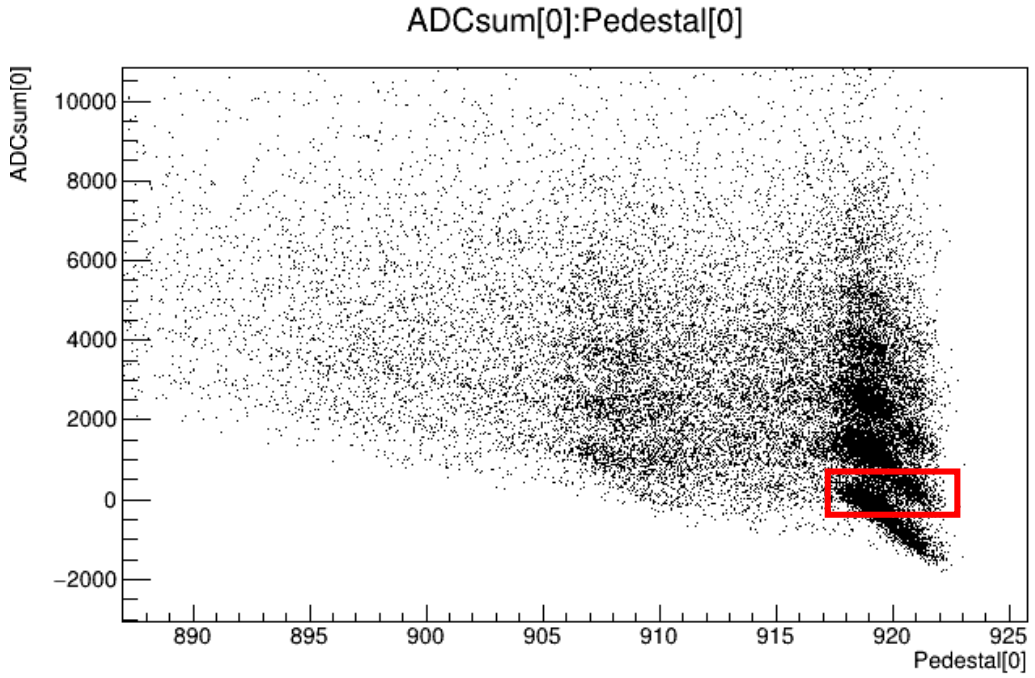
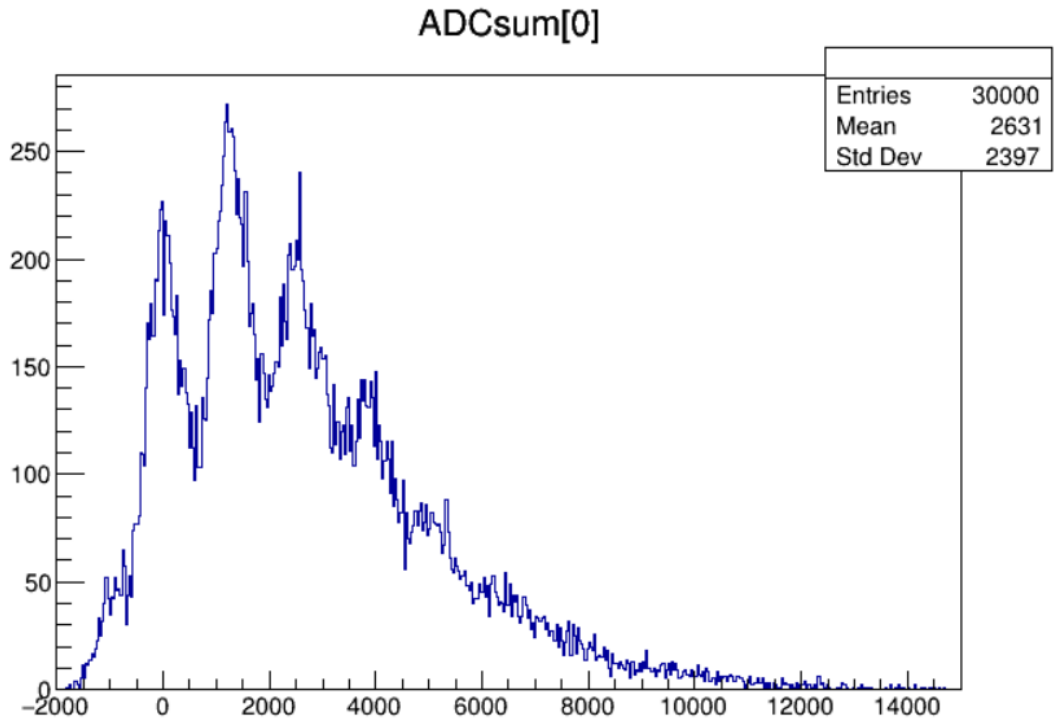
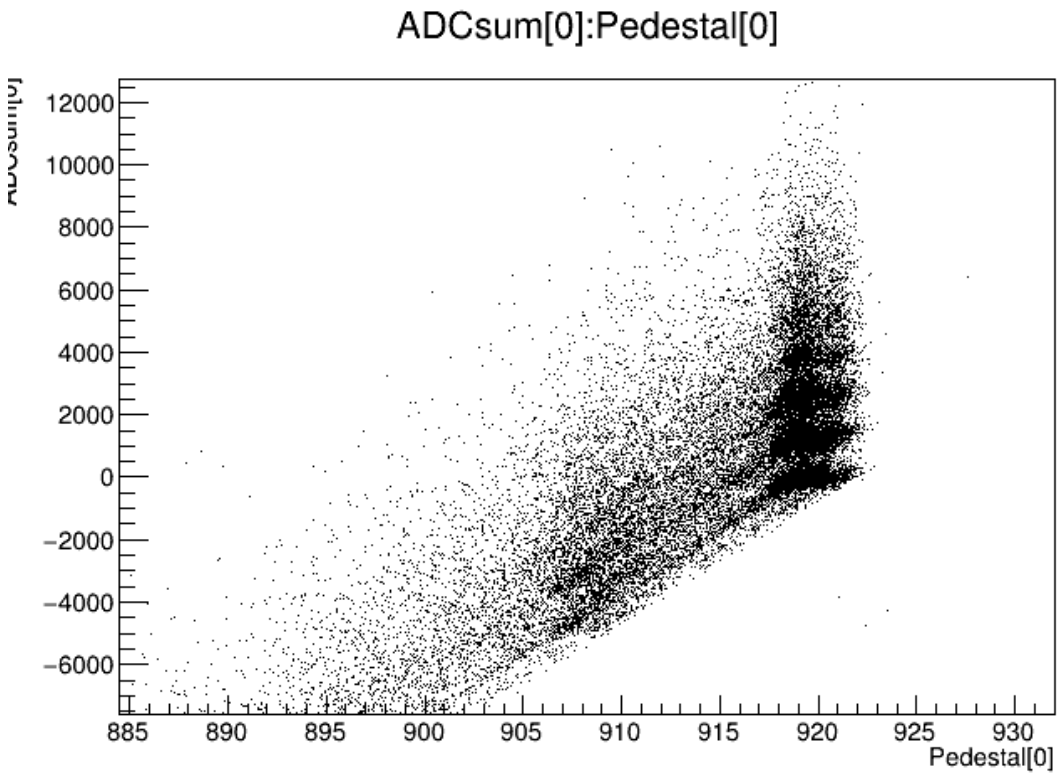
200318

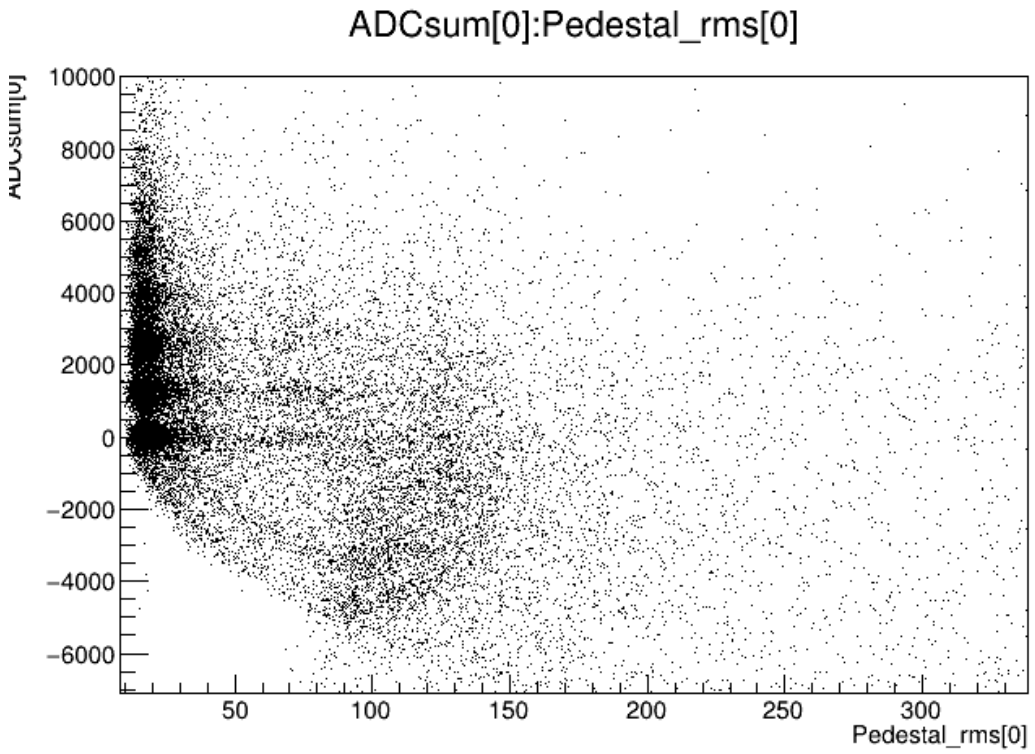
ADCsum and Pedestal



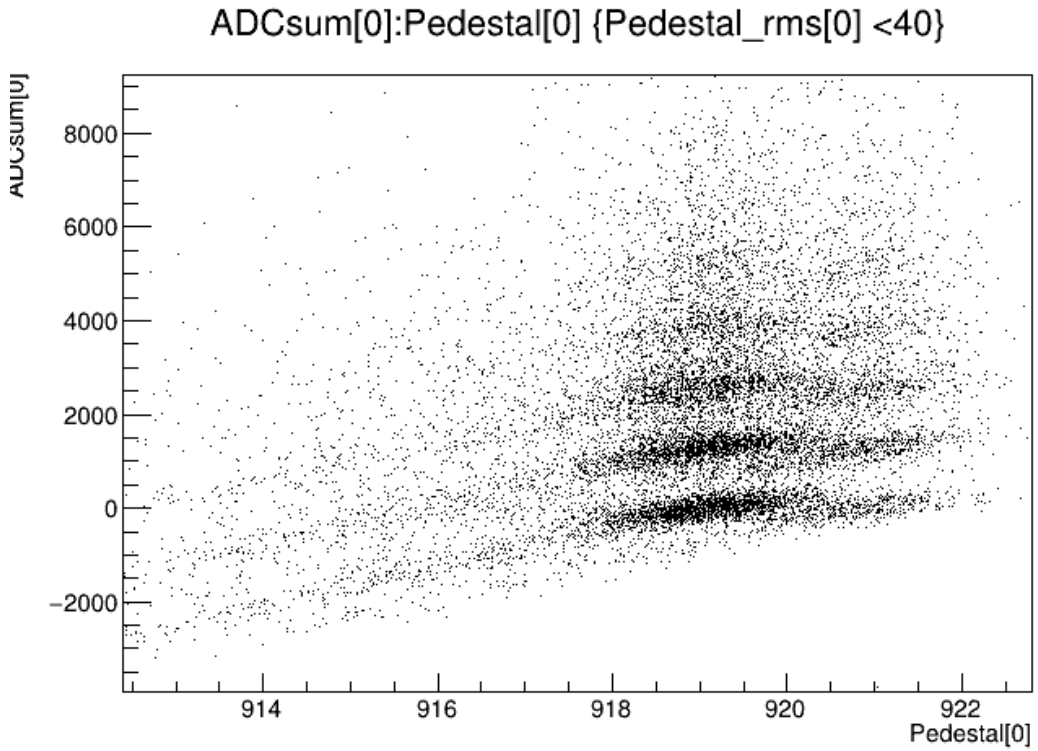


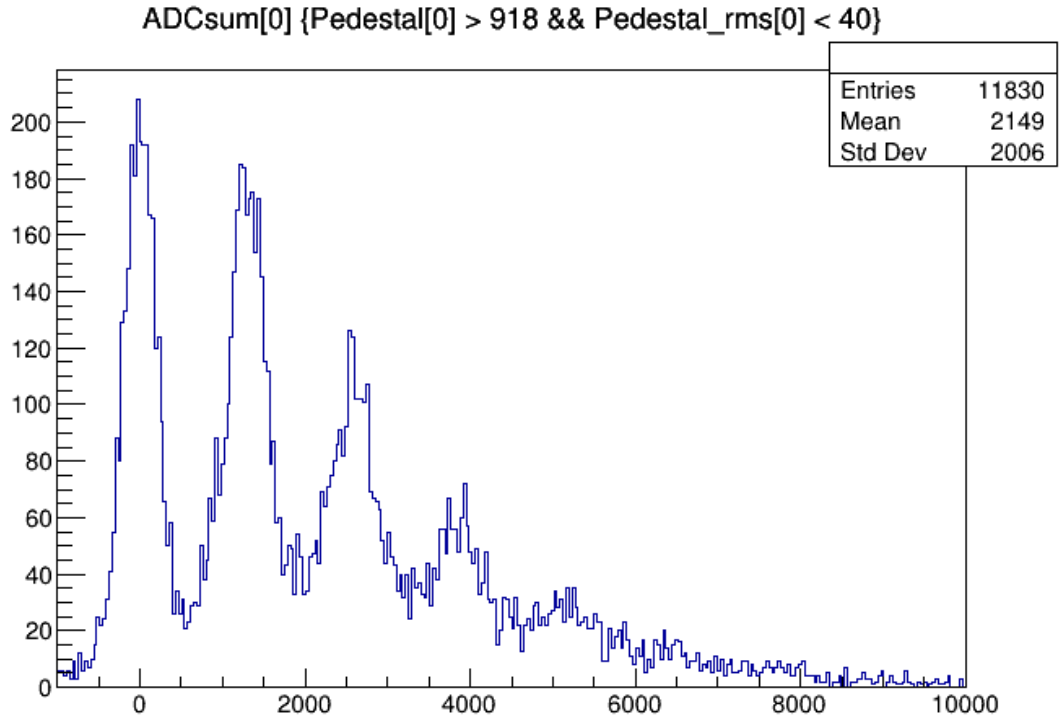
Find cut condition



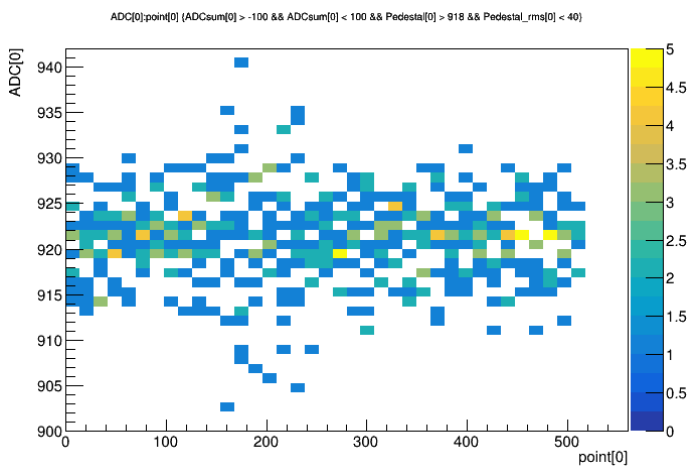


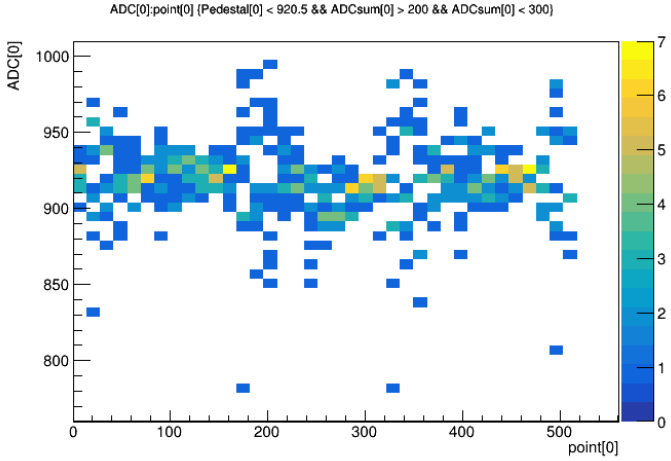
Find cut condition



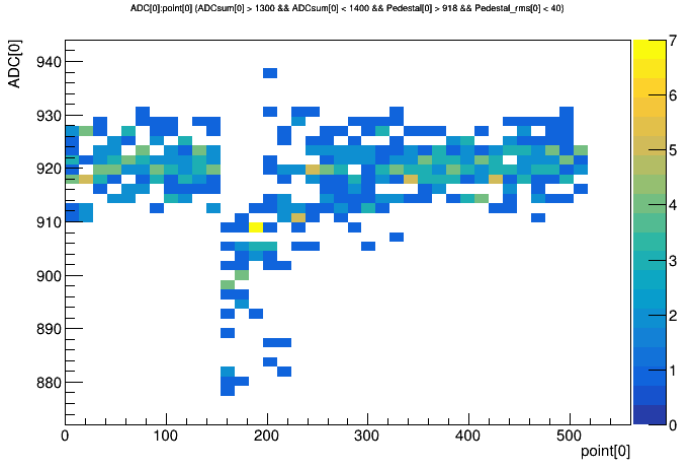


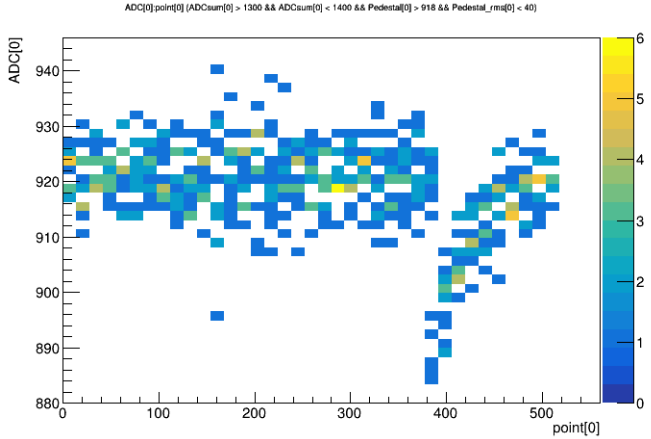
Waveform check



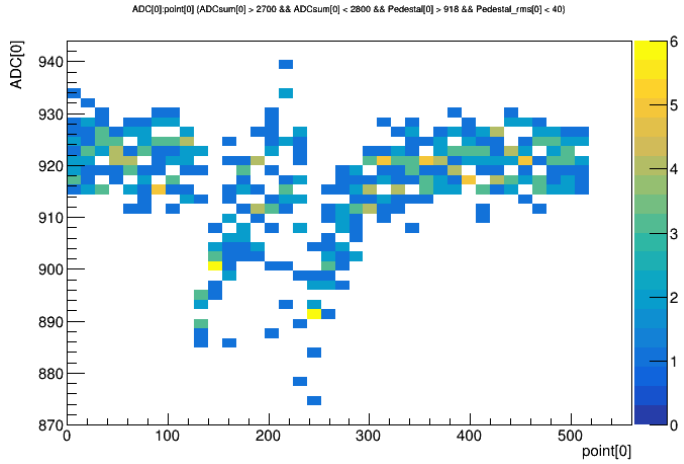


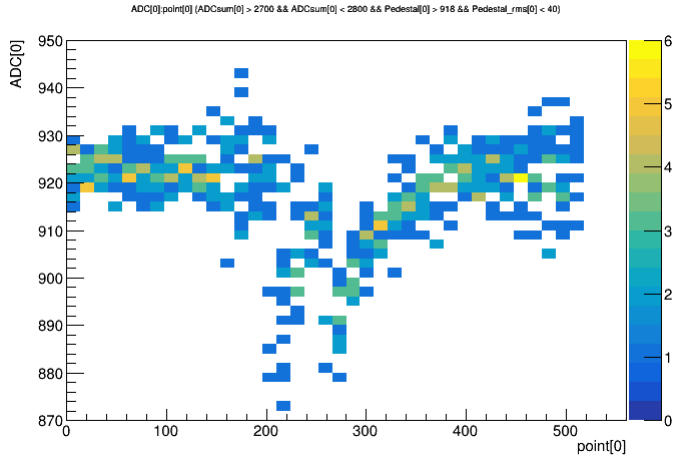
Waveform at 1st peak





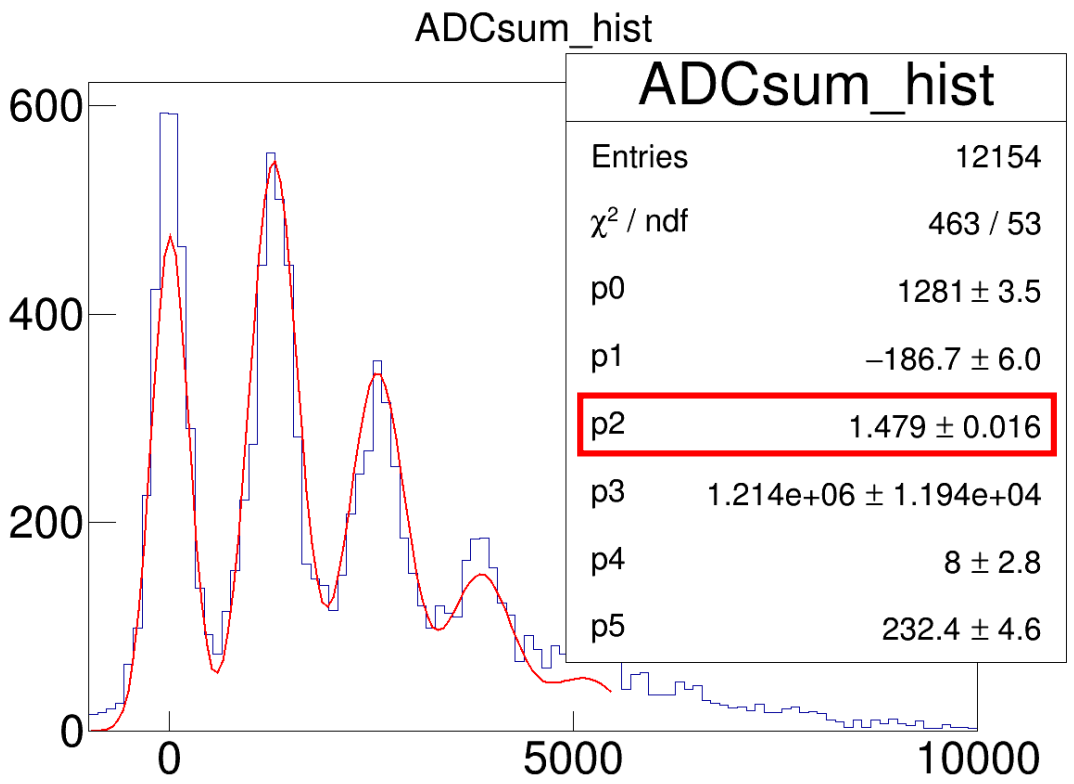
Waveform at 2nd peak





Waveform at 3rd peak

Average of the number of pulses?

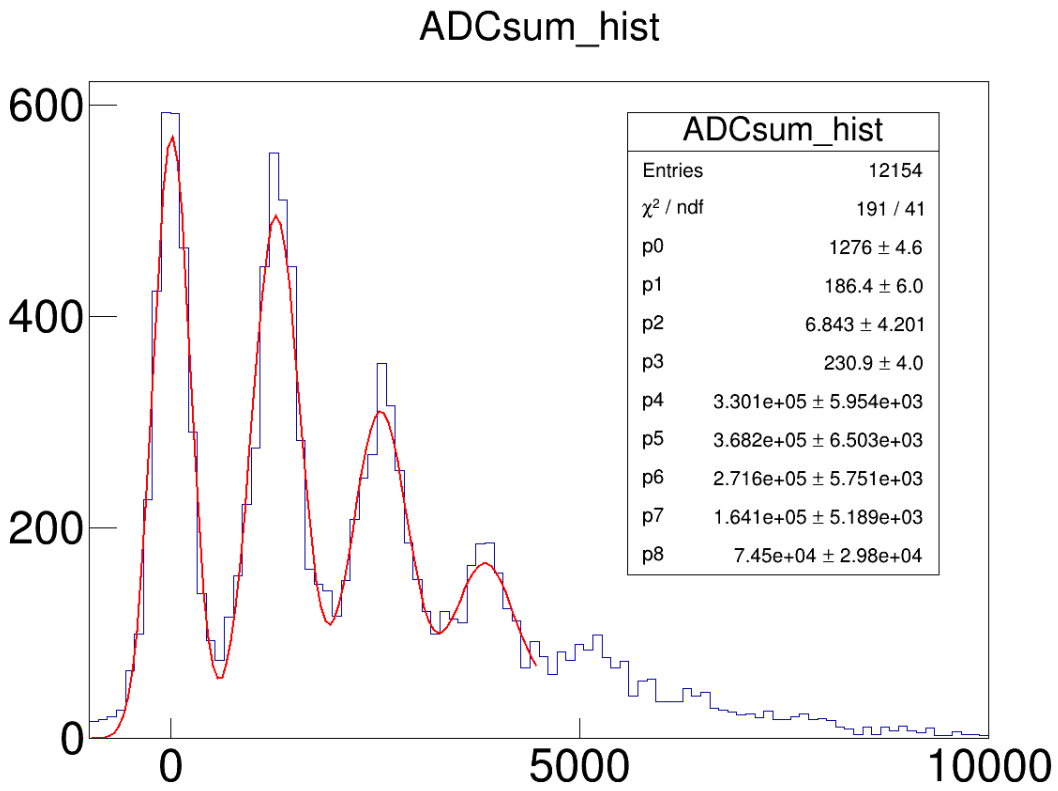



```
double fitting_ft(double *x, double *par)
{
    /*  par[0] : single photon ADC sum
    par[1] : sigma for single photon
    par[2] : parameter for Poisson distribution(mean and variation of Poission distribution)
    par[3] : normalization factor
    par[4] : pedestal mean
    par[5] : pedestal sigma
    par[6] : unknown pedstal    */
    double fitting_function = TMath::Poisson(0, par[2]) * TMath::Gaus(x[0], par[4], par[5], 1);
    for (int i = 1; i < 5; i++)
    {
        double n = (double)i;
        fitting_function += TMath::Poisson(n, par[2]) * TMath::Gaus(x[0], par[4] + n * par[0], TMath::Sqrt(par[5] * par[5] + n * par[1] * par[1]), 1);
    }
    fitting_function = par[3] * fitting_function;

    return fitting_function;
}
```

- Amplitudes of Gaussian은 Poisson distribution으로 결정된다고 생각하고 fitting을 해보았다.
- 이 때 Poisson distribution의 평균은 1.479로, 1280 ns에 평균 1.479개의 dark pulse가 생기는 것을 확인할 수 있다.

Average of the number of pulses?

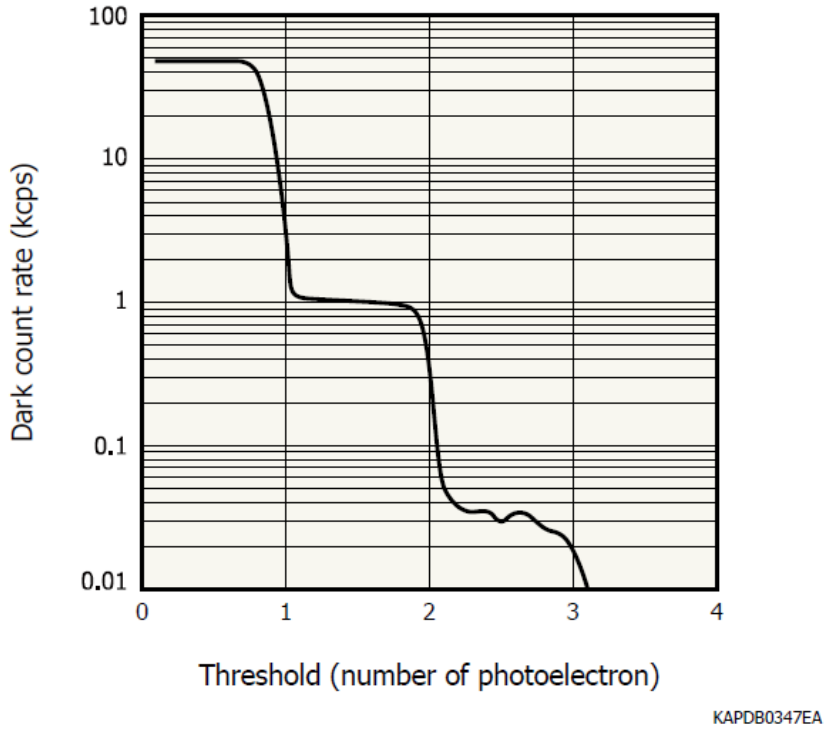


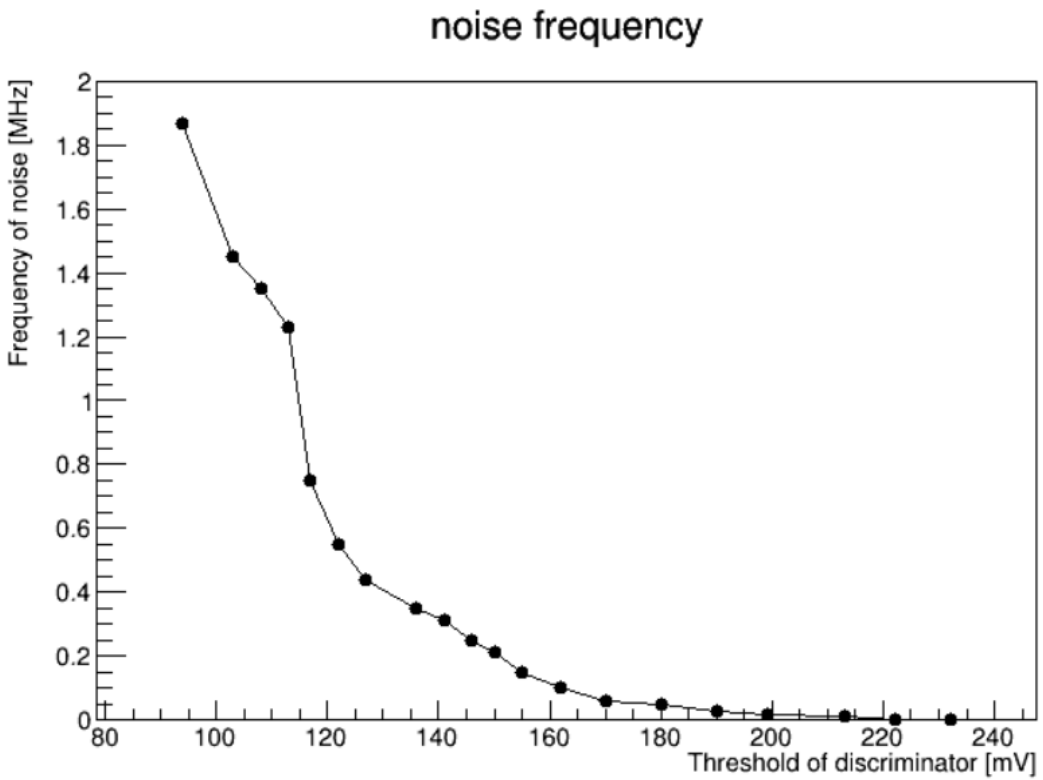

```
double fitting_ft(double *x, double *par)
{
    /*  par[0] : single photon ADC sum
    par[1] : sigma for single photon
    par[2] : Pedestal mean
    par[3] : pedestal sigma
    par[4]/[5]/[6]/[7]/[8] : normalization factor
    */
    double fitting_function = 0;
    for (int i = 0; i < 5; i++)
    {
        double n = (double)i;
        fitting_function += par[i+4] * TMath::Gaus(x[0], par[2] + n * par[0], TMath::Sqrt(par[3] * par[3] + n * par[1] * par[1]), 1);
    }
    return fitting_function;
}
```

- Gaussain의 Amplitude를 단순히 변수로 받아서 fitting을 진행하였다.
- 이 때, 평균 pulse의 수는 Gaussian의 amplitud로 가중치를 둔 평균으로 계산하였으며, 그 결과 약 1.40815 개로 나타났다.
- 앞의 100 point를 제외한 412 point의 시간은 1030 ns 로, Typically 2 MHz의 noise signal에 비교했을 때 작게 나타났음을 확인할 수 있다.

Dark Count Rate(data sheet)

[Figure 1-24] DCR vs. counting discriminator threshold level

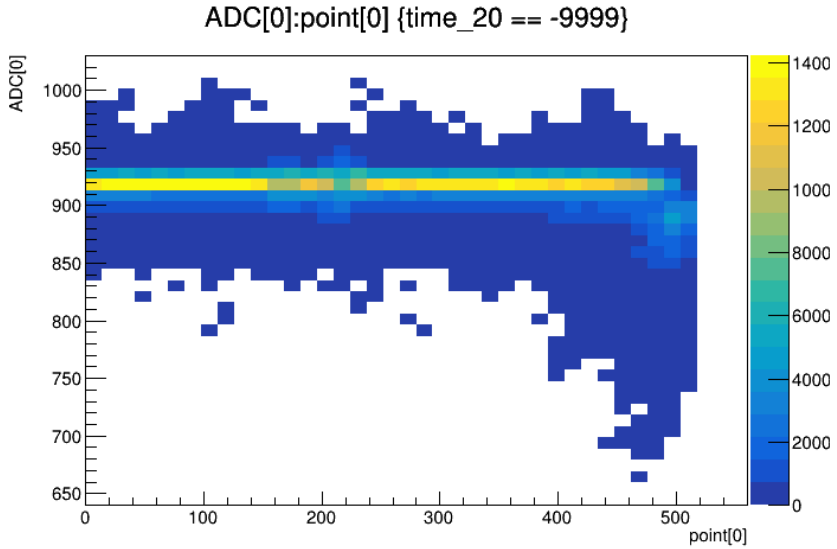


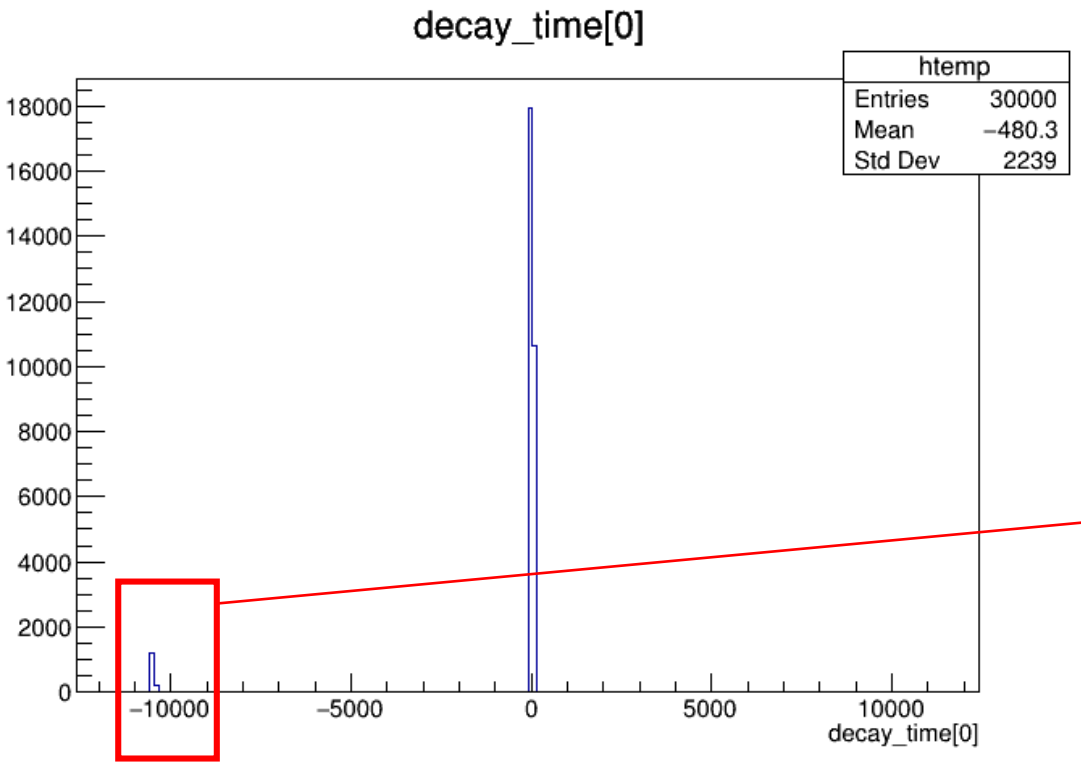


Peak finding algorithm

1. Signal에는 fall time(최대 높이의 20%와 80% 사이의 시간)이 있어 바로 pedestal range로 가지 않는다. 따라서 fall time이 작은 경우 event에서 제외한다.

fall time

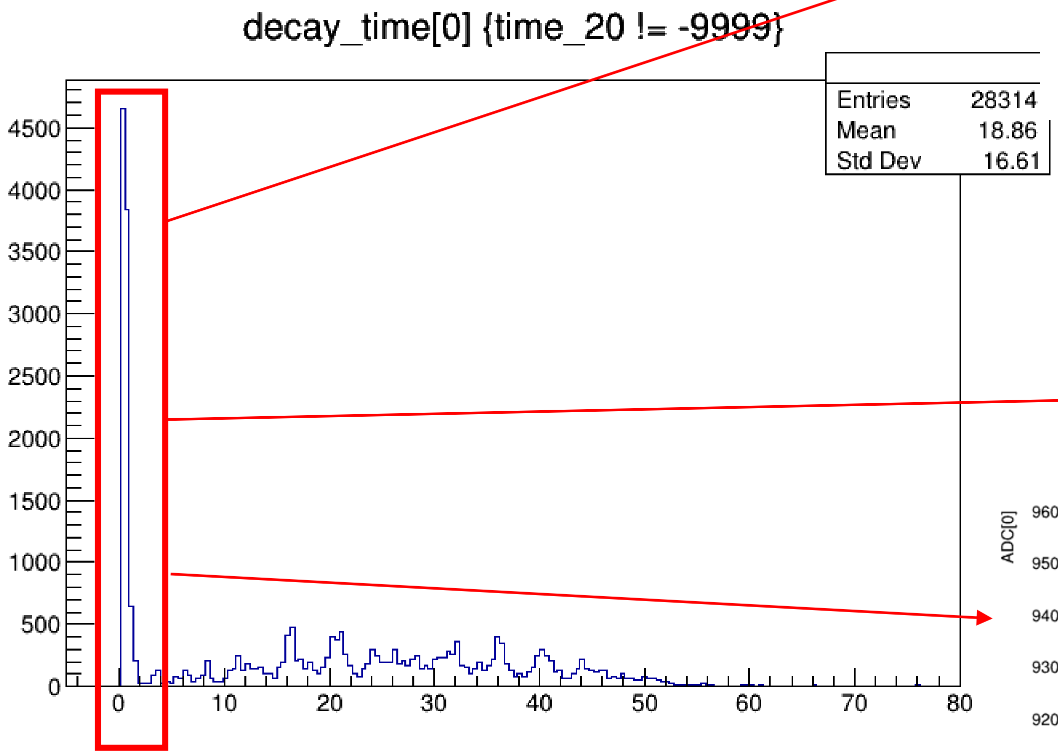


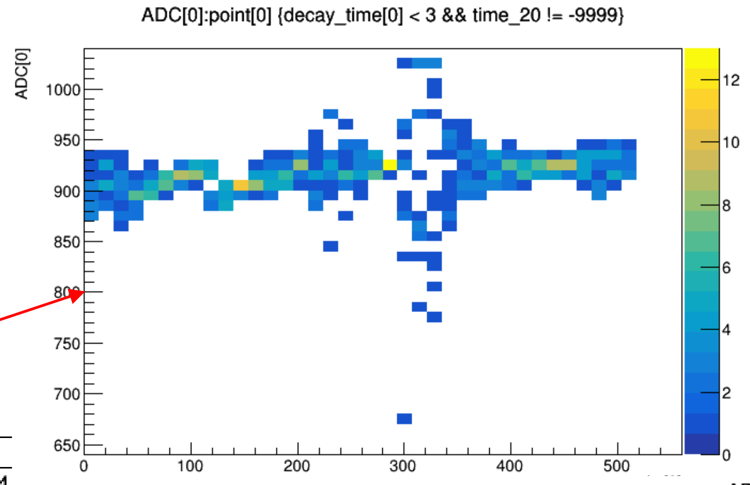


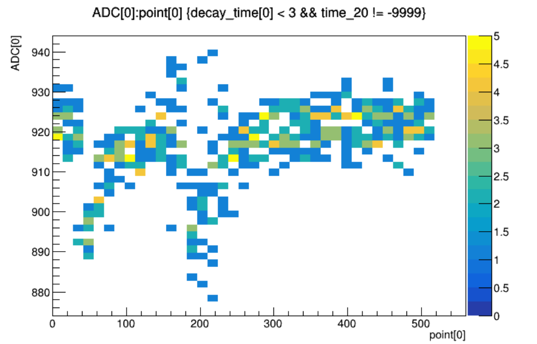
Row	time_20	time_80
40	-9999	499.46293
42	-9999	499.38866
45	-9999	500.5476
46	-9999	500.21133
54	-9999	500.71971
74	-9999	491.70494
83	-9999	495.1488
111	-9999	507.562
119	-9999	483.72230
140	-9999	469.56742
157	-9999	500.83163
167	-9999	487.3292
175	-9999	508.50042
233	-9999	471.29233
252	-9999	483.77909
260	-9999	497.42333
264	-9999	500.65533
280	-9999	492.50085
323	-9999	495.62233
346	-9999	471.01816
361	-9999	505.98871
390	-9999	507.9238
405	-9999	493.14
422	-9999	509.58737
431	-9999	507.9415

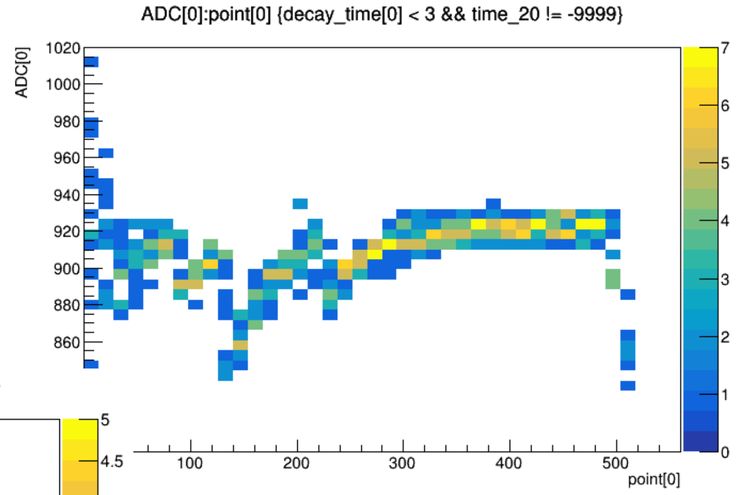
즉, peak가 뒤쪽에 생겨서 파형이 끝까지 나타나지 않은 pulse

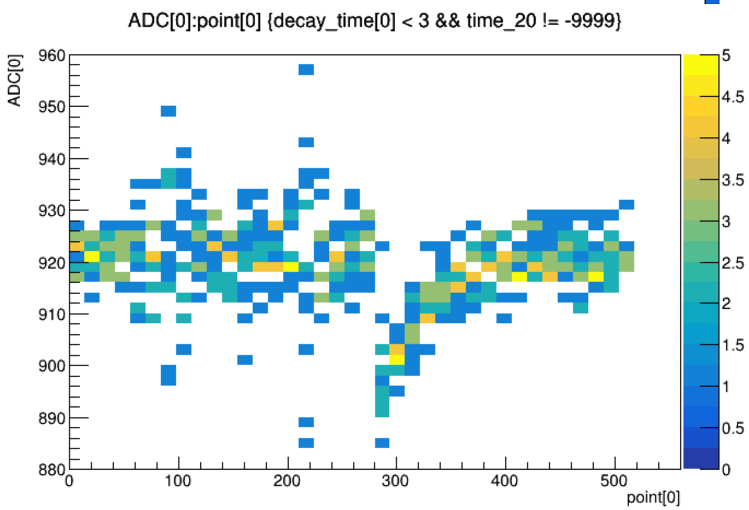
Fall time


decay_time[0] {time_20 != -9999}
Entries 28314
Mean 18.86
Std Dev 16.61

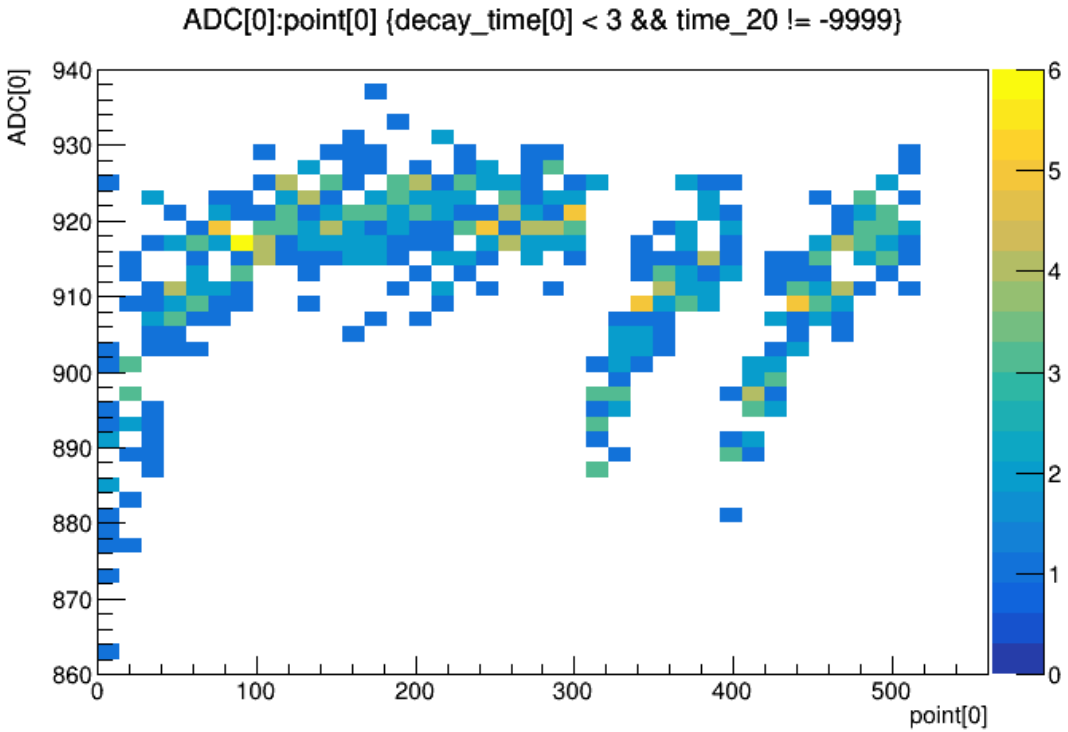

ADC[0]:point[0] {decay_time[0] < 3 && time_20 != -9999}
ADC[0]

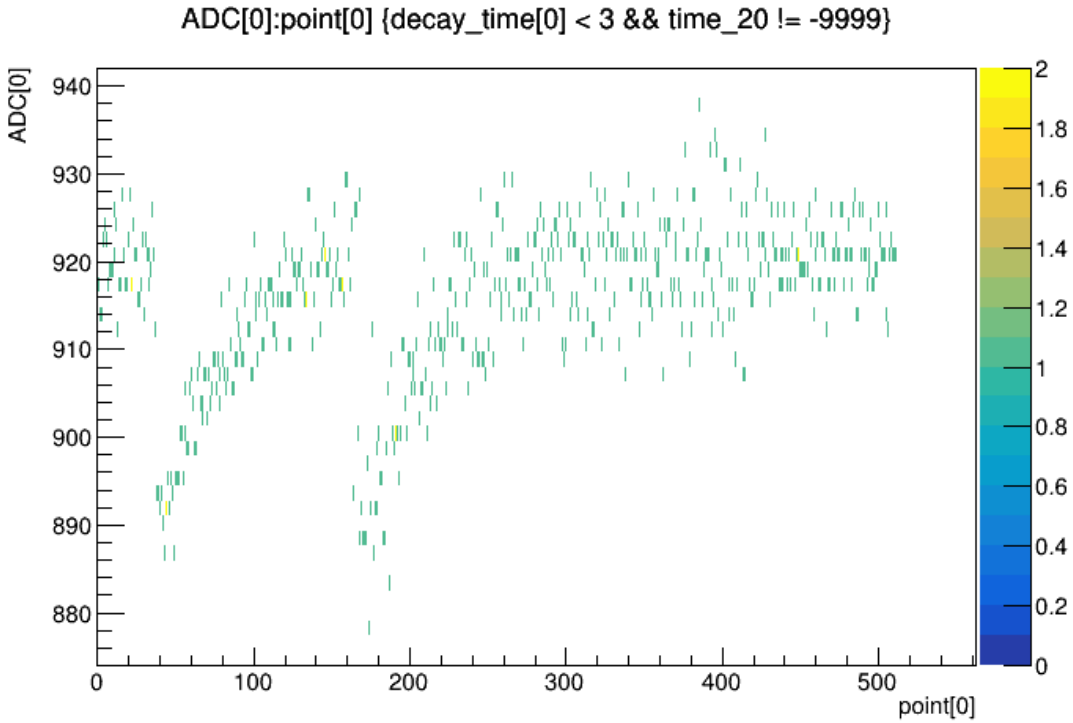

ADC[0]:point[0] {decay_time[0] < 3 && time_20 != -9999}
ADC[0]
point[0]


ADC[0]:point[0] {decay_time[0] < 3 && time_20 != -9999}
ADC[0]
point[0]


ADC[0]:point[0] {decay_time[0] < 3 && time_20 != -9999}
ADC[0]
point[0]

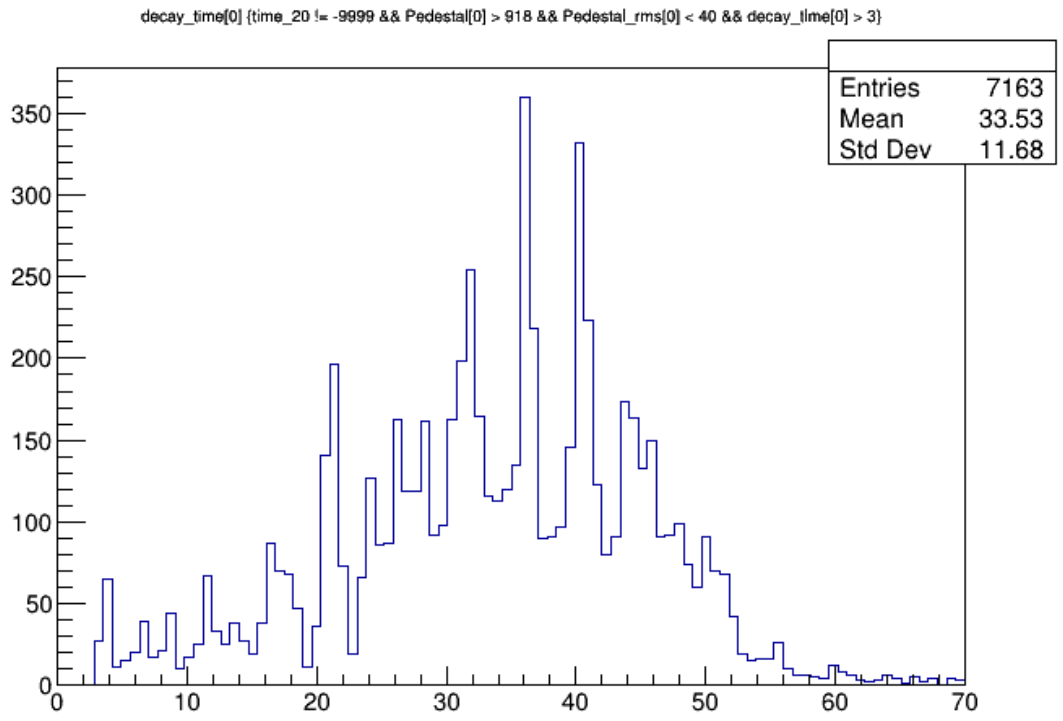
not noise





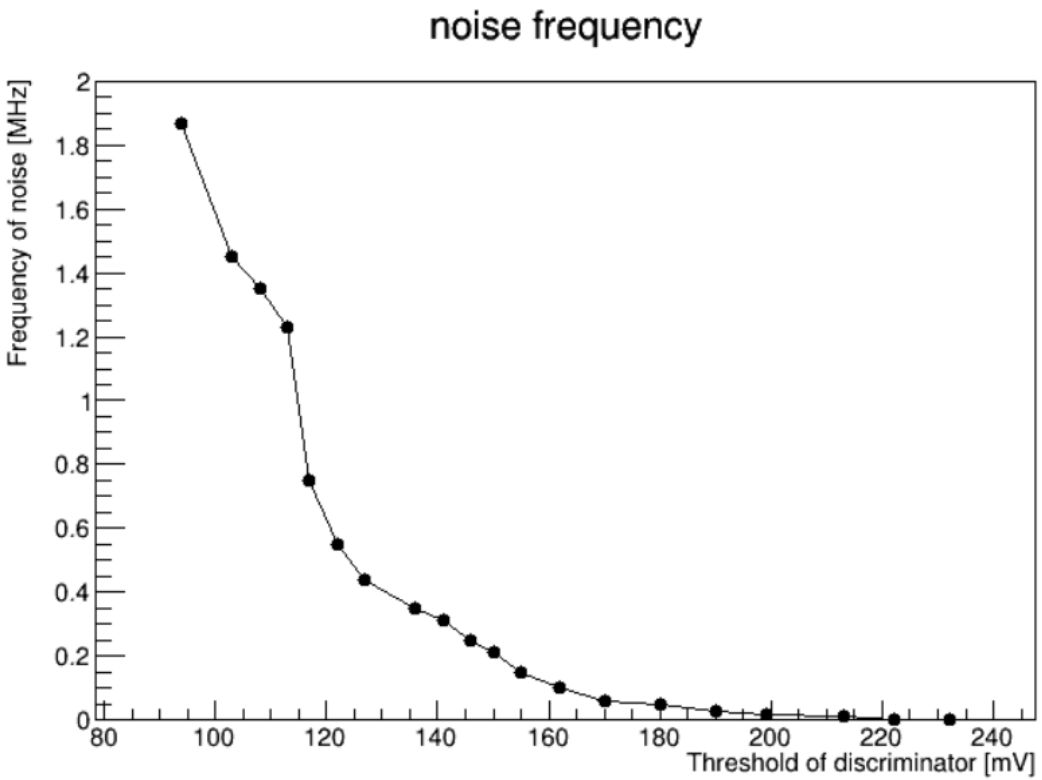
앞에서 100 point에서 pedesta을 지정하게 되는데 pedestal이 잘못 지정되어 decay time이 굉장히 짧게 설정된 경우
즉, 제대로 된 decay time을 보기 위해서는 pedestal이 잘 지정되었음을 확인하는 것도 중요하다.

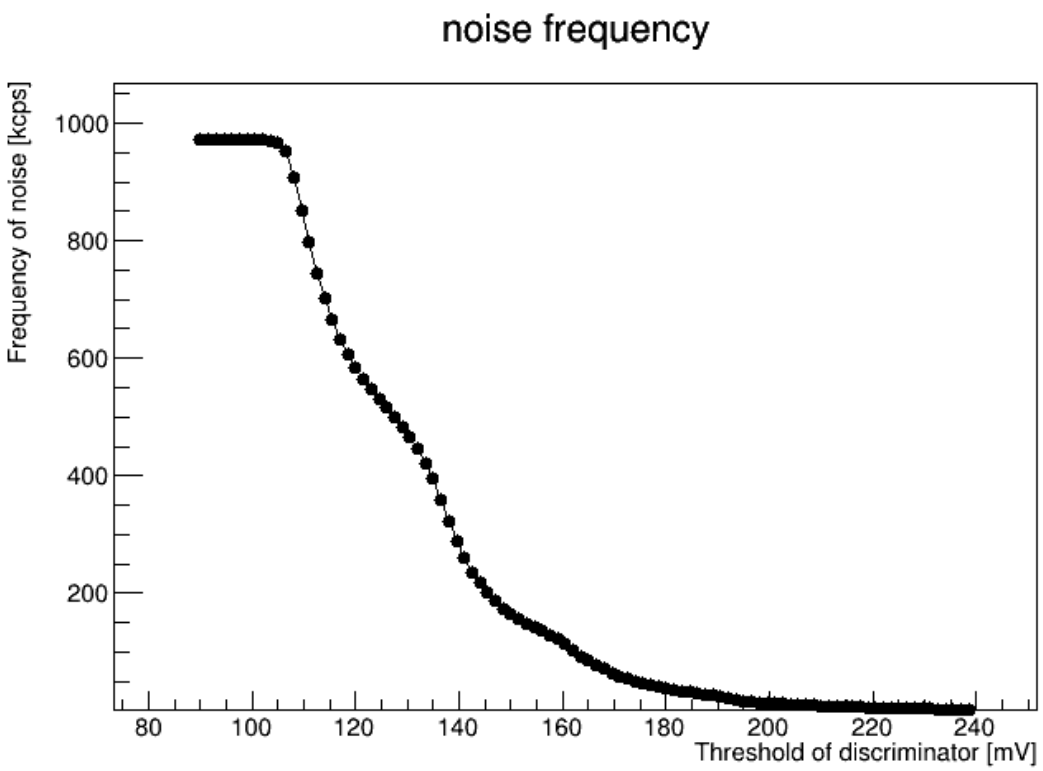
Cut condition for fall time



- 왼쪽의 그림에서 준 cut은 20%가 되는 지점에서의 시간이 지정될 것, pedestal이 918보다 클 것, pedestal region에 pulse가 없어서 rms 값이 작을 것, 이다.

Noise Frequency





문제점

- 각 Event마다 Count되는 횟수를 살펴봤을 때, 모두 0 아니면 1 밖에 나타나지 않음.
- 하지만 실제로는 2개의 pulse 가 들어온 경우도 있었기 때문에 수정할 필요가 있다고 생각함.