Poisson Statistics

Sungur Özkan
Boğaziçi University • Physics Department

***Abstract*—In this experiment, we have shown that Radioactive Decay is a random process that can be modeled with Poisson distribution. We have used Caesium and Barium as Gamma-ray Sources and recorded the emissions of radiation with a Geiger Counter for 1-s and 10-s intervals. We have build the histogram from the data and found that both Gaussian and Poisson distributions fit the data very well. In the last part, we have observed that the time elapsed between successive decaying events are distributed exponentially, which is another fundamental property of Poisson statistics and a proof for our initial assumption.**

I. Theory

The Poisson distribution was first introduced by Simeon Dennis Poisson, in 1837[1]. It is a discrete probability distribution that describes the probability of a certain number of events happening in a specific time interval. One can observe the Poisson distribution almost in all counting processes[2]. The Poisson distribution is a suitable model if the following assumptions are true:

- The occurence of one event does not affect the occurence of another within the same interval.
- The rate at which events occur is constant, which means you know the mean number of events occurring within a given interval of time.
- Events must be countable; they cannot be continuous. For example, the number of accidents at a corner in a year.

An important property of the Poisson distribution is that the variable "n" must be a random variable, meaning the events that happen must occur randomly. An example for this is the radioactive decay, which we have used in this experiment. Radioactive decay is the process by which an unstable atomic nucleus loses its energy by radiation. Radioactive decay is a random process at the level of single atoms. According to quantum theory, it is impossible to predict when a particular atom will decay, regardless of how long the atom has existed. However, for a significant number of identical atoms, the overall decay rate can be expressed as half-life[3].

The normalized Poisson distribution function is typically denoted as:

$$P_{\mathrm{P}}(n,\mu) = \frac{e^{-\mu}\mu^n}{n!}$$

where n is the number of events, and μ is the mean. The normalization condition for the Poisson distribution is expressed as:

$$\sum_{n=0}^{\infty} P_{\mathrm{P}}(n,\mu) = 1$$

In the Poisson distribution, the standard deviation is equal to the square root of the mean (μ), expressed as:

$$\sigma = \sqrt{\mu}$$

where σ represents the standard deviation.

The Poisson distribution is a special case of the binomial distribution, similar to the Gaussian distribution being a special case. When the average number of events gets larger, the Poisson distribution approaches to the Gaussian distribution. The probability density function of the Gaussian distribution with mean μ and standard deviation σ evaluated at x is given by:

$$P_G(\mu,\sigma,x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

But, the Poisson distribution shows an exponential behavior when it comes to the distribution of successive events [2]. The probability of observing n counts during a time interval t is:

$$P(\mu,n) = P(\alpha,t,n) = \frac{(\alpha t)^n e^{-\alpha t}}{n!} \tag{5}$$

where $\alpha = \mu/t$. Then, the probability of having one event in a time interval dt is:

$$P(\alpha,dt,1) = \frac{(\alpha dt)e^{-\alpha dt}}{1!} \tag{6}$$

Next, the probability of having n events in an interval t followed by an event within dt is found by multiplying the two probabilities in the previous equations:

$$P_p(n+1,t)\,dt = P(\alpha,t,n)\,P(\alpha,dt,1) \tag{7}$$

Then, we have:

$$P_p(n+1,t)\,dt = \frac{(\alpha t)^n e^{-\alpha t}}{n!} \cdot \frac{(\alpha dt)e^{-\alpha dt}}{1!} \tag{8}$$

Now, since dt is infinitesimal, we can make the approximation that $e^{-\alpha dt} \approx 1$. So, (8) becomes:

$$P_p(n+1,t)\,dt = \frac{(\alpha t)^n e^{-\alpha t}}{n!} \cdot \frac{(\alpha dt)}{1!} \tag{9}$$

If we divide both sides with dt, we get:

$$P_p(n+1,t) = \frac{(\alpha t)^n e^{-\alpha t}\alpha}{n!} \tag{10}$$

In the experiment, we will have successive events for n = 0 and 1. Then,

$$P_p(n=0+1,t) = e^{-\alpha t}\alpha \tag{11}$$

$$P_p(n=1+1,t) = te^{-\alpha t}\alpha^2 \tag{12}$$

II. The Experimental Setup

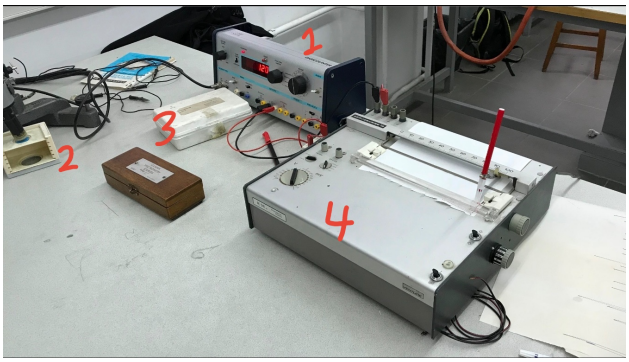

Fig. 1. The setup

- Geiger Counter with a scaler: An instrument used for detecting and measuring ionizing radiation such as alpha particles, beta particles and gamma rays. It consists of a Geiger–Müller tube (the sensing element which detects the radiation) and the processing electronics, which display the result. The tube is filled with an inert gas such as helium, neon, or argon at low pressure, to which a high voltage is applied. The tube briefly conducts electrical charge when high energy particles or gamma radiation make the gas conductive by ionization[4].
- Sample holder
- Caesium and Barium gamma ray sources
- Chart recorder

III. Method

The procedure of the experiment is as follows:

Part One:

1) A Caesium gamma ray source is placed on the sample holder to decide the operating voltage of the Geiger counter. The counter is set to 100 s, radioactivity and single mode. The number of counts are recorded in 100 s intervals for High Voltage in the range from 300V to 500V at 20V intervals. We expect a graph like this:

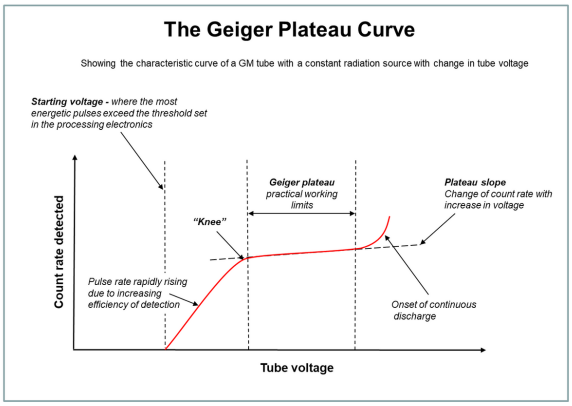

Fig. 2. Geiger counter graph

A HV value on the plateau (flat region) is selected. This is the region where the counter behaves the most stable.

2) The counter is set to the decided HV value.
3) The counter is set to 10s intervals, and the gamma ray source on the sample holder is placed such that you get 100 counts in 10s intervals.
4) The number of counts in 10s intervals are recorded for 100 times.
5) The counter is set to 1s intervals and the number of counts are recorded for 100 times again.
6) Gamma ray source is changed from Caesium to Barium.
7) The sample holder is placed such that you get 20-30 counts in 10s intervals.
8) Step 4 and Step 5 are repeated.
9) We end up with 4 data sets.

Part Two:

10) In this part, we examine the time intervals between consecutive counts by using the chart recorder. Barium gamma ray source is placed on the holder such that you get approximately 1 count each second.
11) The chart recorder is started and run for about 2 minutes. The distance between successive counts are measured. (take 1mm = 1s)

IV. The Data

Here is an example table of how some of our data looks like:

TABLE I
The Number Of Counts Table for Source/Interval

Caesium/1sec	Caesium/10sec	Barium/1sec	Barium/10sec
10	140	3	17
20	138	2	26
9	147	2	31
10	132	1	18
19	140	2	16
12	152	4	19
15	137	2	20
24	152	2	28
14	147	4	22
11	151	5	31

We filled histograms to observe our data, here is an example, the rest can be found in the appendix:

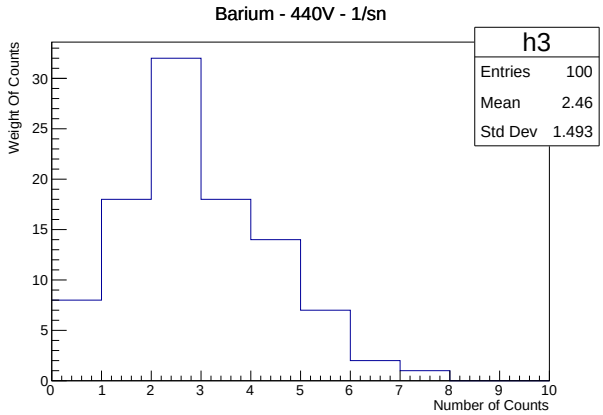

Fig. 3. Barium / 1 second intervals

V. The Analysis

The analysis was made using CERN's framework ROOT, version v6.30.0 The plot obtained for calibrating the Geiger counter is:

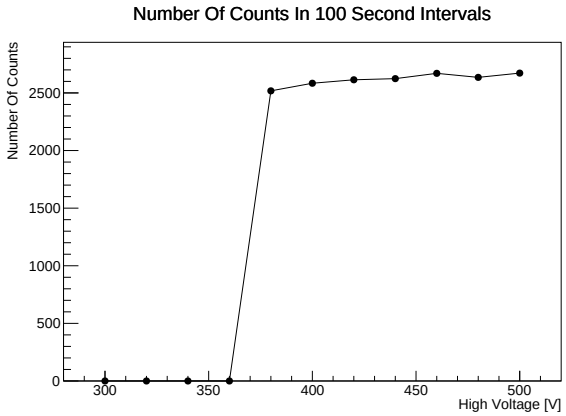

Fig. 4. Number of counts as a function of High Voltage

The range from 400V to 480V looks ideal to choose for the High Voltage. We have chosen to set 440V.

After taking the datasets and obtaining their means and standard deviations, we calculated $\sqrt{\mu}/\sigma$ to see how it looks as a function of σ.

TABLE II
$\sqrt{\mu}/\sigma$ Data Table

$\sqrt{\mu}/\sigma$	σ
1.08708	3.48103
1.01106	11.5918
1.05068	1.49278
0.948039	5.16318

Here is the plot of the values:

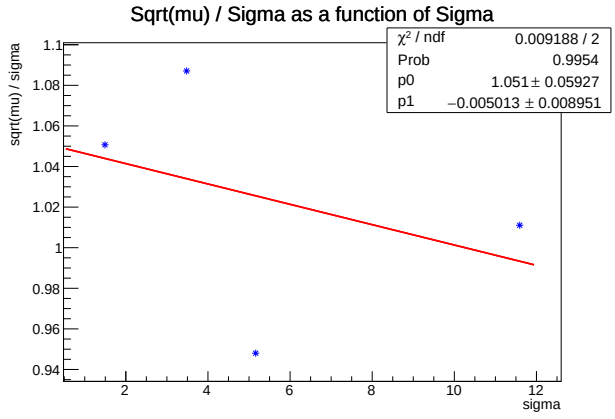

Fig. 5. $\sqrt{\mu}/\sigma$ as a function of σ

These data and plot shows that $\sqrt{\mu}/\sigma$ values are near 1 for all datasets, which means that the datasets show Poisson Distribution behavior.

Then, we have fitted Gaussian (red) and Poisson (blue) distribution functions to our histograms. The best fit results for Gaussian distribution are printed on the legends.

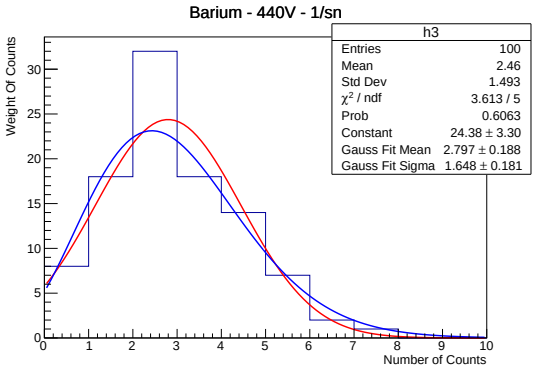

Fig. 6. Barium / 1s intervals

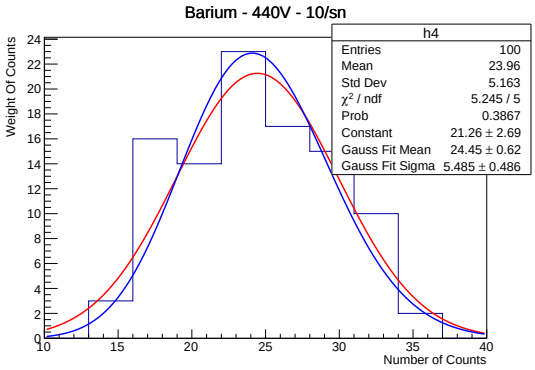

Fig. 7. Barium / 10s intervals

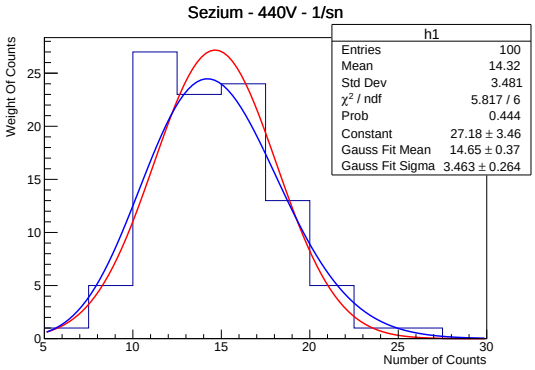

Fig. 8. Caesium / 1s intervals

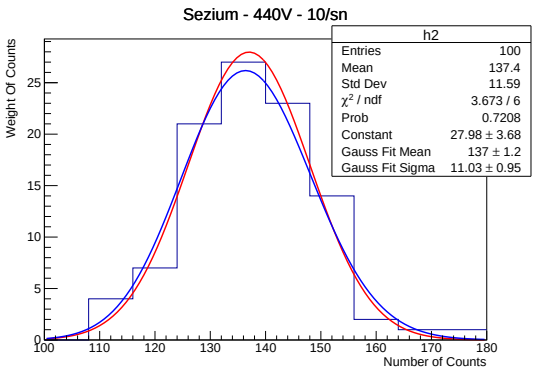

Fig. 9. Caesium / 10s intervals

TABLE III
POISSON ALPHA VALUES

	Poisson α values
Caesium/1sec	14.7178
Caesium/10sec	136.88
Barium/1sec	2.9451
Barium/10sec	24.6013

TABLE IV
χ^2 VALUES

	Gaussian χ^2	Poisson χ^2
Caesium/1sec	0.9695	0.911
Caesium/10sec	0.612	0.644
Barium/1sec	0.722	0.664
Barium/10sec	1.049	1.102

For the second part, we have plotted the time differences between successive pulses for n=0 and n=1. Then, we have used the poisson functions (11, 12) in the theory part to fit these histograms. The best fit stats and the χ^2 values are printed on the legends.

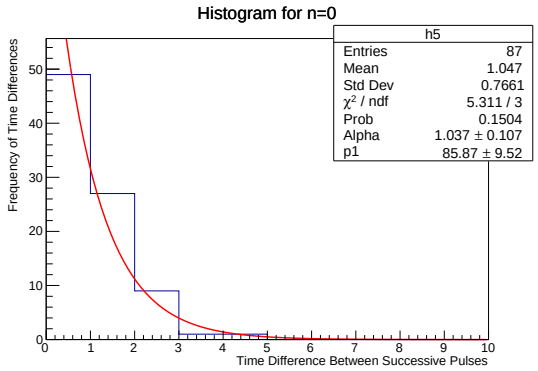

Fig. 10. Time Differences for n=0

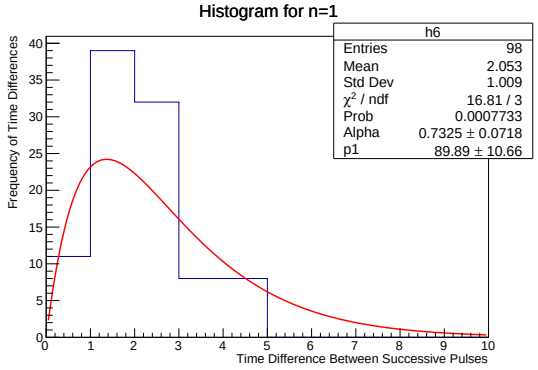

Fig. 11. Time Differences for n=1

VI. THE RESULT

The total time elapsed is 102.9 seconds and we have 99 peaks, so, the expected value of alpha is 1.039 for n=0 and 1.05 for n=1. Our results are:

$$\alpha_{n=0} = 1.037 \pm 0.107 \quad (1)$$

$$\alpha_{n=1} = 0.7325 \pm 0.0718 \quad (2)$$

Our results are 0.002σ away for n=0 and 0.31σ away for n=1.

VII. THE CONCLUSION

In the first part of the experiment we have seen that the Gaussian and Poisson fits are very similar to each other and they fit the data very well. When we check the values of χ^2, they are close, but Gaussian fits better for Caeisum-10s and Barium-10s, which have larger means. We have also seen that this radioactive decay can be modeled with Poisson distribution because it shows the strong correlation between $\sqrt{\mu}$ and σ which we observe in Poisson distribution. So our assumption in the beginning of the experiment holds. For the second part, Our result is almost identical for n=0, whereas it is 0.31σ away from the real value. The cause of error here can be the wrong measurement of the distances between the peaks. Second part has shown us that the time intervals between successive emission of radiations show exponential distribution. This is another proof for our assumption that radioactive decay is a random process governed by Poisson statistics.

REFERENCES

[1] *Poisson Distribution - Wikipedia*. URL: https://en.wikipedia.org/wiki/Poisson_distribution.

[2] E. Gülmez. *Advanced Physics Experiments*. 1st. Boğaziçi University Publications, 1999.

[3] *Radioactive decay - Wikipedia*. URL: https://en.wikipedia.org/wiki/Radioactive_decay.

[4] *Geiger counter - Wikipedia*. URL: https://en.wikipedia.org/wiki/Geiger_counter.

VIII. Appendix

Other histograms for raw data:

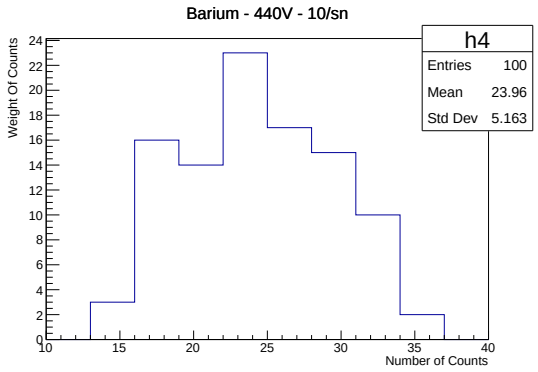

Fig. 12. Barium - 440V - 10s

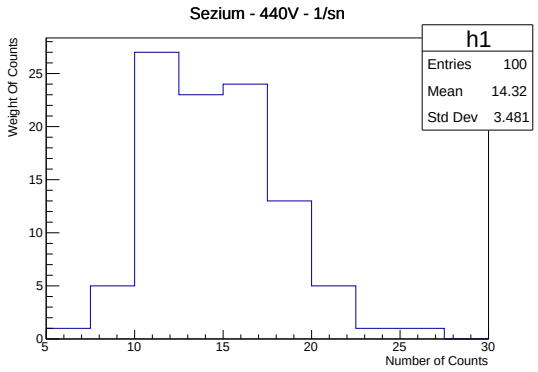

Fig. 13. Caesium - 440V - 1s

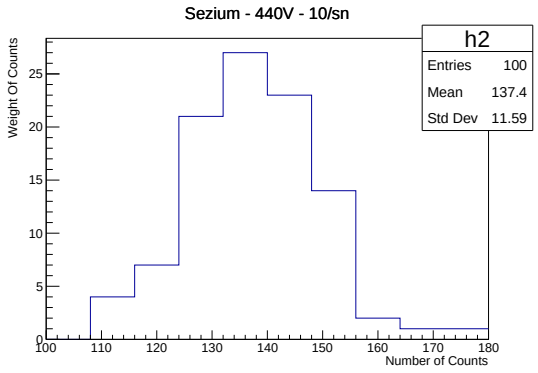

Fig. 14. Caesium - 440V - 10s

```
{
#include <iostream>
#include <vector>
#include <string>
#include <cmath>
#include <utility>

{
gStyle->SetOptFit(1111);
TF1 *fnew = new TF1("fnew", "gaus");
TF1 *f2 = new TF1("f2",
    "[0]*TMath::Poisson(x,[1])",0,30);
f2->SetLineColor(kBlue);

TCanvas *c1 = new TCanvas();
TGraphErrors cal("data_Sheet1.txt");
cal.GetXaxis()->SetTitle("High Voltage [V]");
cal.GetYaxis()->SetTitle("Number Of Counts");
cal.SetTitle("Number Of Counts In 100 Second
    Intervals");
cal.SetLineColor(kBlack);
cal.SetMarkerStyle(20);
cal.Draw();
c1->Print("calibration.pdf");

f2->SetParName(0, "Amplitude Scale");
f2->SetParName(1, "Alpha");
fnew->SetParName(1, "Gauss Fit Mean");
fnew->SetParName(2, "Gauss Fit Sigma");

TCanvas *c2 = new TCanvas();
TH1F *h1 = new TH1F("h1","Sezium - 440V -
    1/sn",10, 5, 30);
std::ifstream file1("data_Sheet2.txt");
h1->GetXaxis()->SetTitle("Number of Counts");
h1->GetYaxis()->SetTitle("Weight Of Counts");

float datum1;
while (file1>>datum1) h1->Fill(datum1);
double meanSezyum1 = h1->GetMean();
double stdSezyum1 = h1->GetStdDev();
fnew->SetRange(5,30);
h1->Fit("fnew", "S");
f2->SetParameters(h1->GetMaximum(),
    h1->GetMean());
f2->SetRange(5,30);
h1->Fit("f2", "S+");
h1->Draw();
//fnew->Draw("same");
//f2->Draw("same");
double chiSquare11 = fnew->GetChisquare();
int ndf11 = fnew->GetNDF();
double chindf11 = chiSquare11 / ndf11;
double alpha1 = f2->GetParameter(1);

double chiSquare12 = f2->GetChisquare();
int ndf12 = f2->GetNDF();
double chindf12 = chiSquare12 / ndf12;
c2->Print("Seizum-1.pdf");



TCanvas *c3 = new TCanvas();
TH1F *h2 = new TH1F("h2","Sezium - 440V -
    10/sn",10, 100, 180);
std::ifstream file2("data_Sheet3.txt");
h2->GetXaxis()->SetTitle("Number of Counts");
h2->GetYaxis()->SetTitle("Weight Of Counts");
float datum2;
while (file2>>datum2) h2->Fill(datum2);
double meanSezyum10 = h2->GetMean();
double stdSezyum10 = h2->GetStdDev();
fnew->SetRange(100,180);
h2->Fit("fnew", "S");
f2->SetParameters(h2->GetMaximum(),
    h2->GetMean());
f2->SetRange(100,180);
h2->Fit("f2", "S+");
h2->Draw();
```

```
//fnew->Draw("same");
//f2->Draw("same");
double chiSquare21 = fnew->GetChisquare();
int ndf21 = fnew->GetNDF();
double chindf21 = chiSquare21 / ndf21;

double chiSquare22 = f2->GetChisquare();
int ndf22 = f2->GetNDF();
double chindf22 = chiSquare22 / ndf22;
c3->Print("Seizum-10.pdf");
double alpha2 = f2->GetParameter(1);



TCanvas *c4 = new TCanvas();
TH1F *h3 = new TH1F("h3","Barium - 440V -
    1/sn",10, 0, 10);
h3->GetXaxis()->SetTitle("Number of Counts");
h3->GetYaxis()->SetTitle("Weight Of Counts");
std::ifstream file3("data_Sheet4.txt");
float datum3;
while (file3>>datum3) h3->Fill(datum3);
//h3->Rebin(10);
double meanBarium1 = h3->GetMean();
double stdBarium1 = h3->GetStdDev();
fnew->SetRange(0,10);
h3->Fit("fnew", "S");
f2->SetParameters(h3->GetMaximum(),
    h3->GetMean());
f2->SetRange(0,10);
h3->Fit("f2", "S+");
h3->Draw();
//fnew->Draw("same");
//f2->Draw("same");
double chiSquare3 = fnew->GetChisquare();
int ndf3 = fnew->GetNDF();
double chindf3 = chiSquare3 / ndf3;
double chiSquare32 = f2->GetChisquare();
int ndf32 = f2->GetNDF();
double chindf32 = chiSquare32 / ndf32;
c4->Print("Barium-1.pdf");
double alpha3 = f2->GetParameter(1);



TCanvas *c5 = new TCanvas();
TH1F *h4 = new TH1F("h4","Barium - 440V -
    10/sn",10, 10, 40);
std::ifstream file4("data_Sheet5.txt");
h4->GetXaxis()->SetTitle("Number of Counts");
h4->GetYaxis()->SetTitle("Weight Of Counts");
float datum4;
while (file4>>datum4) h4->Fill(datum4);

double meanBarium10 = h4->GetMean();
double stdBarium10 = h4->GetStdDev();
fnew->SetRange(10,40);
h4->Fit("fnew", "S");
f2->SetParameters(h4->GetMaximum(),
    h4->GetMean());
f2->SetRange(10,40);
h4->Fit("f2", "S+");
h4->Draw();
//fnew->Draw("same");
//f2->Draw("same");
double chiSquare4 = fnew->GetChisquare();
int ndf4 = fnew->GetNDF();
double chindf4 = chiSquare4 / ndf4;
double chiSquare42 = f2->GetChisquare();
int ndf42 = f2->GetNDF();
double chindf42 = chiSquare42 / ndf42;
c5->Print("Barium-10.pdf");
double alpha4 = f2->GetParameter(1);


double means[] = {meanSezyum1, meanSezyum10,
    meanBarium1, meanBarium10};
double sigmas[] = {stdSezyum1, stdSezyum10,
    stdBarium1, stdBarium10};
double sqr[4];
  for (int i = 0; i < 4; i++) {
      double x = std::sqrt(means[i]) /
          sigmas[i];
      sqr[i] = x;
   }

TCanvas *c6 = new TCanvas();
TGraphErrors gr(4, sigmas, sqr);
gr.SetTitle("Sqrt(mu) / Sigma as a function of
    Sigma");
gr.SetLineColor(kBlack);
gr.SetMarkerColor(kBlue);
gr.SetMarkerStyle(30);
gr.GetXaxis()->SetTitle("sigma");
gr.GetYaxis()->SetTitle("sqrt(mu) / sigma");
TF1 *fitFunc = new TF1("fitFunc", "[0] +
    [1]*x",0,12);
//fitFunc->SetParameter(0,
    initialGuessForParameter0);
//fitFunc->SetParameter(1,
    initialGuessForParameter1);
gr.Fit("fitFunc", "R");
gr.Draw("A*");
fitFunc->Draw("same");
c6->Print("line.pdf");


// PART2


TF1 *f3 = new TF1("f3",
    "[1]*[0]*TMath::Exp(-[0]*x)");
TF1 *f4 = new TF1("f4",
    "[1]*[0]*[0]*x*TMath::Exp(-[0]*x)");
f3->SetParName(0, "Alpha");
f4->SetParName(0, "Alpha");
TCanvas *c7 = new TCanvas();
TH1F *h5 = new TH1F("h5","n=0",10, 0.0, 10.0);
h5->SetTitle("Histogram for n=0");

h5->GetXaxis()->SetTitle("Time Difference
    Between Successive Pulses");
h5->GetYaxis()->SetTitle("Frequency of Time
    Differences");
std::ifstream file5("data_Sheet9.txt");
float datum5;
while (file5>>datum5) h5->Fill(datum5);
f3->SetParameters(h5->GetMean(),h5->GetMaximum());
f3->SetRange(0.1 ,10);
h5->Fit("f3");
h5->Draw();
c7->Print("n0.pdf");

TCanvas *c8 = new TCanvas();
TH1F *h6 = new TH1F("h6","n=1",10, 0.0, 10.0);
h6->GetXaxis()->SetTitle("Time Difference
    Between Successive Pulses");
```

```
h6->GetYaxis()->SetTitle("Frequency of Time
    Differences");
h6->SetTitle("Histogram for n=1");
std::ifstream file6("data_Sheet10.txt");
float datum6;
while (file6>>datum6) h6->Fill(datum6);
f4->SetParameters(h6->GetMean(),
    h6->GetMaximum());
f4->SetRange(0,10);
h6->Fit("f4");
h6->Draw();
c8->Print("n1.pdf");


}

}
```