

Solution Notes for Data Science

Example Sheet 2

Question 1

There are many ways to approach this problem. A common approach is to split the figure into four regions: (0) the boundary of the face, (1) the smile, (2) the left eye and (3) the right eye. Then, generate a categorical random variable $K \in [0, 3]$, which indicates which of the four regions to sample. The probability for each region should be roughly proportional to the area of each region. Now we have to specify a distribution for each of the regions. Some simple ones are the following:

1. For the boundary of the face, pick the radius $R \sim \mathcal{N}(1, 0.07^2)$ and the angle $\Theta \sim \mathcal{U}(0, 2\pi)$ with the x -axis. Then get the x, y coordinates by projecting on the axes, i.e. $x = R \sin(\Theta)$ and $y = R \cos(\Theta)$.
2. For the smile, pick $R \sim \mathcal{N}(0.5, 0.07^2)$ and the angle $\Theta \sim \mathcal{U}(-\pi/2.5, \pi/2.5)$ and use the same process to convert the polar into Cartesian coordinates.
3. For the right eye, choose a normal distribution centred at $(0.35, 0.3)$ with .
4. For the left eye, choose a normal distribution centred at $(-0.35, 0.3)$.

```
import numpy as np
import matplotlib.pyplot as plt
```

```
def rxy():
    k = np.random.choice(4, p=[.7, .2, .05, .05])
    if k == 0:
        # Generate a sample from the boundary of the face.
        r = np.random.normal(loc=1, scale=0.07)
        theta = np.random.uniform(0, 2 * np.pi)
        x = r * np.cos(theta)
        y = r * np.sin(theta)
    elif k == 1:
        # Generate a sample from the smile on the face.
        r = np.random.normal(loc=0.5, scale=0.07)
        theta = np.random.uniform(-np.pi * 4 / 5, -np.pi / 5)
        x = r * np.cos(theta)
        y = r * np.sin(theta)
    elif k == 2:
        # Generate a sample from the right eye.
        s = np.random.normal(loc=[0.35, 0.3], scale=0.08)
        x, y = s[0], s[1]
    elif k == 3:
        # Generate a sample from the left eye.
        s = np.random.normal(loc=[-0.35, 0.3], scale=0.08)
        x, y = s[0], s[1]
    return x, y
```

```
n = 5000
xy = [rxy() for _ in range(n)]
x, y = [x for x, _ in xy], [y for _, y in xy]
```

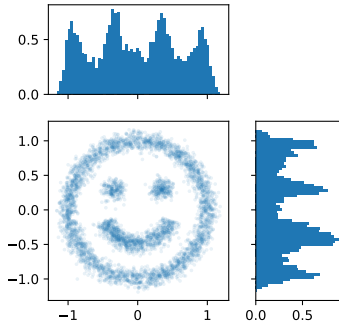
```
fig, ((ax_x, dummy), (ax_xy, ax_y)) = plt.subplots(2, 2, figsize=(4, 4), sharex='col',
↪ sharey='row',
```

```

dummy.remove()
ax_xy.scatter(x, y, s=3, alpha=.1)
ax_x.hist(x, density=True, bins=60)
ax_y.hist(y, density=True, bins=60, orientation='horizontal')
plt.show()

```

This code gives the following figure.



Follow-up questions:

1. How would you the the smile more thin near the ends?
2. What is the minimal change you can make in the code above to get a winky face?
3. How does the figure change as you take more samples? Is there a problem if you take too many samples?
4. How do you expect the figure to change with the size of the marker?
5. How can you deal with the problem that the inner part of the face boundary can appear more dense?
6. Investigate the posterior of K given two samples between the right eye and the smile.

Question 2

- (a) We obtain the pdf for the distribution by differentiation (without forgetting about the indicator function),

$$\Pr(\Theta = \theta) = -\frac{b_0}{\theta^{\alpha_0+1}} \cdot (-\alpha_0) \cdot \mathbf{1}_{\theta \geq b_0} = \alpha_0 \cdot \frac{b_0}{\theta^{\alpha_0+1}} \cdot \mathbf{1}_{\theta \geq b_0}$$

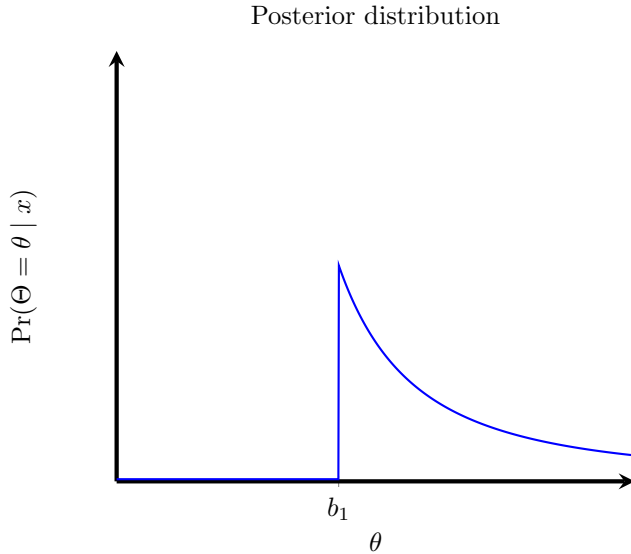
- (b) To calculate the posterior,

$$\begin{aligned}
\Pr(\theta \mid x) &= \frac{\Pr(x \mid \theta) \cdot \Pr(\theta)}{\Pr(x)} \\
&= (\text{const}) \cdot \left(\prod_{i=1}^n \Pr(x_i \mid \theta) \right) \cdot \Pr(\theta) \\
&= (\text{const}) \cdot \left(\prod_{i=1}^n \frac{1}{\theta} \mathbf{1}_{x_i \in [0, \theta]} \right) \cdot \Pr(\theta) \\
&= (\text{const}) \cdot \frac{1}{\theta^n} \cdot \mathbf{1}_{\min_i x_i \geq 0} \cdot \mathbf{1}_{\max_i x_i \leq \theta} \cdot \Pr(\theta) \\
&= (\text{const}') \cdot \left(\frac{1}{\theta^n} \cdot \mathbf{1}_{\max_i x_i \leq \theta} \right) \cdot \left(\frac{b_0}{\theta^{\alpha_0+1}} \cdot \mathbf{1}_{\theta \geq b_0} \right) \\
&= (\text{const}') \cdot \frac{1}{\theta^{n+\alpha_0+1}} \cdot \mathbf{1}_{\theta \geq \max(b_0, \max_i x_i)}
\end{aligned}$$

By denoting $b_1 = \max(b_0, \max_i x_i)$ and $\alpha_1 = n + \alpha_0 + 1$ we see that this density is proportional to the density of $\text{Pareto}(b_1, \alpha_1)$ and it is defined over the same domain. Hence, the posterior is $\text{Pareto}(b_1, \alpha_1)$.

- (c) The Pareto pdf has a peak on the left boundary and is decreasing in its domain. So, we will determine a confidence interval of the form $[b_1, r]$ such that $\Pr(\Theta \in [b_1, r] \mid X) = 0.95$, i.e.

$$1 - \left(\frac{b_1}{r}\right)^{\alpha_1} = 0.95 \Rightarrow r = \frac{b_1}{0.05^{1/\alpha_1}}.$$



- (d) Any interval for which there is 0.95 probability of Θ being in it, is a valid confidence interval. So, a valid confidence interval would also be one of the form $[\ell, \infty)$. For this we have,

$$\left(\frac{b_1}{\ell}\right)^{\alpha_1} = 0.95 \Rightarrow \ell = \frac{b_1}{0.95^{1/\alpha_1}}.$$

However, the first interval is more natural as (1) it is the shortest possible confidence interval and (2) it also contains the peaks of the distribution.

Some follow-up questions:

1. How many valid confidence intervals are there?
2. How should one handle bi-modal distributions?
3. Can you think of a symmetric distribution where the shortest possible interval is not symmetric?

Question 3

We start by writing out Bayes' rule for n independent samples,

$$\begin{aligned}
 \Pr(\mu \mid x_1, \dots, x_n) &= \frac{\Pr(x_1, \dots, x_n \mid \mu) \Pr(\mu)}{\Pr(x_1, \dots, x_n)} \\
 &= (\text{const}_1) \cdot \left(\prod_{i=1}^n \Pr(x_i \mid \mu) \right) \Pr(\mu) \\
 &= (\text{const}_1) \cdot \left(\prod_{i=1}^n (\text{const}_2) \exp\left(-\frac{(x_i - \mu)^2}{2\sigma_0^2}\right) \right) \cdot (\text{const}_3) \cdot \exp\left(-\frac{(\mu - \mu_0)^2}{2\rho_0^2}\right) \\
 &= (\text{const}_4) \cdot \exp\left(-\sum_{i=1}^n \frac{(x_i - \mu)^2}{2\sigma_0^2} - \frac{(\mu - \mu_0)^2}{2\rho_0^2}\right) \\
 &= (\text{const}_4) \cdot \exp\left(-\sum_{i=1}^n \frac{x_i^2 - 2x_i\mu + \mu^2}{2\sigma_0^2} - \frac{\mu^2 - 2\mu\mu_0 + \mu_0^2}{2\rho_0^2}\right) \\
 &= (\text{const}_4) \cdot \exp\left(-\mu^2 \left(\frac{n}{2\sigma_0^2} + \frac{1}{2\rho_0^2}\right) + \mu \left(\frac{\sum_{i=1}^n x_i}{\sigma_0^2} + \frac{\mu_0}{\rho_0^2}\right) + (\text{const}_5)\right)
 \end{aligned}$$

Now we have an expression in the form $-a\mu^2 + b\mu$, where $a = \frac{n}{2\sigma_0^2} + \frac{1}{2\rho_0^2}$ and $b = \frac{\sum_{i=1}^n x_i}{\sigma_0^2} + \frac{\mu_0}{\rho_0^2}$ and we want to bring in the form $-\frac{(\mu-c)^2}{2\tau^2}$.

$$\begin{aligned}
 -a\mu^2 + b\mu &= -a \left(\mu^2 - \frac{b}{a}\mu \right) = -a \left(\mu^2 - 2 \cdot \frac{b}{2a} \cdot \mu + \left(\frac{b}{2a}\right)^2 - \left(\frac{b}{2a}\right)^2 \right) \\
 &= -a \left(\mu - \frac{b}{2a} \right)^2 + (\text{const}_6) = -\frac{(\mu - \frac{b}{2a})^2}{1/(\sqrt{2a})} + (\text{const}_6).
 \end{aligned}$$

Hence, the mean and st. deviation are given by

$$c = -\frac{b}{2a} = \frac{\frac{\sum_{i=1}^n x_i}{\sigma_0^2} + \frac{\mu_0}{\rho_0^2}}{\frac{n}{\sigma_0^2} + \frac{1}{\rho_0^2}} \quad \text{and} \quad \tau = \frac{1}{2a} = \frac{1}{\sqrt{\frac{n}{\sigma_0^2} + \frac{1}{\rho_0^2}}}.$$

So, the posterior is of the form

$$\Pr(\mu \mid x_1, \dots, x_n) = (\text{const})_7 \cdot \exp\left(-\frac{(\mu - c)^2}{2 \cdot \tau^2}\right).$$

Two densities that are defined over the same domain and are proportional to each other must be equal. Hence, the posterior is a Normal distribution $\mathcal{N}(c, \tau^2)$.

Question 4

We start by writing out Bayes' rule for these n independent samples,

$$\Pr(\mu \mid x_1, \dots, x_n) = \frac{\Pr(x_1, \dots, x_n \mid \mu) \cdot \Pr(\mu)}{\Pr(x_1, \dots, x_n)} = (\text{const}) \cdot \left(\prod_{i=1}^n \Pr(x_i \mid \mu) \right) \cdot \Pr(\mu).$$

We know that the prior follows a Normal distribution $\mathcal{N}(0, 5^2)$, so it remains to find an expression for $\Pr(x_i \mid \mu)$. Let H_i be the indicator random variable for whether the i -th sample is an outlier or not,

$$\begin{aligned}
 \Pr(x_i \mid \mu) &= \Pr(x_i, H_i = 1 \mid \mu) + \Pr(x_i, H_i = 0 \mid \mu) \\
 &= \Pr(x_i \mid H_i = 1, \mu) \Pr(H_i = 1 \mid \mu) + \Pr(x_i, H_i = 0 \mid \mu) \Pr(H_i = 0 \mid \mu) \\
 &= \Pr(x_i \mid H_i = 1, \mu) \Pr(H_i = 1) + \Pr(x_i, H_i = 0 \mid \mu) \Pr(H_i = 0) \\
 &= 0.01 \cdot p_{\text{cauchy}}(x_i) + 0.99 \cdot p_{\text{normal}}(x_i).
 \end{aligned}$$

where in the second step we used the chain rule for conditional probability $\Pr(A, B \mid C) = \Pr(A \mid B, C) \cdot \Pr(B \mid C)$ and the third step follows from independence of μ and H_i . Combining all of these, we get

$$\Pr(\mu \mid x_1, \dots, x_n) = (\text{const}) \cdot \left(\prod_{i=1}^n (0.01 \cdot p_{\text{cauchy}}(x_i) + 0.99 \cdot p_{\text{normal}}(x_i)) \right) \cdot \Pr(\mu)$$

Now, we can estimate the posterior distribution using sampling and normalisation. The code below implements this.

```
import numpy as np
import scipy.stats
import matplotlib.pyplot as plt

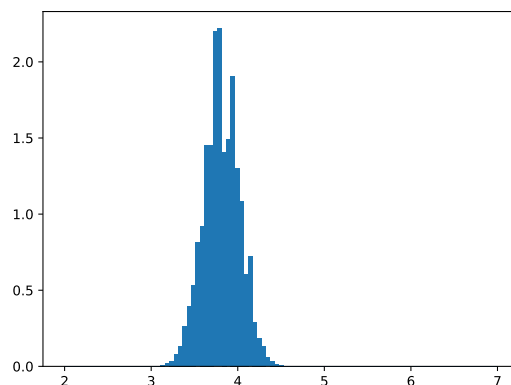
x = [4.3, 2.8, 3.9, 4.1, 9, 4.5, 3.3] # The dataset
samples = 10000 # Number of samples to use.
prior_mu, prior_sigma = 0, 5 # Parameters for the Normal prior.
p_cauchy = 0.01
likelihood_sigma = 0.5

def prx(x, mu):
    p1 = scipy.stats.norm.pdf(x, loc=mu, scale=likelihood_sigma)
    p2 = scipy.stats.cauchy.pdf(x)
    return np.prod((1-p_cauchy) * p1 + p_cauchy * p2)

mus = np.random.normal(loc=prior_mu, scale=prior_sigma, size=samples)
w = np.array([prx(x, mu) for mu in mus])
w = w / np.sum(w)

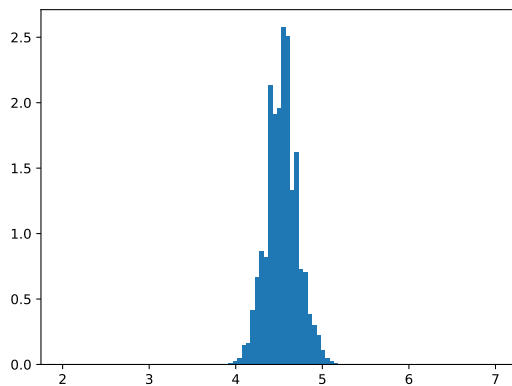
plt.hist(mus, weights=w, density=True, bins=np.linspace(2, 7, 100))
plt.show()
```

Running it gives the following plot.



Note 1: Your results may differ because you used different binning.

Note 2: If we had not used the outlier distribution, then the posterior distribution would be quite different (why? - which samples contribute to this?):



Question 5

We assume that the model has the form $\text{temp}_i \sim \mathcal{N}(y_i, \sigma^2)$, where $y_i = \alpha + \beta_1 \cdot \sin(2\pi t_i) + \beta_2 \cdot \cos(2\pi t_i)$.

The choice of priors is not a formal procedure and consists of a collection of heuristics (You may want to read more about this process in Chapter 7 of “Probability and Statistics: The Science of Uncertainty” by Michael J. Evans and Jeffrey S. Rosental).

In this case, we choose $a \sim \mathcal{U}(8, 12)$ and $\gamma \sim \mathcal{U}(-0.01, 0.07)$. We choose β_1 and β_2 to be dependent in the following way, choose a radius $R \sim \mathcal{U}(6, 8)$ and an angle $\Phi \sim \mathcal{U}(0, 2\pi)$, then set $\beta_1 = R \cos(\Theta)$ and $\beta_2 = R \sin(\Theta)$. We also choose $\sigma \sim \mathcal{N}(1.41, 0.1^2)$, i.e. close to the observed value.

Now, in order to compute the confidence interval we need to estimate empirically the distribution $\Gamma \mid X$. We do this by marginalising over the parameters α, β_1, β_2 , i.e.

$$\begin{aligned} \Pr(\Gamma \mid X) &= \int_{\alpha, \beta_1, \beta_2} \Pr(\alpha, \beta_1, \beta_2, \gamma \mid X) d\alpha d\beta_1 d\beta_2 \\ &= (\text{const}) \cdot \int_{\alpha, \beta_1, \beta_2} \Pr(X \mid \alpha, \beta_1, \beta_2, \gamma) \cdot \Pr(\alpha, \beta_1, \beta_2, \gamma) d\alpha d\beta_1 d\beta_2, \end{aligned}$$

which we estimate by sampling the values of the parameters $\alpha, \beta_1, \beta_2, \gamma$ using their priors and computing $\Pr(X \mid \alpha, \beta_1, \beta_2, \gamma)$ which is just the likelihood of a linear model with Normal noise. From this process we accumulate tuples of the form $(\alpha, \beta_1, \beta_2, \gamma, \Pr(X \mid \alpha, \beta_1, \beta_2, \gamma))$. We sort these tuples by their γ value, normalise to find the unknown constant and finally we find the 2.5% and 97.5% quantiles to estimate a symmetric confidence interval.

An example implementation is the following,

```
import numpy as np
import pandas

climate =
↳ pandas.read_csv('https://www.cl.cam.ac.uk/teaching/2021/DataSci/data/climate.csv')
climate['t'] = climate.yyyy + (climate.mm - 1) / 12
climate['temp'] = (climate.tmin + climate.tmax) / 2

# Let's look at Oxford, which has longer records.
df = climate.loc[(climate.station == 'Oxford') & (~pandas.isna(climate.temp))]
t, temp = df['t'], df['temp']

# Sample from the prior distributions.
samples = 100000
alpha = np.random.uniform(8, 12, size=samples)
# Introduce a dependence between beta1 and beta2.
beta_magnitude = np.random.uniform(6, 8, size=samples)
beta_angle = np.random.uniform(0, 2 * np.pi, size=samples)
```

```

beta_1 = beta_magnitude * np.cos(beta_angle)
beta_2 = beta_magnitude * np.sin(beta_angle)

gamma = np.random.uniform(-1 / 100, 7 / 100, size=samples)
sigma = np.random.normal(loc=1.41, scale=0.1, size=samples)

# Compute weights
def logpr(temp, a, b1, b2, gamma, t, sigma):
    pred = a + b1 * np.sin(2 * np.pi * t) + b2 * np.cos(2 * np.pi * t) + gamma * (t - 2000)
    n = len(temp)
    return - n / 2 * np.sqrt(2 * np.pi * (sigma ** 2)) - np.sum((temp - pred) ** 2) / (2 *
    ↪ (sigma ** 2))

w = np.array([logpr(temp, a, b1, b2, g, t, s) for (a, b1, b2, g, s) in zip(alpha, beta_1,
    ↪ beta_2, gamma, sigma)])
# Log-exp trick in order to avoid underflow.
w = np.exp(w - max(w))
w = w / np.sum(w)

# Find a 95% confidence interval (see section 6.2 of lecture notes)
i = np.argsort(gamma)
gamma, w = gamma[i], w[i]
F = np.cumsum(w)
print(f'[{gamma[F < 0.025] [-1]}, {gamma[F > 0.975] [0]}]')

```

Note 1: For many reasonable choices of priors most of the samples will have a negligible contribution. So, we have to take a large number of samples.

Note 2: We have used the log-sum-exp trick to avoid underflows.

Question 6

- (a) We know that the random variables B_e (for $e \in E = \{\text{As, Blk, Mix, Oth, Wh}\}$) follow a Beta distribution with parameters $\alpha = 1/2, \beta = 1/2$ and are independent. So, we can write the prior as

$$\begin{aligned} \Pr(\beta_{\text{As}}, \beta_{\text{Blk}}, \beta_{\text{Mix}}, \beta_{\text{Oth}}, \beta_{\text{Wh}}) &= \prod_{e \in E} \Pr(B_e = \beta_e) = \prod_{e \in E} (\text{const}) \cdot (1 - \beta_e)^{-1/2} \beta_e^{-1/2} \cdot \mathbf{1}_{\beta_e \in [0,1]} \\ &= (\text{const}') \prod_{e \in E} (1 - \beta_e)^{-1/2} \beta_e^{-1/2} \cdot \mathbf{1}_{\beta_e \in [0,1]}. \end{aligned}$$

- (b) As usual, in order to compute the posterior of the parameters given independent samples, we use Bayes' rule, where for convenience we write $\beta = (\beta_{\text{As}}, \beta_{\text{Blk}}, \beta_{\text{Mix}}, \beta_{\text{Oth}}, \beta_{\text{Wh}})$,

$$\Pr(\beta \mid y) = \frac{\Pr(y \mid \beta) \cdot \Pr(\beta)}{\Pr(y)} = (\text{const}) \cdot \Pr(\beta) \cdot \prod_{i=1}^n \Pr(y_i \mid \beta).$$

We already know the expression for the prior from (a), so we need to determine the expression for the likelihood term. The outcome of the i -th investigation is assumed to be drawn from the distribution $Y_i \sim \text{Bin}(1, \beta_{\text{eth}_i})$ (i.e. a Binomial distribution whose parameter is determined from the ethnicity of the i -th person). Therefore the probability of an outcome being y given the parameters is

$$\Pr(y_i \mid \beta) = \binom{1}{y_i} (\beta_{\text{eth}_i})^{y_i} (1 - \beta_{\text{eth}_i})^{1-y_i} = (\beta_{\text{eth}_i})^{y_i} (1 - \beta_{\text{eth}_i})^{1-y_i},$$

by noticing that $\binom{1}{y_i} = 1$ for both $y_i = 0$ ($\binom{1}{0} = 1$) and $y_i = 1$ ($\binom{1}{1} = 1$). Now, we can write the posterior as

$$\Pr(\beta \mid y) = (\text{const}'') \prod_{e \in E} (1 - \beta_e)^{-1/2} \beta_e^{-1/2} \cdot \mathbf{1}_{\beta_e \in [0,1]} \prod_{i=1}^n (\beta_{\text{eth}_i})^{y_i} (1 - \beta_{\text{eth}_i})^{1-y_i}.$$

Normally we would be done, but in this case we can determine exactly the distribution for the posterior. To see this we group the terms by β_e and notice that the term β_e appears in the product as many times as people with ethnicity $\text{eth}_i = e$ and $y_i = 1$ (minus $-1/2$). Let's call this number n_e . Similarly, the term $1 - \beta_e$ appears as many times as the number of people with ethnicity $\text{eth}_i = e$ and $y_i = 1$ (minus $-1/2$). Call this number s_e .

$$\Pr(\beta | y) = (\text{const}'') \prod_{e \in E} (1 - \beta_e)^{-1/2+n_e} \beta_e^{-1/2+s_e} \cdot \mathbf{1}_{\beta_e \in [0,1]}.$$

Now we can recognise that this is proportional to the density of the five joint independent Beta distributions and defined over the same domain. So, the posterior for B_e is $\text{Beta}(n_e + 1/2, s_e + 1/2)$.

Note 1: The random variable Y_i follows a Bernoulli distribution. By expressing $Y_i \sim \text{Bin}(1, p)$ hints that you should express $\Pr(Y_i = y) = p^y(1 - p)^{1-y}$. The alternative way to express this probability is using the indicator function, i.e. $\Pr(Y_i = y) = \mathbf{1}_{y=1} \cdot p + \mathbf{1}_{y=0} \cdot (1 - p)$. (We can check these two are equivalent by setting $y = 0$ into both equations and getting $1 - p$ or by setting $y = 1$ and getting p).

Note 2: It is interesting that we had a Beta prior and we ended up with a shifted Beta posterior (when the likelihood is Bernoulli). In Bayesian statistics, we say that the Beta distribution is conjugate to itself with respect to the Bernoulli likelihood.

Question 7

- (a) From Question 3, we know that a prior of $\mathcal{N}(\mu_0 = 5, \rho_0^2 = 3^2)$ for the mean of a Normal distribution likelihood with $\sigma_0 = 3.2$, gives a Normal distribution posterior with

$$\mu_1 = \frac{\frac{m_0}{\rho_0^2} + \frac{\sum x_i}{\sigma_0^2}}{\frac{1}{\rho_0^2} + \frac{n}{\sigma_0^2}} = \frac{\frac{5}{3^2} + \frac{10 \cdot 3 \cdot 30}{3 \cdot 3.2^2}}{\frac{1}{3^2} + \frac{30}{3.2^2}} = 10.106 \quad \text{and} \quad \sigma_1 = \frac{1}{\sqrt{\frac{1}{\rho_0^2} + \frac{n}{\sigma_0^2}}} = \frac{1}{\sqrt{\frac{1}{3^2} + \frac{30}{3.2^2}}} = 0.573$$

- (b) There are two unknown parameters M and H . We assume that the mean of the sick patients is independent of whether the 31-rst patient is sick or healthy. (Think about why this assumption is reasonable). Hence,

$$\Pr(M = \mu, H = h) = \Pr(M = \mu) \cdot \Pr(H = h) = \frac{1}{\sqrt{2\pi\rho_0^2}} \cdot e^{-(\mu-\mu_0)^2/(2\rho_0^2)} \cdot (0.99 \cdot \mathbf{1}_{h=\text{healthy}} + 0.01 \cdot \mathbf{1}_{h=\text{sick}})$$

- (c) Let x_1, \dots, x_{30} be the test results for the 30 sick patients and x_{31} be the result for the last patient. As always we start with Bayes' rule and take advantage of the independence between samples,

$$\begin{aligned} \Pr(M = \mu, H = h | x_1, \dots, x_{31}) &= (\text{const}) \cdot \Pr(x_1, \dots, x_{31} | M = \mu, H = h) \cdot \Pr(M = \mu, H = h) \\ &= (\text{const}) \left(\prod_{i=1}^{31} \Pr(x_i | M = \mu, H = h) \right) \cdot \Pr(M = \mu) \cdot \Pr(H = h) \end{aligned}$$

By noticing that x_i for $i < 31$ is independent of x_{31} (since the last patient being sick is independent of another patient's test result, given the parameters of the model). Hence,

$$\begin{aligned} \Pr(M = \mu, H = h | x_1, \dots, x_{31}) &= (\text{const}) \left(\prod_{i=1}^{30} \Pr(x_i | M = \mu) \right) \cdot \Pr(x_{31} | M = \mu, H = h) \cdot \Pr(M = \mu) \cdot \Pr(H = h) \end{aligned}$$

By regrouping the terms, we recognise the posterior from Question 3, so it can be simplified as follows,

$$\begin{aligned} \Pr(M = \mu, H = h | x_1, \dots, x_{31}) &= (\text{const}) \underbrace{\left(\prod_{i=1}^{30} \Pr(x_i | M = \mu) \right)}_{\text{proportional to posterior from Q3}} \cdot \Pr(M = \mu) \cdot \Pr(x_{31} | M = \mu, H = h) \cdot \Pr(H = h) \\ &= (\text{const}') \exp\left(-\frac{(\mu_1 - \mu)^2}{2 \cdot \sigma_1^2}\right) \cdot \Pr(x_{31} | M = \mu, H = h) \cdot \Pr(H = h). \end{aligned}$$

Notice that μ_1 depends on the mean of the samples x_1, \dots, x_{30} . Now, we look at the remaining terms

$$\Pr(x_{31} \mid M = \mu, H = h) \cdot \Pr(H = h) = \begin{cases} 0.99 \cdot \frac{1}{\sqrt{2\pi \cdot 2.1^2}} \exp\left(-\frac{x_{31}^2}{2 \cdot 2.1^2}\right) & \text{if } h = \text{healthy} \\ 0.01 \cdot \frac{1}{\sqrt{2\pi \cdot 3.2^2}} \exp\left(-\frac{(x_{31} - \mu)^2}{2 \cdot 3.2^2}\right) & \text{if } h = \text{sick}. \end{cases}$$

Combining all of these we get,

$$\begin{aligned} & \Pr(M = \mu, H = h \mid x_1, \dots, x_{31}) \\ &= (\text{const}') \exp\left(-\frac{(\mu_1 - \mu)^2}{2 \cdot \sigma_1^2}\right) \cdot \begin{cases} 0.99 \cdot \frac{1}{\sqrt{2\pi \cdot 2.1^2}} \exp\left(-\frac{x_{31}^2}{2 \cdot 2.1^2}\right) & \text{if } h = \text{healthy} \\ 0.01 \cdot \frac{1}{\sqrt{2\pi \cdot 3.2^2}} \exp\left(-\frac{(x_{31} - \mu)^2}{2 \cdot 3.2^2}\right) & \text{if } h = \text{sick}. \end{cases} \end{aligned}$$

- (d) Now, we need to slightly change our posterior since we are given the mean $\bar{x}_{1:30}$ of the samples x_1, \dots, x_{30} and not the actual samples. Since the average of normal distributions follows a normal distribution, so $\bar{X}_{1:30} \sim \mathcal{N}(\mu, \frac{3.2^2}{30})$.

$$\begin{aligned} & \Pr(M = \mu, H = h \mid \bar{x}_{1:30}, x_{31}) \\ &= (\text{const}') \exp\left(-\frac{(\bar{x}_{1:30} - \mu)^2}{2 \cdot 3.2^2/30}\right) \cdot \begin{cases} 0.99 \cdot \frac{1}{\sqrt{2\pi \cdot 2.1^2}} \exp\left(-\frac{x_{31}^2}{2 \cdot 2.1^2}\right) & \text{if } h = \text{healthy} \\ 0.01 \cdot \frac{1}{\sqrt{2\pi \cdot 3.2^2}} \exp\left(-\frac{(x_{31} - \mu)^2}{2 \cdot 3.2^2}\right) & \text{if } h = \text{sick}. \end{cases} \end{aligned}$$

We want to compute $\Pr(H = h \mid \bar{x}_{1:30}, x_{31})$, which we do using marginalisation and sampling. The first step is,

$$\Pr(H = h \mid \bar{x}_{1:30}, x_{31}) = \int_{\mu=-\infty}^{\infty} \Pr(M = \mu, H = h \mid \bar{x}_{1:30}, x_{31}) d\mu.$$

The second step is to evaluate this integral using Monte-Carlo estimates, i.e. we sample M and H using the given priors, compute the unnormalised probabilities and then normalise to find the desired probability. The code below does this,

```
import numpy as np
import scipy.stats

samples = 100000 # The number of samples to take.
p_healthy = 0.99 # Probability of a patient being healthy.
mu_healthy, sigma_healthy = 0, 2.1 # Parameters of the Normal distribution for the test
↳ result of healthy patient.
sigma_sick = 3.2 # The st. deviation of the Normal distribution for the test result of
↳ sick patient.
prior_mu_sick, prior_sigma_sick = 5, 3 # Parameters of the Normal prior distribution for
↳ mean of sick patients.
x_31 = 8.8 # The result of the unknown patient.
x_mean, n = 10.3, 30 # The mean result for the 30 sick patients.

# Collecting the samples. We need to sample both parameters: health status and mean for
↳ sick patients.
health_status = np.random.choice(['healthy', 'sick'], p=[p_healthy, 1.0 - p_healthy],
↳ replace=True, size=samples)
mu = np.random.normal(loc=prior_mu_sick, scale=prior_sigma_sick, size=samples)

# The likelihood of the 31st patient P(x_31 | mu, h) for the two cases of h.
likelihood_x31_healthy = scipy.stats.norm.pdf(x=x_31, loc=mu_healthy, scale=sigma_healthy)
likelihood_x31_sick = scipy.stats.norm.pdf(x=x_31, loc=mu, scale=sigma_sick)

# P(x_{1:30} | mu, h) * P(x_31 | mu, h)
# Note: This is proportional to the posterior (the prior terms are added through sampling).
weights = np.exp(-(x_mean - mu) ** 2 / (2 * n * sigma_sick ** 2)) \
* np.where(health_status == 'healthy', likelihood_x31_healthy,
↳ likelihood_x31_sick)
```

```

# By summing for both healthy and sick, we can marginalise out "mu".
posterior_healthy = np.sum(weights[health_status == 'healthy']) / np.sum(weights)
print((posterior_healthy, 1 - posterior_healthy))

```

Question 9

(a) By defining $Z = \mathbf{1}_{Y \in [0.5, 0.7]}$, we want to estimate $\Pr(Z | X)$.

$$\Pr(X | Z) = \frac{\Pr(Z | X) \cdot \Pr(X)}{\Pr(Z)} = (\text{const}) \cdot \Pr(Z | X) \cdot \Pr(X)$$

We can therefore generate samples of X , then generate samples of Y and finally count the samples for which $Z = 1$ and normalise to find the probability. This is shown in the code below

```

import numpy as np
import matplotlib.pyplot as plt

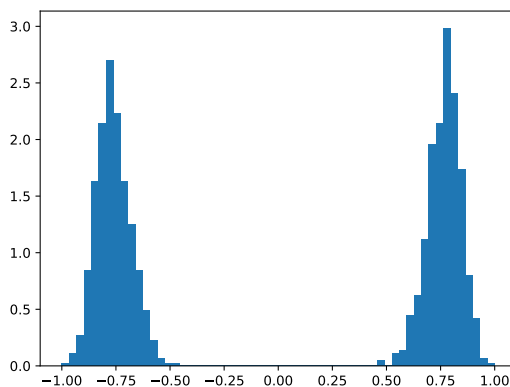
# Generate the samples.
samples = 10000
xs = np.random.uniform(-1, 1, size=samples)
ys = np.random.normal(loc=xs ** 2, scale=0.1, size=samples)

# Compute which ones satisfy the criteria.
z = np.array(np.logical_and(ys >= 0.5, ys <= 0.7), dtype=int)

plt.hist(xs, weights=z, density=True, bins=np.linspace(-1, 1, 60))
plt.show()

```

This gives the following figure.



(b) By the definition of conditional probability,

$$\Pr(X | X \geq 0) = \frac{\Pr(X \geq 0, X)}{\Pr(X \geq 0)}$$

If $X < 0$, then the probability is zero. Otherwise, $\Pr(X \geq 0, X) = \Pr(X)$, hence

$$\Pr(X | X \geq 0) = \frac{\text{scipy.stats.normal.pdf}(x, \mu, \sigma)}{1 - \text{scipy.stats.normal.cdf}(0, \mu, \sigma)}$$

```

import numpy as np
import scipy.stats

```

```
def compute_prob(x, mu, sigma):
    if x < 0:
        return 0
    prob = scipy.stats.norm.pdf(x, loc=mu, scale=sigma) / (
        1 - scipy.stats.norm.cdf(0, loc=mu, scale=sigma))
    print(f'prob: {prob}')
    print(f'Compare with: {scipy.stats.norm.pdf(x, loc=mu, scale=sigma)}')
```

```
compute_prob(x=4, mu=2.4, sigma=6)
compute_prob(x=1, mu=-1, sigma=3)
```

Question 12

Let H_i be the indicator random variable for whether the i -th point is an outlier or not. Then,

$$\begin{aligned}
 \Pr(H_5 \mid x_1, \dots, x_n) &= \int_{\mu=-\infty}^{\infty} \Pr(H_5, \mu \mid x_1, \dots, x_n) d\mu \\
 &= \int_{\mu=-\infty}^{\infty} \frac{\Pr(H_5, \mu, x_1, \dots, x_n)}{\Pr(x_1, \dots, x_n)} d\mu \\
 &= \int_{\mu=-\infty}^{\infty} \frac{\Pr(H_5, \mu, x_1, \dots, x_n)}{\Pr(x_1, \dots, x_n)} \cdot \frac{\Pr(H_5, \mu)}{\Pr(H_5, \mu)} d\mu \\
 &= (\text{const}) \cdot \int_{\mu=-\infty}^{\infty} \Pr(x_1, \dots, x_n \mid H_5, \mu) \Pr(H_5, \mu) d\mu \\
 &= (\text{const}) \cdot \int_{\mu=-\infty}^{\infty} \prod_{i=1}^n \Pr(x_i \mid H_5, \mu) \Pr(H_5, \mu) d\mu \\
 &= (\text{const}) \cdot \int_{\mu=-\infty}^{\infty} \left(\prod_{i \neq 9} \Pr(x_i \mid \mu) \right) \cdot \Pr(x_i \mid H_5, \mu) \Pr(H_5) \Pr(\mu) d\mu \\
 &= (\text{const}) \cdot \int_{\mu=-\infty}^{\infty} \left(\prod_{i \neq 9} (0.01 \cdot p_{\text{cauchy}}(x_i) + 0.99 \cdot p_{\text{norm}}(x_i)) \right) \Pr(\mu) \cdot \begin{cases} 0.01 \cdot p_{\text{cauchy}}(x_5) & \text{if } H_5 = 1 \\ 0.99 \cdot p_{\text{normal}}(x_5) & \text{if } H_5 = 0 \end{cases} d\mu
 \end{aligned}$$

Now, we estimate the integrals for $\Pr(H_5 = 1 \mid x_1, \dots, x_n)$ and $\Pr(H_5 = 0 \mid x_1, \dots, x_n)$ up to the (const) factor. Then by normalising we get an estimate for $\Pr(H_5 = 1 \mid x_1, \dots, x_n)$. The following code implements this,

```
import numpy as np
import scipy.stats

x = [4.3, 2.8, 3.9, 4.1, 4.5, 3.3] # Dataset without x5.
x_5 = 9 # The fifth sample in the original dataset.
samples = 10000 # Number of samples to use.
prior_mu, prior_sigma = 0, 5 # Parameters for the Normal prior.
p_cauchy = 0.01
likelihood_sigma = 0.5

def prx(x, mu, h_5):
    p1 = scipy.stats.norm.pdf(x, loc=mu, scale=likelihood_sigma)
    p2 = scipy.stats.cauchy.pdf(x)
    # P(x_1, .., x_4, x_6, x_7 | mu)
    p_others = np.prod((1 - p_cauchy) * p1 + p_cauchy * p2)
```

```

# P(x_5 | h_5, mu)
if h_5 == 'cauchy':
    p_5 = scipy.stats.cauchy.pdf(x_5)
else:
    p_5 = scipy.stats.norm.pdf(x_5, loc=mu, scale=likelihood_sigma)
return p_others * p_5

# Generate the samples.
mu = np.random.normal(loc=prior_mu, scale=prior_sigma, size=samples)
h_5 = np.random.choice(['cauchy', 'normal'], p=[p_cauchy, 1 - p_cauchy], replace=True,
    ↪ size=samples)

# Compute P(h_9, mu | x1, ... , xn).
posterior = np.array([prx(x, m, h) for m, h in zip(mu, h_5)])

# Marginalise out mu.
w_cauchy = np.sum(posterior[h_5 == 'cauchy'])
w_normal = np.sum(posterior[h_5 == 'normal'])

# Normalise to cancel out the constant.
print(w_cauchy / (w_cauchy + w_normal))

```