
Subject: indexing arrays with arrays

Posted by [Conor](#) on Tue, 07 Aug 2007 13:39:54 GMT

[View Forum Message](#) <> [Reply to Message](#)

It seems to me that a highly useful syntax for IDL would be something like this:

```
res = randomu(seed,10)
start_ind = indgen(5)
end_ind = start_ind + 1
```

```
extract = res[start_ind:end_ind]
```

In this case, I would envision extract being a 10 element array. Namely, it would be the equivalent of:

```
extract = [res[start_ind[0]:end_ind[0]], res[start_ind[1]:end_ind[1]],
res[start_ind[2]:end_ind[2]], etc...]
```

Does anyone else think something like this would be useful? Anyone know how I can bug RSI and request something like this? I couldn't find a "feature request" option, or anything similar, on their site. Anyway, that point aside, can anyone think of a robust way to do this right now? I've been trying to come up with a speedy, robust, for-loop-less way to do this, but haven't come up with anything. Any thoughts would be appreciated.

Subject: Re: indexing arrays with arrays

Posted by [JD Smith](#) on Thu, 09 Aug 2007 16:45:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Tue, 07 Aug 2007 06:39:54 -0700, Conor wrote:

> It seems to me that a highly useful syntax for IDL would be something
> like this:

```
>
> res = randomu(seed,10)
> start_ind = indgen(5)
> end_ind = start_ind + 1
```

```
>
> extract = res[start_ind:end_ind]
```

```
>
> In this case, I would envision extract being a 10 element array.
> Namely, it would be the equivalent of:
```

```
>
> extract = [res[start_ind[0]:end_ind[0]], res[start_ind[1]:end_ind[1]],
> res[start_ind[2]:end_ind[2]], etc...]
```

- >
- > Does anyone else think something like this would be useful? Anyone
- > know how I can bug RSI and request something like this? I couldn't
- > find a "feature request" option, or anything similar, on their site.
- > Anyway, that point aside, can anyone think of a robust way to do this
- > right now? I've been trying to come up with a speedy, robust, for-
- > loop-less way to do this, but haven't come up with anything. Any
- > thoughts would be appreciated.

Well, if end_ind is always a constant offset from start_ind, it's easy just to construct the index vector yourself:

```
t=[end_ind[0]-start_ind[0]+1,n_elements(start_ind)]
extract=res[reform(rebin(1#start_ind,t)+rebin(indgen(t[0]),t ),t[0]*t[1])]
```

If the start-end difference can be any variable amount, this is a problem for HISTOGRAM related to chunk indexing. See the HISTOGRAM tutorial and:

http://www.dfanning.com/idl_way/chunkindex.html

JD
