Assignment 4

In this assignment you are required to write 3 functions:

Q1: A function that simulates from the posterior distribution for Weibull observations.

Q2: A function that simulates a 3D Ising model in a magnetic field.

Q3: A function that conducts Bayesian analysis in regression.

The template for the functions in given in the file “Assignment_4_123456789.R”. You should change (only) the number “123456789” in the file name to your ID number. Do not make any other changes in the name of the file.

In each one of the functions that are in the file you are required to replace the place-holder “return(NA)” by the body of the code of the function that you write. You must not change the name of the function nor the arguments of the function. Do not add any comments to the code and do not add any code outside the body of the functions. Unless otherwise stated, your code cannot assume availability of or call any package besides the standard packages in the base distribution of R that are uploaded automatically when R session is opened.

Question 1 (Q1):

The Weibull distribution is popular in survival analysis. The density function of a 2-parameter Weibull observation is:

$$f_X(x|\alpha,\lambda) = \alpha\lambda^{\alpha}x^{\alpha-1}e^{-\lambda x^{\alpha}}, \qquad x > 0.$$

The two parameters, α and λ, are nonnegative. Assume the prior distribution

$$f(\alpha,\lambda) = f_1(\alpha)f_2(\lambda),$$

where f_1 and f_2 are two density functions with positive supports.

Given independent observations $X_1, X_2, \dots, X_n$ from the Weibull distribution, and given the two prior densities f_1 and f_2, you are required to write a function that applies the Metropolis-Hastings algorithm in order to produce a sequence of pairs (α_i, λ_i) with the posterior distribution as the stationary distribution of the sequence.

Given the current pair $(\alpha_{i-1}, \lambda_{i-1})$, the proposal for the new pair should be independent Exponential random variables with rates $1/\alpha_{i-1}$ and $1/\lambda_{i-1}$, respectively. (I.e., their expectations are α_{i-1} and λ_{i-1}, respectively). The acceptance probability of the proposal is computed using both the proposal density, the likelihood of the observations, and the prior density of the parameters.

The arguments of the function are:

x = a numeric vector of the observations.

f.alpha = a function. Computes the prior density of α.

f.lambda = a function. Computes the prior density of λ.

start = a numeric vector of length 2. The initial values (α_0, λ_0).

N = The number of iterations of the Metropolis-Hastings algorithm.

The output of the function should be a matrix with $(N + 1)$ rows and 2 columns that contains the sequence of simulated pairs, with the first row occupied by the initial values.

Question 2 (Q2):

The Ising model is a popular model in statistical mechanics. It is also used in in image reconstruction. Algorithm 4.31 implements the Ising model in the 2D setting and Algorithm 4.32 applies this model for denoising in the context of image processing. In the current assignment you are required to extend Algorithm 4.32 to 3D setting, which is relevant for example in video processing (where time is the extra dimension).

The description of the Ising model in the 2D setting is given in the book. It involves, for the grid points $i = (i_1, i_2)$, $1 \leq i_1, i_2 \leq L$, the random states $X_i \in \{-1, +1\}$, together with the observations Y_i, where $Y_i \sim N(X_i, \sigma^2)$. This model is illustrated in Exercise E4.8.

The main difference in the 3D setting is the fact that grid points $i = (i_1, i_2, i_3)$ are characterized by 2 integers and that the neighborhood of a grid point ($j \sim i$) involves the 6 points with the property that $|j_1 - i_1| + |j_2 - i_2| + |j_3 - i_3| = 1$.

In this question you are asked to write a function called “a4q2” that denoises a video and generalizes the function of Exercise E4.8. The input to the function is a series of images (stored as a 3D array) and the output is the denoised sequence.

The arguments of the function are:

> M = An integer. The number of sweeps of the Gibbs sampler over the entire array (following the burn-in sweeps).
>
> beta = A numerical value. The interaction parameter.
>
> Y = An array. Your function should work when Y is a matrix and when Y is a 3D array. It should also work when the components of the “dim” attribute are not equal to each other.
>
> sigma = A numerical value. The standard deviation of the observations Y_i.
>
> burn.in = an integer. The number of burn-in sweeps, with a default value of 100.

The output of the function should be an array of the same dimension as Y with the denoised video.

Question 3 (Q3):

Consider the regression model:

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \;\; 1 \le i \le n, \;\; \epsilon_i \sim N(0, \sigma^2), \qquad i.i.d.$$

A Bayesian model may use: $\beta_0 \sim N(\mu_0, \sigma_0^2)$, $\beta_1 \sim N(\mu_1, \sigma_1^2)$, and $\frac{1}{\sigma^2} \sim Gamma(r, \lambda)$, independent.

Given observation: $(y_1, x_1), (y_2, x_2), \dots, (y_n, x_n)$. It can be shown that in the conditional distribution, given the observations:

$$\beta_0 | \{\beta_1 = b_1, \sigma^2 = v\} \sim N\left(\frac{\mu_0/\sigma_0^2 + \Sigma_{i=1}^n (y_i - b_1 x_i)/v}{1/\sigma_0^2 + n/v}, \qquad \frac{1}{1/\sigma_0^2 + n/v}\right)$$

$$\beta_1 | \{\beta_0 = b_{10}, \sigma^2 = v\} \sim N\left(\frac{\mu_1/\sigma_1^2 + \Sigma_{i=1}^n x_i (y_i - b_0)/v}{1/\sigma_1^2 + \Sigma_{i=1}^n x_i^2/v}, \qquad \frac{1}{1/\sigma_1^2 + \Sigma_{i=1}^n x_i^2/v}\right)$$

$$\left(\frac{1}{\sigma^2}\right) | \{\beta_0 = b_0, \beta_1 = b_1\} \sim Gamma(r + n/2, \qquad \lambda + \Sigma_{i=1}^n (y_i - b_0 - b_1 x_i)^2/2).$$

(Notice that the Gamma distribution in the last equation corresponds to the distribution of $1/\sigma^2$).

Write a function "a4q3" that implements the Gibbs algorithm in order to produce a sample from the posterior distribution of $(\beta_0, \beta_1, \sigma^2)$, given the observations. The output of the function should contain the paths of the entire Markov process that is produced by the algorithm, including the initiating values $\theta^{(0)} = \left(b_0^{(0)}, b_1^{(0)}, v^{(0)}\right)$. A sweep that updates the 3 values should move from left to right. Hence, when producing $\theta^{(k)} = \left(b_0^{(k)}, b_1^{(k)}, v^{(k)}\right)$ you should simulate first $b_0^{(k)}$, then $b_1^{(k)}$, and then $v^{(k)}$.

The arguments of the function are:

y = a numerical vector. The observed values of the response in the regression model.

x = a numerical vector. The observed values of the explanatory variable in the regression model..

theta = a numerical vector of length 3 that contains the initiating values $\theta^{(0)}$.

param = a numerical vector of length 6 that contains, in this order: μ_0, σ_0^2, μ_1, σ_1^2, r, and λ.

n.sweep = an integer. The total number of sweeps of the Gibbs algorithm.

The function should return a data frame of size $(\text{n.sweep} + 1) \times 3$. The first row of the data frame should be value of the argument theta.