Week 6 Problems

1. Load the `macro` dataset from the `Zelig` library. Implement a Bayesian linear regression of `unem` on `gdp` and `trade` using a Gibbs sampler with the following priors:

$$
\begin{aligned}
\boldsymbol{\beta} &\sim N(\mathbf{m}, \mathbf{V}) \\
\sigma^2 &\sim \text{Inv-Gamma}\left(\frac{\nu}{2}, \frac{\delta}{2}\right)
\end{aligned}
$$

   Assume $\boldsymbol{\beta}$ and σ^2 are a priori independent. Use diffuse priors (let $\nu = 1$ and $\delta = 1$).

   The full conditionals for the Gibbs sampler are:

$$
\begin{aligned}
\boldsymbol{\beta}|\sigma^2, \mathbf{y} &\sim N(\mathbf{m}^*, \mathbf{V}^*) \\
\sigma^2|\boldsymbol{\beta}, \mathbf{y} &\sim \text{Inv-Gamma}\left(\frac{n+\nu}{2}, \frac{(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})+\delta}{2}\right)
\end{aligned}
$$

   where

$$
\begin{aligned}
\mathbf{V}^* &= (\mathbf{X}'(\boldsymbol{\sigma^2}\mathbf{I})^{-\mathbf{1}}\mathbf{X} + \mathbf{V}^{-\mathbf{1}})^{-\mathbf{1}} \\
\mathbf{m}^* &= \mathbf{V}^*(\mathbf{X}'(\boldsymbol{\sigma^2}\mathbf{I})^{-\mathbf{1}}\mathbf{y} + \mathbf{V}^{-\mathbf{1}}\mathbf{m})
\end{aligned}
$$

   Check for convergence both visually and statistically.

2. Recall that a probit regression model can be expressed with a latent variable formulation in the following way:

$$
\begin{aligned}
y_i &= \begin{cases} 1 & \text{if } y_i^* \geq 0 \\ 0 & \text{if } y_i^* < 0 \end{cases} \\
y_i^* &\sim N(\mathbf{X}_i\boldsymbol{\beta}, 1)
\end{aligned}
$$

   Suppose that $\mathbf{y}^*$ is a random variable that we want to draw along with $\boldsymbol{\beta}$ ($\mathbf{y}^*$ is unknown and all unknowns are random variables in the Bayesian setting). Use this formulation and a Gibbs sampler to implement a probit regression model of `vote` on `age` and `income` in the `turnout` dataset in `Zelig`. That is, you want to sample from

$$
\begin{aligned}
\mathbf{y}^* &\mid \boldsymbol{\beta}, \mathbf{y} \\
\boldsymbol{\beta} &\mid \mathbf{y}^*, \mathbf{y}
\end{aligned}
$$

   You may want to use the truncated normal functions `rtnorm()` in the `msm` package. Be sure to check for convergence.