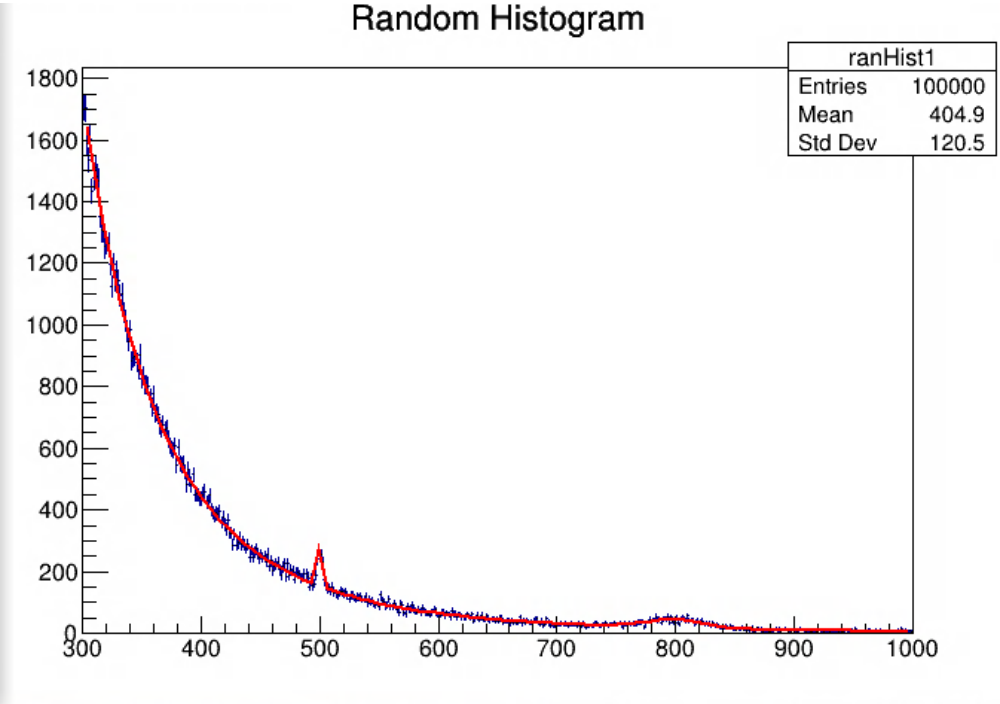
FIT2 : Sorawich Maichum – sm9cq

First plot:




```
Compiled based function: f1  based on a functor object.  Ndim = 1, Npar = 9
fit parameters
0       5.92814e+07 +- 2.1303e+06
1       4.7558 +- 0.0106997
2       33.4113 +- 0.25323
3       1.56478e+10 +- 2.06993e+10
4       500.199 +- 0.00192546
5       0.114271 +- 0.00402928
6       27.9137 +- 1.33492
7       799.332 +- 1.28057
8       25.2448 +- 1.13384
chi2 / ndf: 596.901 / 491 reduced chi2: 1.21568
 Pvalue: 0.000726216
```

The first plot of a given distribution which show the best fitting with above optimized fitting parameters. The p-value of this fitter is 0.000726216 which is very low.

Second plot:

```
// Functions with this interface may be used to construct a "TFunction"
Double_t myfunction(Double_t *xin, Double_t *par) {
  Double_t x=xin[0];
  Double_t bkgScale=par[0];
  Double_t alpha=par[1];
  Double_t beta=par[2];
  Double_t background =TMath::Exp(-0.5*(x-alpha)*(x-alpha)/beta/beta);/
s/ pow(x/beta,-1.0*alpha);
  Double_t A1=par[3];
  Double_t mu1=par[4];
  Double_t sig1=par[5];
  Double_t peak1=A1*TMath::Exp(-0.5*(x-mu1)*(x-mu1)/sig1/sig1);
  Double_t A2=par[6];
  Double_t mu2=par[7];
  Double_t sig2=par[8];
  Double_t peak2=A2*exp(-0.5*(x-mu2)*(x-mu2)/sig2/sig2);
//Double_t A3=par[9];
  //Double_t mu3=par[10];
  //Double_t sig3=par[11];
  //Double_t peak3=A3*exp(-0.5*(x-mu3)*(x-mu3)/sig3/sig3);
  Double_t peaks=par[9]*sin(par[10]*x)*exp(-0.5*(x-par[9])*(x-par[9])/p
sar[10]/par[10]);
  return bkgScale*background+peak1+peak2+peaks;
```

This is my guessing function.

$$F(x) = Ae^{\frac{(x-a)^2}{2b^2}} + Be^{\frac{(x-c)^2}{2d^2}} + Ce^{\frac{(x-e)^2}{2d^2}} + Dsin(wx) * e^{\frac{(x-D)^2}{2w^2}}$$

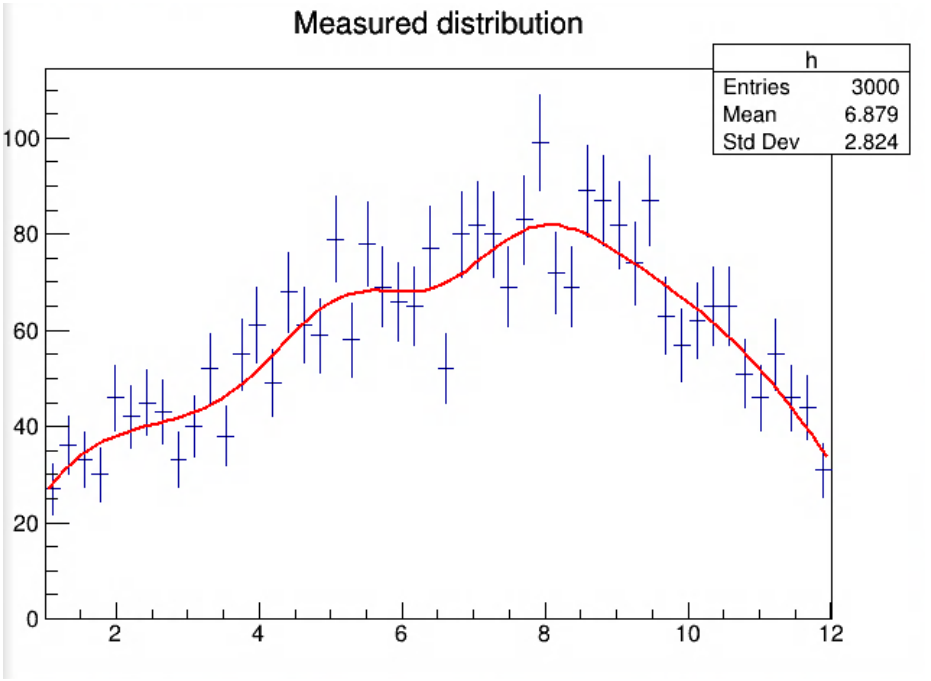
When I looked at the distribution, I believed that it might be compose of a Big Gaussian distribution and some oscillations.

On my first try, it wasn't good. Afterward, I add 2 gaussians and hope that will make a plot of fitting function better.

Then, I set those parameters like this.

```
/**/
//f1->SetParameters(1,1,1,1,1,1,1,1,1,1);
f1->SetParameters(7,9,5,10,12,2,9,9,5,8,1.5);
```

This is what I get after use the above function and parameters.

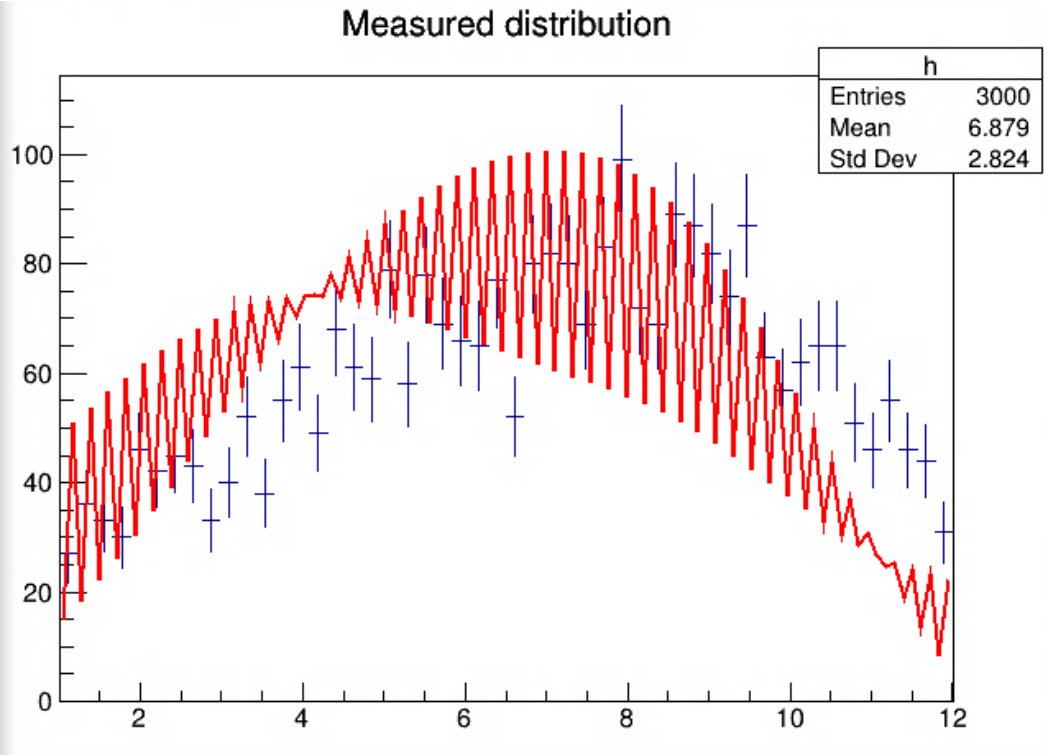



```
Compiled based function: f1  based on a functor object.  Ndim = 1, Npar = 11
fit parameters
0       71.7697 +- 17.2539
1       0.262257 +- 0.565127
2       1.83805 +- 0.324284
3       6479.11 +- 9.73379
4       7.67461 +- 0.159606
5       35.081 +- 1.67691
6       -6418.79 +- 9.72594
7       -65.3215 +- 16.0496
8       948.158 +- 182.355
9       6.24271 +- 1.2092
10      1.67554 +- 0.0718284
chi2 / ndf: 48.3267 / 39 reduced chi2: 1.23915
 Pvalue: 0.145456
```

I think this one is quite a good plot for this distribution with the p-value is 0.145456.

But I tried some adjustment of those parameters to play around and hope I can get better plot with smaller p-value.

Fortunately, I found that the p-value is 0.0658615 which is smaller than a previous one. But the fitting of distribution is very peculiar and seem not reasonable. (The plot is provided below)



This is what I get for the smaller p-value with the starting parameters which are all 1. It should be better but, in my opinion, the previous one should be better.

```
/**/
f1->SetParameters(1,1,1,1,1,1,1,1,1,1,1);
//f1->SetParameters(7,9,5,10,12,2,9,9,5,8,1.5);
```

```
Compiled based function: f1  based on a functor object.  Ndim = 1, Npar = 11
fit parameters
0       75.4369 +- 1.60541
1       4.93453 +- 0.0865228
2       2.97456 +- 0.107089
3       35.9378 +- 3.0003
4       8.88663 +- 0.153083
5       1.87017 +- 0.125657
6       14.5135 +- 15.9165
7       29.2214 +- 0.103193
8       2.30298 +- 12.1207
9       25.6496 +- 3.98252
10      28.1104 +- 0.0162328
chi2 / ndf: 53.0691 / 39 reduced chi2: 1.36075
 Pvalue: 0.0658615
```

The optimized parameters and it p-value.