

CS 109A, AC 209A, STAT 121A: Data Science

Lab #0: Python Cheatsheet

Fall, 2016

A list of other tutorials:

1. `numpy`:
<http://www.scipy-lectures.org/intro/numpy/operations.html>
2. `beautiful soup`:
http://web.stanford.edu/~zlotnick/TextAsData/Web_Scraping_with_Beautiful_Soup.html
3. `matplotlib`:
<http://www.labri.fr/perso/nrougier/teaching/matplotlib/>
4. `python tutorial/cheatsheet`:
https://realpython.com/files/python_cheat_sheet_v1.pdf
http://www.astro.up.pt/~sousasag/Python_For_Astronomers/Python_qr.pdf

1 Strings

	So you want to...	Code
Get substrings	Get n -th character in a string	<code>string[n]</code>
	Get the characters between position n and position m (not including m) in a string	<code>string[n:m]</code>
	Get first n characters (not including n) in a string	<code>string[:n]</code>
	Get the last character in a string	<code>string[-1]</code>
	Get the last n characters in a string	<code>string[-n:]</code>
Find/count substrings	Find the first position where a substring occurs (-1 if substring doesn't exist in string)	<code>string.find(substr)</code>
	Count the number of times a substring occurs	<code>string.count(substr)</code>
Concatenate	Concatenate two strings	<code>new_string = string_1 + string_2</code>
	Concatenate a list of strings	<code>new_string = ''.join(list_of_strings)</code>
	Concatenate a list of strings, separated by a space	<code>new_string = ' '.join(list_of_strings)</code>
Split	Split a string into pieces based on some substring delimiter (delimiter not included in pieces)	<code>list_of_substrs = string.split(delim)</code>
Modify	Change the character at the n -th position to, say, 'Z'	<code>string = string[:n] + 'Z' + string[n+1:]</code>
	Replace all occurrences of substring A in a string with substring B	<code>string = string.replace(substr_A, substr_B)</code>
	Delete all occurrences of substring A	<code>string = string.replace(substr_A, '')</code>

2 Looping

So you want to...	Code
Repeat for n number of times	<pre>for i in range(n): #do some stuff</pre>
Do something for each element in a list	<pre>for item in my_list: #do some stuff to item</pre>
Repeat until some condition (logical statement with true false value) is no longer met	<pre>while cond_A: #do some stuff</pre>

3 Lists

So you want to...	Code
Build a list of integers from n to m (not including m) counting by k numbers	<pre>range(n, m, k)</pre>
Get the length of a list	<pre>len(my_list)</pre>
Build a list from scratch	<pre>my_list = [] my_list.append(new_item_1) ...</pre>
Getting first, last, or which ever in between items	<i>(exactly like working with strings)</i>
Modify the n -th item in the list	<pre>my_list[n] = new_item</pre>
Concatenate two lists	<pre>new_list = list_1 + list_2</pre>
Filter the list for items that satisfy some condition	Ex: <pre>sub_list = [] for item in my_list: if item < 1: sub_list.append(item)</pre>
Filter the list for items that satisfy some condition (list comprehension style)	Ex: <pre>sub_list = [item for item in my_list if item < 1]</pre>

4 numpy

	So you want to...	Code
Make an array	Make an (2D) array from a list of (list of) values	<code>np.array(my_list)</code>
	Load tabular data from a file, whose values are separated by some character delimiter, into any array	<code>np.loadtxt(file_name, delimiter=delim_char, [optional things, like skip rows or columns])</code>
Compute with Arrays (1D)	Sum up the values in a array	<code>my_array.sum()</code>
	Sum up the values in a array	<code>my_array.sum()</code>
	Return the minimum value in an array	<code>my_array.min()</code>
	Return the mean of values in an array	<code>my_array.mean()</code>
Get Array Values (2D)	Get the value at the n -th row and m -th column	<code>my_array[n, m]</code>
	Get all values between rows n_1 and n_2 (not including n_2) and columns m_1 and m_2 (not including m_2 as a 2D array	<code>my_array[n_1:n_2, m_1:m_2]</code>
	Get the values in the n -row as an 1D array	<code>my_array[n, :]</code>
	Get the values in the m -column as an 1D array	<code>my_array[:, m]</code>
Filter the Array (2D)	Get values satisfying some condition	Ex: <code>my_array[my_array < 1]</code>
	Get rows with entries that satisfy some condition	Ex: <code>my_array[my_array[:, 0] < 1]</code>
	Get values that satisfy multiple conditions	Ex: <code>my_array[(my_array[:, 0] < 1) & (my_array[:, 0] > 0)]</code>
	Get values that satisfy <i>one of</i> multiple conditions	Ex: <code>my_array[(my_array[:, 0] < 1) (my_array[:, 0] > 0)]</code>

5 matplotlib

So you want to...	Code
Make a grid with n rows and m columns of subplots	<pre>fig = plt.figure() axes_1 = fig.add_subplot(n, m, 1) #do some plotting on the first axes... axes_2 = fig.add_subplot(n, m, 2) #do some plotting on the second axes... ...</pre>
Add a 3D subplot	<pre>axes_k = fig.add_subplot(n, m, k, projection='3d')</pre>
Adding a scatter plot to a set of axes	<pre>axes.scatter(x_array, y_array)</pre>
Adding a curve (given by a set of coordinates on the curve) to a set of axes	<pre>axes.plot(x_array, y_array)</pre>
Adding a histogram of an 1D array of values to a set of axes	<pre>axes.hist(value_array)</pre>
Adding color to your graph	<pre>axes.scatter(x_array, y_array, c='red')</pre>

6 beautifulsoup

So you want to...	Code
Read an html file at some URL	<pre>page = urllib.urlopen("some_url").read()</pre>
Parse an html file (turn it into parse tree)	<pre>soup = BeautifulSoup(page, "lxml")</pre>
Turn the parse tree into a formatted string	<pre>soup.prettify()</pre>
Get the first instance of a tag in the parse tree	<pre>soup.tag</pre>
Get all the text displayed on a page	<pre>soup.get_text()</pre>
Get the tag called “child_tag” from its parent	<pre>parent.child_tag</pre>
Get all the content from a tag	<pre>tag.contents</pre>
Get the string from a tag	<pre>tag.string</pre>
Get a “list” of all child tags of a tag	<pre>tag.children</pre>
Get all tags named “tag”	<pre>soup.find_all('tag')</pre>