Subject: Re: Reading and Plotting big txt. File
Posted by Conor on Thu, 02 Aug 2007 17:27:05 GMT
View Forum Message <> Reply to Message

On Aug 2, 12:55 pm, Conor <cmanc...@gmail.com> wrote:
> The problem is your format statement. What's going on is that with a
> format, IDL doesn't actually read columns. It is more of directions
> where to find the data. In your case, you aren't telling it where the
> spaces are, so it assumes that everything is a data column. If you
> specify 10(a4), it is really reading:
>
> aaaabbbbccccddddeeeffffgggghhhhiiiijjjj
>
> where aaaa = column1, bbbb = column2, etc...
>
> You need to give it the appropriate number of spaces, otherwise the
> data get's all messed up. For example, apply the above "filter" to
> the data below (from your file)
>
> 7 -1848 -1792 -1718 -1678 -1638 -1576 -1517
> -1446 -1372 -1322
>
> The first four columns ' 7 ' are assigned to the first column in your
> data array. The second four columns ' ' go to the second column in
> your data array, etc.. In the end you get:
>
> data = [ 7 ',' ',' -1','848 ',' -17','92 ',' -17','18 ',' -16']
>
> (or something along those lines, anyway)
>
> What you need to do is actually specify where the spaces are:
>
> format = '(a2, 7x, a4, 2x, a4, 7( 3x, a4 ) )'
>
> I don't think that's quite it, but it probably needs to be something
> along those lines. I can't quite get it to work myself,
> unfortunately. I wish someone better informed about formats would
> join in the conversation here...

Okay, here's a solution. I didn't want to have to go here, because it is possibly the worst way to solve this problem, but since I can't figure out the formats and no one else has any suggestions, we'll just do it the "bad" way. It's bad because it is not a general solution (this will only work this one sort of file), it's worse because it is really slow, and it is even worse because neither of us is going to figure out what is wrong with what we've been trying. Oh well. The plan is to manually parse the file. Rather than relying on format statements, I wrote a program that reads the file in line by line and

parses it according to rules I give it.  Specifically, this program
works by telling it where each column starts and how long each column
is.  There's a couple caveats with this program.  First, it should
only read actual data - you'll have to remove the header to run this
program on it (or, you can leave the header in and add a couple
generic readf statements right after opening the file to read out the
header data before entering the main program loop).  Anyway, here's
the program, and I've tested it succesfully on the above text file.
Also, you can download the source directly here:
http://astro.ufl.edu/~cmancone/pros/parse_bigfile.pro

```
function parse_bigfile,filename

openr,lun,filename,/get_lun

st = [0,9,16,24,32,40,48,56,64,72,80]
len = [2,5,5,5,5,5,5,5,5,5,5]
num = n_elements(len)

line = ''
data = intarr(num)

l = 0
while not( eof(lun) ) do begin

 ; read in the line and see how long it is
 readf,lun,line
 data = intarr(num)
 length = strlen(line)

 for i=0,num-1 do begin
  ; if we've moved past the end of the line, we are done with this
line
  if st[i] gt length-1 or length eq 0 then break

  ; read and process the current element
  data[i] = float( strmid( line, st[i], len[i] ) )
 endfor

 ; if this is the first line, create our data result.  Otherwise, just
append the new data
 if l eq 0 then result = data else result = [[result],[data]]

 ; increment our line counter
 ++l
endwhile
```

```
close,lun
free_lun,lun

return,result

end
```

Now, the biggest problem with something like this is that you have to specify where every column stars. For 1000 columns, this is not a simple task. What you will have to do is see what the repeating pattern is (hopefully there is one). So, if the above file is any indication, columns are always 5 characters long with 3 spaces in between. That means that you can initialize the start array to something like:

```
st = findgen(1000)*8
```

of course, it won't be exactly that. If I take the above file as a guide, it would be more like this:

```
st = [0,9,findgen(1000)*8 + 16]
len = fltarr(1002) + 5
```

since the first two columns don't follow the same pattern as the rest of them. Just make sure that len and st have the same number of elements in them. Also, remember that starting positions for strings are zero-indexed too, so the first text column is '0', and the tenth text column is '9', etc... Let me know how it goes.

---