Programming with Python

EOAS Software Carpentry Workshop

September 24nd, 2015

Getting started

For our Python introduction we're going to pretend to be a researcher named Harold Bridge (user id `hbridge`) who is studying inflammation in patients who have been given a new treatment for arthritis.

You need to download some files to follow this lesson:

1. Make a new folder in your Desktop called `python-novice-inflammation`.
2. Download python-novice-inflammation-data.zip and move the file to this folder.
3. If it's not unzipped yet, double-click on it to unzip it. You should end up with a new folder called data.
4. You can access this folder from the Unix shell with:

```
$ cd && cd Desktop/python-novice-inflammation/data
```

Launching Ipython (Jupyter) Notebook

There are several ways that we can use Python. We’re going to start with a tool called Python Notebook that runs in the browser. In a shell window enter these commands:

```
$ cd
$ cd Desktop/python-novice-inflammation/data
$ ipython notebook
```

The shell window is now running a local web server for you. Don’t close it. You will need to open another shell window to do other command line things. Your browser should open to an ”Jupyter: Notebook” page showing a list of directories.

Analyzing patient data

1. Explain what a library is, and what libraries are used for.
2. Load a Python library and use the things it contains.
3. Read tabular data from a file into a program.
4. Assign values to variables.
5. Select individual values and subsections from data.
6. Perform operations on arrays of data.
7. Display simple graphs.

What does the following program print out?

```
first, second = Grace’ , ‘Hopper’
third, fourth = second, first
print (third, fourth)
```

What does the following program print out?

```
first, second = ‘Grace’ , ‘Hopper’
third, fourth = second, first
print (third, fourth)
```

ANSWER:

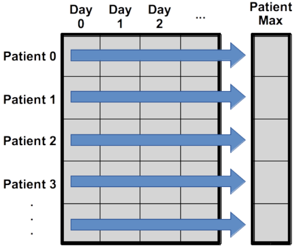
```
Hopper, Grace
```

A subsection of an array is called a slice. We can take slices of character strings as well:

```
element = ‘oxygen’
print (‘first three characters:’, element[0:3])
print (‘last three characters:’, element[3:6])
```

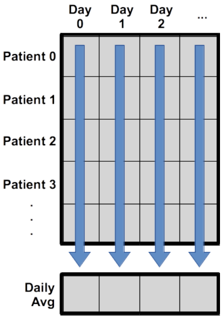
1. What is the value of element[:4]? What about element[4:]? Or element[:]?
2. What is element[-1]? What is element[-2]? Given those answers, explain what element[1:-1] does.
3. The expression element[3:3] produces an empty string, i.e., a string that contains no characters. If data holds our array of patient data, what does data[3:3, 4:4] produce? What about data[3:3, :]?

Operations across an axis



Max for each patient

`data.max(axis=1)`



Average for each day

`data.mean(axis=0)`

1. Why do all of our plots stop just short of the upper end of our graph? Why are the vertical lines in our plot of the minimum inflammation per day not vertical?
2. Create a plot showing the standard deviation of the inflammation data for each day across all patients.

Modify the program to display the three plots on top of one another instead of side by side.

Python has a built-in function called range that creates a list of numbers: range(3) produces [0, 1, 2], range(2, 5) produces [2, 3, 4], and range(2, 10, 3) produces [2, 5, 8]. Using range, write a loop that prints the first three natural numbers:

```
1
2
3
```

Python has a built-in function called range that creates a list of numbers: range(3) produces [0, 1, 2], range(2, 5) produces [2, 3, 4], and range(2, 10, 3) produces [2, 5, 8]. Using range, write a loop that prints the first three natural numbers:

One solution:

```
for num in range(1,4,1):
 print(num)
```

Exponentiation is built into Python:

```
print(5**3)
125
```

Write a loop that calculates the same result using multiplication (without exponentiation).

Exponentiation is built into Python:

```
print(5**3)
125
```

Write a loop that calculates the same result using multiplication (without exponentiation)

One possible answer:

```
ans=1
for ii in range(1,4,1):
 ans=ans*5
print(ans)
```