

ctools - Change request #2490

csphagen: generation of true energy binning

06/04/2018 01:56 PM - Specovius Andreas

Status:	Closed	Start date:	06/04/2018
Priority:	Normal	Due date:	
Assigned To:	Knödlseider Jürgen	% Done:	100%
Category:		Estimated time:	0.10 hour
Target version:	1.6.0		

Description

When setting the true energy binning for csphagen the final number of bins is computed via:

$(\text{bins_per_decade} * \text{num_of_decades}) + 1$

Does this "+1" really make sense?

.. in fact, the user wants a specific binning but gets a slightly different one!

(my example: etruemin=0.1 etruemax=100 etruebins=24 leads to >72 bins instead of exactly 72)

History

#1 - 06/04/2018 02:28 PM - Knödlseider Jürgen

user#291 wrote:

When setting the true energy binning for csphagen the final number of bins is computed via:

$(\text{bins_per_decade} * \text{num_of_decades}) + 1$

Does this "+1" really make sense?

.. in fact, the user wants a specific binning but gets a slightly different one!

(my example: etruemin=0.1 etruemax=100 etruebins=24 leads to >72 bins instead of exactly 72)

The +1 is just there to make sure that the number of bins is always >0.

#2 - 06/07/2018 02:32 PM - Specovius Andreas

user#3 wrote:

user#291 wrote:

When setting the true energy binning for csphagen the final number of bins is computed via:

$(\text{bins_per_decade} * \text{num_of_decades}) + 1$

Does this "+1" really make sense?

.. in fact, the user wants a specific binning but gets a slightly different one!

(my example: etruemin=0.1 etruemax=100 etruebins=24 leads to >72 bins instead of exactly 72)

The +1 is just there to make sure that the number of bins is always >0.

Hence, do you think it makes sense to care about this valid number of bins check in a different way?

I mean it is at least a bit confusing that the number of true energy bins generated is not like expected via the input parameter, right.. and things get worse when the user expects a certain binning!

#3 - 06/07/2018 02:42 PM - Tibaldo Luigi

Perhaps you could elaborate on the reason why you care about the exact binning in True energy? This is only an aspect internal to the tool, and, as long as the binning is reasonable and fine enough, it should have no impact on the final results.

#4 - 06/07/2018 03:37 PM - Specovius Andreas

user#266 wrote:

Perhaps you could elaborate on the reason why you care about the exact binning in True energy? This is only an aspect internal to the tool, and, as long as the binning is reasonable and fine enough, it should have no impact on the final results.

As you may know I am dealing with the comparison of ctools analysis results to the HESS analysis pipeline using HESS data. I needed a certain true energy binning with which a direct comparison of intermediate results of both tools (eg the rmf) would have been easily possible and could not achieve that binning via setting the corresponding user parameters.

Indeed this use case is a very special one which may not occur in every day life. I already adjusted the code of my local installation so that I get the desired binning and if you think for that purpose that's the way we should go it's fine for me.

What I wanted to highlight with this issue is more the general case: that the set of parameters that basically fully defines the true energy binning and which obviously exists to allow the user to define a desired binning does not result in the desired binning.

#5 - 06/11/2018 04:17 PM - Knödseder Jürgen

- Status changed from New to Pull request

- Assigned To set to Knödseder Jürgen

- Target version set to 1.6.0

- % Done changed from 0 to 100

I changed the code so that csphagen etruemin=0.1 etruemax=100 etruebins=24 now leads to exactly 72 energy bins.

Here specifically what is now in the code:

```
# Determine number of energy bins
n_decades = (emax.log10TeV() - emin.log10TeV())
n_bins = int(n_decades * float(self['etruebins'].integer()) + 0.5)
if n_bins < 1:
    n_bins = 1
```

Note that the +0.5 in the parentheses assures correct rounding towards the nearest integer. The positivity of n_bins is now assured using a value test.

#6 - 06/11/2018 05:57 PM - Knödseder Jürgen

- *Status changed from Pull request to Closed*

Merged into devel.