Subject: Re: Reading complicated ASCII data
Posted by David Fanning on Tue, 29 Jun 2010 13:11:05 GMT
View Forum Message <> Reply to Message

Tone M R writes:

> The table of figures is actually in straight columns, a column per
> month, with a dot wherever a measurement is zero. (There are also
> blank spaces at the bottom of each table, for dates such as feb 30th.)
> I've managed to work around the headers and identify where a table
> starts, and what I wanted to do was to read the entire thing into a
> nice structure array I've prepared. However, when using READF, IDL
> stops when trying to convert a dot to a float (understandably), and I
> haven't managed to solve it with a format code. I have thought about
> using STRSPLIT and WHERE to replace them, but then I have to go one
> line at a time, and I was rather hoping to make something a little
> more elegant.
>
> Does anyone see a way around these dots?

No. :-)

Cheers,

David

P.S. Let's just say, when inelegant is the ONLY way,
it is usually elegant enough. :-)

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Reading complicated ASCII data
Posted by Andy Heaps on Tue, 29 Jun 2010 13:29:12 GMT
View Forum Message <> Reply to Message

Have you tried using something like nedit or gedit to replace the dots with NaN? In this case it looks like you'll need to search and replace on a space followed by a dot. You should then be able to read in the data as normal.

Cheers
Andy

Subject: Re: Reading complicated ASCII data
Posted by Paul Van Delst[1] on Tue, 29 Jun 2010 14:13:52 GMT
View Forum Message <> Reply to Message

Tone M R wrote:
> Does anyone see a way around these dots?

Use regular expressions to change them to "0.0". I.e. if a "." is not preceded and followed by a digit, then it becomes "0.0".

Although you could, I wouldn't do the above "preprocessing" in IDL. A scripting language like ruby/python/perl would be the go; e.g.

```
#!/usr/bin/env ruby
# Define regular expression for search
re = %r{\s\.\s}
# Inplace edit the file
ARGF.each do |line|
  line.gsub!(re,"0.0")
  puts(line)
end
```

I created a file of text from your example containing:

```
 ------------------------------------------------------------ -----
[block of not-so-interesting information]

   Date   Jan   Feb   Mar   Apr  May   Jun   Jul
     1   0.5   1.4     .   4.7    .     .   0.1
     2   0.6   0.3   3.9     .    .     .     .
     3   5.8   1.6   4.9   0.1  3.1   3.4   4.4
     4   2.0   5.1   1.9   0.2  0.5   6.7   3.3
     5   6.8   0.6   9.7     .  2.7   0.8   1.6
... and so forth, for an entire year. - a 13x31 table of floats.

[new block of non-helpful stuff]

[new block of data for another year]
 ------------------------------------------------------------ ------------
etc..., for a total of ten years.
```

ran it through the above script like so

```
  $ ruby testit.rb blah.txt
```

and got the result:

---------- -----
[block of not-so-interesting information]

Date	Jan	Feb	Mar	Apr	May	Jun	Jul
1	0.5	1.4	0.0	4.7	0.0	0.0	0.1
2	0.6	0.3	3.9	0.0	0.0	0.0	0.0
3	5.8	1.6	4.9	0.1	3.1	3.4	4.4
4	2.0	5.1	1.9	0.2	0.5	6.7	3.3
5	6.8	0.6	9.7	0.0	2.7	0.8	1.6

... and so forth, for an entire year. - a 13x31 table of floats.

[new block of non-helpful stuff]

[new block of data for another year]
---------- ----------
etc..., for a total of ten years.

So there are some spacing issues to be ironed out, but works easypeasy.

cheers,

paulv

Subject: Re: Reading complicated ASCII data
Posted by Chris W on Tue, 29 Jun 2010 18:36:38 GMT
View Forum Message <> Reply to Message

On Jun 29, 8:05 am, Tone M R <tone...@gmail.com> wrote:
> Hi!
>
> I've been racking my brains and the web for the best part of a day,
> but have not managed to find anything useful to solve my problem,
> which is this:
>
> I've got an automatically generated .txt file of rainfall measurements
> which I need to read. I'm having trouble with the format of the file,
> which looks more or less like this:
> ---------- -----
> [block of not-so-interesting information]
>
> Date Jan Feb Mar Apr May Jun Jul Aug Sep
> Oct Nov Dec
> 1 0.5 1.4 . 4.7 . .
> 0.1 . . . .
> 2 0.6 0.3 3.9 . . . . .
> 4.0 . .

> 3 5.8 1.6 4.9 0.1 3.1 3.4 4.4 0.2 0.9
> 1.4 .
> 4 2.0 5.1 1.9 0.2 0.5 6.7 3.3 . 1.1
> 0.1 .
> 5 6.8 0.6 9.7 . 2.7 0.8 1.6 2.4
> 0.7 . .
> ... and so forth, for an entire year. - a 13x31 table of floats.
>
> [new block of non-helpful stuff]
>
> [new block of data for another year]
> ---------- ----------
> etc..., for a total of ten years.
>
> The table of figures is actually in straight columns, a column per
> month, with a dot wherever a measurement is zero. (There are also
> blank spaces at the bottom of each table, for dates such as feb 30th.)
> I've managed to work around the headers and identify where a table
> starts, and what I wanted to do was to read the entire thing into a
> nice structure array I've prepared. However, when using READF, IDL
> stops when trying to convert a dot to a float (understandably), and I
> haven't managed to solve it with a format code. I have thought about
> using STRSPLIT and WHERE to replace them, but then I have to go one
> line at a time, and I was rather hoping to make something a little
> more elegant.
>
> Does anyone see a way around these dots?

How about reading the whole file into one string,
Use strsplit and split at " . " (assuming those are spaces not tabs)
then strjoin with " 0 "

Chris

Subject: Re: Reading complicated ASCII data
Posted by Tone M R on Wed, 30 Jun 2010 08:34:52 GMT
View Forum Message <> Reply to Message

On Jun 29, 8:36 pm, Chris W <cwood1...@gmail.com> wrote:
> On Jun 29, 8:05 am, Tone M R <tone...@gmail.com> wrote:
>
>
>
>
>
>
>> Hi!
>

>> I've been racking my brains and the web for the best part of a day,
>> but have not managed to find anything useful to solve my problem,
>> which is this:
>
>> I've got an automatically generated .txt file of rainfall measurements
>> which I need to read. I'm having trouble with the format of the file,
>> which looks more or less like this:
>> ---------- -----
>> [block of not-so-interesting information]
>
>> Date Jan Feb Mar Apr May Jun Jul Aug Sep
>> Oct Nov Dec
>> 1 0.5 1.4 . 4.7 . .
>> 0.1 . . . .
>> 2 0.6 0.3 3.9 . . . . .
>> 4.0 . .
>> 3 5.8 1.6 4.9 0.1 3.1 3.4 4.4 0.2 0.9
>> 1.4 .
>> 4 2.0 5.1 1.9 0.2 0.5 6.7 3.3 . 1.1
>> 0.1 .
>> 5 6.8 0.6 9.7 . 2.7 0.8 1.6 2.4
>> 0.7 . .
>> ... and so forth, for an entire year. - a 13x31 table of floats.
>
>> [new block of non-helpful stuff]
>
>> [new block of data for another year]
>> ---------- ----------
>> etc..., for a total of ten years.
>
>> The table of figures is actually in straight columns, a column per
>> month, with a dot wherever a measurement is zero. (There are also
>> blank spaces at the bottom of each table, for dates such as feb 30th.)
>> I've managed to work around the headers and identify where a table
>> starts, and what I wanted to do was to read the entire thing into a
>> nice structure array I've prepared. However, when using READF, IDL
>> stops when trying to convert a dot to a float (understandably), and I
>> haven't managed to solve it with a format code. I have thought about
>> using STRSPLIT and WHERE to replace them, but then I have to go one
>> line at a time, and I was rather hoping to make something a little
>> more elegant.
>
>> Does anyone see a way around these dots?
>
> How about reading the whole file into one string,
> Use strsplit and split at " . " (assuming those are spaces not tabs)
> then strjoin with " 0 "
>

> Chris- Hide quoted text -
>
> - Show quoted text -

Everyone, thanks a lot! Now I know which way to go, which is reassuring, even though this might get messy;)

Tone
