实验物理中的统计方法 作业9

1.

习题 7.1. *Galileo* 研究运动的实验之一是小球和斜坡的实验。在离开斜坡边缘之前，小球的轨迹变成水平，如图7.1所示。对于不同的高度 h，测量从斜坡边缘到落地点的水平距离 d. *1608* 年，*Galileo* 测量了 *5* 组数据，如表7.1所示[1]。假设高度 h 的误差可以忽略，水平距离 d 可以看做独立的标准差 $\sigma = 15\,\text{punti}$ 的高斯

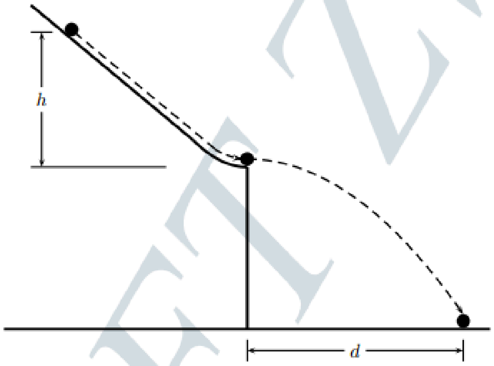

图 7.1: Galileo 小球和斜坡实验的示意图。

h	d
1000	1500
828	1340
800	1328
600	1172
300	800

表 7.1: Galileo 斜坡实验的 5 组数据。给定初始高度 h，d 为落地前的水平距离。单位为punti，1punto≃1mm。

随机变量。(实际上我们不清楚 *Galileo* 如何估计测量误差，但是 $1-2\%$ 的误差是可以接受的。) 此外，我们知道如果 $h=0$，则水平距离 d 将为零，即，如果球从斜坡边缘出发，它将垂直落到地上。

(a) 考虑 h 和 d 的关系为如下形式

$$d = \alpha h \tag{7.1}$$

[1]参见 Stillman Drake and Maclachlan, Galileo's discovery of the parabolic trajectory, *Scientific American* **232** (March 1975) 102; Stillman Drake, *Galileo at Work*, University of Chicago Press, Chicago (1978).

以及

$$d = \alpha h + \beta h^2. \tag{7.2}$$

求参数 α 和 β 的最小二乘估计量。对应于这两个假设的最小 χ^2 和 P-值分别为多少?

(b) 假设 d 和 h 的关系为如下形式

$$d = \alpha h^{\beta}. \tag{7.3}$$

写一段程序对 α 和 β 进行最小二乘拟合。注意这是参数的非线性函数，必须数值求解。

(c)Galileo 认为运动是水平分量和垂直分量的叠加，其中水平运动是匀速运动，垂直速度在斜坡的最低处为零，随后随时间线性增加。证明这将导致关系式

$$d = \alpha\sqrt{h}. \tag{7.4}$$

求 α 的最小二乘估计量以及最小 χ^2。该假设的 P-值是多少?

解:

Part I. 公式推导

(a)

$$d = \alpha h$$

$$\chi^2 = \sum_{i=1}^{n} \frac{(d_i - \alpha h_i)^2}{\sigma^2}$$

$$\hat{\alpha} = \frac{\sum_{i=1}^{n} d_i h_i}{\sum_{i=1}^{n} h_i^2}$$

$$d = \alpha h + \beta h^2$$

$$\chi^2 = \sum_{i=1}^{n} \frac{(d_i - \alpha h_i - \beta h_i^2)^2}{\sigma^2}$$

$$\frac{\partial \chi^2}{\partial \alpha} = 0 \Rightarrow -\sum_{i=1}^{n} d_i h_i + \hat{\alpha}\sum_{i=1}^{n} h_i^2 - \hat{\beta}\sum_{i=1}^{n} h_i^3 = 0$$

$$\frac{\partial \chi^2}{\partial \beta^2} = 0 \Rightarrow -\sum_{i=1}^{n} d_i h_i^2 - \hat{\alpha}\sum_{i=1}^{n} h_i^3 + \hat{\beta}\sum_{i=1}^{n} h_i^4 = 0$$

$$\hat{\alpha} = \frac{\sum_{i=1}^{n} d_i h_i \sum_{i=1}^{n} h_i^4 + \sum_{i=1}^{n} d_i h_i^2 \sum_{i=1}^{n} h_i^3}{\sum_{i=1}^{n} h_i^2 \sum_{i=1}^{n} h_i^4 - \sum_{i=1}^{n} h_i^3 \sum_{i=1}^{n} h_i^3}$$

$$\hat{\beta} = \frac{\sum_{i=1}^n d_i h_i \sum_{i=1}^n h_i^3 + \sum_{i=1}^n d_i h_i^2 \sum_{i=1}^n h_i^4}{\sum_{i=1}^n h_i^4 \sum_{i=1}^n h_i^2 - \sum_{i=1}^n h_i^3 \sum_{i=1}^n h_i^3}$$

(c)

$$d = vt, h = \frac{1}{2}gt^2 \Rightarrow d = \alpha\sqrt{h}$$

所有计算结果见Part II.

Part II. 整体代码实现:

```
from scipy.optimize import minimize
from scipy.stats import chi2

hs     = [1000, 828, 800, 600, 300]
ds     = [1500, 1340, 1328, 1172, 800]
sigma  = 15

def f(alpha, d, h):
    return (d - alpha * h) ** 2 / sigma ** 2

def g(alpha, beta, d, h):
    return (d - alpha * h - beta * h ** 2) ** 2 / sigma ** 2

def s(alpha, beta, d, h):
    return (d - alpha * h ** beta) ** 2 / sigma ** 2

def t(alpha, d, h):
    return (d - alpha * h ** 0.5) ** 2 / sigma ** 2

def chi_squared_f(params):
    alpha = params
    return sum(f(alpha, d, h) for d, h in zip(ds, hs))

def chi_squared_g(params):
    alpha, beta = params
    return sum(g(alpha, beta, d, h) for d, h in zip(ds, hs))

def chi_squared_s(params):
    alpha, beta = params
    return sum(s(alpha, beta, d, h) for d, h in zip(ds, hs))

def chi_squared_t(params):
    alpha = params
    return sum(t(alpha, d, h) for d, h in zip(ds, hs))

result_f  = minimize(chi_squared_f, [1], method='Nelder-Mead')
f_alpha   = result_f.x[0]
chi2_min  = result_f.fun
p_value_f = chi2.sf(chi2_min, df=4)
print(f'f:d=alpha h\nalpha = {f_alpha:.4f}, chi2 = {chi2_min:.2f}, p =
{p_value_f:.4e}')

result_g  = minimize(chi_squared_g, [1, 1], method='Nelder-Mead')
g_alpha, g_beta = result_g.x
```

```
chi2_min  = result_g.fun
p_value_g = chi2.sf(chi2_min, df=3)
print(f'g:d=alpha h + beta h ^2\nalpha = {g_alpha:.4f}, beta = {g_beta:.4f}, chi2
= {chi2_min:.2f}, p = {p_value_g:.4e}')

result_s  = minimize(chi_squared_s, [1, 1], method='Nelder-Mead')
s_alpha, s_beta = result_s.x
chi2_min  = result_s.fun
p_value_s = chi2.sf(chi2_min, df=3)
print(f's:d = alpha h^beta\nalpha = {s_alpha:.4f}, beta = {s_beta:.4f}, chi2 =
{chi2_min:.2f}, p = {p_value_s:.4e}')

result_t  = minimize(chi_squared_t, [1], method='Nelder-Mead')
t_alpha   = result_t.x[0]
chi2_min  = result_t.fun
p_value_t = chi2.sf(chi2_min, df=4)
print(f't:d = alpha sqrt(h)\nalpha = {t_alpha:.4f}, chi2 = {chi2_min:.2f}, p =
{p_value_t:.4e}')
```

结果:

```
f:d=alpha h
alpha = 1.6628, chi2 = 661.99, p = 5.9128e-142
g:d=alpha h + beta h ^2
alpha = 2.7929, beta = -0.0014, chi2 = 64.74, p = 5.6962e-14
s:d = alpha h^beta
alpha = 43.7606, beta = 0.5111, chi2 = 3.76, p = 2.8905e-01
t:d = alpha sqrt(h)
alpha = 47.0857, chi2 = 4.21, p = 3.7864e-01
```

2.

习题 7.5. 考虑随机变量 x 的两个部分重叠的样本，它们的样本容量分别为 n 和 m，共有部分的样本容量为 c。假设已知 x 的方差 $V[x]=\sigma^2$。考虑样本均值

$$y_1=\frac{1}{n}\sum_{i=1}^{n}x_i \tag{7.11}$$

和

$$y_2=\frac{1}{m}\sum_{i=1}^{m}x_i. \tag{7.12}$$

(a) 证明协方差为

$$\mathrm{cov}[y_1,y_2]=\frac{c\sigma^2}{nm}. \tag{7.13}$$

(b) 利用 7.6 节的结果，求 y_1 和 y_2 的加权平均和方差。

解:

(a)

$$cov[y_1,y_2]=E[y_1y_2]-E[y_1]E[y_2]=\frac{c(\mu^2+\sigma^2)-c\mu^2}{nm}=\frac{c\sigma^2}{nm}$$

(b)

$$\mathbf{V} = \begin{bmatrix} \frac{\sigma^2}{n} & \frac{c\sigma^2}{nm} \\ \frac{c\sigma^2}{nm} & \frac{\sigma^2}{m} \end{bmatrix}$$

$$\mathbf{V}^{-1} = \frac{n^2m^2}{nm-c^2} \begin{bmatrix} \frac{\sigma^2}{m} & -\frac{c\sigma^2}{nm} \\ -\frac{c\sigma^2}{nm} & \frac{\sigma^2}{n} \end{bmatrix}$$

$$w_1 = \frac{\frac{\sigma^2}{m} - \frac{c\sigma^2}{nm}}{\frac{\sigma^2}{m} - \frac{c\sigma^2}{nm} - \frac{c\sigma^2}{nm} + \frac{\sigma^2}{n}} = \frac{n-c}{n+m-2c}$$

$$w_2 = \frac{-\frac{c\sigma^2}{nm} + \frac{\sigma^2}{n}}{\frac{\sigma^2}{m} - \frac{c\sigma^2}{nm} - \frac{c\sigma^2}{nm} + \frac{\sigma^2}{n}} = \frac{m-c}{n+m-2c}$$

$$\hat{\lambda} = w_1y_1 + w_2y_2 = \frac{(n-c)\overline{y_1} + (m-c)\overline{y_2}}{n+m-2c}$$

$$\sigma_{\hat{\lambda}^2} = \sum_{i,j} w_iV_{ij}w_j = \frac{(n-c)^2m + (m-c)^2n + 2(n-c)(m-c)c}{(n+m-2c)^2nm}\sigma^2 = \frac{mn-c^2}{(n+m-2c)mn}\sigma^2$$

3.

习题 7.6. 天文学家托勒密（*Claudius Ptolemy*）利用圆盘做过光折射的实验。他把圆盘的一半浸入水中，圆心正好位于水面处，如图7.2所示。大约公元 *140* 年，*Ptolemy* 对 *8* 组不同的入射角 θ_i 测量了相应的折射角

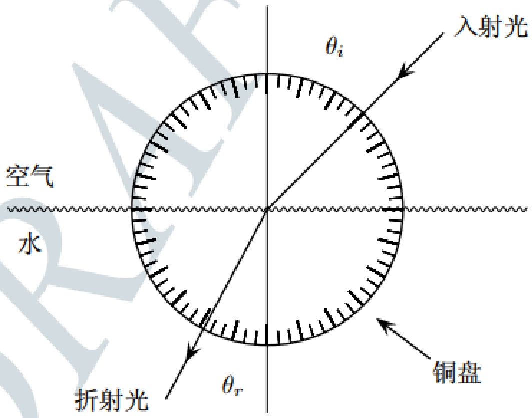

图 7.2: Ptolemy 用来研究光折射的设备。

θ_r，结果如表7.2所示[2]。本练习中，我们认为入射角已知且误差可以忽略，而把折射角看作标准差为 $\sigma = \frac{1}{2}^\circ$ 的高斯随机变量。（这是一个合理的假设，记录的角度精确到最邻近的半度。注意，我们可以将 θ_i 的误差吸收到 θ_r 的有效误差中。）

（*a*）直到 *17* 世纪才发现正确的折射定律，在此之前，通常的假设是

$$\theta_r = \alpha\theta_i, \tag{7.14}$$

[2]取自 Pedersen and Mogens Pihl, *Early Physics and Astronomy: A Historical Introduction*, MacDonald and Janes, London, 1974

θ_i	θ_r
10	8
20	$15\frac{1}{2}$
30	$22\frac{1}{2}$
40	29
50	35
60	$40\frac{1}{2}$
70	$45\frac{1}{2}$
80	50

表 7.2: 入射角和折射角（单位：度）。

然而 *Ptolemy* 更喜欢用下面的形式

$$\theta_r = \alpha\theta_i - \beta\theta_i^2. \tag{7.15}$$

对这两种不同的假设，求参数的最小二乘估计量，并计算最小 χ^2 值。评论一下两个假设的拟合优度。是否可以相信所有的数据都是从实际测量得来的？[3]

(*b*) *1621* 年 *Snell* 发现了折射定律

$$\theta_r = \sin^{-1}\left(\frac{\sin\theta_i}{r}\right), \tag{7.16}$$

其中 $r = n_r/n_i$ 为两种介质的折射率之比。求 r 的最小二乘估计量并计算出最小 χ^2 值。评价对 θ_r 作 $\sigma = \frac{1}{2}^\circ$ 假设的合理性。

解：

Part I

代码实现：

```
from scipy.optimize import minimize
from scipy.stats import chi2
from math import sin, asin, pi

theta_i = [10, 20, 30, 40, 50, 60, 70, 80]
theta_r = [8, 15.5, 22.5, 29, 35, 40.5, 45.5, 50]
sigma = 0.5

def f(r, i, alpha):
    return (r-alpha*i)**2/sigma**2

def g(r, i, alpha, beta):
    return (r-alpha*i+beta*i**2)**2/sigma**2

def h(r, i , alpha):
    return (r-asin(sin(i)/alpha))**2/(sigma*pi/180)**2

def chi_squared_f(params):
    alpha = params[0]
    return sum(f(r, i, alpha) for r, i in zip(theta_r, theta_i))

def chi_squared_g(params):
    alpha, beta = params
    return sum(g(r, i, alpha, beta) for r ,i in zip(theta_r, theta_i))

def chi_squared_h(params):
```

```python
    alpha = params[0]
    return sum(h(r*pi/180, i*pi/180, alpha) for r, i in zip(theta_r, theta_i))

result_f  = minimize(chi_squared_f, [1], method='Nelder-Mead')
f_alpha   = result_f.x[0]
chi2_min  = result_f.fun
p_value_f = chi2.sf(chi2_min, df=7)
print(f'f:theta_r=alpha theta_i\nalpha = {f_alpha:.4f}, chi2 = {chi2_min:.2f}, p = {p_value_f:.4e}')

result_g  = minimize(chi_squared_g, [1, 1], method='Nelder-Mead')
f_alpha, f_beta = result_g.x
chi2_min  = result_g.fun
p_value_g = chi2.sf(chi2_min, df=6)
print(f'g:theta_r = alpha theta_i - beta theta_i^2\nalpha = {f_alpha:.4f}, beta = {f_beta:.4f}, chi2 = {chi2_min:.2f}, p = {p_value_g:.4e}')

result_h  = minimize(chi_squared_h, [1.5], method='Nelder-Mead')
f_alpha   = result_h.x[0]
chi2_min  = result_h.fun
p_value_h = chi2.sf(chi2_min, df=7)
print(f'h:theta_r = arcsin(sin(theta_i)/alpha)\nalpha = {f_alpha:.4f}, chi2 = {chi2_min:.2f}, p = {p_value_h:.4e}')
```

结果：

```
f:theta_r=alpha theta_i
alpha = 0.6662, chi2 = 134.65, p = 6.7109e-26
g:theta_r = alpha theta_i - beta theta_i^2
alpha = 0.8250, beta = 0.0025, chi2 = 0.00, p = 1.0000e+00
h:theta_r = arcsin(sin(theta_i)/alpha)
alpha = 1.3116, chi2 = 14.00, p = 5.1143e-02
```

Part II.

对于题目其他部分的回答：

(a)对于线性的假设，拟合的不佳，但是对于二次函数的假设拟合的太好了，这显然是不符合斯涅尔定律的，不能认为所有数据都是测量得到的。

(b)

$$\chi^2/7 \approx 2$$

量级正确，说明估计是合理的。