

MIDTERM EXAM

ISyE6420

Fall 2020

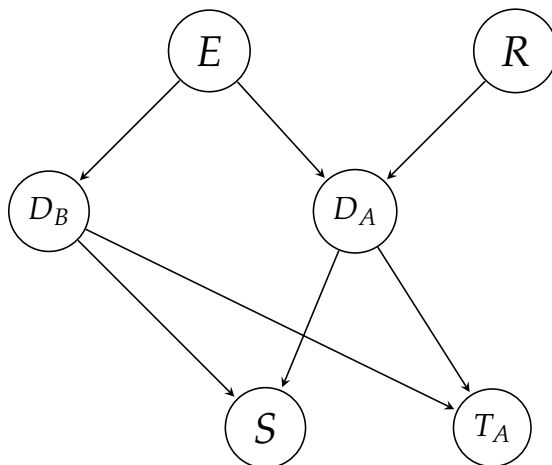
Released October 16, 12:00pm – due October 25, 11:55pm. This exam is not proctored and not time limited except the due date. Late submissions will not be accepted.

Use of all available electronic and printed resources is allowed except direct communication that violates Georgia Tech Academic Integrity Rules.

Name Chen-Yang(Jim), Liu

Problem	1	2	3	Total
Score	/33	/33	/34	/100

1. Bayes Network. Incidences of diseases A and B (D_A, D_B) depend on the exposure (E). Disease A is additionally influenced by risk factors (R). Both diseases lead to symptoms (S). Results of the test for disease A (T_A) are affected also by disease B. Positive test will be denoted as $T_A = 1$, negative as $T_A = 0$. The Bayes Network is shown in Figure 1. Needed conditional probabilities are shown in Table 1.



E	0	1
	0.8	0.2

R	0	1
	0.7	0.3

D_A	0	1
$E^c R^c$	0.9	0.1
$E^c R$	0.4	0.6
ER^c	0.5	0.5
ER	0.3	0.7

D_B	0	1
E^c	0.8	0.2
E	0.3	0.7

S	0	1
$D_A^c D_B^c$	0.95	0.05
$D_A^c D_B$	0.6	0.4
$D_A D_B^c$	0.4	0.6
$D_A D_B$	0.1	0.9

T_A	0	1
$D_A^c D_B^c$	0.92	0.08
$D_A^c D_B$	0.8	0.2
$D_A D_B^c$	0.15	0.85
$D_A D_B$	0.03	0.97

- (a) What is the probability of disease A ($D_A = 1$), if disease B is not present ($D_B = 0$), but symptoms are present ($S = 1$).
- (b) What is the probability of exposure ($E = 1$), if symptoms are present ($S = 1$) and test is positive ($T_A = 1$).

Answers:

```
[17]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
%matplotlib inline
```

1 Bayes Network

Incidences of diseases A and B (DA,DB) depend on the exposure (E). Disease A is additionally influenced by risk factors (R). Both diseases lead to symptoms (S). Results of the test for disease A (TA) are affected also by disease B. Positive test will be denoted as TA = 1, negative as TA = 0. The Bayes Network is shown in Figure 1. Needed conditional probabilities are shown in Table 1.

```
[2]: from pomegranate import DiscreteDistribution,␣  
      ConditionalProbabilityTable, Node, State, BayesianNetwork
```

```
[42]: riskFactor = DiscreteDistribution({"R0": 0.7, "R1": 0.3})  
      exposure = DiscreteDistribution({"E0": 0.8, "E1": 0.2})  
  
      Db = ConditionalProbabilityTable([["E0", "Db0", 0.8],  
                                       ["E0", "Db1", 0.2],  
                                       ["E1", "Db0", 0.3],  
                                       ["E1", "Db1", 0.7]], [exposure])  
  
      Da = ConditionalProbabilityTable([["E0", "R0", "Da0", 0.9],  
                                       ["E0", "R0", "Da1", 0.1],  
                                       ["E0", "R1", "Da0", 0.4],  
                                       ["E0", "R1", "Da1", 0.6],  
                                       ["E1", "R0", "Da0", 0.5],  
                                       ["E1", "R0", "Da1", 0.5],  
                                       ["E1", "R1", "Da0", 0.3],  
                                       ["E1", "R1", "Da1", 0.7]], [exposure,␣  
                                       riskFactor])  
  
      S = ConditionalProbabilityTable([["Da0", "Db0", "S0", 0.95],  
                                       ["Da0", "Db0", "S1", 0.05],  
                                       ["Da0", "Db1", "S0", 0.6],  
                                       ["Da0", "Db1", "S1", 0.4],  
                                       ["Da1", "Db0", "S0", 0.4],  
                                       ["Da1", "Db0", "S1", 0.6],  
                                       ["Da1", "Db1", "S0", 0.1],  
                                       ["Da1", "Db1", "S1", 0.9]], [Da, Db])  
  
      Ta = ConditionalProbabilityTable([["Da0", "Db0", "Ta0", 0.92],  
                                       ["Da0", "Db0", "Ta1", 0.08],  
                                       ["Da0", "Db1", "Ta0", 0.8],
```

```

["Da0", "Db1", "Ta1", 0.2],
["Da1", "Db0", "Ta0", 0.15],
["Da1", "Db0", "Ta1", 0.85],
["Da1", "Db1", "Ta0", 0.03],
["Da1", "Db1", "Ta1", 0.97]], [Da, Db])

s1 = State(riskFactor, name="riskFactor")
s2 = State(exposure, name="exposure")
s3 = State(Da, name="Da")
s4 = State(Db, name="Db")
s5 = State(S, name="S")
s6 = State(Ta, name="Ta")

model = BayesianNetwork("Disease")
model.add_nodes(s1, s2, s3, s4, s5, s6)
model.add_edge(s1, s3)
model.add_edge(s2, s3)
model.add_edge(s2, s4)
model.add_edge(s3, s5)
model.add_edge(s3, s6)
model.add_edge(s4, s5)
model.add_edge(s4, s6)

model.bake()

```

```
[46]: model.predict_proba({})
```

```

[46]: array([{'class' : "Distribution",
  "dtype" : "str",
  "name" : "DiscreteDistribution",
  "parameters" : [
    {
      "R0" : 0.7,
      "R1" : 0.30000000000000001
    }
  ],
  "frozen" : false
},
  {
    "class" : "Distribution",
    "dtype" : "str",
    "name" : "DiscreteDistribution",
    "parameters" : [

```

```

        {
            "E0" :0.7999999999999997,
            "E1" :0.20000000000000004
        }
    ],
    "frozen" :false
},
    {
        "class" : "Distribution",
        "dtype" : "str",
        "name" : "DiscreteDistribution",
        "parameters" : [
            {
                "Da1" :0.312000000000000044,
                "Da0" :0.6879999999999996
            }
        ],
        "frozen" :false
    },
    {
        "class" : "Distribution",
        "dtype" : "str",
        "name" : "DiscreteDistribution",
        "parameters" : [
            {
                "Db0" :0.6999999999999996,
                "Db1" :0.30000000000000004
            }
        ],
        "frozen" :false
    },
    {
        "class" : "Distribution",
        "dtype" : "str",
        "name" : "DiscreteDistribution",
        "parameters" : [
            {
                "S1" :0.32192000000000004,
                "S0" :0.6780799999999996
            }
        ],
        "frozen" :false
    },
    {

```

```

    "class" : "Distribution",
    "dtype" : "str",
    "name" : "DiscreteDistribution",
    "parameters" : [
        {
            "Ta1" : 0.35624000000000002,
            "Ta0" : 0.6437599999999999
        }
    ],
    "frozen" : false
}], dtype=object)

```

1.1 Performing Inference

- (a) What is the probability of disease A ($DA = 1$), if disease B is not present ($DB = 0$), but symptoms are present ($S = 1$).
- (b) What is the probability of exposure ($E = 1$), if symptoms are present ($S = 1$) and test is positive ($TA = 1$).

```

[47]: # (a)
model.predict_proba({"Db": "Db0", "S": "S1"})

```

```

[47]: array([
    {
        "class" : "Distribution",
        "dtype" : "str",
        "name" : "DiscreteDistribution",
        "parameters" : [
            {
                "R0" : 0.4289652247667518,
                "R1" : 0.5710347752332482
            }
        ],
        "frozen" : false
    },
    {
        "class" : "Distribution",
        "dtype" : "str",
        "name" : "DiscreteDistribution",
        "parameters" : [
            {
                "E0" : 0.8481764206955046,
                "E1" : 0.1518235793044955
            }
        ]
    }
])

```

```

    ],
    "frozen" :false
},
    {
    "class" : "Distribution",
    "dtype" : "str",
    "name" : "DiscreteDistribution",
    "parameters" : [
        {
            "Da1" : 0.821034775233248,
            "Da0" : 0.1789652247667519
        }
    ],
    "frozen" :false
},
    'Db0', 'S1',
    {
    "class" : "Distribution",
    "dtype" : "str",
    "name" : "DiscreteDistribution",
    "parameters" : [
        {
            "Ta1" : 0.712196776929601,
            "Ta0" : 0.28780322307039885
        }
    ],
    "frozen" :false
}], dtype=object)

```

So based on above the Bayesian Network, $P(D_A = 1 | D_B = 0, S = 1) = 0.821034775233248$

```

[48]: #(b) What is the probability of exposure ( $E = 1$ ), if symptoms are present  $\hookrightarrow (S = 1)$  and test is positive ( $TA = 1$ ).
      model.predict_proba([{'S': "S1", 'Ta': "Ta1"}])

```

```

[48]: [array([
    "class" : "Distribution",
    "dtype" : "str",
    "name" : "DiscreteDistribution",
    "parameters" : [
        {
            "R0" : 0.46352398227589187,
            "R1" : 0.5364760177241081

```

```

    }
  ],
  "frozen" :false
},
{
  "class" : "Distribution",
  "dtype" : "str",
  "name" : "DiscreteDistribution",
  "parameters" : [
    {
      "E0" : 0.5779343661124118,
      "E1" : 0.4220656338875882
    }
  ],
  "frozen" :false
},
{
  "class" : "Distribution",
  "dtype" : "str",
  "name" : "DiscreteDistribution",
  "parameters" : [
    {
      "Da1" : 0.9230809724366711,
      "Da0" : 0.07691902756332904
    }
  ],
  "frozen" :false
},
{
  "class" : "Distribution",
  "dtype" : "str",
  "name" : "DiscreteDistribution",
  "parameters" : [
    {
      "Db0" : 0.4589910240916621,
      "Db1" : 0.541008975908338
    }
  ],
  "frozen" :false
},
('S1', 'Ta1'], dtype=object)]

```

So based on above the Bayesian Network, $P(E = 1|T_A = 1, S = 1) = 0.4220656338875882$

2. Times to Failure. Three devices are monitored until failure. The observed lifetimes are 0.9, 1.8, and 0.3 years. If the lifetimes are modeled as exponential distribution with rate λ ,

$$T_i \sim \text{Exp}(\lambda), \quad f(t|\lambda) = \lambda e^{-\lambda t}, t > 0, \lambda > 0.$$

Assume exponential prior on λ ,

$$\lambda \sim \text{Exp}(2), \quad \pi(\lambda) = 2e^{-2\lambda}, \lambda > 0.$$

- (a) Find the posterior distribution of λ .
- (b) Find the Bayes estimator for λ .
- (c) Find the MAP estimator for λ .
- (d) Numerically find 95% equitailed confidence interval for λ .
- (e) Find the posterior probability of hypothesis $H_0 : \lambda \leq 1/2$.

Answers:

(a)

$$\begin{aligned}
 \pi(\lambda|T) &\propto f(T|\lambda)\pi(\lambda) \\
 &\equiv \prod_{i=1}^3 \lambda e^{-\lambda t_i} 2e^{-2\lambda}, \quad t > 0, \lambda > 0 \\
 &= \lambda^3 e^{-\lambda \sum_{i=1}^3 t_i} 2e^{-2\lambda} \\
 &= 2\lambda^3 e^{-\lambda(\sum_{i=1}^3 t_i) - 2\lambda} \\
 &= 2\lambda^3 e^{-\lambda(\sum_{i=1}^3 t_i + 2)} \\
 &= 2\lambda^{4-1} e^{-\lambda(\sum_{i=1}^3 t_i + 2)} \\
 &\propto \frac{(\sum_{i=1}^3 t_i + 2)^4}{\Gamma(4)} \lambda^{4-1} e^{-\lambda(\sum_{i=1}^3 t_i + 2)} \\
 &\sim \text{Gamma}(\alpha = 4, \beta = \sum_{i=1}^3 t_i + 2) = \text{Gamma}(\alpha = 4, \beta = 0.9 + 1.8 + 0.3 + 2)
 \end{aligned}$$

(b)

$$E[(\pi(\lambda|T))] = \frac{\alpha}{\beta} = \frac{4}{5}$$

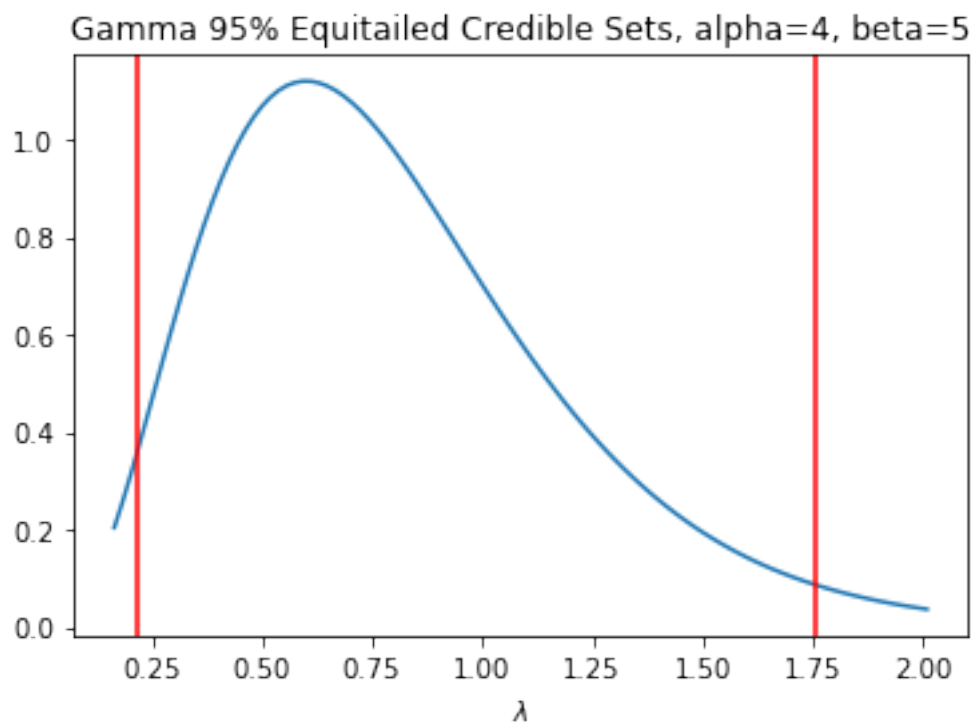
(c)

$$\begin{aligned}\frac{\partial \pi(\lambda|T)}{\partial \lambda} &= \frac{\partial}{\partial \lambda} \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta \lambda} \\ &= (\alpha - 1) \lambda^{\alpha-2} e^{-\beta \lambda} - \beta \lambda^{\alpha-1} e^{-\beta \lambda} = 0 \\ &= e^{-\beta \lambda} ((\alpha - 1) \lambda^{\alpha-2} - \beta \lambda^{\alpha-1}) = 0, 0 \leq \lambda < \infty, \alpha, \beta > 0 \\ &= e^{-\beta \lambda} \lambda^{\alpha-2} (\alpha - 1 - \beta \lambda) = 0, e^{-\beta \lambda} > 0 \\ &= \lambda = 0 \text{ or } \frac{\alpha - 1}{\beta} \\ &= \alpha \geq 1, \frac{\alpha - 1}{\beta} > 0, 0 \text{ is not a mode in this problem} \\ \frac{\alpha - 1}{\beta} &= \frac{4 - 1}{5} = \frac{3}{5}\end{aligned}$$

(d)

```
[117]: # (d) Numerically find 95% equitailed confidence interval for .
from scipy.stats import gamma
fig, ax = plt.subplots(1, 1)
alpha = 4
beta = 5
x = np.linspace(gamma.ppf(0.01, a=alpha, scale=1/beta), gamma.ppf(0.99,
↪a=alpha, scale=1/beta), 100)

ax.plot(x, gamma.pdf(x, a=alpha, scale=1/beta))
plt.axvline(x=gamma.ppf(0.025, a=alpha, scale=1/beta), color='r')
plt.axvline(x=gamma.ppf(0.975, a=alpha, scale=1/beta), color='r')
plt.title("Gamma 95% Equitailed Credible Sets, alpha=4, beta=5")
plt.xlabel("$\lambda$")
plt.show()
```



```
[123]: print("lambda range in 95% Equitailed Credible Set: [{}, {}]"
         .format(gamma.
         ↪ppf(0.025, a=alpha, scale=1/beta),
         gamma.
         ↪ppf(0.975, a=alpha, scale=1/beta)))
```

```
lambda range in 95% Equitailed Credible Set: [0.21797307472526498,
1.7534546139484648]
```

```
[121]: # (e) Find the posterior probability of hypothesis H0 : 1/2.
gamma.cdf(1/2, a=alpha, loc=0, scale=1/beta)
```

```
[121]: 0.2424238668669339
```

3. Gibbs and High/Low Protein Diet in Rats. Armitage and Berry (1994, p. 111) report data on the weight gain of 19 female rats between 28 and 84 days after birth. The rats were placed on diets with high (12 animals) and low (7 animals) protein content.

High protein	Low protein
134	70
146	118
104	101
119	85
124	107
161	132
107	94
83	
113	
129	
97	
123	

We want to test the hypothesis on dietary effect. Did a low protein diet result in significantly lower weight gain?

The classical t test against one sided alternative will be significant. We will do the test Bayesian way using Gibbs sampler.

Assume that high-protein diet measurements $y_{1i}, i = 1, \dots, 12$ are coming from normal distribution $\mathcal{N}(\theta_1, 1/\tau_1)$, where τ_1 is precision parameter,

$$f(y_{1i}|\theta_1, \tau_1) \propto \tau_1^{1/2} \exp \left\{ -\frac{\tau_1}{2} (y_{1i} - \theta_1)^2 \right\}, i = 1, \dots, 12.$$

Low-protein diet measurements $y_{2i}, i = 1, \dots, 7$ are coming from normal distribution $\mathcal{N}(\theta_2, 1/\tau_2)$,

$$f(y_{2i}|\theta_2, \tau_2) \propto \tau_2^{1/2} \exp \left\{ -\frac{\tau_2}{2} (y_{2i} - \theta_2)^2 \right\}, i = 1, \dots, 7.$$

Assume that θ_1 and θ_2 have normal priors $\mathcal{N}(\theta_{10}, 1/\tau_{10})$ and $\mathcal{N}(\theta_{20}, 1/\tau_{20})$, respectively. Take prior means as $\theta_{10} = \theta_{20} = 110$ (apriori no preference) and precisions as $\tau_{10} = \tau_{20} = 1/100$.

Assume that τ_1 and τ_2 have the gamma $\mathcal{Ga}(a_1, b_1)$ and $\mathcal{Ga}(a_2, b_2)$ priors with shapes $a_1 = a_2 = 0.01$ and rates $b_1 = b_2 = 4$.

(a) Construct Gibbs sampler that will sample $\theta_1, \tau_1, \theta_2$, and τ_2 from their posteriors.

(b) Find sample differences $\theta_1 - \theta_2$. Proportion of positive differences approximates posterior probability of hypothesis $H_0 : \theta_1 > \theta_2$. What is this proportion?

(c) Using sample quantiles find the 95% equitailed credible set for $\theta_1 - \theta_2$. Does this set contain 0?

Answers:

(a) Construct Gibbs sampler that will sample $\theta_1, \tau_1, \theta_2$, and τ_2 from their posteriors.

- High Protein Posterior $\propto f(y_{1i}|\theta_1, \tau_1)\pi(\theta_1)\pi(\tau_1)$, $\pi(\theta_1) \sim N(110, 100), \pi(\tau_1) \sim \text{Gamma}(\alpha = 0.01, \text{beta} = 4)$
- Low Protein Posterior $\propto f(y_{2i}|\theta_2, \tau_2)\pi(\theta_2)\pi(\tau_2)$, $\pi(\theta_2) \sim N(110, 100), \pi(\tau_2) \sim \text{Gamma}(\alpha = 0.01, \text{beta} = 4)$

1. joint distribution of High Protein

$$\text{joint} \propto [\prod_{i=1}^{12} \tau_1^{0.5} e^{-\frac{\tau_1}{2}(y_{1i}-\theta_1)^2}] \frac{1}{\sqrt{2\pi 100^2}} e^{-\frac{(\theta_1-110)^2}{2*100^2}} \frac{b_1^{a_1}}{\Gamma(0.01)} \tau_1^{0.01-1} e^{-b_1 \tau_1}$$

$$\begin{aligned} \pi(\theta_1|Y_{1i}, \tau_1) &\propto e^{-\frac{\tau_1}{2} \sum_{i=1}^{12} (y_{1i}-\theta_1)^2} e^{-\frac{1}{2*100^2} (\theta_1-110)^2} \\ &\propto e^{-\frac{\tau_1}{2} \sum_{i=1}^{12} (y_{1i}^2 - 2\theta_1 y_{1i} + \theta_1^2) - \frac{1}{2*100^2} (\theta_1-110)^2} \\ &\propto e^{-\frac{\tau_1}{2} [\sum_{i=1}^{12} y_{1i}^2 - 2\theta_1 \sum_{i=1}^{12} y_{1i} + 12\theta_1^2] - \frac{1}{2*100^2} [\theta_1^2 - 220\theta_1 + 110^2]} \\ &\propto e^{\theta_1 \tau_1 \sum_{i=1}^{12} y_{1i} - 6\tau_1 \theta_1^2 - \frac{\theta_1^2}{2*100^2} + \frac{110}{100^2} \theta_1} e^{-\frac{\tau_1 \sum_{i=1}^{12} y_{1i}^2}{2} - \frac{110^2}{2*100^2}} \\ &\propto e^{\theta_1 \tau_1 \sum_{i=1}^{12} y_{1i} - 6\tau_1 \theta_1^2 - \frac{\theta_1^2}{2*100^2} + \frac{110}{100^2} \theta_1} \times \text{Constant} \\ &\propto e^{\theta_1 \tau_1 \sum_{i=1}^{12} y_{1i} - 6\tau_1 \theta_1^2 - \frac{\theta_1^2}{2*100^2} + \frac{110}{100^2} \theta_1} \\ &\propto e^{\theta_1^2 (-\frac{12}{2} \tau_1 - \frac{1}{2*100^2}) + \theta_1 (\tau_1 \sum_{i=1}^{12} y_{1i} + \frac{220}{2*100^2})} \\ &\propto e^{-\frac{1}{2} (12\tau_1 + \frac{1}{100^2}) [\theta_1^2 - \frac{\tau_1 \sum_{i=1}^{12} y_{1i} + \frac{220}{2*100^2}}{-0.5(12\tau_1 + \frac{1}{100^2})} \theta_1]} \\ &\propto e^{-\frac{1}{2} (12\tau_1 + \frac{1}{100^2}) [\theta_1 - 0.5 (\frac{\tau_1 \sum_{i=1}^{12} y_{1i} + \frac{220}{2*100^2}}{-0.5(12\tau_1 + \frac{1}{100^2})})]^2 + \text{constant}} \\ &\propto e^{-\frac{1}{2} (12\tau_1 + \frac{1}{100^2}) [\theta_1 - 0.5 (\frac{\tau_1 \sum_{i=1}^{12} y_{1i} + \frac{220}{2*100^2}}{-0.5(12\tau_1 + \frac{1}{100^2})})]^2} \\ &\sim N(\frac{1}{2} (\frac{\tau_1 \sum_{i=1}^{12} y_{1i} + \frac{220}{2*100^2}}{-0.5(12\tau_1 + \frac{1}{100^2})}), \frac{1}{12\tau_1 + \frac{1}{100^2}}) \\ \pi(\tau_1|y_{1i}, \theta_1) &\propto \tau_1^{\frac{12}{2}} e^{-\frac{\tau_1}{2} \sum_{i=1}^{12} (y_{1i}-\theta_1)^2} \tau_1^{0.01-1} e^{-4\tau_1} \\ &\propto \tau_1^{\frac{12}{2} + 0.01 - 1} e^{-\tau_1 (\frac{1}{2} \sum_{i=1}^{12} (y_{1i}-\theta_1)^2 + 4)} \\ &\sim \text{Gamma}(\frac{12}{2} + 0.01, \frac{1}{2} \sum_{i=1}^{12} (y_{1i} - \theta_1)^2 + 4) \end{aligned}$$

Similarly,

2. joint distribution of Low Protein

$$\begin{aligned}
\pi(\theta_2|Y_{2i}, \tau_2) &\propto e^{-\frac{\tau_2}{2} \sum_{i=1}^7 (y_{2i} - \theta_2)^2} e^{-\frac{1}{2 \cdot 100^2} (\theta_2 - 110)^2} \\
&\propto e^{-\frac{\tau_2}{2} \sum_{i=1}^7 (y_{2i}^2 - 2\theta_2 y_{2i} + \theta_2^2) - \frac{1}{2 \cdot 100^2} (\theta_2 - 110)^2} \\
&\propto e^{-\frac{\tau_2}{2} [\sum_{i=1}^7 y_{2i}^2 - 2\theta_2 \sum_{i=1}^7 y_{2i} + 7\theta_2^2] - \frac{1}{2 \cdot 100^2} [\theta_2^2 - 220\theta_2 + 110^2]} \\
&\propto e^{\theta_2 \tau_2 \sum_{i=1}^7 y_{2i} - \frac{7}{2} \tau_2 \theta_2^2 - \frac{\theta_2^2}{2 \cdot 100^2} + \frac{220}{2 \cdot 100^2} \theta_2} \times \text{constant} \\
&\propto e^{\theta_2 \tau_2 \sum_{i=1}^7 y_{2i} - \frac{7}{2} \tau_2 \theta_2^2 - \frac{\theta_2^2}{2 \cdot 100^2} + \frac{220}{2 \cdot 100^2} \theta_2} \\
&\propto e^{\theta_2^2 (-\frac{7}{2} \tau_2 - \frac{1}{2 \cdot 100^2}) + \theta_2 (\tau_2 \sum_{i=1}^7 y_{2i} + \frac{220}{2 \cdot 100^2})} \\
&\propto e^{-\frac{1}{2} (7\tau_2 + \frac{1}{100^2}) [\theta_2^2 - \frac{\tau_2 \sum_{i=1}^7 y_{2i} + \frac{220}{2 \cdot 100^2}}{-0.5(7\tau_2 + \frac{1}{100^2})} \theta_2]} \\
&\sim N\left(\frac{1}{2} \left(\frac{\tau_2 \sum_{i=1}^7 y_{2i} + \frac{220}{2 \cdot 100^2}}{-0.5(7\tau_2 + \frac{1}{100^2})} \right), \frac{1}{7\tau_2 + \frac{1}{100^2}}\right) \\
\pi(\tau_2|y_{2i}, \theta_2) &\propto \tau_2^{\frac{7}{2}} e^{-\frac{\tau_2}{2} \sum_{i=1}^7 (y_{2i} - \theta_2)^2} \tau_2^{0.01-1} e^{-4\tau_2} \\
&\propto \tau_2^{\frac{7}{2} + 0.01 - 1} e^{-\tau_2 (\frac{1}{2} \sum_{i=1}^7 (y_{2i} - \theta_2)^2 + 4)} \\
&\sim \text{Gamma}\left(\frac{7}{2} + 0.01, \frac{1}{2} \sum_{i=1}^7 (y_{2i} - \theta_2)^2 + 4\right)
\end{aligned}$$

(b) Find sample differences $\theta_1 - \theta_2$. Proportion of positive differences approximates posterior probability of hypothesis $H_0 : \theta_1 > \theta_2$. What is this proportion?

```
[10]: high = np.array([134, 146, 104, 119, 124, 161, 107, 83, 113, 129, 97,
    ↪123])
low = np.array([70, 118, 101, 85, 107, 132, 94])
```

```
[11]: ## setup for theta1, tau1
theta1 = 1
tau1 = 1
y1_sum = high.sum() # 1440
# iterations
burn = 1000
ntotal = 10000 + burn
# update list
theta1s = np.zeros(ntotal)
tau1s = np.zeros(ntotal)

theta1s[0] = theta1
tau1s[0] = tau1
```

```
[12]: for i in range(1, ntotal):
        mu = - (tau1 * y1_sum + 220/(2 * 100**2)) / (12 * tau1 + 1/(100**2))
        sigma2 = 1 / (12 * tau1 + 1/100**2)
        theta1 = np.random.normal(mu, sigma2)

        var_ga = ((high-theta1) ** 2).sum()
        tau1 = np.random.gamma(shape=12/2+0.01, scale=1/(var_ga/2+4))
        theta1s[i] = theta1
        tau1s[i] = tau1
```

```
[13]: ## setup for theta1, tau1
        theta2 = 1
        tau2 = 1
        y2_sum = low.sum() # 1440
        # iterations
        burn =1000
        ntotal = 10000 + burn
        # update list
        theta2s = np.zeros(ntotal)
        tau2s = np.zeros(ntotal)

        theta2s[0] = theta2
        tau2s[0] = tau2
```

```
[14]: for i in range(1, ntotal):
        mu = - (tau2 * y2_sum + 220/(2 * 100**2)) / (7 * tau2 + 1/(100**2))
        sigma2 = 1 / (7 * tau2 + 1/100**2)
        theta2 = np.random.normal(mu, sigma2)

        var_ga = ((low-theta2) ** 2).sum()
        tau2 = np.random.gamma(shape=7/2+0.01, scale=1/(var_ga/2+4))
        theta2s[i] = theta2
        tau2s[i] = tau2
```

(b) Find sample differences $\theta_1 - \theta_2$

```
[15]: # sample differences
        theta1s[burn:] - theta2s[burn:]
```

```
[15]: array([ -943.98558989, -18917.61517935, -6459.13272254, ...,
        -7902.84099889, -15112.87964215,  8479.85905783])
```

```
[16]: # proportion
```

```
((theta1s[burn:] - theta2s[burn:])>0).sum() / len(theta1s[burn:] -  
→theta2s[burn:])
```

Proportion of positive differences approximates posterior probability of hypothesis $H_0 : \theta_1 > \theta_2$. What is this proportion?

```
[16]: 0.5034
```

The proportion is 0.5034.

(c) Using sample quantiles find the 95% equitailed credible set for $\theta_1 - \theta_2$. Does this set contain 0?

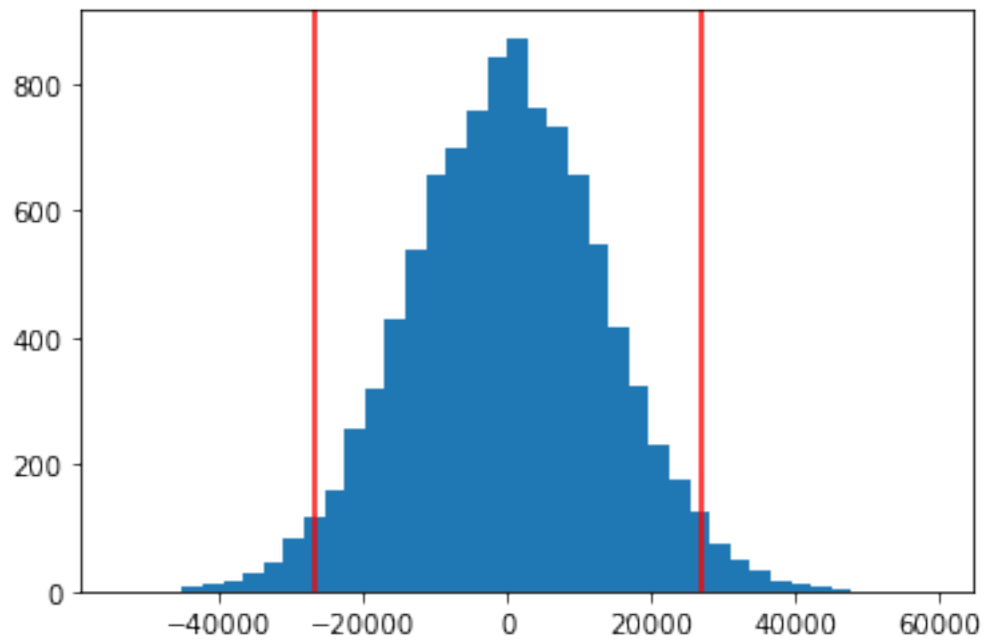
```
[17]: # create frequency  
__freq = {}  
for i in (theta1s[burn:] - theta2s[burn:]):  
    if i not in __freq:  
        __freq[i] = 1  
    else:  
        __freq[i] += 1
```

```
[18]: sum(__freq.values()) # number of observations
```

```
[18]: 10000
```

```
[19]: freq = {i: j/ 10000 for i, j in __freq.items()}  
freq_ = dict(sorted(freq.items(), key=lambda x: x[0], reverse=False))  
  
p_0025 = 0  
p_0975 = 0  
cum = 0  
for i, j in freq_.items():  
    cum += j  
    if 0.026 > cum >= 0.025:  
        p_0025 = i  
    if 0.976 > cum >= 0.975:  
        p_0975 = i
```

```
[20]: plt.hist((theta1s[burn:] - theta2s[burn:]), bins=40)  
plt.axvline(x=p_0025,color='r')  
plt.axvline(x=p_0975,color='r')  
plt.show()
```

```
[21]: print("95% equi-tailed credible set: [ {} , {} ]".format(p_0025, p_0975))
```

95% equi-tailed credible set: [-26581.915621442815 , 27161.3435941287]

Does this set contain 0?

Based on the above plot, yes. This set contains 0