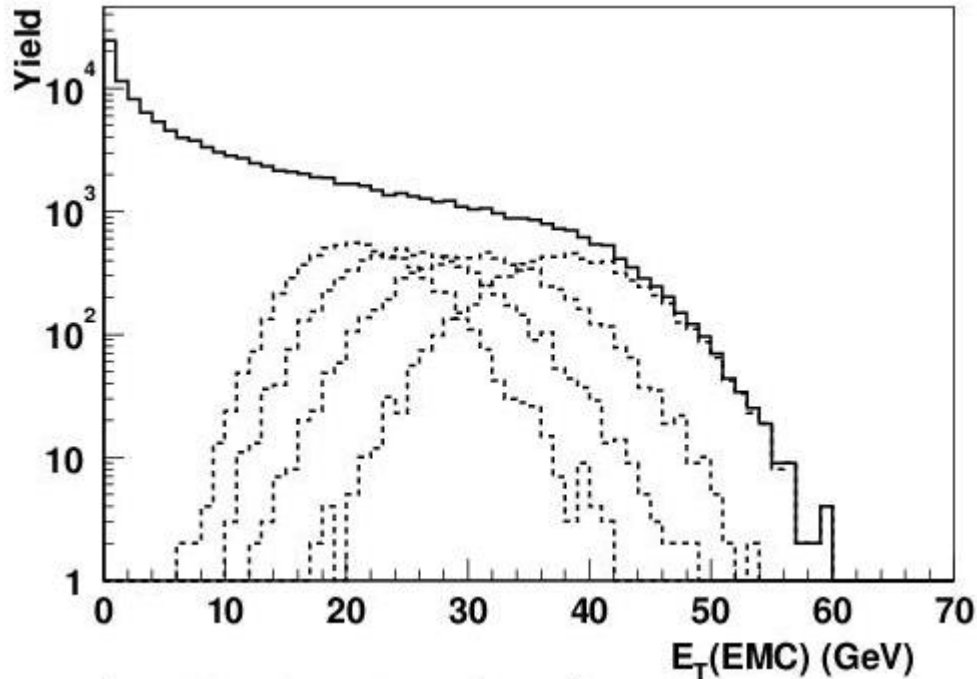


•The Scaledown Story

- What's the deal?
- We cannot select the way collisions happen. We have to take whatever nature gives to us.
- Strategy: Make as many collisions as possible, then pick the ones you want.
- The number is limited by the collider performance, and ultimately by what the detectors can stand - at some point you start loading the detectors or even damage them.
- Unfortunately, the abundance of collisions is of the **uninteresting** type, been there, done that, have seen millions of similar ones.

•Such as E_T (“transverse Energy”)



•The ET distribution has 10^{**4} more entries in the peripheral area than in the central -- for one central collision you would take 10000 peripheral ones, which, by and large, all look the same.

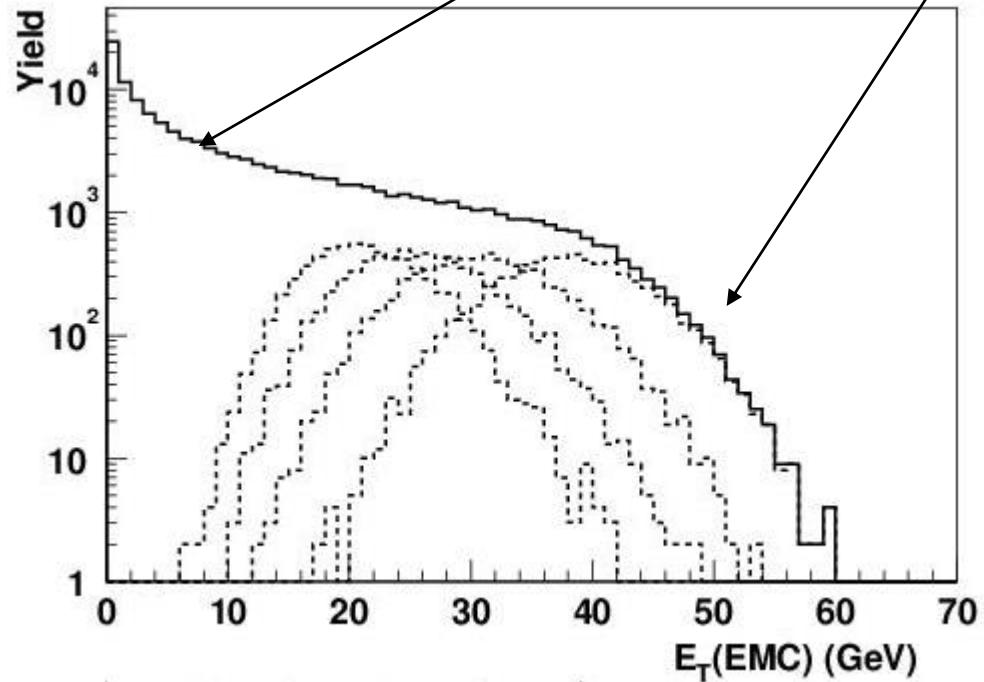
- Throw in more selective triggers and scaledowns -- scale down the abundant ones and make room for the rare ones.
- The idea is that you take one out of every, say, 20 of the abundant ones as a representative for all 20. Later, this one event is *weighted* with 20 in the analysis. Easy, right? Hang on...

•Some basics

- So why not just take all events? Just takes longer, right?
- Well, it may actually take months or years longer, even if we had so much beam time. Because if you are not selective about events, the DAQ runs at a high dead time, which can easily be 99%.
- Now say you do get 99% dead time, and during your run you have 300 J/Psi events. Under those circumstances, you get only 3 of those events to tape.
- Solution: Decrease the dead time by cutting out the abundant triggers and make room for the others. Just another way of saying “look more closely in this region”. If we run at about 15% dead, we record 255 of the 300 J/Psi’s.
- By the way: a high dead time is not always a bad thing. If after your trigger selection you still have more than enough events left (so they are not that “rare” after all), just take what you can get. (E.g., WA98 ran with ~60%).

• Scaledowns

- So scale the events.
- Take only every, say, 20th here but each one here



•One event representing many others

- That looks easy. You count, say, 20 events of a given type and then take only the 20th. But it implies that you count those 20 events that arrived **during the live time of the DAQ**. Can't count those that came during the dead time - those wouldn't have been recorded anyway.
- “This event represents x other events that *could have been taken*”. (We just decided not to).
- Only then is that representing thing true and can be done right. (Note: Large collaborations usually get this right).

• Pitfalls of the weight

We assume that we take a run (1) with 4500 events total:

Run 1: 1 1 2 20 20 events 445, 786, 1894, 522, 852 (4499)

Now let's assume for simplicity that all events add one entry to one bin of a histogram. Your math book tells you to calculate, and ROOT's TH1F will calculate

$$c = \sum_{i=0}^{n_{\text{Events}}} w_i \pm \sqrt{\sum_{i=0}^{n_{\text{Events}}} w_i^2}$$

$$\begin{aligned} c &= 445 * 1 + 786 * 1 + 1894 * 2 + 522 * 20 + 852 * 20 \\ &\pm \sqrt{445 * 1^2 + 786 * 1^2 + 1894 * 2^2 + 522 * 20^2 + 852 * 20^2} \\ &= 32499 \pm \sqrt{558407} \\ &= 32499 \pm 747.2 \end{aligned}$$

- **Now let's change the scaledown...**

Wait. Why would we change the scaledowns?

You never quite know what the “mix” of interesting events is up front (it is an experiment after all)

So at the end of the day you find that you didn't take enough events of the types that were scaled down by a factor of 20.

So a meeting is held, and it is decided that we make up the shortfall by relaxing the scaledowns in the next run.

In practice, you will find that such scaledowns change *all the time* in order to adapt to changing beam conditions – you usually want to max out your DAQs capabilities.

•Cont'ed

Let's change the scaledown:.

Run 1:	1	1	2	20	20	events	445,	786,	1894,	522,	852	(4499)
Run 2:	1	1	2	2	2	events	121,	214,	516,	1827,	2982	(5660)

You see that you get many more events here!

$$c = \sum_{i=0}^{nEvents} w_i \pm \sqrt{\sum_{i=0}^{nEvents} w_i^2}$$

$$\begin{aligned} c &= 445 * 1 + 786 * 1 + 1894 * 2 + 522 * 20 + 852 * 20 \\ &\quad + 121 * 1 + 214 * 1 + 516 * 2 + 1827 * 2 + 2982 * 2 \\ &\pm \text{sqrt}(445 * 1^2 + 786 * 1^2 + 1894 * 2^2 + 522 * 20^2 + 852 * 20^2 \\ &\quad + 121 * 1^2 + 214 * 1^2 + 516 * 2^2 + 1827 * 2^2 + 2982 * 2^2) \\ &= 43825 \pm \sqrt{580042} \\ &= 43825 \pm 761.6 \end{aligned}$$

•Ok? Or not?

•Watch this.

Let's see how many events we had originally.

	445, 786, 3788,	10440,	17040	before scaledown
scaledowns	1 1 2	20	20	
Run 1: events	445, 786, 1894,	522,	852	
	121, 214, 1032,	3654,	5964	before scaledown
scaledowns	1 1 2	2	2	
Run 2: events	121, 214, 516,	1827,	2982	

Remember the scaledown definition: “Events offered” / “Events accepted”

$$\frac{(10440 + 3654) \text{ events offered}}{14094} \bigg/ \frac{(522 + 1827) \text{ events offered}}{2349} = 6$$

$$\frac{(17040 + 5964) \text{ events offered}}{23004} \bigg/ \frac{(852 + 2982) \text{ events offered}}{3824} = 6$$

•Watch this.

Now we have

566, 1000, 4820, 14094, 23004 before scaledown
1 1 2 6 6

•Run 1* : 566, 1000, 2410, 2349, 3834

$$\begin{aligned}c &= 566 * 1 + 1000 * 1 + 2410 * 2 + 2349 * 6 + 3834 * 6 \\ &\pm \sqrt{566 * 1^2 + 1000 * 1^2 + 2410 * 2^2 + 2349 * 6^2 + 3834 * 6^2} \\ &= 43825 \pm \sqrt{233794} \\ &= 43825 \pm 483.5\end{aligned}$$

•Remember? We had +/- 761.6 here before!

•What's the difference?

•Run 1:	1	1	2	20	20	events	445,	786,	1894	522,	852	(4500)
•Run 2:	1	1	2	2	2	events	121,	214,	516,	1827,	2982	(5660)

The problem: Each event from a given trigger in run 1 and 2 has the exact same statistical significance, as long as we don't change the trigger, which we didn't. The physics is the same in the morning and in the afternoon.

The fact that we took 1827 of them (2982 of them) a while later has no bearing on our result.

Yet, we assign 522 of them the weight 20 and 1827 the weight 2 (ditto with 852 and 2982 events from “peripheral”).

All I did is to assign each event the same statistical weight (yes, I tweaked the numbers so that an even “6” would come out for added drama, but for any scaledown setting there is such an equal weight). It's even easy:

weight = events that were offered / events that were taken
= sum of event*weight_for_the_run / number of events taken

•How to do that practically?

As your analysis progresses and burns through the runs, the weights are constantly changing. Takes some getting-used to. And you don't know the numbers up front!

maintain two numbers **per trigger**:

for each event that you consider to be triggered by that trigger, add up the current weight (in that run).

And click another counter +1 (one more accepted event)

The first is (number of events offered [to the daq])

The second is number of events taken

At any time you can calculate the trigger weight as the ratio of the two

Unfortunately, you need to book each histogram you want to have in the end **once per trigger!** Then fill each one with $\text{weight}=1$.

Then, when you want to look at “the histogram”, you need to add those histograms with the current trigger-specific weight.

Easiest is to “abuse” yet another TH1F as a spreadsheet to keep the sum of weights and event counts (nice since when you add up histos from different job outputs, you add up those as well, easier on the bookkeeping).

•Oh my god.

It's tedious, really.

However, if you don't do it right, your error bars can be significantly too large, especially with a wide variation in scaledowns.

Remember, too large error bars make fits work that wouldn't work otherwise. We may not be able to rule out one model over another, for example.

Make the best use of your always limited statistics!