Subject: Reading in date & time from csv file
Posted by justinclouds on Sun, 18 May 2014 20:25:45 GMT
View Forum Message <> Reply to Message

I want to read a csv file where the first column is date and time in the format YYYY_MM_DD_HH_MM_SS (e.g. 2014_05_12_18_39_33.194). I proceed as follows:

data = READ_CSV(file, N_TABLE_HEADER=30)

time=string(data.field01[*], format='(I4, "_", I2, "_", I2, "_", I2, "_", I2, "_", I2.5)')

When I print the variable time the result I get is:
2014_**_**_**_**_**

Can anybody help with what needs to be changed here to read this correctly or suggest an alternate method.

- D

Subject: Re: Reading in date & time from csv file
Posted by Matthew Argall on Sun, 18 May 2014 23:31:21 GMT
View Forum Message <> Reply to Message

How about StrSplit?

```
IDL> tstr = '2014_05_12_18_39_33.194'
IDL> time = strsplit(tstr, '_', /EXTRACT)
IDL> help, time
TIME            STRING    = Array[6]
IDL> print, time
2014 05 12 18 39 33.194
```

You can then convert them from strings to integers, floats, etc. using the Fix, Long, Float, Double, etc. functions.

Subject: Re: Reading in date & time from csv file
Posted by justinclouds on Mon, 19 May 2014 01:40:23 GMT
View Forum Message <> Reply to Message

This would work. However, the output to:

time=string(data.field01[*])

is:

IDL> print, time[0]
      2014.0000

and the value in the csv file is 2014_05_12_18_39_33.194

So I need to fix how the data is read into 'time' vector.

Any ideas?

- D

Subject: Re: Reading in date & time from csv file
Posted by Fabzi on Mon, 19 May 2014 06:59:05 GMT
View Forum Message <> Reply to Message

On 19.05.2014 03:40, justinclouds wrote:
> This would work. However, the output to:
>
> time=string(data.field01[*])
>
> is:
>
> IDL> print, time[0]
>   2014.0000
>
> and the value in the csv file is 2014_05_12_18_39_33.194
>
> So I need to fix how the data is read into 'time' vector.
>
> Any ideas?
>
> - D

It does not make much sense. If I have a csv file which looks like this:

2014_05_12_18_39_33.194, 12.25
2014_05_13_18_39_33.194, 12.25

and I read it with read_csv i get:

IDL> data = READ_CSV('test.csv')
IDL> print, data.FIELD1[0]
2014_05_12_18_39_33.194

then I would do what mMtthew said and maybe use the julday() function

Cheers,

Fabien

Subject: Re: Reading in date & time from csv file
Posted by Matthew Argall on Mon, 19 May 2014 11:57:25 GMT
View Forum Message <> Reply to Message

What is the result when you do

IDL> help, data.field01
IDL> print, data.field01[0]

If it is not a string like '2014_05_12_18_39_33.194', then what is the exact line of the file you are reading?

You do not want to use the String function because data.field01 ought to be a string already. You should be doing

IDL> time = strsplit(data.field01, '_', /EXTRACT)

Subject: Re: Reading in date & time from csv file
Posted by Phillip Bitzer on Mon, 19 May 2014 15:27:32 GMT
View Forum Message <> Reply to Message

On Sunday, May 18, 2014 3:25:45 PM UTC-5, justinclouds wrote:
>
> Can anybody help with what needs to be changed here to read this correctly or suggest an alternate method.
>
>

So, the problem seems to be READ_CSV doesn't seem to detect the first column is a string, correct?

IDL> f = 'test.txt'
IDL> data = READ_CSV(f)

IDL> help, data
** Structure <247e2b8>, 2 tags, length=64, data length=64, refs=1:
  FIELD1 DOUBLE Array[4]
  FIELD2 DOUBLE Array[4]
IDL> print, data.field1
      2014.0000   2014.0000   2014.0000   2014.0000

Not sure why - but I don't use READ_CSV much.

Using READCOL works just fine, as I can specify the format of each column easily:
IDL> READCOL, f, date, val, FORMAT='A, F'

IDL> print, date
2014_05_12_18_39_33.194 2014_05_12_19_39_33.194 2014_05_12_20_39_33.194
2014_05_12_21_39_33.194

Then, you can use STR_SPLIT as Matt suggested...if you're on IDL 8. Otherwise, you'll have to get fancy with STREGEX.

Here's the test file I used:

2014_05_12_18_39_33.194, 1.0
2014_05_12_19_39_33.194, 2.0
2014_05_12_20_39_33.194, 3.0
2014_05_12_21_39_33.194, 4.0

Subject: Re: Reading in date & time from csv file
Posted by justinclouds on Mon, 19 May 2014 15:37:17 GMT
View Forum Message <> Reply to Message

I changed the code so that time=data.field01

IDL> help, data.field01
<Expression>   DOUBLE   = Array[157105]
IDL> print, data.field01[0]
      2014.0000

So, I did find a work around by using the readcol function from
http://idlastro.gsfc.nasa.gov/ftp/pro/misc/readcol.pro

readcol,file,time, FORMAT = 'A', SKIPLINE = 30
year = strmid(time, 0,4)
month = strmid(time, 5,2)
day = strmid(time, 8,2)
hour = strmid(time, 11,2)
minute = strmid(time, 14,2)
second = strmid(time, 17,2)

This is working. However, I still don't understand why the READ_CSV cannot do the job. I also tried READ_ASCII with the same results as READ_CSV.

- D

Subject: Re: Reading in date & time from csv file
Posted by justinclouds on Mon, 19 May 2014 15:39:00 GMT
View Forum Message <> Reply to Message

On Monday, May 19, 2014 9:27:32 AM UTC-6, Phillip Bitzer wrote:
> On Sunday, May 18, 2014 3:25:45 PM UTC-5, justinclouds wrote:
>
>>
>
>> Can anybody help with what needs to be changed here to read this correctly or suggest an alternate method.
>
>>
>
>>
>
>
>
> So, the problem seems to be READ_CSV doesn't seem to detect the first column is a string, correct?
>
>
>
> IDL> f = 'test.txt'
>
> IDL> data = READ_CSV(f)
>
>
>
> IDL> help, data
>
> ** Structure <247e2b8>, 2 tags, length=64, data length=64, refs=1:
>
> FIELD1 DOUBLE Array[4]
>
> FIELD2 DOUBLE Array[4]
>
> IDL> print, data.field1
>
> 2014.0000 2014.0000 2014.0000 2014.0000
>
>
>
> Not sure why - but I don't use READ_CSV much.
>
>
>
> Using READCOL works just fine, as I can specify the format of each column easily:
>

> IDL> READCOL, f, date, val, FORMAT='A, F'
>
>
>
> IDL> print, date
>
> 2014_05_12_18_39_33.194 2014_05_12_19_39_33.194 2014_05_12_20_39_33.194
2014_05_12_21_39_33.194
>
>
>
> Then, you can use STR_SPLIT as Matt suggested...if you're on IDL 8. Otherwise, you'll have to
get fancy with STREGEX.
>
>
>
> ----------
>
> Here's the test file I used:
>
> ----------
>
> 2014_05_12_18_39_33.194, 1.0
>
> 2014_05_12_19_39_33.194, 2.0
>
> 2014_05_12_20_39_33.194, 3.0
>
> 2014_05_12_21_39_33.194, 4.0

Thanks. I did just what you suggested and we must have been posting this at the same time!
Thanks!
