Advanced Bayesian Methods - Assignment 1

통계데이터사이언스학과
박주연
2022311137

Q1

Regression problem

$X \in \mathbb{R}^{n \times p}$, $y \in \mathbb{R}^p$, $\sigma^2 \in (0, \infty)$

$y \sim N(X\beta, \sigma^2 Q)$, Q is known p.d matrix

$\beta | \sigma^2 \sim N(\mu_0, \sigma^2 k_0^{-1})$

$\sigma^2 \sim \text{Inv-}\chi^2(\nu_0, \sigma_0^2)$

(a) Derive $p(\beta | \sigma^2, y)$

Since Q is $n \times n$ positive definite matrix, we can write Q as $Q = VV'$ for some nonsingular matrix V. And the above regression can expressed as

$$y = X\beta + \varepsilon, \quad \varepsilon \sim N_n(0_n, \sigma^2 Q)$$

Multiply both sides of the regression model by V^{-1}

$$y_* = X_* \beta + \varepsilon_*, \quad \varepsilon_* \sim N(0, \sigma^2 I_n)$$

$\because \text{Var}(\varepsilon_*) = \text{Var}(V^{-1}\varepsilon) = V^{-1}\text{Var}(\varepsilon)V^{-1'}$
$= \sigma^2 V^{-1}(VV')V^{-1'}$
$= \sigma^2$

where $y_* = V^{-1}y$, $X_* = V^{-1}X$, $\varepsilon_* = V^{-1}\varepsilon$

Hence, the transformed model is identical to ordinary linear regression.

- ordinary linear regression

$y | \beta, \sigma^2 \sim N(X\beta, \sigma^2 I_n)$

$\beta | \sigma^2 \sim N(\mu_0, \sigma^2 k_0^{-1})$, $\sigma^2 \sim \text{Inv-}\chi^2(\nu_0, \sigma_0^2)$

$$\Rightarrow \begin{cases} \beta | \sigma^2, y \sim N(\tilde{\beta}, \sigma^2(X'X + k_0)^{-1}), & \tilde{\beta} = (X'X + k_0)^{-1}(X'y + k_0\mu_0) \\ \sigma^2 | y \sim \text{Inv-}\chi^2\left(\nu_0 + n, \dfrac{\nu_0\sigma_0^2 + SSE_B}{\nu_0 + n}\right), & SSE_B = (y - X\mu_0)'[I_n - X(X'X + k_0)^{-1}X'](y - X\mu_0) \end{cases}$$

- Transformed linear regression

$y_* | \beta, \sigma^2 \sim N(X_*\beta, \sigma^2 I_n)$

$\beta | \sigma^2 \sim N(\mu_0, \sigma^2 k_0^{-1})$, $\sigma^2 \sim \text{Inv-}\chi^2(\nu_0, \sigma_0^2)$

$$\Rightarrow \beta | \sigma^2, y_* \sim N(\tilde{\beta}, \sigma^2(X_*'X_* + k_0)^{-1}), \quad \tilde{\beta} = (X_*'X_* + k_0)^{-1}(X_*'y_* + k_0\mu_0)$$

$$\sigma^2(X_*'X_* + k_0)^{-1} = \sigma^2(X'\underbrace{V^{-1'}V^{-1}}_{Q^{-1}}X + k_0)^{-1}$$
$$= \sigma^2(X'Q^{-1}X + k_0)^{-1}$$

$$(X_*'X_* + k_0)^{-1}(X_*'y_* + k_0\mu_0) = (X'\underbrace{V^{-1'}V^{-1}}_{Q^{-1}}X + k_0)^{-1}(X'\underbrace{V^{-1'}V^{-1}}_{Q^{-1}}y + k_0\mu_0)$$
$$= (X'Q^{-1}X + k_0)^{-1}(X'Q^{-1}y + k_0\mu_0)$$

$$\Rightarrow \sigma^2 | y_* \sim \text{Inv-}\chi^2\left(\nu_0 + n, \frac{\nu_0\sigma_0^2 + SSE_B}{\nu_0 + n}\right)$$

$$SSE_B = (y_* - X_*\mu_0)'[I_n - X_*(X_*'X_* + k_0)^{-1}X_*'](y_* - X_*\mu_0)$$
$$= [V^{-1}(y - X\mu_0)]'[I_n - V^{-1}X(X'\underbrace{V^{-1'}V^{-1}}_{Q^{-1}}X + k_0)^{-1}X'V^{-1'}][V^{-1}(y - X\mu_0)]$$
$$= (y - X\mu_0)'V^{-1'}[I_n - V^{-1}X(X'Q^{-1}X + k_0)^{-1}X'V^{-1'}][V^{-1}(y - X\mu_0)]$$
$$= (y - X\mu_0)'[\underbrace{V^{-1'}V^{-1}}_{Q^{-1}} - \underbrace{V^{-1'}V^{-1}}_{Q^{-1}}X(X'Q^{-1}X + k_0)^{-1}X'\underbrace{V^{-1'}V^{-1}}_{Q^{-1}}](y - X\mu_0)$$
$$= (y - X\mu_0)'Q^{-1}[I_n - X(X'Q^{-1}X + k_0)^{-1}X'Q^{-1}](y - X\mu_0)$$

Therefore

$$\beta \mid \sigma^2, y \sim N(\tilde{\beta}, \sigma^2(X'Q^{-1}X + k_0)^{-1}), \quad \tilde{\beta} = (X'Q^{-1}X + k_0)^{-1}(X'Q^{-1}y + k_0\mu_0)$$

and

$$\sigma^2 \mid y \sim \text{Inv-}\chi^2\left(\nu_0 + n, \frac{\nu_0\sigma_0^2 + SSE_B}{\nu_0 + n}\right), \quad SSE_B = (y - X\mu_0)'Q^{-1}[I_n - X(X'Q^{-1}X + k_0)^{-1}X'Q^{-1}](y - X\mu_0)$$

(b) Derive $p(\sigma^2 \mid \beta, y)$

$$p(\sigma^2 \mid \beta, y) \propto p(\sigma^2 \mid \beta)\, p(y \mid \beta, \sigma^2)$$

$$= p(\sigma^2)\, p(y \mid \beta, \sigma^2) \qquad (\because p(\beta, \sigma^2) = p(\beta)p(\sigma^2))$$

$\sigma^2 \sim \text{Inv-}\chi^2(\nu_0, \sigma_0^2)$ $\qquad$ $y \mid \beta, \sigma^2 \sim N(X\beta, \sigma^2 Q)$

$$\propto (\sigma^2)^{-\nu_0/2-1} \exp\left(-\frac{\nu_0\sigma_0^2}{2\sigma^2}\right) \times (\sigma^2)^{-n/2} \exp\left(-\frac{1}{2\sigma^2}(y - X\beta)'Q^{-1}(y - X\beta)\right)$$

$$= (\sigma^2)^{-(\nu_0+n)/2-1} \exp\left(-\frac{1}{2\sigma^2}(\nu_0\sigma_0^2 + (y - X\beta)'Q^{-1}(y - X\beta))\right)$$

Shape parameter $\nu_0 + n$

Scale parameter : $\frac{1}{\nu_0 + n}(\nu_0\sigma_0^2 + (y - X\beta)'Q^{-1}(y - X\beta))$

$$\therefore \sigma^2 \mid \beta, y \sim \text{Inv-}\chi^2\left(\nu_0 + n, \frac{1}{\nu_0 + n}(\nu_0\sigma_0^2 + (y - X\beta)'Q^{-1}(y - X\beta))\right)$$

(c) Derive $p(\beta \mid y)$

From Q1-(a), we found the distribution of $\beta \mid \sigma^2, y$ and $\sigma^2 \mid y$. And the marginal posterior $p(\beta \mid y)$, obtained by integral out σ^2 in the joint posterior, is multivariate t.

(i.e. $z \mid u \sim N(\mu, uI)$ and $u \sim \text{Inv-}\chi^2(\nu, \tau^2)$ then $z \sim t_\nu(\mu, \tau^2 I)$)

We can apply it to derive $p(\beta \mid y)$.

$$\beta \mid \sigma^2, y \sim N(\tilde{\beta}, \sigma^2(X'Q^{-1}X + k_0)^{-1}), \quad \tilde{\beta} = (X'Q^{-1}X + k_0)^{-1}(X'Q^{-1}y + k_0\mu_0)$$

$$\sigma^2 \mid y \sim \text{Inv-}\chi^2\left(\nu_0 + n, \frac{\nu_0\sigma_0^2 + SSE_B}{\nu_0 + n}\right), \quad SSE_B = (y - X\mu_0)'Q^{-1}[I_n - X(X'Q^{-1}X + k_0)^{-1}X'Q^{-1}](y - X\mu_0)$$

$$\Rightarrow \beta \mid y \sim t_{\nu_0+n}\left(\tilde{\beta}, \frac{1}{\nu_0 + n}(\nu_0\sigma_0^2 + SSE_B)(X'Q^{-1}X + k_0)^{-1}\right)$$

(d) Derive $p(\beta_1 \mid y)$

It's know that the element of multivariate t distribution is univariate t distribution.

Suppose that

$X = [X_1\ X_2 \cdots X_p]' \sim$ multivariate $t_\nu(\mu, \Sigma)$

the ith element of $\mu = (\mu)_{(i)}$

the (i,j)th element of $\Sigma = (\Sigma)_{(i,j)}$

Then

$$X_i \sim t_\nu((\mu)_i, (\Sigma)_{(i,j)})$$

Therefore $\beta_1 \mid y$ also follows t distribution.

$$\beta_1 \mid y \sim t_{\nu_0+n}\left((\tilde{\beta})_{(1)}, \sqrt{\left(\frac{1}{\nu_0+n}(\nu_0\sigma_0^2 + SSE_B)(X'Q^{-1}X + k_0)^{-1}\right)_{(1,1)}}\right)$$

(e) Derive $p(\sigma^2|y)$

Since the detailed derivation was presented in Q1-(a), I will provide a brief idea and result here.

Let

$Q = VV'$ ($\because$ Q : positive definite matrix)

$y_* = V^{-1}y$

$X_* = V^{-1}X$

$\varepsilon_* = V^{-1}\varepsilon$

Then

$y_* \sim N(X_*\beta, \sigma_2 I_n)$: same as ordinary linear regression

The posterior distribution of σ^2 is

$$\sigma^2|y_* \sim Inv\text{-}\chi^2\left(\nu_0+n, \frac{\nu_0\sigma_0^2+SSE_B}{\nu_0+n}\right)$$

$$SSE_B = (y_*-X_*\mu_0)'[I_n - X_*(X_*'X_*+K_0)^{-1}X_*'](y_*-X_*\mu_0)$$

$$= (y-X\mu_0)'Q^{-1}[I_n - X(X'Q^{-1}X+K_0)^{-1}X'Q^{-1}](y-X\mu_0)$$

$$\therefore \sigma^2|y \sim Inv\text{-}\chi^2\left(\nu_0+n, \frac{1}{\nu_0+n}(\nu_0\sigma_0^2+SSE_B)\right)$$

where $SSE_B = (y-X\mu_0)'Q^{-1}[I_n - X(X'Q^{-1}X+K_0)^{-1}X'Q^{-1}](y-X\mu_0)$

(f) Draw independent sample of (β,σ^2) from joint posterior

- Sampling from posterior $p(\beta,\sigma^2|y)$

  Step 1. Draw σ^2 from $p(\sigma^2|y)$

  Step 2. Draw β from $p(\beta|\sigma^2,y)$ conditional on σ^2

- In more detail...

  Step 1

  By definition of scaled inverse χ^2 distribution, we can draw samples using random chi-square sample.

  1. Draw $w \sim \chi^2(\nu)$
  2. Transformation

  $$\sigma^2 = \frac{\nu\tau^2}{w} \sim Inv\text{-}\chi^2(\nu,\tau^2)$$

  Step 2

  We can make random multivariate sample using univariate random standard normal samples.

  1. Find matrix A s.t. $Q = AA'$ (by cholesky decomposition)
  2. Draw $z_i \overset{iid}{\sim} N(0,1)$, $i = 1, 2, \cdots, p$
  3. $Z = [z_1, \cdots, z_p] \in \mathbb{R}^p \sim MVN(0_p, I_p)$ $\because z_i$ is iid sample.
  4. Transformation

  $$y = AZ + \underset{\in\mathbb{R}^p}{\underline{\mu}} \sim MVN(\mu, AA')$$ ($\because$ Linear combination of gaussian also gaussian)

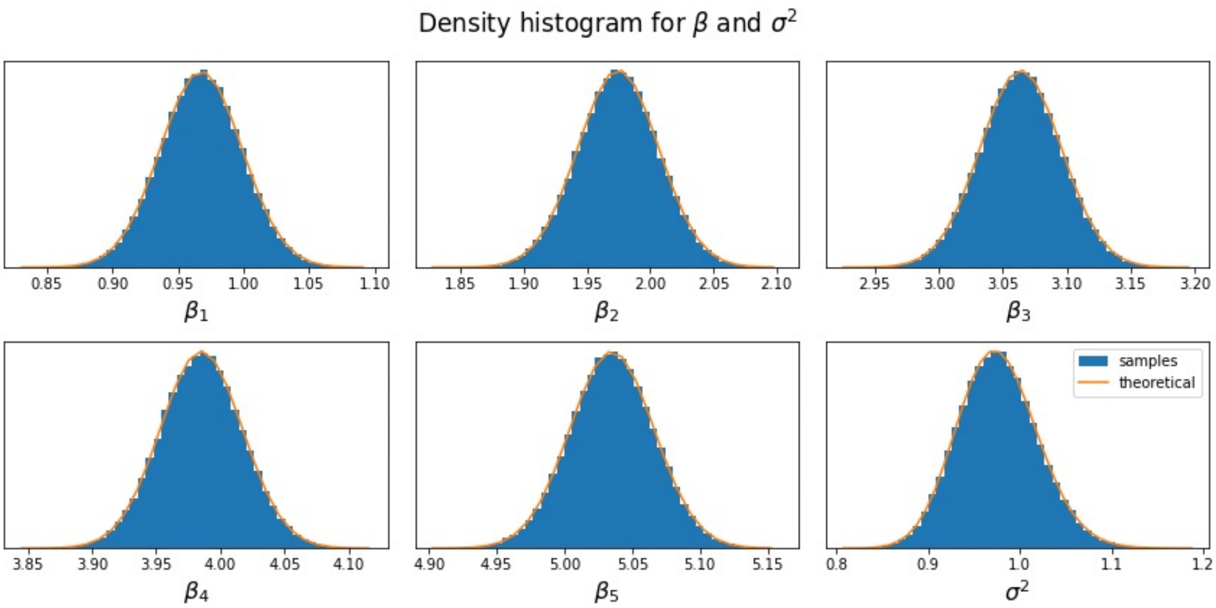
Thus, we can get independent samples (β,σ^2) from joint posterior dist using univariate standard normal dist and chi-square dist.

(g)

Let's draw (β, σ^2) samples using the idea described in Q1-(f).

The marginal curves overlay on the histograms of β and σ^2.

For the marginal curves, I use the t distribution and inverse chi-square distribution.



* The code is attached in Appendix

Q2

(a)

likelihood $\quad y|\beta,\Sigma_y \sim N(X\beta, \Sigma_y)$

prior $\quad p(\beta,\Sigma_y) = \underbrace{p(\beta|\Sigma_y)}_{\propto 1}\, p(\Sigma_y) \propto p(\Sigma_y)$

posterior $\quad p(\beta,\Sigma_y|y) = p(\beta|\Sigma_y, y)\, p(\Sigma_y|y)$

$$\propto \underbrace{p(y|\beta,\Sigma_y)}_{N(X\beta,\Sigma_y)}\underbrace{p(\beta,\Sigma_y)}_{\propto p(\Sigma_y)}$$

$$\propto p(\Sigma_y)\underbrace{|\Sigma_y|^{-1/2}\exp(-\tfrac{1}{2}\overbrace{(y-X\beta)'\Sigma_y^{-1}(y-X\beta)}^{=A})}_{\text{from } N}$$

$$A = (y-X\beta)'\Sigma_y^{-1}(y-X\beta) = (y-X\hat\beta + X\hat\beta - X\beta)'\Sigma_y^{-1}(y-X\hat\beta + X\hat\beta - X\beta)$$

$$= (y-X\hat\beta)'\Sigma_y^{-1}(y-X\hat\beta)$$

$$+ (X\hat\beta - X\beta)'\Sigma_y^{-1}(X\hat\beta - X\beta)$$

$$+ \underbrace{(y-X\hat\beta)'\Sigma_y^{-1}(X\hat\beta - X\beta) + (X\hat\beta - X\beta)'\Sigma_y^{-1}(X\hat\beta - X\beta)}_{=0}$$

$$= (y-X\hat\beta)'\Sigma_y^{-1}(y-X\hat\beta) + (\beta-\hat\beta)'(X'\Sigma_y^{-1}X)(\beta-\hat\beta)$$

$$\therefore\ p(\beta,\Sigma_y|y) \propto p(\Sigma_y)|\Sigma_y|^{-1/2}\exp(\tfrac{-1}{2}(y-X\hat\beta)'\Sigma_y^{-1}(y-X\hat\beta)) \times \exp(-\tfrac{1}{2}(\beta-\hat\beta)'(X'\Sigma_y^{-1}X)(\beta-\hat\beta))$$

↳ Depend on β (kernel of N)

$\therefore\ \beta \sim N(\hat\beta, (X'\Sigma_y^{-1}X)^{-1})$

$$p(\Sigma_y|y) = \int p(\beta,\Sigma_y|y)\,d\beta$$

$$\propto p(\Sigma_y)|\Sigma_y|^{-1/2}\exp(\tfrac{-1}{2}(y-X\hat\beta)'\Sigma_y^{-1}(y-X\hat\beta)) \times |X'\Sigma_y^{-1}X|^{-1/2}$$

$$\because \underbrace{\int \exp(-\tfrac{1}{2}(\beta-\hat\beta)'(X'\Sigma_y X)(\beta-\hat\beta))\,d\beta}_{\text{kernel of normal dist.}} \propto |(X'\Sigma_y^{-1}X)^{-1}|^{1/2}$$

$$\therefore\ p(\Sigma_y|y) \propto p(\Sigma_y)|\Sigma_y|^{-1/2}|X'\Sigma_y^{-1}X|^{-1/2}\exp(-\tfrac{1}{2}(y-X\hat\beta)'\Sigma_y^{-1}(y-X\hat\beta))$$

(b)

Suppose

$$\underset{n\times 1}{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} y_{1,1} \\ \vdots \\ y_{1,n_1} \\ y_{2,1} \\ \vdots \\ y_{2,n_2} \\ \vdots \\ y_{I,n_I} \end{bmatrix}, \qquad \underset{n\times p}{X} = \begin{bmatrix} x_{1,1} \\ \vdots \\ x_{1,n_1} \\ x_{2,1} \\ \vdots \\ x_{2,n_2} \\ \vdots \\ x_{I,n_I} \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_I \end{bmatrix}$$

1st group's y

where $\underset{n_j\times p}{X_j} = \begin{bmatrix} x_{j,1} \\ \vdots \\ x_{j,n_j} \end{bmatrix}$, $\beta \in \mathbb{R}^p$

We can simplify the expression of $p(\Sigma_y|y)$ as

$$p(\Sigma_y|y) \propto p(\Sigma_y)|\Sigma_y|^{-1/2}|X'\Sigma_y X|^{-1/2}\exp\{-\tfrac{1}{2}(y-X\hat\beta)'\Sigma_y^{-1}(y-X\hat\beta)\}$$

$$\propto \prod_{j=1}^{I}\sigma_j^{-2} \times \left(\prod_{j=1}^{I}(\sigma_j^2)^{n_j}\right)^{-1/2}\left|\sum_{j=1}^{I}\frac{1}{\sigma_j^2}X_j'X_j\right|^{-1/2}\exp\left\{-\sum_{j=1}^{I}\sum_{i=1}^{n_j}\frac{1}{2\sigma_j^2}(y_{ij} - x_{ij}\hat\beta)^2\right\}$$

Since,

$$|\Sigma_y|^{-1/2} = ((\sigma_1^2)^{n_1} \times \cdots \times (\sigma_I^2)^{n_I})^{-1/2} \qquad (\because \Sigma_y \text{ is diagonal matrix})$$

$$= \left(\prod_{j=1}^{I}(\sigma_j^2)^{n_j}\right)^{-1/2}$$

$$\underbrace{X'\Sigma_y^{-1}X}_{p\times p} = \begin{bmatrix} X_1 \\ \vdots \\ \underset{n_I\times p}{X_I} \end{bmatrix}' \begin{bmatrix} \frac{1}{\sigma_1^2}I_{n_1} & & \\ & \ddots & \\ & & \underset{p\times p}{\frac{1}{\sigma_I^2}I_{n_I}} \end{bmatrix} \begin{bmatrix} X_1 \\ \vdots \\ \underset{n_I\times p}{X_I} \end{bmatrix} = \Sigma_{j=1}^{I} \frac{1}{\sigma_j^2} X_j' X.$$

$$\Rightarrow |X'\Sigma_y^{-1}X|^{-1/2} = \left| \Sigma_{j=1}^{I} \frac{1}{\sigma_j^2} X_j' X_j \right|^{-1/2}$$

$$-\frac{1}{2}(y-X\hat\beta)'\Sigma_y^{-1}(y-X\hat\beta) = -\frac{1}{2}(y-X\hat\beta)'\begin{pmatrix} 1/\sigma_1^2 & & 0 \\ & \ddots & \\ 0 & & 1/\sigma_I^2 \end{pmatrix}(y-X\hat\beta)$$

$$= -\Sigma_{j=1}^{I} \frac{1}{2\sigma_j^2}(y_j - X_j\hat\beta)^2$$

$$= -\Sigma_{j=1}^{I}\Sigma_{i=1}^{n_j} \frac{1}{2\sigma_j^2}(y_{ij} - x_{ij}\hat\beta)^2$$

(c)

We can use collasped gibbs sampling for sampling algorithm of $p(\beta, \sigma_1^2, \dots, \sigma_I^2 | y)$.

At $(t+1)$th iteration,

<u>Step 1</u> Sampling $\sigma_1^2, \dots, \sigma_I^2$ (collasped gibbs)

1. Draw $\sigma_1^{2(t+1)}$ from $p(\sigma_1^2 | \sigma_2^{2(t)}, \dots, \sigma_I^{2(t)}, y)$
2. Draw $\sigma_2^{2(t+1)}$ from $p(\sigma_2^2 | \sigma_1^{2(t+1)}, \sigma_3^{2(t)}, \dots, \sigma_I^{2(t)}, y)$

$\vdots$

I. Draw $\sigma_I^{2(t+1)}$ from $p(\sigma_I^2 | \sigma_1^{2(t+1)}, \dots, \sigma_{I-1}^{2(t+1)}, y)$

<u>Step 2</u> Sampling β from $p(\beta | \sigma_1^2, \dots, \sigma_I^2, y)$

1. Draw $y^{(t+1)}$ from $p(\beta | \sigma_1^2, \dots, \sigma_I^2, y)$ plugging $\sigma_1^{2(t+1)}, \dots, \sigma_I^{2(t+1)}$

Then $(y^{(t+1)}, \sigma_1^{2(t+1)}, \dots, \sigma_I^{2(t+1)})$ is posterior sample.

(d)

$$p(\Sigma_y | \beta, y) \propto p(y|\beta, \Sigma_y)\, p(\Sigma_y|\beta)$$

$$\propto p(y|\beta,\Sigma_y)\, p(\beta, \Sigma_y)$$

$$\propto \underbrace{p(y|\beta,\Sigma_y)}_{\sim N(X\beta, \Sigma_y)}\, \underbrace{p(\Sigma_y)}_{\propto \prod_{j=1}^{I}\sigma_j^{2}}$$

$$\propto \prod_{j=1}^{I}(\sigma_j^2)^{-n_j/2} \exp\Big(\underbrace{-\tfrac{1}{2}(y-X\beta)'\Sigma_y^{-1}(y-X\beta)}\Big) \times \prod_{j=1}^{I}\sigma_j^{-2}$$

$$= -\frac{1}{2}(y-X\beta)'\begin{bmatrix} \frac{1}{\sigma_1^2} & & \\ & \ddots & \\ & & \frac{1}{\sigma_I^2} \end{bmatrix}(y-X\beta) = -\frac{1}{2}\Sigma_{j=1}^{I}\frac{1}{\sigma_j^2}(y_j - X_j\beta)^2$$

$$= -\Sigma_{j=1}^{I}\Sigma_{i=1}^{n_j}\frac{1}{2\sigma_j^2}(y_{ij} - x_{ij}\beta)^2$$

$$\propto \prod_{j=1}^{I}(\sigma_j^2)^{-n_j/2-1} \times \exp\left(-\Sigma_{j=1}^{I}\Sigma_{i=1}^{n_j}\frac{1}{2\sigma_j^2}(y_{ij}-x_{ij}\beta)^2\right)$$

$$= \prod_{j=1}^{I}\left\{(\sigma_j^2)^{-n_j/2-1}\exp\left(-\frac{1}{2\sigma_j^2}\Sigma_{i=1}^{n_j}(y_{ij}-x_{ij}\beta)^2\right)\right\}$$

$\therefore \Sigma_y | y \sim \prod_{j=1}^{I} \text{Inv-}\chi^2(\nu_n, \tau_n^2)$ where $\nu_n = n_j$ and $\tau_n^2 = \frac{1}{n_j}\Sigma_{i=1}^{n_j}(y_{ij}-x_{ij}\beta)^2)$

[Appendix]

Q1-(g)

```
### library
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import warnings

from scipy.stats import chi2, norm
from numpy.linalg import inv
from scipy.special import loggamma

warnings.filterwarnings('ignore')
plt.rcParams['figure.facecolor'] = 'white'
np.set_printoptions(suppress=True)

## pdf of scaled noncentral t distribution
def scaled_noncentral_t(x, mu, sigma, nu):
    prob = np.exp(loggamma((nu + 1) / 2) - loggamma(nu / 2) - np.log(np.sqrt(nu * np.pi) * sigma)) * \
        (1 + 1 / nu * ((x - mu) / sigma)**2)**(-(nu + 1) / 2)
    return prob

## pdf of scaled inverse chi2 distribution
def scaled_inverse_chi2(x, nu, tau2):
    log = (nu / 2) * np.log(nu * tau2 / 2) - loggamma(nu / 2) - (nu / 2 +1) * np.log(x) - (nu * tau2 / (2 * x))
    return np.exp(log)

## draw sigma2 sample given y
def Draw_sigma2_sample(X, hyper, iter):
    """ Draw sigma2 from p(sigma2 | y) """

    # parameters
    n, p = X.shape
    df = hyper['nu0'] + n                                 # df of Scaled-Inv-chi2
    SS_inv = inv(np.matmul(X.T, X) + hyper['K0'])
    inside = np.eye(n) - np.matmul(X, np.matmul(SS_inv, X.T))
    SSE = np.matmul(y.T, np.matmul(inside, y))
    tau2 = (hyper['nu0'] * hyper['sigma2_0'] + SSE) / df  # scaling paramter of Scaled-Inv-chi2

    # for multivariate t
```

```python
    cov = (hyper['nu0'] * hyper['sigma2_0'] + SSE) / df * SS_inv

    # sampling sigma2
    sigma2_samples = np.zeros(iter)
    for i in range(iter):
        # 1. Draw chi2 random sample
        w = chi2.rvs(df, size=1)
        # 2. Scaled and inverse
        sigma2 = df * tau2 / w  # scaled inverse Chi2 r.v.
        sigma2_samples[i] = sigma2

    return sigma2_samples, df, tau2, cov

## draw beta conditional on sigma2, y
def Draw_beta_sample(X, y, sigma2):
    """ Draw beta from p(beta | sigma2, y) conditional on sigma2 """

    # parameters
    n, p = X.shape
    SS = np.matmul(X.T, X) + hyper['K0']
    beta_hat = np.matmul(inv(SS), np.matmul(X.T, y))
    iter = len(sigma2)

    # sampling beta conditional on sigma2
    beta_samples = np.zeros(shape=(iter, p))
    for i in range(iter):
        beta_cov = sigma2[i] * inv(np.matmul(X.T, X) + hyper['K0'])

        ## 1. Cholesky decomposition
        A = np.linalg.cholesky(beta_cov)
        # Sanity check; np.sum(beta_cov.round(5) != np.matmul(A, A.T).round(5))

        ## 2. Draw standard normal random samples
        Z = np.random.normal(0, 1, p)
        beta_samples[i] = np.matmul(A, Z) + beta_hat

    return beta_samples, beta_hat


## 0. Initial setting
n = 1000; p = 5
true = dict({
    'beta': [1, 2, 3, 4, 5],
    'sigma': 1
})
```

```python
# Conjugate prior that provide weak information
hyper = dict({
    'm0':0,
    'K0':np.eye(p) * 0.0001,
    'nu0': 0.0001,
    'sigma2_0':0.0001
})

## 1. Draw X from standard normal distribution
X = np.random.normal(0, 1, size=(n, p))

## 2. Generate y
# 1) Draw z_i from z ~ N(0, 1) (i = 1, ..., n) and Z = [z_1, ..., z_n] ~ MVN(0, I)
Z = np.random.normal(0, 1, n)

# 2) Transformation: y = Z + mu => y ~ MVN(mu, I)
y_mean = np.matmul(X, true['beta'])
y = Z + y_mean

## 3. Draw sigma2
sigma2_samples, df, tau2, cov = Draw_sigma2_sample(X, hyper, 10**5)

## 4. Draw beta conditional on sigma2
beta_samples, beta_hat = Draw_beta_sample(X, y, sigma2_samples)

## 5. Plotting
plt.figure(figsize=(6*2, 3*2))

# 1) for beta
for i in range(X.shape[1]):
    plt.subplot(2, 3, i+1)
    plt.hist(beta_samples[:, i], density=True, bins=45, label='samples')

    # Marginal posterior of beta ~ multivariate t; t_df, t_cov, beta_hat
    grid = np.arange(min(beta_samples[:, i]), max(beta_samples[:, i]), 0.01)
    theoretical = scaled_noncentral_t(x=grid, mu=beta_hat[i], sigma =
np.sqrt(cov.diagonal()[i]), nu = df)
    plt.plot(grid, theoretical, label='theoretical')
    plt.xlabel(f'$\\beta_{i+1}$', fontsize=16)
    plt.yticks(())

# 2) for sigma2
plt.subplot(2, 3, 6)
plt.hist(sigma2_samples, density=True, bins=45, label='samples')
```

```
# Marginal posterior of sigma2 ~ scaled-inv-chi2; nu=df, tau2=tau2
grid = np.arange(min(sigma2_samples), max(sigma2_samples), 0.01)
theoretical = scaled_inverse_chi2(x=grid, nu=df, tau2=tau2)
plt.plot(grid, theoretical, label='theoretical')
plt.xlabel(f'$\\sigma^2$', fontsize=16)
plt.yticks(())

plt.legend()
plt.suptitle(f'Density histogram for $ \\beta$ and $ \\sigma^2$', fontsize=17)
plt.tight_layout()
plt.savefig('./hw1')
plt.show()
```