual
• RowMonth_day=Raineff.convert('M') #Each month is separate row
• RowYear_month=Month_mean.convert('Y')
• MonthYear_mean=RowYear_month.mean(axis=0).data #Monthly means over all the years
```

```
>>> Rain
timeseries([ 0.  0.  0. ...,  0.  1.8  0.5],
            dates = [01-Jan-1996 ... 31-Aug-2004],
            freq = D)
```

- GDAL for manipulating geospatial raster data and OGR for manipulating geospatial vector data
- <http://www.gis.usu.edu/~chrisg/python/2009/>

```
1 | import gdal
2 | from gdalconst import *
3 | # Open the raster dataset
4 | dataset = gdal.Open(filename, GA_ReadOnly)
5 | # Print the projection of the data
6 | print dataset.GetProjection()
```

```
1 | import ogr
2 | # Get the driver
3 | driver = ogr.GetDriverByName('ESRI Shapefile')
4 | # Open a shapefile
5 | dataset = driver.Open(shapefileName, 0)
```

- » EXCEL-interaction
- » Read and write excel-files...

```
1 import xlrd
2 # Open the Excel file
3 book = xlrd.open_workbook("excelFile.xls")
4 # Read the first sheet in the Excel workbook
5 sheet = book.sheet_by_index(0)
6 # Read the first row from column A to E
7 rowValues = sheet.row_values(0, start_colx=0, end_colx=4)
8 # Print the row values
9 for value in rowValues:
10     print value

1 import xlwt
2 # Create a new workbook
3 book = xlwt.Workbook()
4 # Add a new sheet
5 sheet = book.add_sheet("My Sheet")
6 # Write the number 5 in the first row, first column
7 sheet.write(0, 0, 5)
8 # Save the file
9 book.save("myExcelFile.xls")
```

Further:

- Link to other languages:
 - R: Rpy, Rpy2, pyRserve 0.3
 - Fortran: f2py, fwrap
 - C/C++ : Weave, Cython
- SetupTools
 - Download, build, install, upgrade, and uninstall Python packages
- Distributions:
 - Pythonxy: <http://www.pythonxy.com/>
- Python Package Index...
 - <http://pypi.python.org/pypi>

Python	2.6.5		
xy	1.2.3	ETS	3.5.0
xydoc	1.0.3	VTK	5.6.1
PyQt	4.5.4	ITK	3.16
Spyder	2.0.5	mx	3.1.3
formlayout	1.0.9	pydicom	0.9.5
QtHelp	4.5.2	PyOpenGL	3.0.1
PyQt4	5.2.0	VPython	5.32
wxPython	2.8.10.1	SymPy	0.6.7
NumPy	1.5.1	cxopt	1.1.3
SciPy	0.8.0	PyWavelets	0.2.0
numexpr	1.4.1	scikits.timeseries	0.9.1.3
guidata	1.2.5	pyopency	2.1.0.wr
guiqwt	2.0.8.1	NetworkX	1.3
Matplotlib	1.0.0	MDP	2.6
gnuplot	1.8	PyTables	2.2.1
PIL	1.1.7	vttables	2.0
IPython	0.10.1	h5py	1.3.1.bet
Pyreadline	1.5	pyhdf	0.8.1
SetupTools	0.6.11	netcdf4	0.9
		GDAL	1.6.1
		PP	1.6.0
		Pywin32	2.14
		SendKeys	0.3
		pywinauto	0.4.0
		pyvisa	1.3
		PyParallel	0.2.0
		PySerial	2.5.0
		Cython	0.13
		psyc0	1.6
		py2exe	0.6.9
		Sphinx	1.0.4
		docutils	0.7
		jinja2	2.5.2
		pygments	1.3.1
		ReportLab	2.5
		rst2pdf	0.16
		simplejson	2.1.1
		xlrd	0.7.1
		xlwt	0.7.2
		nose	0.11.3
		pylint	0.22.0
		winpdb	1.4.8

- Python Data Science Handbook:
<https://github.com/jakevdp/PythonDataScienceHandbook>
 - Introduction to **NumPy**
 - Data manipulation with **Pandas**
 - Visualisation with **Matplotlib**
- Hydropy
<https://stijnvanhoey.github.io/hydropy/>