Please describe all your work in clear terms, before implementing R code. Each question should include a description of your approach with clear indication of where I can download the associated source code. Your code should be attached to your assignment or uploaded online to a repository I can freely access.

Generalized Linear Models.
In R load the package (`survival`) and consider the analysis of the data-set (`infert`). Ignoring dependence due to matching, consider a Bayesian analysis for a probit model relating case status to: age, parity, education, spontaneous and induced.

Assume case status y_i has density $y_i \sim Bern(p_i)$, $p_i = \Phi(X_i'\beta)$, where $\Phi(w)$ is the cdf of a standard Normal distribution evaluated at w. Consider a unit information prior $\beta \sim N(0, n\,(X'X)^{-1})$, where n is the sample size. We are interested in $p(\beta \mid Y)$.

(1) Describe and implement a Metropolis-Hastings algorithm designed to obtain a MC with stationary distribution $p(\beta \mid Y)$.

(2) Describe and implement a data augmented (DA-MCMC) strategy targeting $p(\beta \mid Y)$.

(3) Describe and implement a parameter expanded - data augmentation (PX-DA MCMC) algorithm targeting $p(\beta \mid Y)$.

(4) Assess mixing and convergence of the chains induced by the competing transition schemes implemented in 1,2 and 3. Comment on potential trade-offs involving: coding complexity, storage and cpu time.

Consider now a logit link, by assuming $\log\left(\frac{p_i}{1-p_i}\right) = X_i'\beta$, with $\beta \sim N(0, n\,(X'X)^{-1})$, as before.

(5) Describe and implement a random walk Metropolis-Hastings algorithm designed to obtain a MC with stationary distribution $p(\beta \mid Y)$.

(6) Describe and implement a Langevin-Hastings algorithm designed to obtain a MC with stationary distribution $p(\beta \mid Y)$.

(7) Describe and implement an adaptive Metropolis-Hastings algorithm designed to obtain a MC with stationary distribution $p(\beta \mid Y)$.

(8) Assess mixing and convergence of the chains induced by the competing transition schemes implemented in 5,6 and 7. Comment on potential trade-offs involving: coding complexity, storage and cpu time.