Tips to process large data sets (and some ideas to develop simple simulations)

Cecilia Jarne

cecilia.jarne@unq.edu.ar

IFIBA U B A CONICET

CONICET

Universidad Nacional de Quilmes

WTPC Workshop en técnicas de programación científica

About data analysis

Summary:

- The basic use of the Terminal
- Bash Shell Scripting
- Python Scripts (looping over data sets)
- Where ask for help?

WTPC workshop en técnicas de programación científica

The basic use of the Terminal

Let's start with a basic example: When you need to find your own file

```
>> find ~/wtpc -name '*.pdf'
/home/jarne/wtpc/cuader/cuadernillo.pdf
/home/jarne/wtpc/charlas/architecture.pdf
/home/jarne/wtpc/charlas/optimization.pdf
/home/jarne/wtpc/charlas/linking_compiled.pdf
/home/jarne/wtpc/charlas/linking_python.pdf
/home/jarne/wtpc/charlas/documentation.pdf
/home/jarne/wtpc/charlas/licenses.pdf
/home/jarne/wtpc/charlas/debug_profile.pdf
```

WTPC workshop en técnicas de programación científica

The basic use of the Terminal

Looking inside a file content and print it on screen:

```
>> cat students.txt
surname, name
Alcain, Pablo
Dolina, Alejandro
Singer, Paul
```

The basic use of the Terminal

Also you can use the command more:

```
>> more cuadernillo.tex
\documentclass[10pt]{article}
\usepackage[utf8]{inputenc}
\usepackage[inner=0.5in, outer=1in]{geometry}
\begin{document}
```

The basic use of the Terminal

Sending the result to a new file:

```
>> find ~/wtpc -name '*.pdf' > file_list
```

Or append it in an existing file:

```
>> find ~/wtpc -name '*.pdf' >> lista_archivos
```

WTPC workshop en técnicas de programación científica

The basic use of the Terminal

When you execute ls you will see the file that you created:

```
>> find ~/wtpc -name '*.pdf' > file_list
>> ls
file_list
```

The basic use of the Terminal

```
>> find ~/wtpc -name '*.pdf' > file_list
>> ls
file_list
>> cat file_list
wtpc/cuader/cuadernillo.pdf
wtpc/charlas/architecture.pdf
wtpc/charlas/optimization.pdf
wtpc/charlas/linking_compiled.pdf
wtpc/charlas/linking_python.pdf
wtpc/charlas/documentation.pdf
wtpc/charlas/licenses.pdf
wtpc/charlas/debug_profile.pdf
```

WTPC
Workshop en técnicas de programación científica

The basic use of the Terminal

You can find the lines having certain characters:

```
>>grep linking file_list
/home/jarne/wtpc/charlas/linking_compiled.pdf
/home/jarne/wtpc/charlas/linking_python.pdf
```

Also it is possible to perform the task :

```
>> grep -e < as regular expression >
>> grep -r < recursive: in a directory >
```

The basic use of the Terminal

Other example changing recursively some text inside a file:

```
>> sed 's/palcain/pablohe/g' file_list
/home/jarne/wtpc/cuader/cuadernillo.pdf
/home/jarne/wtpc/charlas/architecture.pdf
/home/jarne/wtpc/charlas/optimization.pdf
/home/jarne/wtpc/charlas/linking_compiled.pdf
/home/jarne/wtpc/charlas/linking_python.pdf
/home/jarne/wtpc/charlas/documentation.pdf
/home/jarne/wtpc/charlas/licenses.pdf
/home/jarne/wtpc/charlas/debug_profile.pdf
```

The command modify the file 'in place'

```
>> sed -i modifies
```

WTPC workshop en técnicas de programación científica

Bash Shell Scripting

Classical programming example:

```
$ cat hello_world
echo 'Hello, world!'
$ bash hello_world
Hello, world!
```

Other example:

```
$ cat hello_name
echo 'Hello, my name is' $1
$ bash hello_world cecilia
Hello, my name is cecilia
```

WTPC workshop en técnicas de programación científica

Bash Shell Scripting

One more:

```
$ cat count
if [ $1 -le 10 ]
then echo $(seq 0 $1)
else echo 'Sorry, I can count upto 10 only'
fi
$ bash count 9
0 1 2 3 4 5 6 7 8 9
$ bash count 31
Sorry, I can count upto 10 only
```

Sometimes:

```
>> ./count
bash: ./count: Permission denied
```

The solution is change the permissions on file

To see the permissions:

```
$ ./count
bash: ./count: Permission denied
$ ls -l count
-rw-r--r-- 1 cecilia cecilia ... hello_world
```

To change them:

```
$ ./count
bash: ./count: Permission denied
$ ls -l count
-rw-r--r-- 1  cecilia cecilia ... hello_world
$ chmod u+x count
-rwxr--r-- 1  cecilia cecilia ... hello_world
$ ./count 4
0 1 2 3 4
```

Bash Shell Scripting

You may find in some scripts:
The exclamation mark (!) is affectionately called "bang". The shell comment symbol ("#") is sometimes called "hash".

```
#!/bin/bash
```

Bash Shell Scripting:

Other example: This was a code I used to copy Files in cluster to my scratch:

```
cd /scratch/jarne
mkdir en20.500
cd en20.500
mkdir th32.000
cd th32.000
Sget -b -v '/Simulations/libraries/en2
0.500/th32.000/*/*/DAT*.small.tar'
```

This other example I used to Extract text from files (bash and awk)

```
#!/bin/sh
# extract text from files
echo Fecha,offset,slope >listado__.txt
     for i in $( find . -type f -name '*.job.out' );
     do
  DA=`cat $i | awk -F 'VERTICAL DEPTH' '{print $2 }' | sed '/^$/d'`
  echo $DA >>listado__.txt
    done
```

Basic Python scripting

Some ideas to create python scripts:

Basic Python scripting

An example: Loop over files in sub-directory (opening .wav files):

```
1  import os
2  import scipy
3  r_dir = 'home/mydir'
4  for root, sub, files in os.walk(r_dir):
5      files = sorted(files)
6      for f in files:
7          w     = scipy.io.wavfile.read(os.path.join(root, f))
8          print (r_dir)
9          base=os.path.basename(f)
10         print (base)
11         dir = os.path.dirname(base)
12         if not os.path.exists(dir):
13            os.mkdir('plots/'+base)
14         print('-------------------------')
```

WTPC Workshop en técnicas de programación científica

Basic Python scripting

Other example: only work with one type of file (.txt)

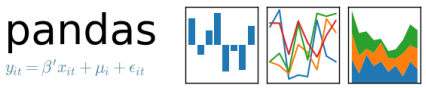
```
r_dir = 'home/mydir'
for root, dirs, files in os.walk(r_dir)
    for file in files:
        if file.endswith('.txt'):
            print(os.path.join(root, file))
            fname    = os.path.join(root, file)
            base     = os.path.basename(file)
            name_part= base[0:5]
```

WTPC Workshop en técnicas de programación científica

Python data handle libraries: Pandas (Python package)

Some ideas on Python data handle libraries

Pandas (Python package)



`http://pandas.pydata.org/`

Fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data.

- Tabular data with heterogeneously-typed columns, as in an SQL table or Excel spreadsheet.
- Ordered and unordered (not necessarily fixed-frequency) time series data.
- Arbitrary matrix data (homogeneously typed or heterogeneous) with row and column labels.
- Any other form of observational / statistical data sets.
- Primary data structures of pandas, Series (1-dimensional) and DataFrame (2-dimensional)

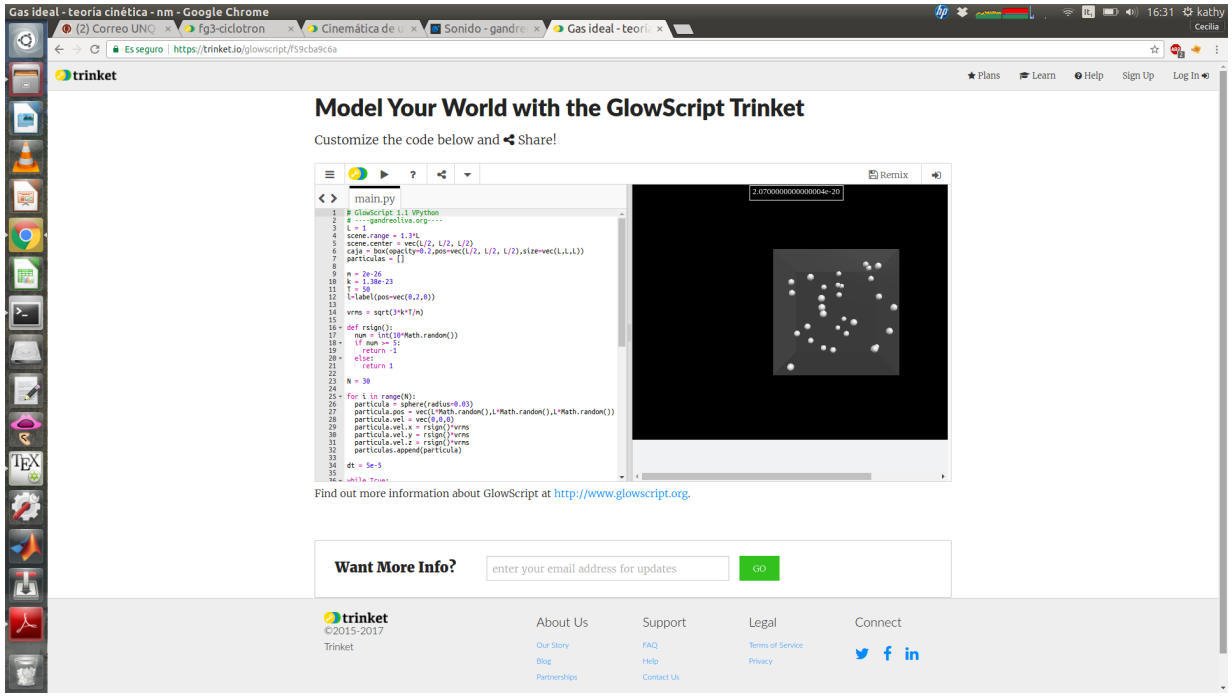
WTPC workshop en técnicas de programación científica

Python Data handle libraries

`http://scikit-learn.org/stable/`

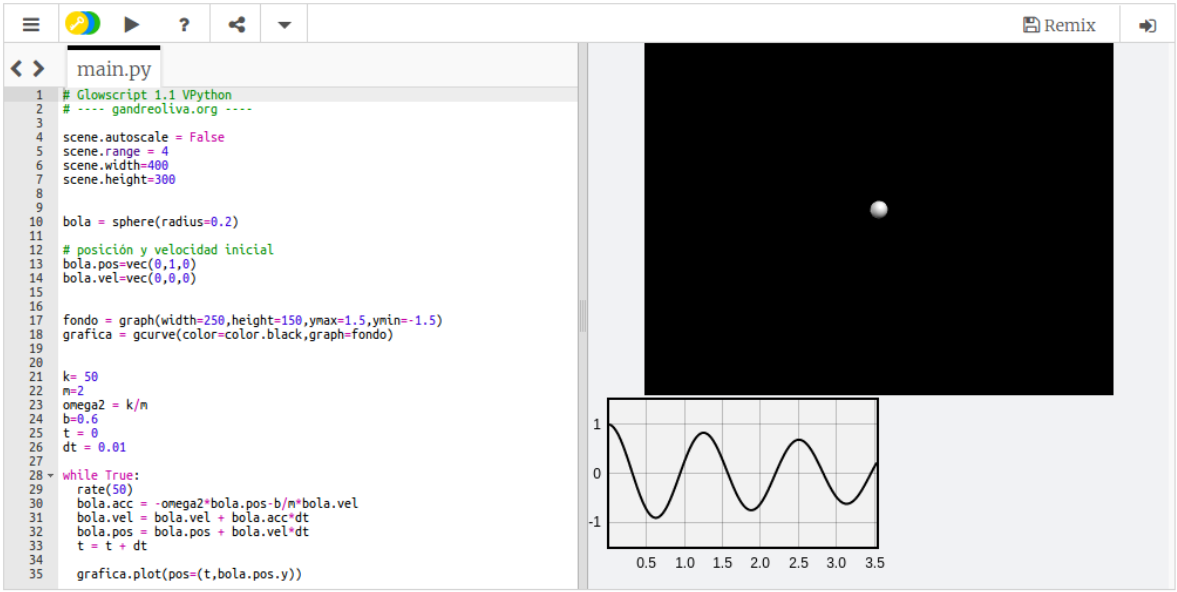
- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Python Simulation Ideas
(From André Oliva Universidad de Costa Rica.)



https://trinket.io/glowscript/f59cba9c6a

Basic Python scripting



Find out more information about GlowScript at http://www.glowscript.org.

https://trinket.io/glowscript/8d527c7d16

Other:

http://gandreoliva.org/cursos/fg2/simulacfg2.php

WTPC
workshop en técnicas de programación científica

Data analysis

- `https://stackoverflow.com/`
- Scikit learn:
  `http://scikit-learn.org`
- Pandas
  `http://pandas.pydata.org/`

Credits:

WTPC group (especially Pablo Alcain) http://wtpc.github.io Andre Oliva website `http://gandreoliva.org`