实验物理中的统计方法 作业7

1.

习题 4.4. 表4.1是实验获取的分区间数据和理论预言值。第二、三列是区间边界，第四列是对应区间的观测事件数 n_i（$i=1,\ldots,20$），服从泊松分布。第五、六列是两种理论对期待值 $\nu_i = E[n_i]$ 的预言，如图4.2所示。

(*a*)写一段程序，将表中20个区间的实验观测值和理论预期值画成直方图，画到一张图上，并根据 *"Statistical Data Analysis"* 的式（*4.39*）式计算 χ^2 统计量。

(*b*) 因为很多区间的事件数很小甚至为零，前面计算的统计量不太可能服从 χ^2 分布。写一段程序，根据两种理论假设（**theory1** 和 **theory2**）给出真实的 χ^2 分布。如果利用（*a*）中的数据计算统计检验，其 *P*-值是多少？如果利用正常的 χ^2 分布计算，*P*-值是多少？

解:

(a)

```
import ROOT
x_min = [0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5,
         5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5, 9.0, 9.5]
x_max = [0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0,
         5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5, 9.0, 9.5, 10.0]
data = [1, 0, 3, 4, 6, 3, 3, 4, 5, 7, 4, 5, 2, 0, 1, 0, 0, 1, 0, 0]
theory1 = [0.2, 1.2, 1.9, 3.2, 4.0, 4.5, 4.7, 4.8, 4.8, 4.5,
           4.1, 3.5, 3.0, 2.4, 1.6, 0.9, 0.5, 0.3, 0.2, 0.1]
theory2 = [0.2, 0.7, 1.1, 1.6, 1.9, 2.2, 2.7, 3.3, 3.6, 3.9,
           4.0, 4.0, 3.9, 3.5, 3.2, 2.2, 2.5, 1.5, 1.0, 0.5]

canvas = ROOT.TCanvas('canvas', 'Canvas',800,600)
ROOT.gStyle.SetOptStat(0)
hist_data = ROOT.TH1F('hist_data', 'data', len(x_min), x_min[0], x_max[-1])
hist_theory1 = ROOT.TH1F('hist_theory1', 'theory1', len(x_min), x_min[0],
x_max[-1])
hist_theory2 = ROOT.TH1F('hist_theory2', 'theory2', len(x_min), x_min[0],
x_max[-1])

for i in range(len(x_min)):
    hist_data.Fill((x_min[i] + x_max[i]) / 2, data[i])
    hist_theory1.Fill((x_min[i] + x_max[i]) / 2, theory1[i])
    hist_theory2.Fill((x_min[i] + x_max[i]) / 2, theory2[i])

hist_data.SetLineColor(ROOT.kBlack)
hist_data.SetXTitle('x')
hist_data.SetYTitle('N(x)')
hist_theory1.SetLineColor(ROOT.kRed)
hist_theory2.SetLineColor(ROOT.kBlue)

hist_data.Draw('HIST')
hist_theory1.Draw('HIST SAME')
hist_theory2.Draw('HIST SAME')

legend = ROOT.TLegend(0.7,0.7,0.9,0.9)
legend.AddEntry(hist_data, 'data')
legend.AddEntry(hist_theory1, 'theory1')
```

```
legend.AddEntry(hist_theory2, 'theory2')
legend.Draw()

canvas.SaveAs('4.4.a.png')

chi2_1 = 0
chi2_2 = 0
for i in range(len(data)):
    chi2_1 += (data[i]-theory1[i])**2/theory1[i]
    chi2_2 += (data[i]-theory2[i])**2/theory2[i]
print(f'chi2 for theory 1 is {chi2_1}\nchi2 for theory 2 is {chi2_2}')
```

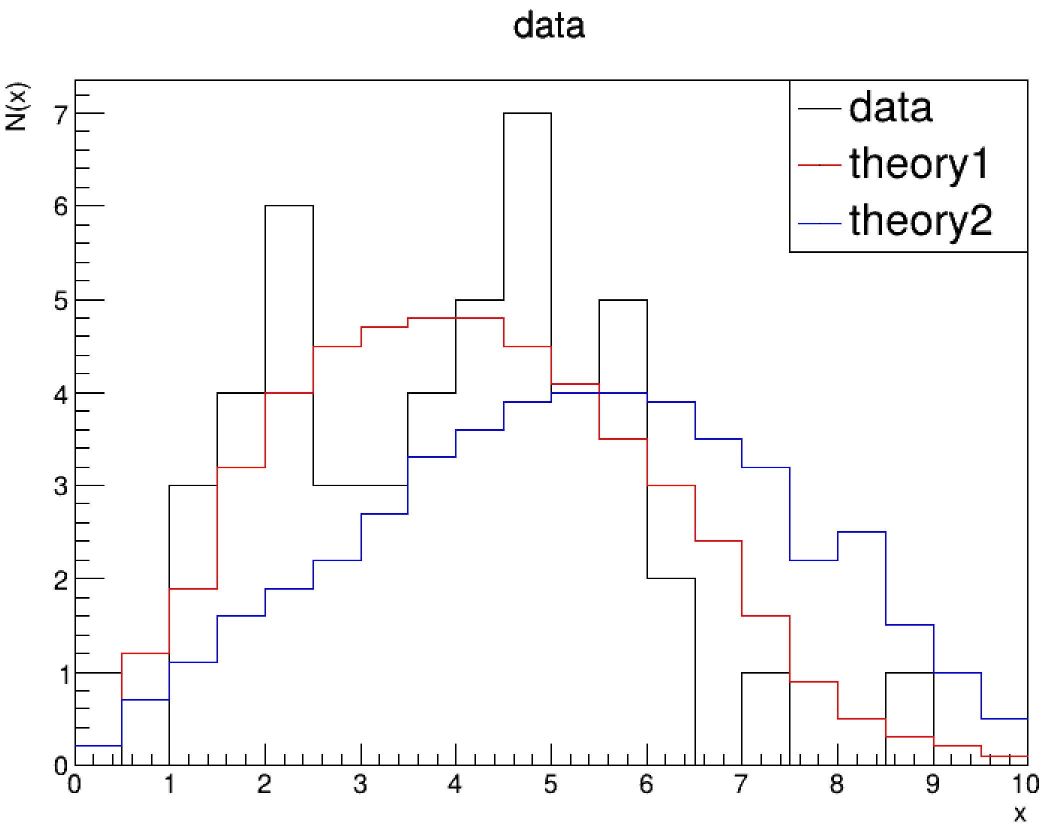

Figure 4.4(a)

结果:

```
chi2 for theory 1 is 15.819254111754042.
chi2 for theory 2 is 35.66526857645278.
```

(b)

```
import ROOT
chi2_1 = 15.819254111754042
chi2_2 = 35.66526857645278
n =10000
x_min = [0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5,
         5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5, 9.0, 9.5]
x_max = [0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0,
         5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5, 9.0, 9.5, 10.0]
theory1 = [0.2, 1.2, 1.9, 3.2, 4.0, 4.5, 4.7, 4.8, 4.8, 4.5,
           4.1, 3.5, 3.0, 2.4, 1.6, 0.9, 0.5, 0.3, 0.2, 0.1]
```

```python
theory2 = [0.2, 0.7, 1.1, 1.6, 1.9, 2.2, 2.7, 3.3, 3.6, 3.9,
           4.0, 4.0, 3.9, 3.5, 3.2, 2.2, 2.5, 1.5, 1.0, 0.5]

canvas = ROOT.TCanvas('canvas', 'Canvas',800,600)
ROOT.gStyle.SetOptStat(0)
hist_theory1 = ROOT.TH1F('hist_theory1', 'theory', 100, 0, 60)
hist_theory2 = ROOT.TH1F('hist_theory2', 'theory', 100, 0, 60)

def chi2_pdf(x, params):
    scale = params[0]
    ndof = params[1]
    return scale * ROOT.Math.chisquared_pdf(x[0], ndof)

f_chi2 = ROOT.TF1("f_chi2", chi2_pdf, 0, 60, 2)
f_chi2.SetParameters(1, 20)

chi_2_list_1 = []
chi_2_list_2 = []

rndm = ROOT.TRandom3()

def calculate_real_chi2(hist, chi_2_list, theory):
    for _ in range(n):
        chi_2 = 0
        for k in range(20):
            mock = rndm.Poisson(theory[k])
            chi_2 += (mock-theory[k])**2/theory[k]
        hist.Fill(chi_2)
        chi_2_list.append(chi_2)
    area = hist.Integral()
    bin_width =  hist.GetBinWidth(1)
    hist.Scale(1/(area*bin_width))

calculate_real_chi2(hist_theory1, chi_2_list_1, theory1)
calculate_real_chi2(hist_theory2, chi_2_list_2, theory2)

hist_theory1.Draw('HIST')
hist_theory2.Draw('HIST SAME')
f_chi2.Draw('SAME')

hist_theory1.SetLineColor(ROOT.kRed)
hist_theory2.SetLineColor(ROOT.kBlue)
f_chi2.SetLineColor(ROOT.kBlack)

legend = ROOT.TLegend(0.7,0.7,0.9,0.9)
legend.AddEntry(hist_theory1, 'theory1')
legend.AddEntry(hist_theory2, 'theory2')
legend.AddEntry(f_chi2, 'ideal distridution')
legend.Draw()
hist_theory1.SetXTitle('x')
hist_theory1.SetYTitle('f(x)')

canvas.SaveAs('4.4.b.png')

def calculate_p_value(chi_2_list, n, chi_2):
    tot = 0
```

```python
    for x in chi_2_list:
        if x >= chi_2:
            tot += 1
    return tot/n

ndof = 20
rough_p_value_1 = ROOT.TMath.Prob(chi2_1, ndof)
rough_p_value_2 = ROOT.TMath.Prob(chi2_2, ndof)
print(f'Rough p-value for theory 1 is {rough_p_value_1}.\nRough p-value for 
theory 2 is {rough_p_value_2}.')
real_p_value_1 = calculate_p_value(chi_2_list_1, n, chi2_1)
real_p_value_2 = calculate_p_value(chi_2_list_2, n, chi2_2)
print(f'Real p-value for theory 1 is {real_p_value_1}.\nReal p-value for theory 2 
is {real_p_value_2}.')
```

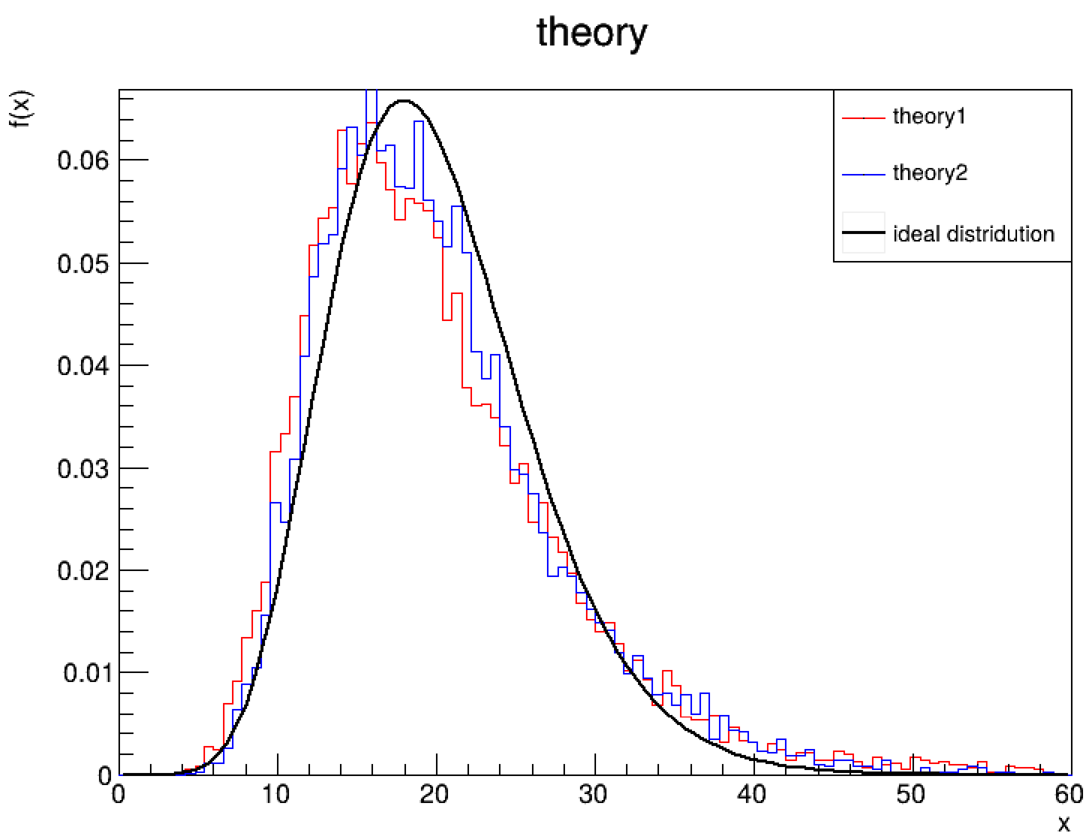

Figure 4.4(b)

结果：

```
Rough p-value for theory 1 is 0.7277722172068071.
Rough p-value for theory 2 is 0.016834198165650262.
Real p-value for theory 1 is 0.6483.
Real p-value for theory 2 is 0.0406.
```

2.

习题 4.5. 在放射性实验中，卢瑟福和盖革记录了固定时间间隔内 α 衰变的次数。数据如表4.2所示。假定放射源中放射性核素的数目非常大，且任意一个核素在小时间间隔内发射一个 α 粒子的概率很小，则可以认为在时间间隔 Δt 内发生衰变的次数 m 服从泊松分布。如果观测结果与泊松分布的假设存在差异，则表明核素的 α 衰变不相互独立，比如某个核素发生 α 衰变可能会引发邻近核素也发生衰变，从而在短时间间隔内形成衰变簇团。

(a) 利用表4.2的数据，计算样本均值

$$\overline{m} = \frac{1}{n_{\mathrm{tot}}}\sum_m n_m m, \tag{4.7}$$

以及样本方差

$$s^2 = \frac{1}{n_{\mathrm{tot}}-1}\sum_m n_m(m-\overline{m}), \tag{4.8}$$

其中 n_m 是发生 m 个衰变的次数，$n_{\mathrm{tot}} = \sum_m n_m = 2608$ 是测量时间内总衰变次数。求和从 $m=0$ 一直到测量时间内最大的衰变次数 (次数为 $m=14$)。利用 $\overline{m}$ 和 s^2，求分散度

$$t = \frac{s^2}{\overline{m}}. \tag{4.9}$$

$\overline{m}$ 和 s^2 分别为 m 的均值和方差的估计量（参见 *Statistical Data Analysis* 第 5 章）；如果 m 服从泊松分布，则 $\overline{m}$ 和 s^2 应该相等，于是可以预期 t 大约为 1。可以证明对于泊松分布，当 n_{tot} 很大时，$(n_{\mathrm{tot}}-1)t$ 服从自由度为 $n_{\mathrm{tot}}-1$ 的 χ^2 分布。而且，当 n_{tot} 很大时，该分布变成均值为 $n_{\mathrm{tot}}-1$，方差为 $2(n_{\mathrm{tot}}-1)$ 的高斯分布。

(b) m 服从泊松分布这一假设的 P-值为多少？为了表征 t 的观测值与泊松假设相符或不相符，应该选取什么样的 t 值（即 t 大表示相符还是 t 小表示相符）？

(c) 写一段蒙特卡罗程序产生很多组数据，每组数据包含 n_{tot} 个服从泊松分布的 m 值。(泊松分布的随机数可以在 ROOT 中直接调用，gRandom− > Possion(ν)。) 对于 m 的均值，可以取表4.2中数据的均值 $\overline{m}$。对于每组数据，计算 t 的值并填充至直方图。利用直方图和从卢瑟福实验数据得到的 t 值，计算泊松假设的 P-值。将该结果与（a）中的结果进行比较。(选作：将 $(n_{\mathrm{tot}}-1)t$ 记录至直方图，与均值为 $n_{\mathrm{tot}}-1$，方差为 $2(n_{\mathrm{tot}}-1)$ 的高斯分布进行比较。)

(a)

```
m = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
n_m = [57, 203, 383, 525, 532, 408, 273, 139, 45, 27, 10, 4, 0, 1, 1]
m_aver = sum(m[i]*n_m[i] for i in range(15))/sum(n_m)
s_square = sum((m[i]-m_aver)**2*n_m[i] for i in range(15))/(sum(n_m)-1)
t = s_square/m_aver
print(f'M_aver is {m_aver}, s_square is {s_square} and t is {t}.')
```

结果：

```
M_aver is 3.871549079754601, s_square is 3.696190618227001 and t is
0.9547058663301989.
```

(b)

```
ndof = sum(n_m)-1
p_value = 2*(1-ROOT.TMath.Prob(ndof*t, ndof))

print(f'P_value is {p_value}.')
```

结果：

```
P_value is 0.09865784328213767.
```

在$t < 1$时，应当认为t大表示相符；在$t > 1$时，应当认为t小表示相符。

(c)

```
N = 10000
n_tot = 2608
m_aver = 3.871549079754601

rnd = ROOT.TRandom3()
hist = ROOT.TH1F('hist', 'hist', 100, 0.75, 1.25)
mock_list = []
canvas = ROOT.TCanvas('canvas', 'canvas', 800, 600)

for i in range(N):
    current_mock_list = []
    for j in range(n_tot):
        current_mock_list.append(rnd.Poisson(m_aver))
    current_m = sum(current_mock_list)/n_tot
    current_s_square = sum((current_mock_list[k]-current_m)**2 for k in
range(n_tot))/(n_tot-1)
    current_t = current_s_square/current_m
    mock_list.append(current_t)
    hist.Fill(current_t)
hist.Draw()
canvas.SaveAs('4.5.png')
cnt = 0
for num in mock_list:
    if abs(num-1) >= abs(t-1):
        cnt += 1
print(f'P-value is {cnt/N}.')
```

结果：

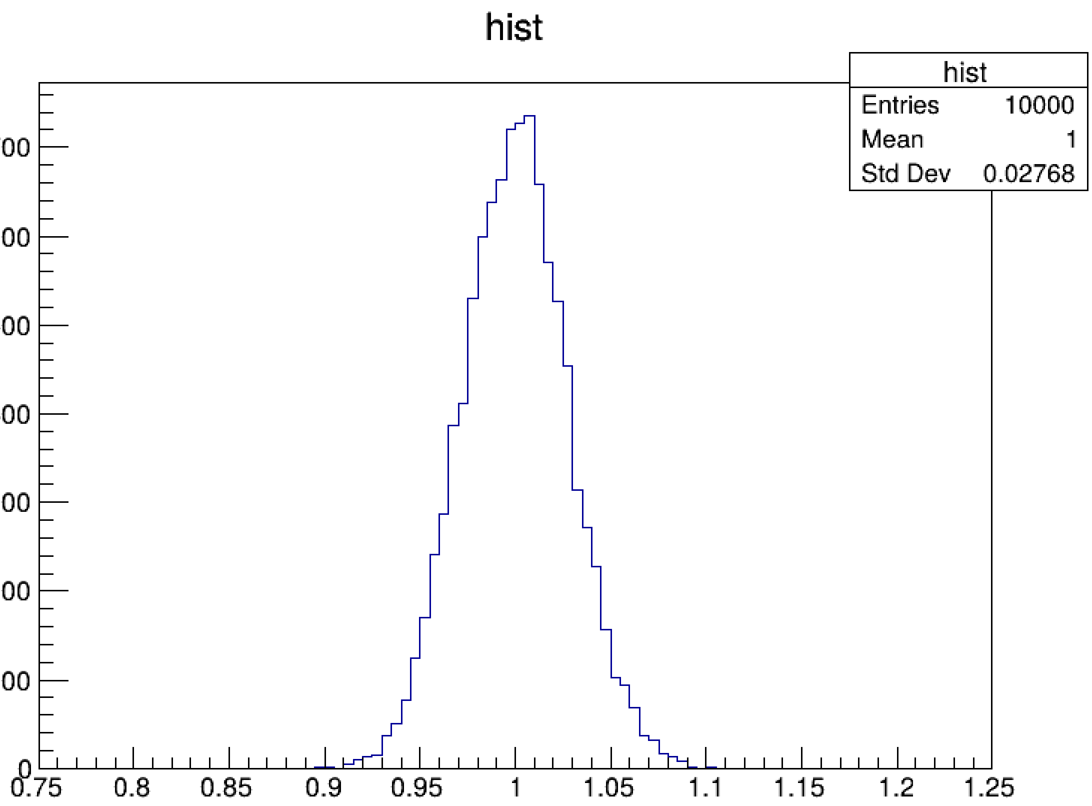

Figure 4.5

```
P-value is 0.1012.
```

(选做)

```
import ROOT
import math
N = 10000
n_tot = 2608
m_aver = 3.871549079754601
mu = n_tot-1
sigma = math.sqrt(2*(n_tot-1))

rnd = ROOT.TRandom3()
hist = ROOT.TH1F('hist', 'hist', 100,1000,4000)
canvas = ROOT.TCanvas('canvas', 'canvas', 800, 600)

for i in range(N):
    current_mock_list = []
    for j in range(n_tot):
        current_mock_list.append(rnd.Poisson(m_aver))
    current_m = sum(current_mock_list)/n_tot
    current_s_square = sum((current_mock_list[k]-current_m)**2 for k in
range(n_tot))/(n_tot-1)
    current_t = current_s_square/current_m
    hist.Fill((n_tot-1)*current_t)

gauss = ROOT.TF1('gauss', 'gaus',1000, 4000)
norm_factor = 1.0 / (sigma * math.sqrt(2 * math.pi))
gauss.SetParameters(norm_factor, mu, sigma)
```

```
hist.Scale(1.0 / hist.Integral(), "width")
hist.Draw('hist')
gauss.Draw('same')
canvas.SaveAs('4.5.1.png')
```

结果:

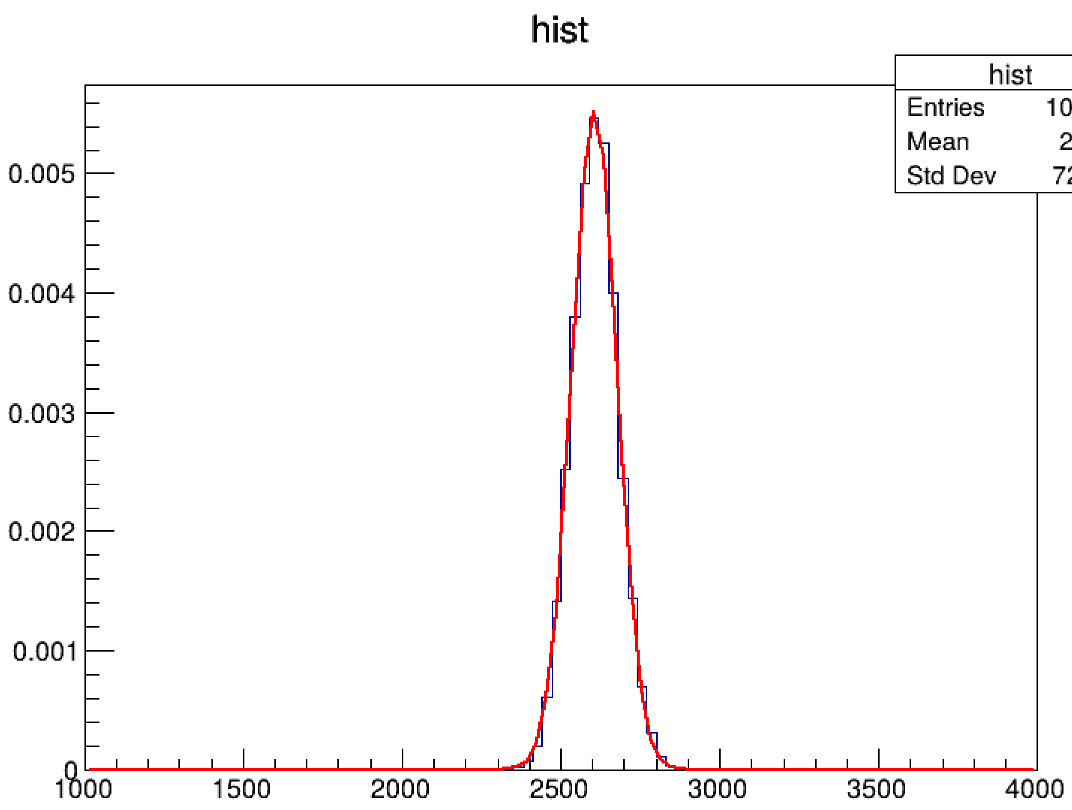

Figure 4.5.1

3.

习题 4.9. 根据某理论，观测到流星表示幸运事件。根据以往的统计，某人每年平均观测到 *10* 颗流星。*2022* 年某人观测到 *5* 颗流星。我们能据此说 *2022* 年对于这个人来说不是幸运年吗？请在 $\alpha = 0.05$ 的显著性水平下给出结论。

解:

观测到的流星的数目分布遵循泊松分布:

$$P(n) = \frac{10^n}{n!}\mathrm{e}^{-10}$$

得到:

$$P(n \leq 5) = 0.067086 > 0.05$$

不能说明这一结论。

4.

习题 4.10. 如果对某个假设进行了几个独立的显著性检验，给出了显著性水平 $P_1, P_2, \ldots, P_n$，总的显著性水平不能通过将这些概率相乘得到。为什么呢？

如果 X 是在 0 和 1 之间均匀分布的随机变量，证明 $-2\ln X$ 是自由度为 2 的 χ^2 变量。我们可以利用这个结果来合并独立的显著性检验的结果。如果三个检验的显著性水平分别为 0.145、0.263 和 0.087，我们应当如何评估总的显著性？

解：

（1）由p值于假设检验之间的关系：

$$P(p \leq \alpha) = \alpha$$

要求随机变量p服从均匀分布

构造的统计量$Q = \Pi_{i=1}^{n} P_i$，均匀分布的乘积显然不满足均匀分布，所以没有这样的关系，不能作为显著性水平。

（2）

记$Y = -2\ln X$，对应概率密度函数为$g(y)$。

已知：

$$X = \mathrm{e}^{-\frac{y}{2}},\ f(x) = 1$$

$$|f(x)\mathrm{d}x| = |g(y)\mathrm{d}y|$$

$$g(y) = \frac{1}{2}\mathrm{e}^{-\frac{y}{2}} = \chi^2(2)$$

$$\chi_1^2 = 3.8620$$

$$\chi_2^2 = 2.6712$$

$$\chi_3^2 = 4.8837$$

三者的和服从自由度为6的卡方分布：

$$\chi^2 = 11.4169$$

$$p = 0.0763$$