
Calculating a P-Value

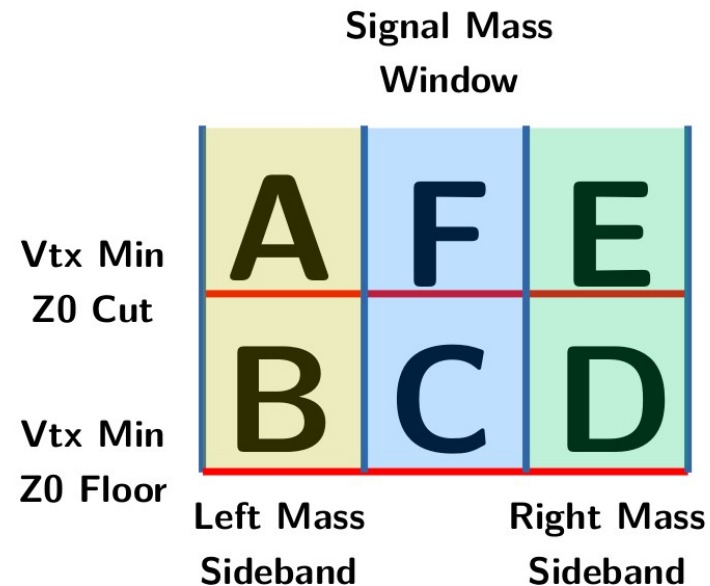
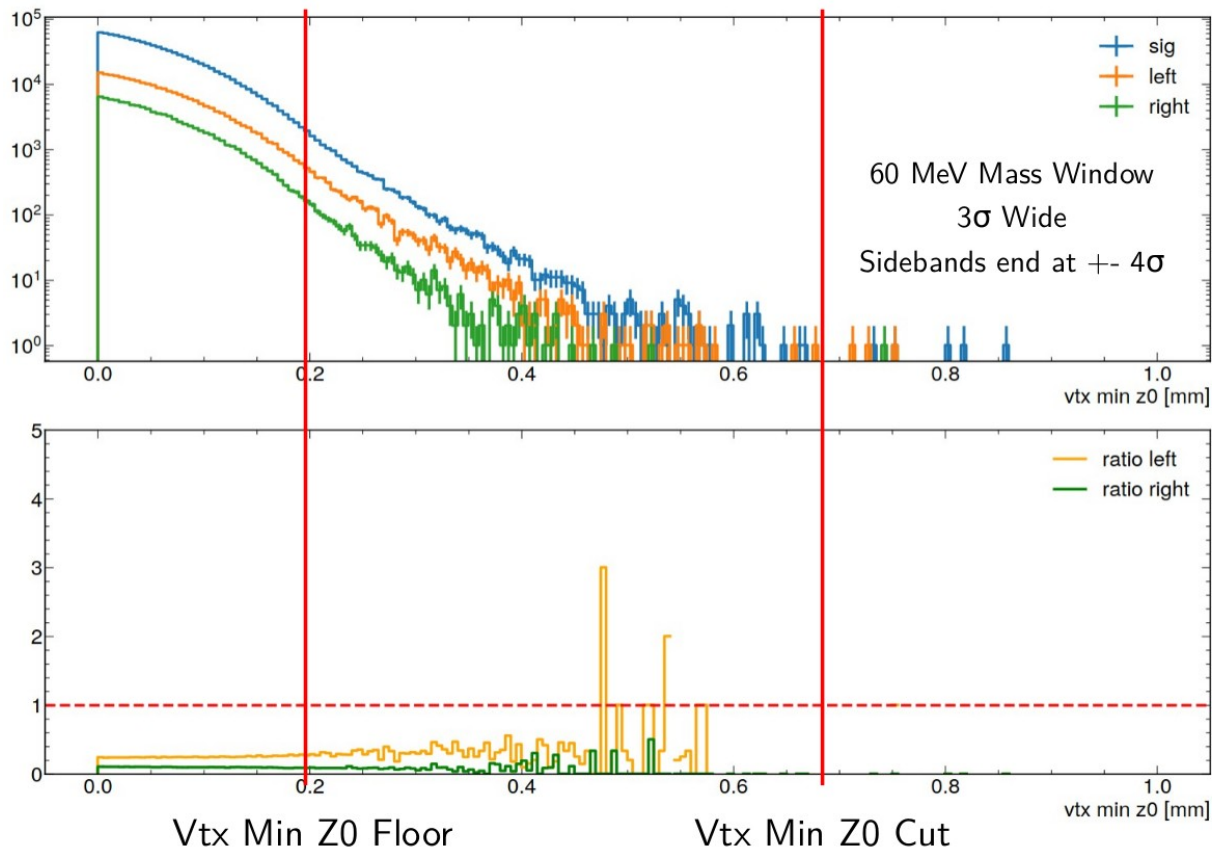
06/11/2024

Alic Spellman

Cameron Bravo



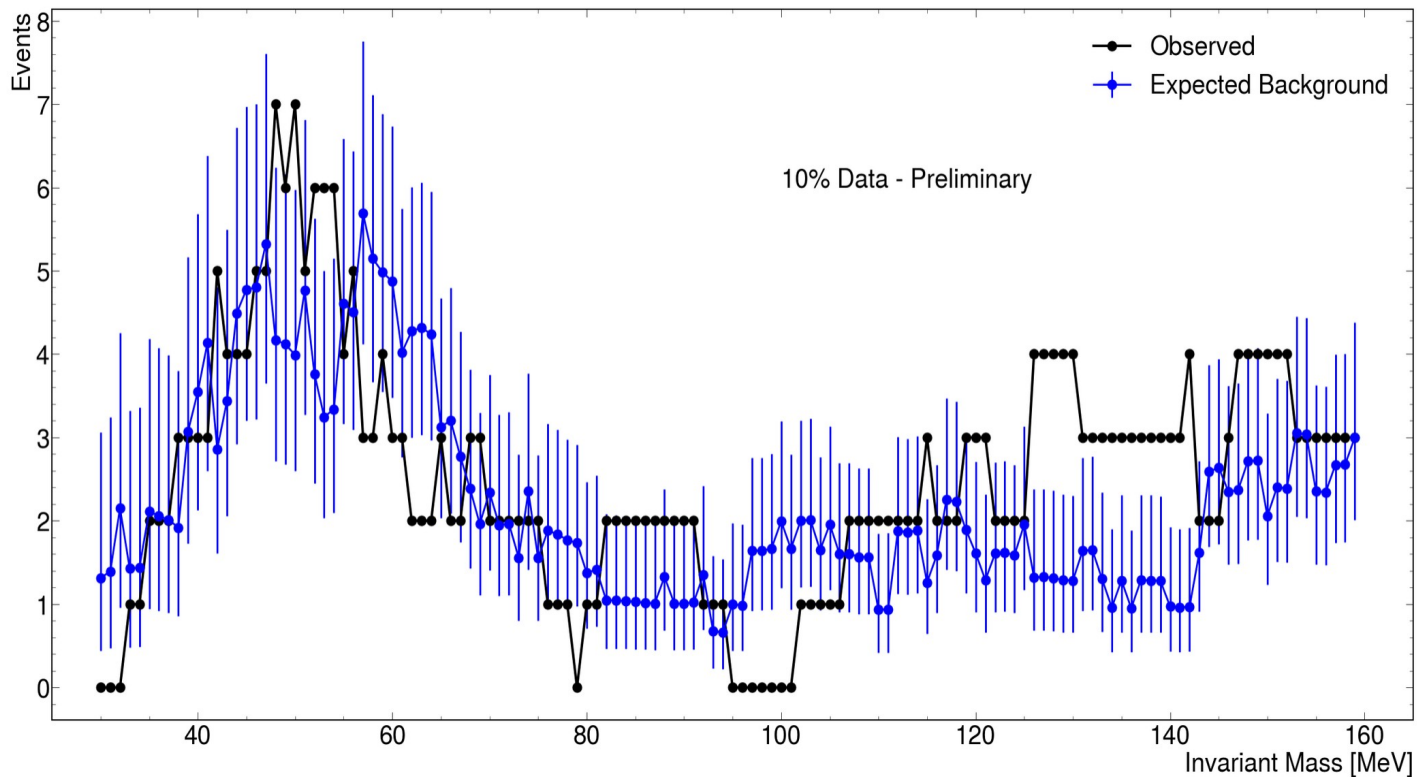
Expected Background Estimate – ABCD Method



$$(A+E)/(B+D)*C = F$$



Expected Background Estimate – 10% Data



Error on (A+E) and (B+D)

```
sigma_AE_up = poisson_up_err(A+E)  
sigma_AE_low = poisson_low_err(A+E)
```

```
sigma_BD = math.sqrt(B+D)  
sigma_C = math.sqrt(C)
```

Calculate the propagated uncertainty

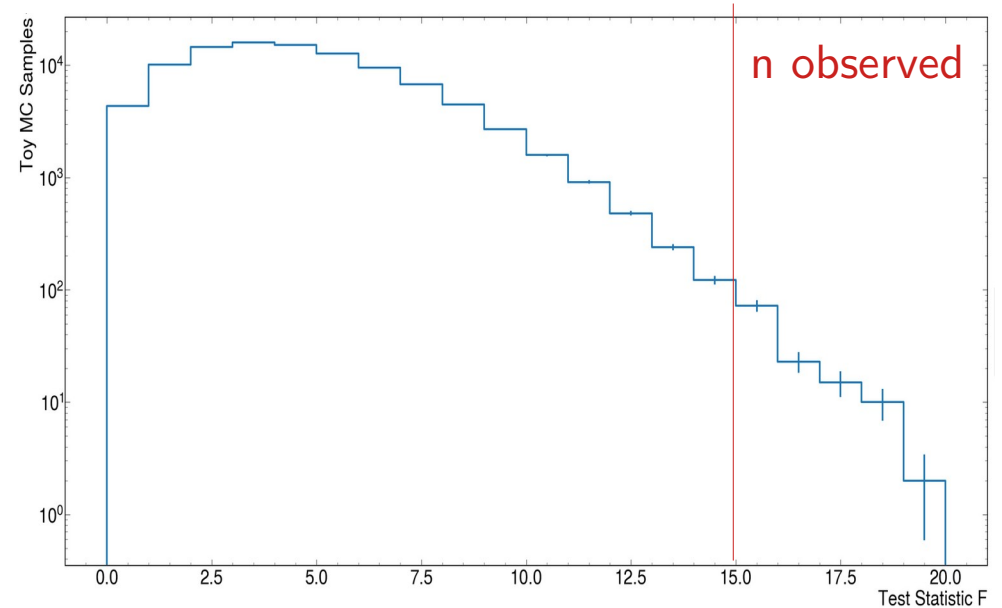
```
sigma_F_up = math.sqrt(  
    (partial_AE * sigma_AE_up) ** 2 +  
    (partial_BD * sigma_BD) ** 2 +  
    (partial_C * sigma_C) ** 2  
)
```

```
sigma_F_low = math.sqrt(  
    (partial_AE * sigma_AE_low) ** 2 +  
    (partial_BD * sigma_BD) ** 2 +  
    (partial_C * sigma_C) ** 2  
)
```



Calculating P-Value using MC Toys

- Run N Toy MC experiments
- Build distribution of test statistic 'F' (expected background)
- Three distributions...
 - Poisson with mean=(A+E)
 - Gaus with mean=(B+D), std=sqrt(B+D)
 - Gaussian with mean=C, std=sqrt(C)
- For each experiment, sample the distributions, calculate expected background mean μ_F
- **Test Statistic: sample Poisson with mean = μ_F**
- Get pvalue by integrating (normalized) test statistic distribution from 'nobs' $\rightarrow \infty$

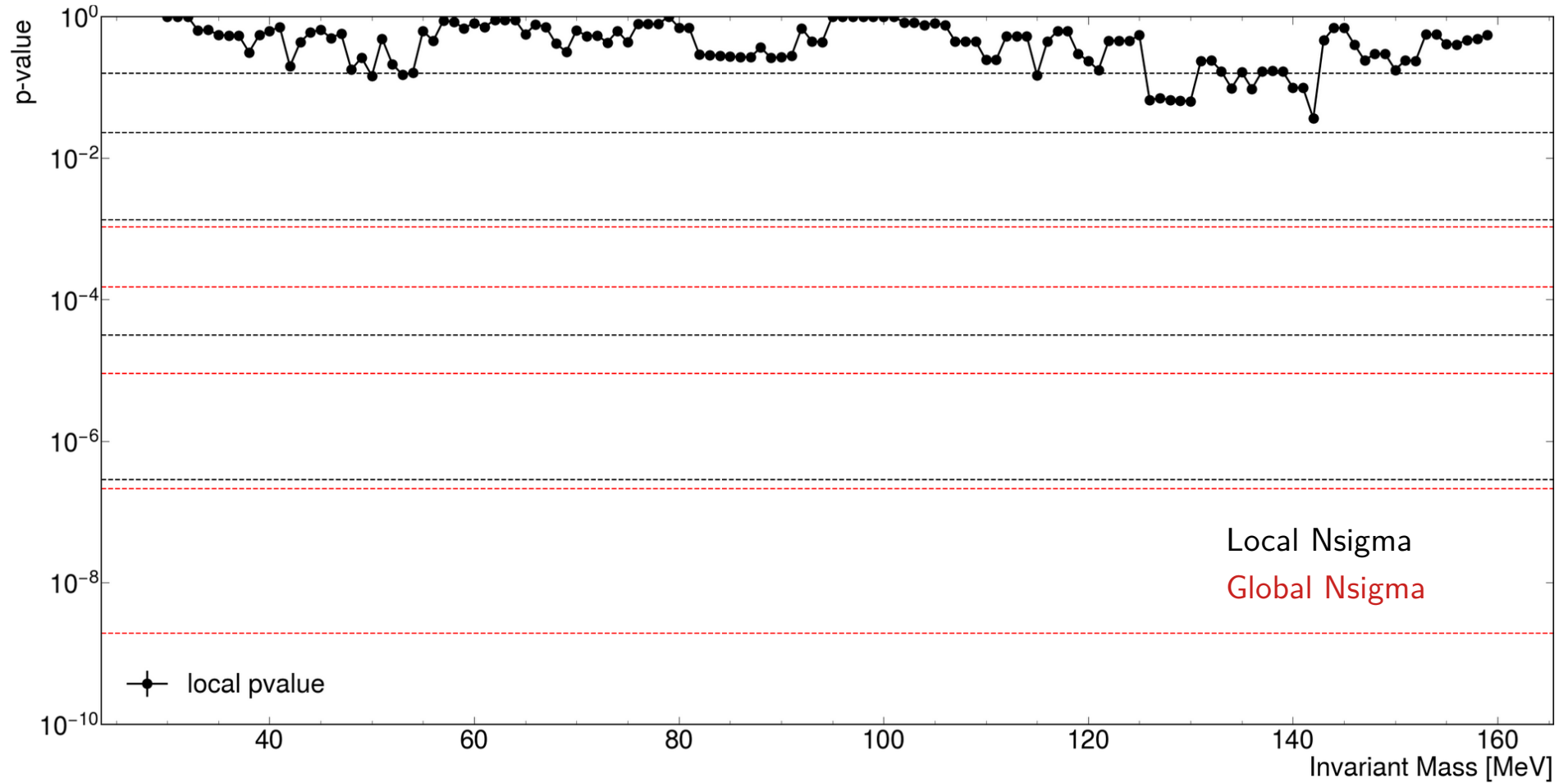


```
for i in range(ntrials):
    A_E_s = np.random.poisson(lam=(A+E))
    B_D_s = np.random.normal(loc=(B+D), scale=np.sqrt((B+D)))
    C_s = np.random.normal(loc=C, scale=np.sqrt(C))

    mu_F = ((A_E_s)/(B_D_s))*C_s
    F = np.random.poisson(lam=mu_F)
    distribution_F.fill(F)
```

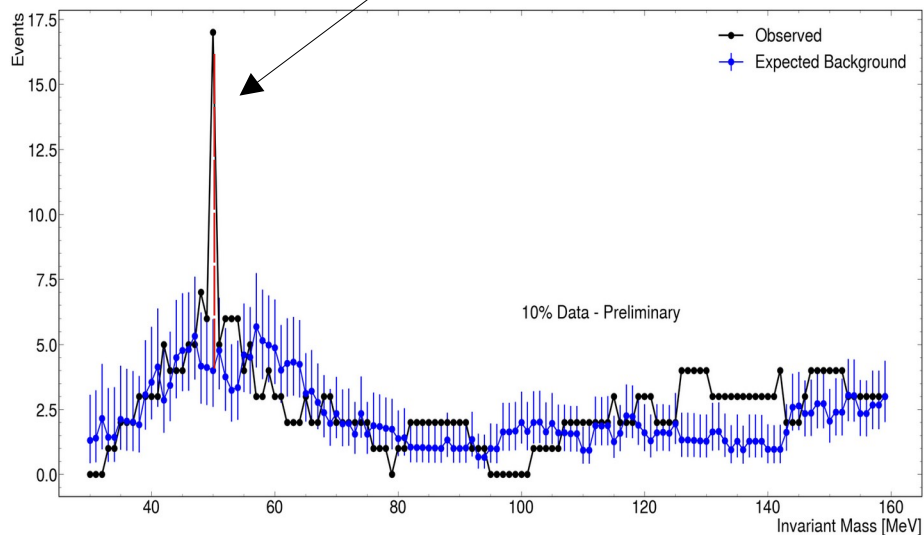


10% Data P-Values

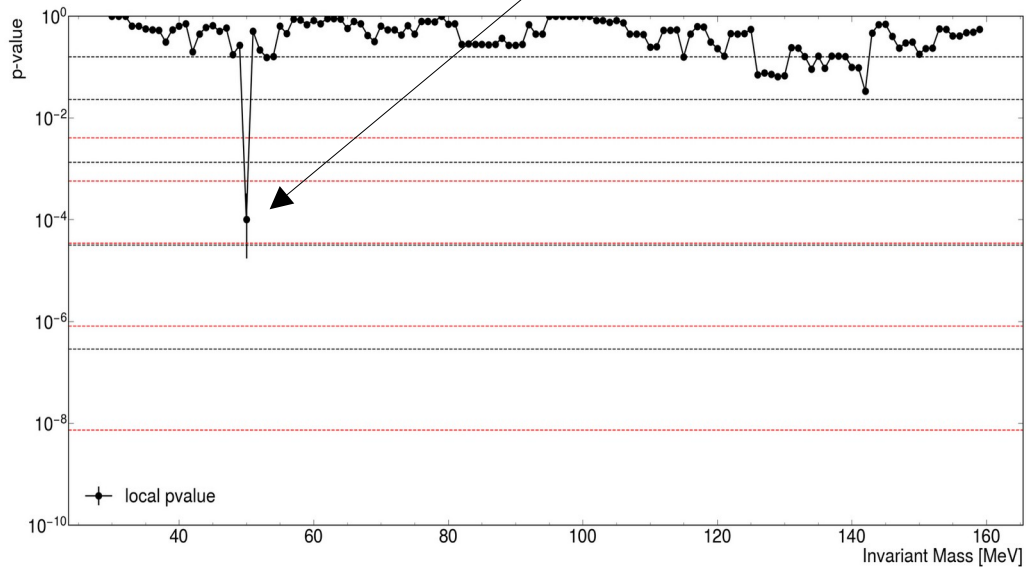


Testing Extremes

Inject 10 events into the
55 MeV bin



Only achieve global 3
sigma



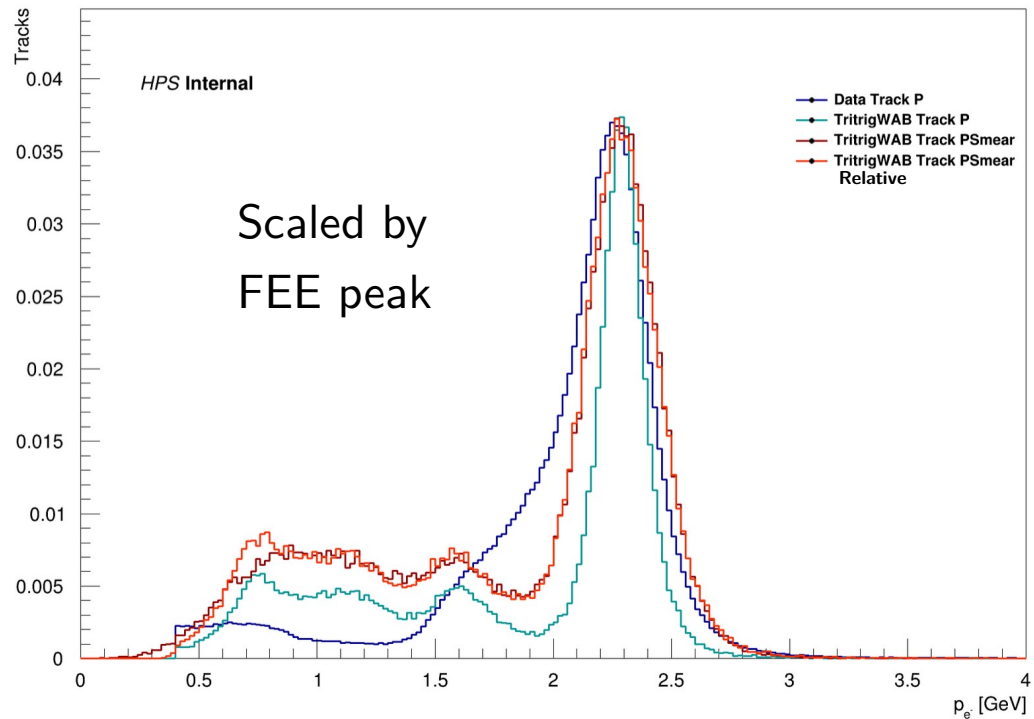
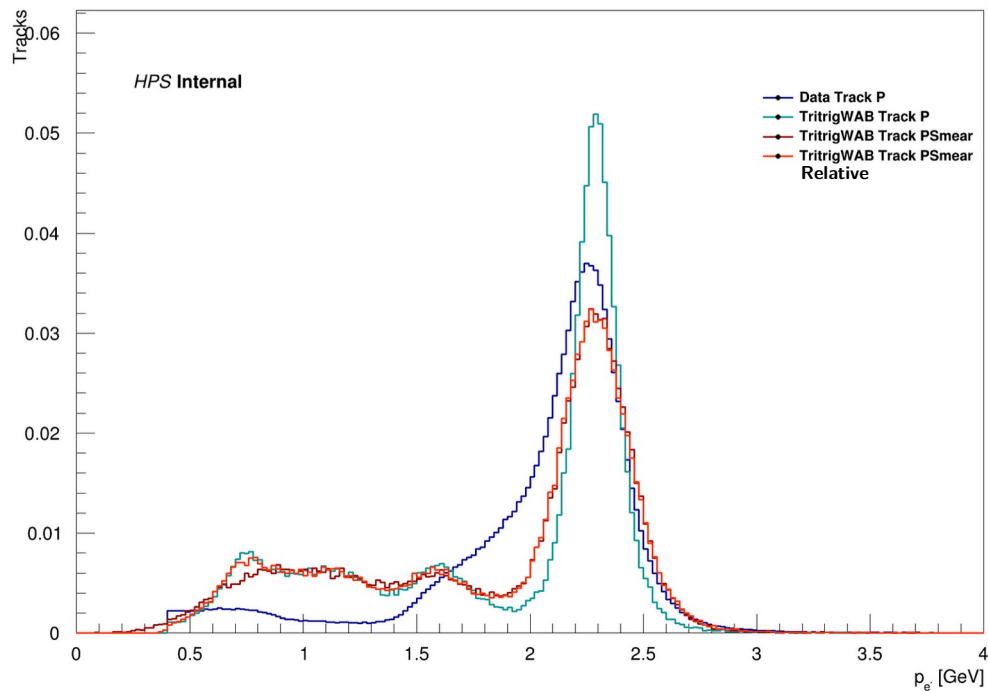
* $1.13e-6$ (global 4 sigma)
if we just throw
 $\text{Poisson}(x=17 \mid \mu=4)$



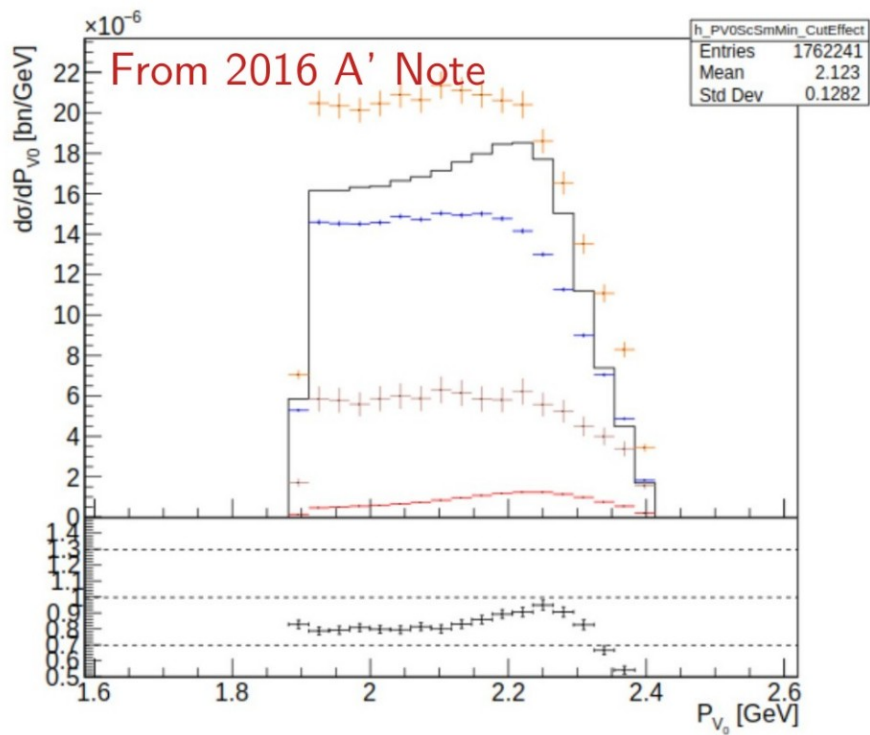
Track Momentum Smearing



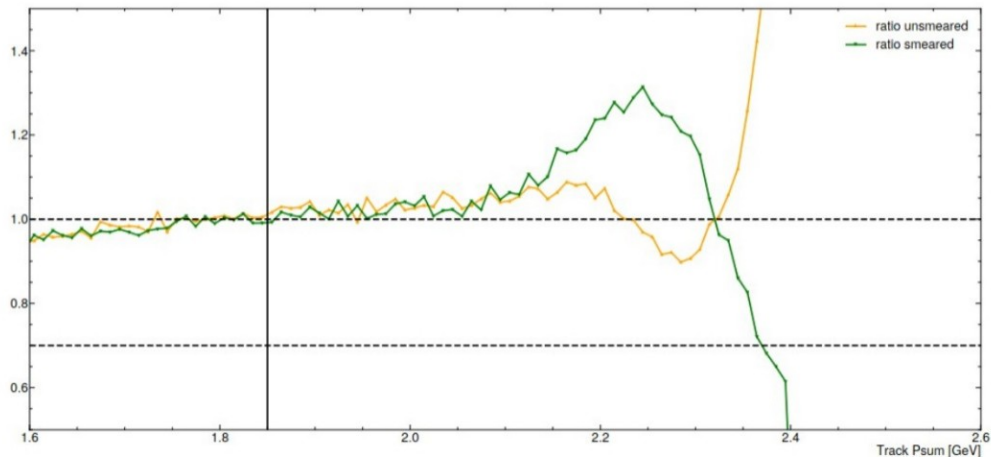
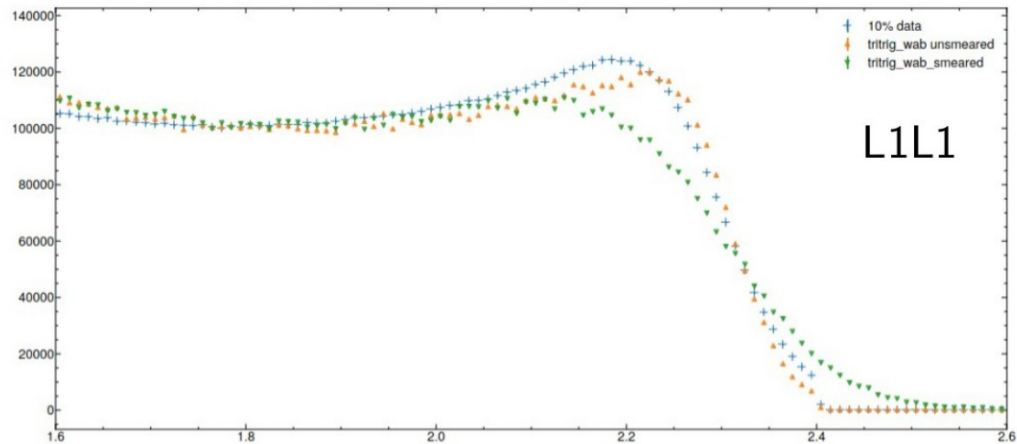
Track Momentum Smearing



Momentum Smearing



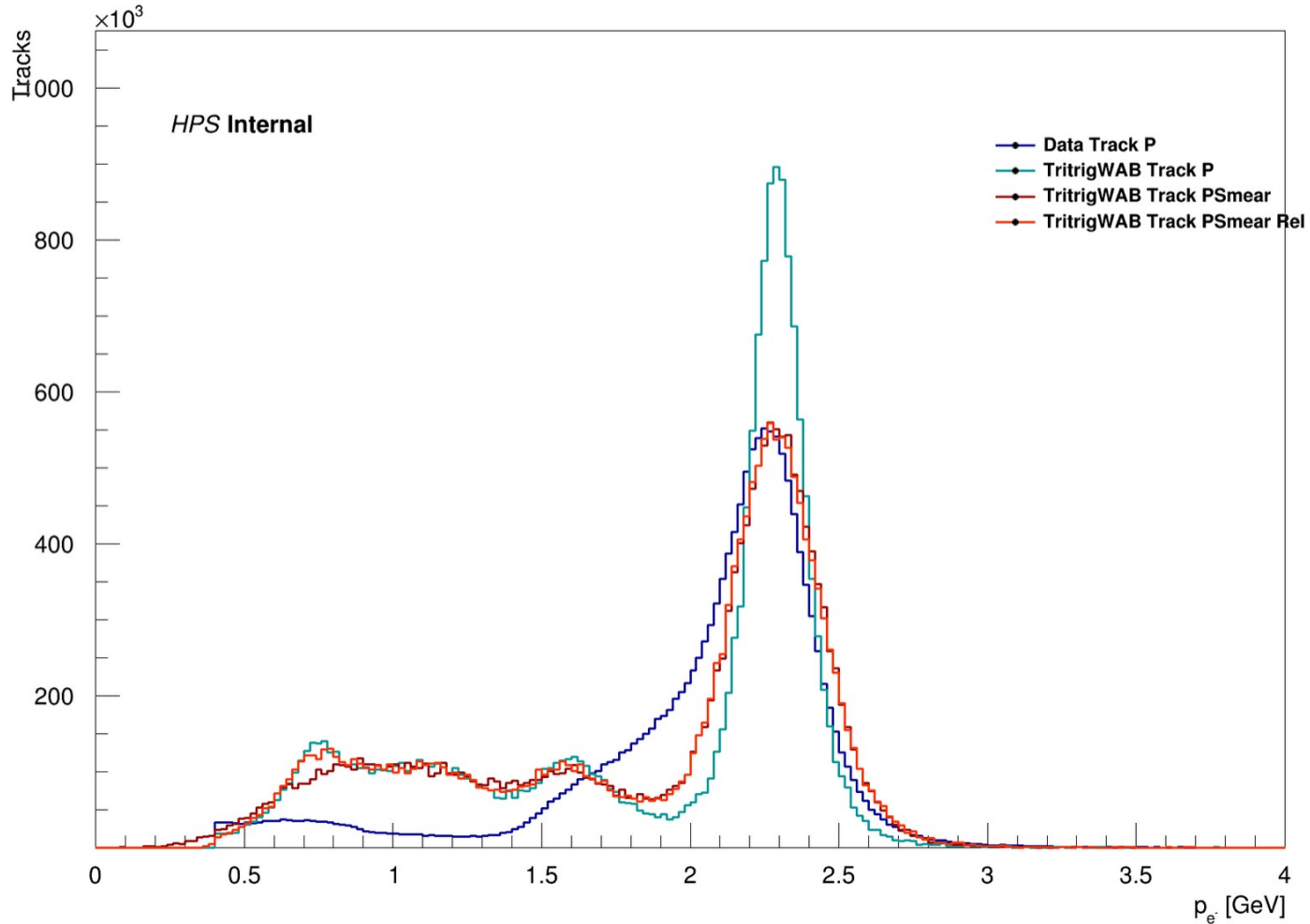
Unclear if momentum smearing is applied when this plot was made...



Backup



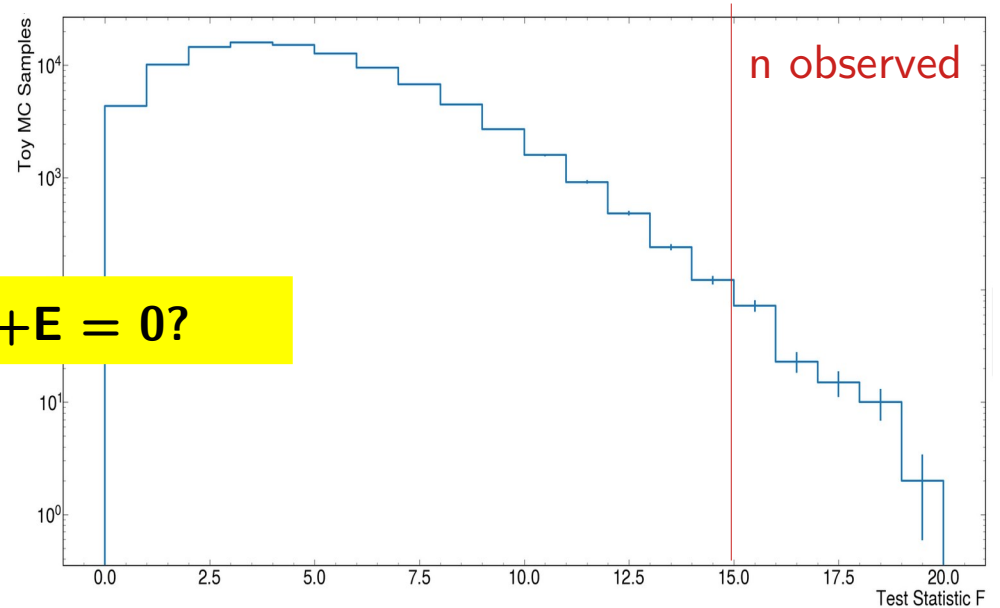
Track Momentum Smearing



Calculating P-Value using MC Toys

- Run N Toy MC experiments
- Build distribution of test statistic 'F' (expected background)
- Three distributions...
 - Poisson with mean=(A+E)
 - Gaus with mean=(B+D), std=sqrt(B+D)
 - Gaussian with mean=C, std=sqrt(C)
- For each experiment, sample the distributions, calculate expected background mean μ_F
- **Test Statistic: sample Poisson with mean = μ_F**
- Get pvalue by integrating (normalized) test statistic distribution from 'nobs' $\rightarrow \infty$

What if A+E = 0?



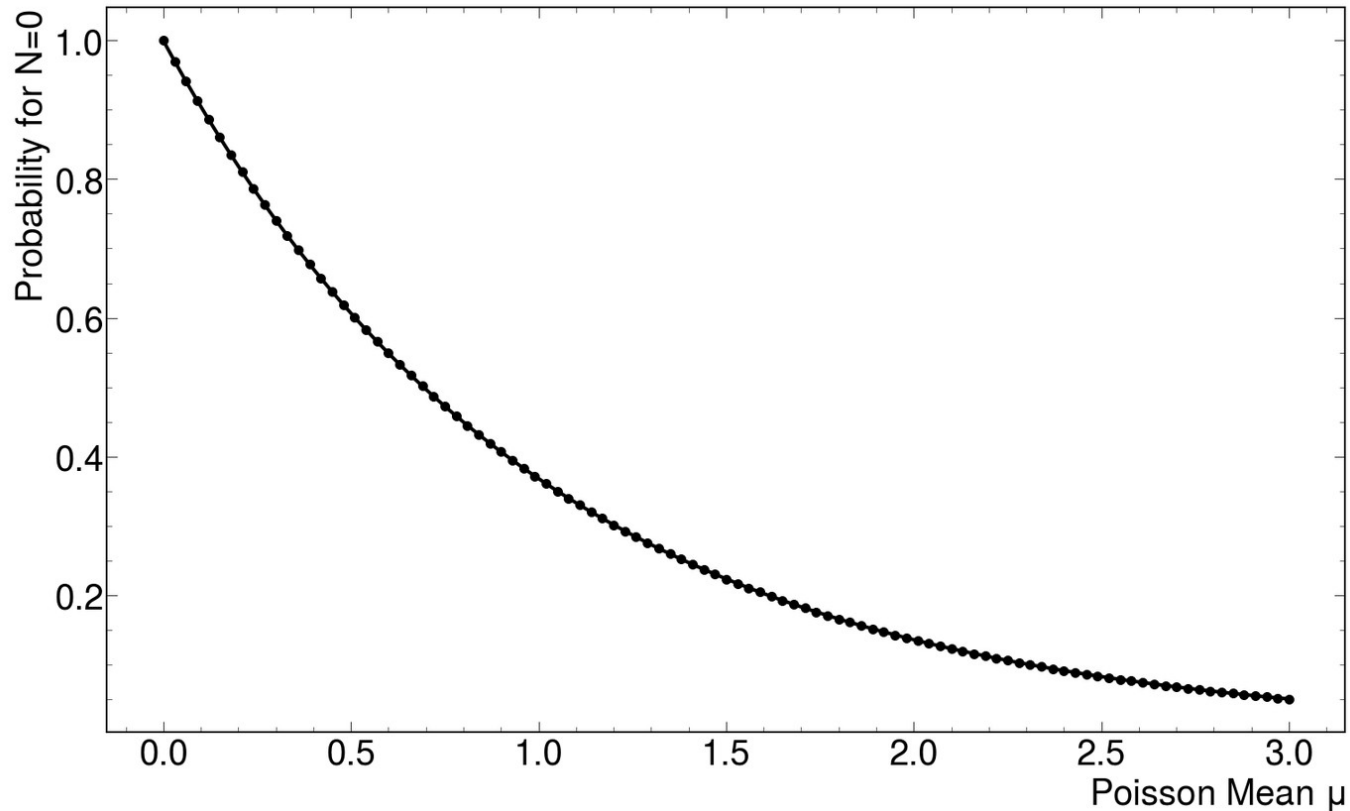
```
for i in range(ntrials):
    A_E_s = np.random.poisson(lam=(A+E))
    B_D_s = np.random.normal(loc=(B+D), scale=np.sqrt((B+D)))
    C_s = np.random.normal(loc=C, scale=np.sqrt(C))

    mu_F = ((A_E_s)/(B_D_s))*C_s
    F = np.random.poisson(lam=mu_F)
    distribution_F.fill(F)
```



Error when $A+E = 0$?

- If $A+E = 0$, we can't build a Poisson distribution for the toys
- We could just force $A+E = 1$, but that's very conservative



Not actually
applicable under
current min z0 cut

