Analysis Syntax

The EPOS analysis tools are quite sophisticated, including the possibility to use macros and loops. Here, we just discuss some very basic applications.

An elementary analysis consists of several command between the keywords **beginanalysis** and **endanalysis**, a simple example being

```
!------------------
!  Define analysis
!------------------
beginanalysis
   histogram
       xvariable
       yvariable
       normalisation
       xmin
       xmax
       nb_of_bins
   trigger variable min max
   idcode id_0 idcode id_1 ...  idcode id_n-1 idcode id_n
   noweak
endanalysis
!---------------------------------------------------
!  Write out final results to output file
!---------------------------------------------------
write title
histoweight
writearray number_of_columns
```

The histogram command

For the histogram we imagine something like "***y* versus *x***", where ***x*** and ***y*** are some variables (*xvariable*, *yvariable*). There are many possible choices for ***xvariable*** (rap = rapidity, pt = transverse momentum, mulevt = event multiplicity, etc), whereas for ***yvariable*** one usually takes a "counter" (like numptl = number of particles or numevt = number of events). There are "particle variables" (rap, pt) and "event variables" (mulevt), and the two variables xvariable and yvariable must be of the same type.

Then one defines the ***normalization***. The analysis procedure performs a sum $\Delta y = \sum y$ of y-values within a bin Δx. Without normalisation (i.e. *normalization* = 0) the output (via writearray 2) will be two columns, the first one (X) being the x-values (bin centers), and the second one (Y) being Δy, so we have "Δy versus x". Usually we want normalized distributions "Y=$K \times \Delta y$ versus X" with some normalization K.

The simplest case is ***normalization*** being equal to a single digit integer ***k***, which gives the following normalization factor K_1:

k	K_1
0	1
1	1 / number of events
2	1 / number of triggered events
4	1 / bin-counts
5	1 / bin sum
6	1 / number of summed bin-counts (yield=1.)
7	uses same normalization as previous histogram

In addition, one often needs to divide by the bin width, which can be done by using a ***normalization*** being equal to a two digit integer ***jk***, where in addition to normalization factor K_1, we have another factor

j	K_2
0	1
1	1 / bin-width
2	sigma_total / bin-width

A normalization factor $K = K_1 \times K_2$ is already enough for most applications. For some special applications, we allow ***normalization*** to be equal to a three digit integer ***ijk***, where in addition to normalization factors K_1 and K_2, we have another factor

i	K_3
0	1
1	x (*x*-bin center)
2	1 / (2 pi x) (modified for mt0)
3	kno-scaling normalization
4	1 / x**1.5
5	1 / x

(where K_3 is strictly speaking not a constant, it may depend on the bin index) and finally one may also use ***normalization*** to be equal to a four digit integer ***hijk***, with two options for ***h***, namely h=0 being the normal case discussed above, giving $K\Delta y_i$ (with i referring to the bins) and h=1, which amounts to $\sum_{j=1}^{i} K\Delta y_j$.

In order to count y-values in x-bins, as discussed above, we need to define the x-interval, which is done via ***xmin*** and ***xmax*** , and the numer of bins, defined via ***nr_of_bins***.

Other analysis command

The **trigger** command is used to select data with variable values between a lower bound (***min***) and an upper bound (***max***).

The **idcode** commands define the particles of interest. Please refer to **src/KWt/idt.dt** to get EPOS identifier values. ***(9970*** means charged particles)

The command **noweak** means that we do not take into account weak decays.

There are many more analysis commands!

Output

Only defining an analysis via **beginanalysis** ... **endanalysis** will not produce any output, one needs to activate the output.

The following commands allow to write out the results in a "histogram file" named **${HTO}z-*name*.histo**, where ***name*** corresponds to ***name*** chosen for the optns file ***name*.optns**. The command ***write*** writes litterally to the output file : it can be used to write simply the histogram title, or much more (the EPOS developers use it to provide a particular histogram syntax, to be interpreted by graphics tools like python or gnuplot, but this will not be discussed here).

The command **histoweight** prints the "weight" of the histogram, which is needed for adding several histgrams. Its precise definition depends on the normalization.

Finally, the command **writearray** followed by a number **n** creates a **n**-column table. The two first columns always contain X and Y values for all the bins, the content of the third column depends on the normalization.

The normalization affects the output:

- For the "normal cases" like ***normalization*** = 11 (or 12), the command **histoweight** writes out the number of (triggered) events, and the third column from **writearray** contains the absolute statistical uncertainty $\mathrm{Y}/\sqrt{N}$ (with N being the number of entries in the considered bin).
- For the "special case" of ***normalization*** = 4, the command **histoweight** writes out 0 and the third column of the results array contains the number of entries in the current bin.