Python programming — exercises

Installation

Install Python

Install python and some libraries

Check that you can write:

```
$ python
>>> import simplejson
>>> import feedparser
>>> import cherrypy
>>> import pymongo
>>> import nltk
>>> nltk.download()
>>> from nltk.corpus import brown
>>> brown.words()
['The', 'Fulton', 'County', 'Grand', 'Jury', 'said', ...]
```

Install Python

Install `ipython` (e.g., by `pip`)

Start with:

```
ipython -pylab
```

Once installed make sure you can write:

```
In [1]: plot(sin(linspace(0,8,100)))
```

Install CGI Python

Copy CGI script from

http://www.student.dtu.dk/~faan/cgi-bin/helloworld

to your own directory:

`~/public_html/cgi-bin/`

and see that it works.

Extra task

After installing Cherrypy see that it works.

Try to get the `bonus-sqlobject.py` from the tutorial to work.

Note that this requires the installation of a SQL database. One of the line in the `bonus-sqlobject.py` file states:

```
# configure your database connection here
__connection__ = 'mysql://root:@localhost/test'
```

If you don't want to install MySQL try installing the simpler sqlite and its python support and then change the connection line.

Extra extra installation tasks

Install spyder

Get a hello world Google App Engine application up and running.

Get a hello world Heroku up and running.

General Python

For loops, str and int

Write a function, `ishashad` that determines whether a number is a *Harshad number* (for number base 10).

A Harshad number "is an integer that is divisible by the sum of its digits" (Wikipedia)

Example: $81 \rightarrow 8 + 1 = 9 \rightarrow 81/9 = 9 \rightarrow$ Harshad!

```
>>> ishashad(81)
True
```

Hint: convert the number to a string.

Dictionaries

Count the number of items in a list with the result in a dictionary.

List example:

```
l = ['a', 'b', 'f', 'f', 'b', 'b']
```

Should give something like:

```
c = {'a': 1, 'b': 3, 'f': 2}
```

What and where is `defaultdict`?

Recursion

Implement a factorial function, $n!$, with recursion:

```
>>> factorial(4)
24
```

$(4! = 1 \times 2 \times 3 \times 4 = 24)$

See what happens with `factorial(1000)`

Classes

Construct a module with a derived dictionary class with sorted keys:

```
>>> s = SortedKeysDict({'a': 1, 'c': 2, 'b': 3, 'd': 4})
>>> s.keys()
['a', 'b', 'c', 'd']
>>> s.items()
[('a', 1), ('b', 3), ('c', 2), ('d', 4)]
```

Also implement `doctest` for the class.

Document it and extract the document with, e.g., `pydoc`

File reading and simple computing

Consider a file with the following matrix $\mathbf{X}$:

```
1 2
3 4
```

Read and compute $\mathbf{Y} = 2 * \mathbf{X}$

Try also using the with statement in this case.

Project Euler

Project Euler is a website with mathematical problems that should/could be solved by computers.

Go to the Web-site http://projecteuler.net/ and solve some of the problems using Python.

As an example the problem number 16 can be solved in one line of Python:

```
>>> sum(map(int, list(str(2**1000))))
1366
```

Encoding

UTF-8 encoding/UNICODE

In terms of UTF-8/UNICODE what is wrong with the following code:

https://raw.github.com/gist/1035399

Hint look at the word “naïve”.

Make a correction.

See also:

http://finnaarupnielsen.wordpress.com/2011/06/20/simplest-sentiment-analysis-in-python-with-af/

UTF-8 encoding/UNICODE

Translate the AFINN sentiment word list with a language translation web service, — or perhaps just a part it — to a language you know and see if it works with with a couple of sentences.

Numerical python

File reading and simple computing

Consider a file with the following matrix $\mathbf{X}$:

```
1 2
3 4
```

Read and compute $\mathbf{Y} = 2 * \mathbf{X}$ now with NumPy!

Matrix rank

Compute the rank of the `array`:

```
>>> from numpy import *
>>> A = array([[1, 0], [0, 0]])
>>> rank(A)
2
```

Hmmmm ??? Not this one.

Find the matrix rank by computing the number of numerical non-zero singular values

Function header:

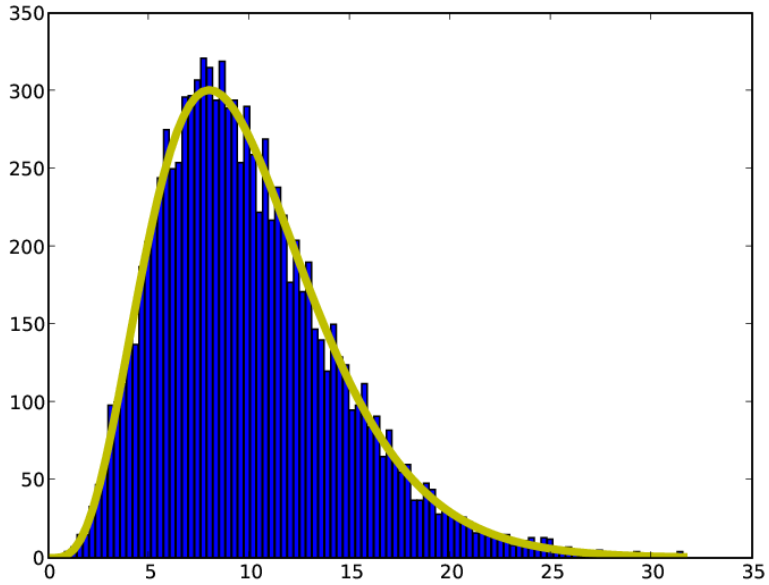
```
def matrixrank(A, tol=None):
    """
    Computes the matrix rank
    >>> matrixrank(array([[1, 0], [0, 0]]))
    1
    """
```

Hint: use the `svd` function in `numpy.linalg`.

Statistical distributions

Generate 10'000 sets with 10 Gaussian distributed samples, square each element and sum over the 10 samples. Plot the histogram of the 10'000 sums together with the teoretically curve of the probability density function.

χ^2_{10} PDF from the `pdf()` function in the `scipy.stats.chi2` class



Coauthors

Read coauthors.csv — a tab-separated file with co-author matrix. Find the author with most coauthoring.

Plot the largest connected component part of the network with `NetworkX`.

Text mining

Word and sentence segmentation

Segment the following short text into sentences and words:

```
>>> s = u"""DTU course 02820 is taught by Mr. Bartlomiej Wilkowski,
Mr. Marcin Marek Szewczyk & Finn Årup Nielsen, Ph.D. Some of aspects
of the course are: machine learning and web 2.0. The telephone to Finn
is (+45) 4525 3921, and his email is fn@imm.dtu.dk. A book
published by O'Reilly called 'Programming Collective
Intelligence' might be useful. It costs $39.99 or 285.00 kroner in
Polyteknisk Boghandle. Is 'Text Processing in Python' appropriate for
the course? Perhaps! The constructor function in Python is called
"__init__()". fMRI will not be a topic of the course."""
```

Try both with the `re` module as well as with a function from `nltk`.

Email mining

Change the feature set to less words or other words.

Code available here: https://gist.github.com/1226214

Web serving

Estimation web service

Create a web service that will take a series of numbers and model the data, e.g., with a linear model.

You can, e.g., use the below pointer for the class which makes the computation.

unimodeler.py

Pandas

“Assignment results” in Pandas

Read in the assignment results Excel sheets (available under File Sharing in CampusNet) with Pandas into several dataframes.

Aggregate the dataframe into one big dataframe.

Compute the correlation between the scores in “Score” columns.

Produce a table/matrix of scatter plots of the score results for the different.