Ph 21.1 – Strings and Webs

-v20150331-

Overview: Manipulation of Strings and Grabbing Data off the Web

One of the strengths of python is its capability for manipulation of strings. While not a numerical technique, string manipulation is a very useful tool for data reduction and analysis. This assignment deals with grabbing data from a web site and putting it into a form that can be manipulated quantitatively. Scientific data is broadly available on the web and python has a plethora of tools and modules to extract data. Broadly, we can distinguish 4 categories of web data:

- Formal relational databases (accesible e.g. via SQL)
- Standardized data formats with well-defined metadata and tags that allow efficient data retrieval
- Data in xml or html format
- Formatted plain text data (e.g. ".csv" files – "comma separated values")

Python tools and modules exist for accessing data at any of these levels. This assignment will mostly deal with low-level access (the latter two categories) although it will also provide an example of a discipline-specific python package belonging to the 2nd category mentioned. Python also has tools both for building and querying SQL databases, however, SQL is a language in itself and beyond the scope of a "toolkit" course like the Ph20-22 sequence.

A Current Research Example: the Catalina Real-Time Transient Survey (CRTS)

Caltech is known for building some of the world's largest telescopes: the Palomar 200-inch, the twin 10-meter Keck Telescopes, and in the future possibly the Thirty Meter Telescope (TMT). These telescopes are primarily used for obtaining spectra of faint and distant objects, some from the time when the universe was just in its infancy. The telescopes typically look at a single object or a small field of objects.

With the advent of modern large-format CCD arrays, a new type of astronomy has become possible – surveys of large areas of the sky for transient and variable astronomical objects. These include stellar explosions such as supernovae, outbursts of so-called "active galactic nuclei" which contain supermassive black holes, exotic binary systems which contain a black hole, neutron star, or white dwarf, and near-earth asteroids (NEOs). Finding these requires large CCD arrays (100 mega-pixels to 1 giga-pixel) which observe several hundred to over a thousand square degrees each night.

Within the last half year, data from the Caltech Catalina Real-Time Transient Survey (CRTS) has become publicly available (url: crts.caltech.edu). Researchers are using CRTS data to discover a broad variety of new astronomical objects. In this assignment, we will attempt to extract data from the CRTS web site so that it can be later analyzed with python routines.

Python Tools for Web Access and String Manipulation

In this assignment you will learn about some of the following python modules/tools:

- *urlilb* and *urllib2*
- The *string* module
  - *string.split*, *string.strip*, *string.rstrip*, and *string.lstrip*
- XML and HTML parsers
  - *HTMLParser*
  - *xml.etree.ElementTree*
  - *BeautifulSoup*
- *astropy*
- *vo.table*
- *AtPy*

and more. Look at the documentation pages for these and become familiar with how to get information about them. The first task will be to go to a web site and instead of manually entering text and clicking buttons, use python to achieve the desired response. That response includes generating some html or xml output that contains the desired data. The next task is to extract the data from the html/xml file. You will be asked to do this in two different ways: (1) a low-level approach in which you will use low-level python tools to parse the html/xml file and extract the data, and (2) a higher-level approach in which you will use a standard package to find and extract the data

It needs to be emphasized that there are many ways to access a given web-accessible data set. Whether to use low-level tools, or high-level tools is a matter of choice, flexibility, and efficiency. However, you should be capable of using either low-level or high-level tools.

Being familiar with low-level string manipulation tools will have significant payoff in many applications beyond just web data access.

The Assignment - Part I

Read the assignment all the way through to get a feel for the entire assignment.

1. Choose your target data set. The default data set is the CRTS data set, with access starting at:

   `http://nesssi.cacr.caltech.edu/cgi-bin/getcssconedbid_release2.cgi`

   If you would like to use an alternate web data set as your target, please discuss your proposal with the TA to determine whether it is a suitable alternative.

   Assuming that you are using the CRTS data, explore the data by hand. Look at some of the advanced parameter options. Type in the Name of a system (e.g. Her X-1 is a famous binary system containing a neutron star). Search for a period in the data (the companion is orbiting the neutron star with 1.7 day period, and the neutron star is stripping material off the companion – see, e.g., *http://en.wikipedia.org/wiki/X-ray_binary*). The data you will be interested in is the intensity of the source versus time. Don't worry too much about the

units at this point. The intensity units are logarithmic (magnitudes) with higher magnitudes being less intense. The time units are in decimal days (modified Julian decimal days to be precise). Astronomers may have invented these units to confuse physicists and others.

Now, using your browser, display the page source for the page given above. It is in *cgi* format. You need not understand *cgi* exhaustively, but you will need to understand it sufficiently to do the next step.

2. Access the web page using python. One possibility is to use *urlib* and *urlib2*, but others exist as well. You may wish to read the tutorial at: *http://docs.python.org/howto/urllib2.html* . Supply a name and specify the type of data output you want by supplying values to, e.g., *urllib2.Request*. Your TA can help with suggestions. The result should be an html or xml web page with a data table in it.

3. Parse the data table, i.e. snip out the parts that you want (the data) using python string manipulations (e.g. *string.split*, *string.strip*, etc.). The end result should be a python array, or a python readable file with the data in simple plain text form. It is also useful to become familiar with the use of "regular expressions" in python (i.e. the module *re*, see *http://docs.python.org/library/re.html*).

4. Produce a python plot of the magnitude versus time, similar to one available from the web site.

The Assignment - Part II

Under the "Advanced parameters" options on the original web page, there are three output options: HTML, ASCII, and VOTable. The first two are "home-grown", while the 3rd is a standard astronomical format called VOTable. Unlike the 1st two options, VOTable is a rigorous XML-based standard for astronomy and there are several python packages that can read VOTable format (e.g. *vo.table*).

Use astropy or some other high-level package to read the VOTable and extract magnitude-time data from it. You can either generate the VOTable in python, or simply download a VOTable manually using the web page. To get you started:

```
from astropy.io.votable import parse
votable = parse("votable.xml", pedantic=False)
```

The keyword "pedantic" controls how fussy the parser should be. In this case, we want it to be lenient. Other options include html or xml parsers such as HTMParser or ElementTree (part of the python distribution).