
Subject: Re: Missing Data Programming Contest
Posted by [David Fanning](#) on Wed, 08 Apr 2009 14:03:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning writes:

- > You should assume:
- >
- > 1. The data can be any data type except complex or string.
- >
- > 2. The missing value *could* be !VALUES.F_NAN.
- >
- > 3. Unsophisticated users might be using your program,
- > so, for example, they might pass in a missing value
- > such as 594.32.

In response to an overwhelming number of questions, it
is permissible to use Coyote Library routines. FLOATS_EQUAL
comes to mind. :-)

Cheers,

David

--

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: Missing Data Programming Contest
Posted by [R.Bauer](#) on Wed, 08 Apr 2009 14:21:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning schrieb:

- > Folks,
- >
- > I don't have time today to think about this, so I've
- > decided to get you to do my thinking for me. :-)
- >
- > Suppose you are expecting a data array and you suspect
- > the data has "missing values" in it that you wish to
- > exclude from further processing. You write a keyword
- > for your routine that allows the user to pass in what
- > he is using for the "missing value".
- >
- > PRO Junker, data, MISSING_VALUE=missing_value
- >

> How would you write the code to assure that this missing
> value would be excluded from further processing?
>
> You should assume:
>
> 1. The data can be any data type except complex or string.
>
> 2. The missing value *could* be !VALUES.F_NAN.
>
> 3. Unsophisticated users might be using your program,
> so, for example, they might pass in a missing value
> such as 594.32.

4. Very unsophisticated users don't care on types and passes a
missing value which can't be found. e.g. 999.9 and the data is
of type double.

>
> No, this is NOT my homework! But I do need it ASAP. ;-)
>
> Cheers,
>
> David

Subject: Re: Missing Data Programming Contest
Posted by [R.Bauer](#) on Wed, 08 Apr 2009 14:27:41 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning schrieb:

> Folks,
>
> I don't have time today to think about this, so I've
> decided to get you to do my thinking for me. :-)
>
> Suppose you are expecting a data array and you suspect
> the data has "missing values" in it that you wish to
> exclude from further processing. You write a keyword
> for your routine that allows the user to pass in what
> he is using for the "missing value".
>
> PRO Junker, data, MISSING_VALUE=missing_value
>
> How would you write the code to assure that this missing
> value would be excluded from further processing?
>
> You should assume:
>

> 1. The data can be any data type except complex or string.
>
> 2. The missing value *could* be !VALUES.F_NAN.
>
> 3. Unsophisticated users might be using your program,
> so, for example, they might pass in a missing value
> such as 594.32.
>
> No, this is NOT my homework! But I do need it ASAP. ;-)
>
> Cheers,
>
> David

That old tool (nearly 10 years) has some more parameters

http://www.fz-juelich.de/icg/icg-1/idl_icglib/idl_source/idl_work/rb_lib/def_valid_for_param.pro

example at

http://www.fz-juelich.de/icg/icg-1/idl_icglib/idl_source/idl_html/examples/def_valid_for_param_example.pro.html

cheers
Reimar

Subject: Re: Missing Data Programming Contest
Posted by [David Fanning](#) on Wed, 08 Apr 2009 18:26:26 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning writes:

> Suppose you are expecting a data array and you suspect
> the data has "missing values" in it that you wish to
> exclude from further processing. You write a keyword
> for your routine that allows the user to pass in what
> he is using for the "missing value".
>
> PRO Junker, data, MISSING_VALUE=missing_value
>
> How would you write the code to assure that this missing
> value would be excluded from further processing?
>
> You should assume:
>
> 1. The data can be any data type except complex or string.
>
> 2. The missing value *could* be !VALUES.F_NAN.
>

> 3. Unsophisticated users might be using your program,
> so, for example, they might pass in a missing value
> such as 594.32.

Humm. The only really useful suggestion I've heard so far (I forgot my German dictionary today, so I'm not sure about Reimar's contribution) is to delete the MISSING_VALUE keyword and let the user worry about it. Surely this can't be the ONLY suggestion!

Cheers,

David

--

David Fanning, Ph.D.

Coyote's Guide to IDL Programming (www.dfanning.com)

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Missing Data Programming Contest
Posted by [wlandsman](#) on Wed, 08 Apr 2009 19:11:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

> Humm. The only really useful suggestion I've heard so
> far (I forgot my German dictionary today, so I'm not
> sure about Reimar's contribution) is to delete the
> MISSING_VALUE keyword and let the user worry about
> it. Surely this can't be the ONLY suggestion!
>

Well, I'll give the naive answer because I am not certain what the problem is. If the routine contains a MISSING = missing keyword, then you can decide whether the user wants a MISSING value or NAN. (Presumably a missing value rather than NAN is only useful for integer data.)

```
do_missing = 0
do_nan = 0
if N_elements(missing) GT 0 then $
    if finite(nan) then do_missing = 1 else do_nan = 1
```

and then you can process data accordingly. For example, to remove missing values from a variable x

```
if do_missing then begin
    g = where(x NE missing,Ng) ;or use FLOATS_EQUAL for testing
    if Ng GT 0 then x = x[g]
endif else if do_nan then x = x[where(finite(x))]
```

I suspect I am missing something in your question. Or are you just looking for more speed or elegance? --Wayne

Subject: Re: Missing Data Programming Contest
Posted by [Homeyer](#) on Wed, 08 Apr 2009 23:02:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Apr 8, 8:23 am, David Fanning <n...@dfanning.com> wrote:

> Folks,
>
> I don't have time today to think about this, so I've
> decided to get you to do my thinking for me. :-)
>
> Suppose you are expecting a data array and you suspect
> the data has "missing values" in it that you wish to
> exclude from further processing. You write a keyword
> for your routine that allows the user to pass in what
> he is using for the "missing value".
>
> PRO Junker, data, MISSING_VALUE=missing_value
>
> How would you write the code to assure that this missing
> value would be excluded from further processing?
>
> You should assume:
>
> 1. The data can be any data type except complex or string.
>
> 2. The missing value **could** be !VALUES.F_NAN.
>
> 3. Unsophisticated users might be using your program,
> so, for example, they might pass in a missing value
> such as 594.32.
>
> No, this is NOT my homework! But I do need it ASAP. ;-)
>
> Cheers,
>
> David
> --
> 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.")

The following should suffice:

```

IF (missing_value GT 0) THEN BEGIN
  type = SIZE(array, /TNAME)

  CASE type OF
    'COMPLEX' : MESSAGE, 'Complex values not allowed for removing
MISSING data.'
    'STRING' : MESSAGE, 'String values not allowed for removing
MISSING data.'
    ELSE      : invalid = WHERE((array EQ missing_value), iv_count,
COMPLEMENT = valid, NCOMPLEMENT = v_count)
  ENDCASE

  IF (v_count GT 0) THEN array = array[valid]
ENDIF

```

That is, if you dont care about overwriting the original data or it is not gridded. If it is gridded (data points with missing values should remain), then you could replace those with NaNs in the final array using the indices above.

Cheers,
Cameron Homeyer

Subject: Re: Missing Data Programming Contest
 Posted by [Homeyer](#) on Wed, 08 Apr 2009 23:26:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Apr 8, 6:02 pm, Homeyer <cam.the.weather....@gmail.com> wrote:
 > On Apr 8, 8:23 am, David Fanning <n...@dfanning.com> wrote:
 >
 >
 >
 >> Folks,
 >
 >> I don't have time today to think about this, so I've
 >> decided to get you to do my thinking for me. :-)
 >
 >> Suppose you are expecting a data array and you suspect
 >> the data has "missing values" in it that you wish to
 >> exclude from further processing. You write a keyword
 >> for your routine that allows the user to pass in what
 >> he is using for the "missing value".
 >
 >> PRO Junker, data, MISSING_VALUE=missing_value
 >
 >> How would you write the code to assure that this missing

```

>> value would be excluded from further processing?
>
>> You should assume:
>
>> 1. The data can be any data type except complex or string.
>
>> 2. The missing value *could* be !VALUES.F_NAN.
>
>> 3. Unsophisticated users might be using your program,
>>    so, for example, they might pass in a missing value
>>    such as 594.32.
>
>> No, this is NOT my homework! But I do need it ASAP. ;-)
>
>> Cheers,
>
>> David
>> --
>> 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.")
>
> The following should suffice:
>
> IF (missing_value GT 0) THEN BEGIN
>   type = SIZE(array, /TNAME)
>
>   CASE type OF
>     'COMPLEX' : MESSAGE, 'Complex values not allowed for removing
> MISSING data.'
>     'STRING'  : MESSAGE, 'String values not allowed for removing
> MISSING data.'
>     ELSE      : invalid = WHERE((array EQ missing_value), iv_count,
> COMPLEMENT = valid, NCOMPLEMENT = v_count)
>   ENDCASE
>
>   IF (v_count GT 0) THEN array = array[valid]
> ENDIF
>
> That is, if you dont care about overwriting the original data or it is
> not gridded. If it is gridded (data points with missing values should
> remain), then you could replace those with NaNs in the final array
> using the indices above.
>
> Cheers,
> Cameron Homeyer

```

oops, missed the NaN part:

This would go before the last IF (v_count...

```
IF (~FINITE(missing)) THEN invalid = WHERE(~FINITE(array), iv_count,  
COMPLEMENT = valid, NCOMPLEMENT = v_count)
```

Subject: Re: Missing Data Programming Contest
Posted by [R.Bauer](#) on Thu, 09 Apr 2009 08:32:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

Homeyer schrieb:

> On Apr 8, 6:02 pm, Homeyer <cam.the.weather....@gmail.com> wrote:

>> On Apr 8, 8:23 am, David Fanning <n...@dfanning.com> wrote:

>>

>>

>>

>>> Folks,

>>> I don't have time today to think about this, so I've

>>> decided to get you to do my thinking for me. :-)

>>> Suppose you are expecting a data array and you suspect

>>> the data has "missing values" in it that you wish to

>>> exclude from further processing. You write a keyword

>>> for your routine that allows the user to pass in what

>>> he is using for the "missing value".

>>> PRO Junker, data, MISSING_VALUE=missing_value

>>> How would you write the code to assure that this missing

>>> value would be excluded from further processing?

>>> You should assume:

>>> 1. The data can be any data type except complex or string.

>>> 2. The missing value *could* be !VALUES.F_NAN.

>>> 3. Unsophisticated users might be using your program,

>>> so, for example, they might pass in a missing value

>>> such as 594.32.

>>> No, this is NOT my homework! But I do need it ASAP. ;-)

>>> Cheers,

>>> David

>>> --

>>> 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.")

>> The following should suffice:

>>

>> IF (missing_value GT 0) THEN BEGIN

>> type = SIZE(array, /TNAME)

>>

```

>> CASE type OF
>>   'COMPLEX' : MESSAGE, 'Complex values not allowed for removing
>> MISSING data.'
>>   'STRING'  : MESSAGE, 'String values not allowed for removing
>> MISSING data.'
>>   ELSE      : invalid = WHERE((array EQ missing_value), iv_count,
>> COMPLEMENT = valid, NCOMPLEMENT = v_count)
>>   ENDCASE
>>
>> IF (v_count GT 0) THEN array = array[valid]
>> ENDIF
>>
>> That is, if you dont care about overwriting the original data or it is
>> not gridded. If it is gridded (data points with missing values should
>> remain), then you could replace those with NaNs in the final array
>> using the indices above.
>>
>> Cheers,
>> Cameron Homeyer
>
> oops, missed the NaN part:
>
> This would go before the last IF (v_count...
>
> IF (~FINITE(missing)) THEN invalid = WHERE(~FINITE(array), iv_count,
> COMPLEMENT = valid, NCOMPLEMENT = v_count)
>

```

What is the meaning of ~ in front of the function name?

Subject: Re: Missing Data Programming Contest
 Posted by [Carsten Lechte](#) on Thu, 09 Apr 2009 08:51:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

Reimar Bauer wrote:
 > What is the meaning of ~ in front of the function name?

Logical negation.

chl

Subject: Re: Missing Data Programming Contest
 Posted by [Homeyer](#) on Thu, 09 Apr 2009 12:41:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Apr 8, 6:26 pm, Homeyer <cam.the.weather....@gmail.com> wrote:
> On Apr 8, 6:02 pm, Homeyer <cam.the.weather....@gmail.com> wrote:
>
>
>
>> On Apr 8, 8:23 am, David Fanning <n...@dfanning.com> wrote:
>
>>> Folks,
>
>>> I don't have time today to think about this, so I've
>>> decided to get you to do my thinking for me. :-)
>
>>> Suppose you are expecting a data array and you suspect
>>> the data has "missing values" in it that you wish to
>>> exclude from further processing. You write a keyword
>>> for your routine that allows the user to pass in what
>>> he is using for the "missing value".
>
>>> PRO Junker, data, MISSING_VALUE=missing_value
>
>>> How would you write the code to assure that this missing
>>> value would be excluded from further processing?
>
>>> You should assume:
>
>>> 1. The data can be any data type except complex or string.
>
>>> 2. The missing value *could* be !VALUES.F_NAN.
>
>>> 3. Unsophisticated users might be using your program,
>>> so, for example, they might pass in a missing value
>>> such as 594.32.
>
>>> No, this is NOT my homework! But I do need it ASAP. ;-)
>
>>> Cheers,
>
>>> David
>>> --
>>> 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.")
>
>> The following should suffice:
>
>> IF (missing_value GT 0) THEN BEGIN
>> type = SIZE(array, /TNAME)

```

>
>> CASE type OF
>>   'COMPLEX' : MESSAGE, 'Complex values not allowed for removing
>> MISSING data.'
>>   'STRING' : MESSAGE, 'String values not allowed for removing
>> MISSING data.'
>>   ELSE      : invalid = WHERE((array EQ missing_value), iv_count,
>> COMPLEMENT = valid, NCOMPLEMENT = v_count)
>> ENDCASE
>
>> IF (v_count GT 0) THEN array = array[valid]
>> ENDIF
>
>> That is, if you dont care about overwriting the original data or it is
>> not gridded. If it is gridded (data points with missing values should
>> remain), then you could replace those with NaNs in the final array
>> using the indices above.
>
>> Cheers,
>> Cameron Homeyer
>
> oops, missed the NaN part:
>
> This would go before the last IF (v_count...
>
> IF (~FINITE(missing)) THEN invalid = WHERE(~FINITE(array), iv_count,
> COMPLEMENT = valid, NCOMPLEMENT = v_count)

```

The first IF statement should be "IF (N_ELEMENTS(missing_value) GT 0) THEN BEGIN" as well.

Subject: Re: Missing Data Programming Contest
 Posted by [David Fanning](#) on Thu, 09 Apr 2009 15:59:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning writes:

```

> You should assume:
>
> 1. The data can be any data type except complex or string.
>
> 2. The missing value *could* be !VALUES.F_NAN.
>
> 3. Unsophisticated users might be using your program,
>    so, for example, they might pass in a missing value
>    such as 594.32.
>
>

```

> No, this is NOT my homework! But I do need it ASAP. ;-)

Thanks all, for your help. This was a more difficult problem than I expected it to be, but this was probably due to my program structure more than anything else. I was hoping for a general solution, but couldn't see my way through to that. Maybe next time.

In the end, I ended up replacing "missing" values with NaNs, and then dealing with those just before processing. This is always complicated, of course, by not knowing a priori what kind of data you are talking about, so there is complicated code to deal with all of that. It's a bit of a dog's dish, but at least it works. :-)

Cheers,

David

--

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

Subject: Re: Missing Data Programming Contest
Posted by [R.Bauer](#) on Thu, 09 Apr 2009 16:22:32 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning schrieb:

> David Fanning writes:

>

>> You should assume:

>>

>> 1. The data can be any data type except complex or string.

>>

>> 2. The missing value *could* be !VALUES.F_NAN.

>>

>> 3. Unsophisticated users might be using your program,
>> so, for example, they might pass in a missing value
>> such as 594.32.

>>

>> No, this is NOT my homework! But I do need it ASAP. ;-)

>

> Thanks all, for your help. This was a more difficult problem
> than I expected it to be, but this was probably due to my
> program structure more than anything else. I was hoping for
> a general solution, but couldn't see my way through to that.

> Maybe next time.
>
> In the end, I ended up replacing "missing" values with
> NaNs, and then dealing with those just before processing.
> This is always complicated, of course, by not knowing
> a priori what kind of data you are talking about, so there
> is complicated code to deal with all of that. It's a bit
> of a dog's dish, but at least it works. :-)
>
> Cheers,
>
> David
>

it works only for float or double but not
for everything of spec 1.

e.g. byte, long, structure ...

```
IDL> help,z
Z          ULONG64  = Array[3]
IDL> z[1] = !values.f_nan
% Program caused arithmetic error: Floating illegal operand
IDL> help,z
Z          ULONG64  = Array[3]
IDL> print,z
          1  9223372036854775808          1
```

that's fun or?

/me personally hates dealing with data when I don't know some more
specs. e.g. units, missing_value, fill_value, valid_min, valid_max

cheers

Reimar

Subject: Re: Missing Data Programming Contest
Posted by [David Fanning](#) on Thu, 09 Apr 2009 16:34:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

Reimar Bauer writes:

> it works only for float or double but not
> for everything of spec 1.

```
>
> e.g. byte, long, structure ...
>
> IDL> help,z
> Z          ULONG64  = Array[3]
> IDL> z[1] = !values.f_nan
> % Program caused arithmetic error: Floating illegal operand
> IDL> help,z
> Z          ULONG64  = Array[3]
> IDL> print,z
>          1  9223372036854775808          1
>
> that's fun or?
```

Yes, I have to convert integer data to float data
to do the assignment. I don't check for structures
and the like. I'm nice, but I'm not crazy. :-)

Cheers,

David

--

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