Basic and advanced python modules

Elements of Programming Languages

Emidio Capriotti
http://biofold.org/

Biomolecules Folding and Disease

Department of Pharmacy and Biotechnology (FaBiT)
University of Bologna

Basic python modules

In the standard python interpreter some useful modules are made available by default.

The *os* module allows the usage of operating system-dependent functionality. The module include many functions to interact with the file system such as:

- os.getcwd()
- os.mkdir(path)
- os.remove(path)
- os.path.isfile(path)

The *sys* module in Python provides various functions and variables that are used to manipulate different parts of the Python runtime environment. Some interesting functions are:

- sys.stdout
- sys.stderr
- sys.path
- sys.argv

Read file content

In a python script the python interpreter is define in the first line with the shebang. To read the content of a file a file handler (fh) is defined. The average of numbers extracted from the column of a file can be calculated dividing the sum of the elements by their number.

```
#!/usr/bin/env python3
import sys

def read_values(filename,ncol):
   nlist=[]
   fh=open(filename,'r')
   for line in fh:
      num=float(line.split()[ncol])
      nlist.append(num)
   return nlist

if __name__ == ‘__main__’:
   filename=sys.argv[1]
   ncol=int(sys.argv[2])-1
   nlist=read_values(filename,ncol)
   print (‘Average:’,sum(nlist)/len(nlist))
```

Numpy

The numpy module can be used to generate an array which can be manipulated to calculate the average and other statistical values

```
#!/usr/bin/env python3
import sys
import numpy as np

def read_values(filename,ncol):
   nlist=[]
   fh=open(filename,'r')
   for line in fh:
      num=float(line.split()[ncol])
      nlist.append(num)
   return np.array(nlist)

if __name__ == ‘__main__’:
   filename=sys.argv[1]
   ncol=int(sys.argv[2])-1
   nlist=read_values(filename,ncol)
   print (‘Average:’,np.mean(nlist),’STD:’,np.std(nlist))
```

Running on the shell

If we consider a *purchase.tsv* file contacting there columns reporting the name, the number of items and the total amount of the purchase, the average amount of the purchase can be calculated running the previous script (*average.py*).

The output of the program can be redirected to a the file average.txt

```
emidio@S968-01D20-W01:~$ cat purchase.tsv
Emidio  1       12.0
Paola   2       20.0
Piero   4       30.0
Kashaf  1       5.0
Young   4       40.0
Emidio  2       20.0
Piero   3       20.5
Nicola  4       30.0

emidio@S968-01D20-W01:~$ python3 average.py purchase.tsv 3 > average.txt

emidio@S968-01D20-W01:~$ cat average.txt
Average: 22.1875 STD: 10.313636301033695
```

Running on the shell

If we consider a *purchase.tsv* file contacting there columns reporting the name, the number of items and the total amount of the purchase, the average amount of the purchase can be calculated running the previous script (*average.py*).

The output of the program can be redirected to a the file average.txt

```
emidio@S968-01D20-W01:~$ cat purchase.tsv
Emidio  1       12.0
Paola   2       20.0
Piero   4       30.0
Kashaf  1       5.0
Young   4       40.0
Emidio  2       20.0
Piero   3       20.5
Nicola  4       30.0

emidio@S968-01D20-W01:~$ python3 average.py purchase.tsv 3 > average.txt

emidio@S968-01D20-W01:~$ cat average.txt
Average: 22.1875 STD: 10.313636301033695
```

Statistics with SciPy

SciPy provides algorithms for solving different class of mathematical problems. The scipy.stats has a *linregress* function that calculates regression between two variables

```
#!/usr/bin/env python3
import sys
import numpy as np
from scipy.stats import linregress

def read_values(filename,xcol,ycol):
     xlist=[]
     ylist=[]
     fh=open(filename,'r')
     for line in fh:
             values=line.split()
             x=float(values[xcol])
             y=float(values[ycol])
             xlist.append(x)
             ylist.append(y)
     return np.array(xlist),np.array(ylist)

if __name__ == '__main__':
     filename=sys.argv[1]
     xcol=int(sys.argv[2])-1
     ycol=int(sys.argv[3])-1
     xlist,ylist=read_values(filename,xcol,ycol)
     fit=linregress(xlist,ylist)
     print (‘slope:',fit[0],'intercept:',fit[1],'r-value',fit[2],'p-value',fit[3])
```

Try and except

If in the purchase.tsv file a number of the items associated to a purchase is changed by mistake with a letter, the conversion to float will result in an error.

```
def read_values(filename,xcol,ycol):
     xlist=[]
     ylist=[]
     fh=open(filename,'r')
     for line in fh:
             values=line.split()
             try:
                     x=float(values[xcol])
                     y=float(values[ycol])
                     xlist.append(x)
                     ylist.append(y)
             except:
                     sys.stderr.write('ERROR: Incorrect line: '+line)
     return np.array(xlist),np.array(ylist)
```

Errors and stderr

Errors can be printed on the stderr using *sys.stderr* and can be redirected to a log file through the appropriate redirection symbol (2>).

```
emidio@S968-01D20-W01:~$ cat purchase.tsv
Emidio  1       12.0
Paola   2       20.0
Piero   4       30.0
Kashaf  1       5.0
Young   A       40.0
Emidio  2       20.0
Piero   3       20.5
Nicola  4       30.0

emidio@S968-01D20-W01:~$ python3 regression.py purchase.tsv 2 3 2> average.log
('slope:', 6.647058823529411, 'intercept:', 3.5000000000000036, 'r-value',
0.9374890122876628, 'p-value', 0.0018140272855587877)

emidio@S968-01D20-W01:~$ cat average.log
ERROR: Incorrect line: Young   A    40.0
```

Errors and stderr

Errors can be printed on the stderr using *sys.stderr* and can be redirected to a log file through the appropriate redirection symbol (2>).

```
emidio@S968-01D20-W01:~$ cat purchase.tsv
Emidio  1       12.0
Paola   2       20.0
Piero   4       30.0
Kashaf  1       5.0
Young   A       40.0
Emidio  2       20.0
Piero   3       20.5
Nicola  4       30.0

emidio@S968-01D20-W01:~$ python3 regression.py purchase.tsv 2 3 2> average.log
('slope:', 6.647058823529411, 'intercept:', 3.5000000000000036, 'r-value',
0.9374890122876628, 'p-value', 0.0018140272855587877)

emidio@S968-01D20-W01:~$ cat average.log
ERROR: Incorrect line: Young   A    40.0
```