MA 578: Final Exam

Benjamin Draves

12/14/2019

Exercise 1

Consider the following equivariance vector $y|\rho \sim N(0, R(\rho))$ where the variance structure is given by the matrix

$$R(\rho) = \begin{bmatrix} 1 & \rho & \dots & \rho \\ \rho & 1 & \dots & \rho \\ \vdots & \vdots & \ddots & \vdots \\ \rho & \rho & \dots & 1 \end{bmatrix} = (1-\rho)I_n + \rho 1_n 1_n^T$$

(a)

In this exercise we look to find expressions for $R(\rho)^{-1}$ and $|R(\rho)|$. First notice that $(1-\rho)I_n$ is invertible with inverse $\frac{1}{1-\rho}I_n$. Moreover, notice that $1 + (\sqrt{\rho}1_n)^T\frac{1}{1-\rho}I_n(\sqrt{\rho}1_n) = 1 + \frac{\rho n}{1-\rho} \neq 0$. Therefore, by the Sherman-Morrison formula, the inverse of $R(\rho)$ can be written as

$$\begin{aligned} R^{-1}(\rho) &= \frac{1}{1-\rho}I_n - \frac{\frac{1}{1-\rho}I_n(\rho 1_n 1_n^T)\frac{1}{1-\rho}I_n}{1+\frac{\rho n}{1-\rho}} \\ &= \frac{1}{1-\rho}I_n - \frac{\frac{\rho}{(1-\rho)^2}}{1+\frac{n\rho}{1-\rho}}1_n1_n^T \\ &= \frac{1}{1-\rho}I_n - \frac{\frac{\rho}{(1-\rho)^2}}{\frac{1-\rho+n\rho}{1-\rho}}1_n1_n^T \\ &= \frac{1}{1-\rho}\left(I_n - \frac{\rho}{1+(n-1)\rho}1_n1_n^T\right) \end{aligned}$$

To find an expression for the determinant, recall as $(1-\rho)I_n$ is invertible, we can use the Determinant Lemma to write an expression for $|R(\rho)|$ as follows.

$$\begin{aligned} |R(\rho)| &= (1 + \frac{n\rho}{1-\rho})\,|(1-\rho)I_n| \\ &= \left(\frac{1+\rho(n-1)}{1-\rho}\right)(1-\rho)^n\,|I_n| \\ &= (1-\rho)^{n-1}\,(1+\rho(n-1)) \end{aligned}$$

Recall by definition we require that $R(\rho)$ be positive definite. Therefore all of the eigenvalues of $R(\rho)$ need to be positive. Recall that the determinant of a matrix is just the product of its eigenvalues. Therefore, the determinant of $R(\rho)$ need be positive. Translating this to a lower bound on ρ we have

$$
\begin{aligned}
|R(\rho)| &> 0 \\
(1-\rho)^{n-1}(1+\rho(n-1)) &> 0 \\
(1+\rho(n-1)) &> 0 \\
\rho &> -\frac{1}{n-1}
\end{aligned}
$$

Therefore we require that $\rho \in (-1/(n-1), 1)$ and not $\rho \in (-1, 1)$. Define the parameter θ that scales ρ to the unit interval.

$$\theta = \frac{\rho - (-1/(n-1))}{1-(-1/(n-1))} = \frac{\rho + 1/(n-1))}{1+1/(n-1))} = n^{-1}(\rho(n-1)+1) \in (0,1)$$

(b)

As we wish to now parameterize this function with respect θ first notice that $\rho = \frac{n\theta-1}{n-1}$ Next recall the full likelihood can be written as

$$
\begin{aligned}
p(y|\rho) &= (2\pi)^{-n/2}|R(\rho)|^{-1/2}\exp\left\{-\frac{1}{2}y^T R(\rho)^{-1}y\right\} \\
&= (2\pi)^{-n/2}\left((1-\rho)^{n-1}(1+\rho(n-1))\right)^{-1/2}\exp\left\{-\frac{1}{2}y^T\left(\frac{1}{1-\rho}\left(I_n - \frac{\rho}{1+(n-1)\rho}1_n1_n^T\right)\right)y\right\}
\end{aligned}
$$

Simplifying $|R(\rho)|$ in terms of θ we have

$$
\begin{aligned}
|R(\rho)| &= \left((1-\rho)^{n-1}(1+\rho(n-1))\right) \\
&= \left((1-\frac{n\theta-1}{n-1})^{n-1}(1+\frac{n\theta-1}{n-1}(n-1))\right) \\
&= \left(\frac{n-1-n\theta+1}{n-1}\right)^{n-1}(1+n\theta-1) \\
&= \left(\frac{n(1-\theta)}{n-1}\right)^{n-1}(n\theta) \\
&\propto (1-\theta)^{n-1}\theta
\end{aligned}
$$

Similarly, we can simplify $y^T R^{-1}(\rho)y$ as follows.

$$\begin{aligned}
y^TR^{-1}(\rho)y &= \frac{y^Ty}{1-\rho} - \frac{\rho}{(1+(n-1)\rho)(1-\rho)}(y^T1_n)(y^T1_n)^T \\
&= \frac{1}{1-\rho}(y-1_n\bar{y}+1_n\bar{y})^T(y-1_n\bar{y}+1_n\bar{y}) - \frac{\rho}{(1+(n-1)\rho)(1-\rho)}n\bar{y}^2 \\
&= \frac{(y-\bar{y}1_n)^T(y-\bar{y}1_n)}{1-\rho} + 2\frac{(y-\bar{y}1_n)^T(1_n\bar{y})}{1-\rho} \\
&+ \frac{(1_n\bar{y})^T(1_n\bar{y})}{1-\rho} - \frac{\rho}{(1+(n-1)\rho)(1-\rho)}n\bar{y}^2
\end{aligned}$$

Next notice

$$(y-1_n\bar{y})^T(1_n\bar{y}) = (y^T1_n - \bar{y}1_n^T1_n)\bar{y} = (n\bar{y}-n\bar{y})\bar{y} = 0$$

Therefore, we can write

$$y^TR^{-1}(\rho)y = \frac{n}{1-\rho}s^2(y) + \frac{n}{1-\rho}\left(1-\frac{n\rho}{1+(n-1)\rho}\right)\bar{y}^2$$

Now rewriting these coefficients in terms of θ we have

$$\frac{n}{1-\rho} = \frac{n}{1-\frac{n\theta-1}{n-1}} = \frac{n}{\frac{n-1-n\theta+1}{n-1}} = \frac{n-1}{1-\theta}$$

Moreover, we have

$$\begin{aligned}
\frac{n}{1-\rho}\left(1-\frac{n\rho}{1+(n-1)\rho}\right) &= \frac{n}{1-\rho}\left(\frac{1+(n-1)\rho-n\rho}{1+(n-1)\rho}\right) \\
&= \frac{n}{1-\rho}\left(\frac{1-\rho}{1+(n-1)\rho}\right) \\
&= \left(\frac{n}{1+(n-1)\rho}\right) \\
&= \left(\frac{n}{1+(n\theta-1)}\right) \\
&= \frac{1}{\theta}
\end{aligned}$$

Therefore, reparameterizing $p(y|\rho)$ with respect to θ yields.

$$p(y|\theta) \propto \left[\theta(1-\theta)^{n-1}\right]^{-1/2}\exp\left\{-\frac{1}{2}\left[\frac{(n-1)s^2(y)}{1-\theta}+\frac{\bar{y}^2}{\theta}\right]\right\}$$

(c)

First we find the conjugate prior for θ. By replacing "data" with "parameters" from the expression we developed in part (b) we see the conjugate prior on θ has the shape.

$$p(\theta) \propto [\theta(1-\theta)^{\kappa}]^{-1/2} \exp\left\{-\frac{1}{2}\left[\frac{\kappa\tau^2}{1-\theta} + \frac{\mu^2}{\theta}\right]\right\}$$

where μ^2, τ^2, κ are the prior parameters. From which, the resulting posterior would have the form

$$p(\theta|y) \propto [\theta(1-\theta)^{(\kappa+n)-1}]^{-1/2} \exp\left\{-\frac{1}{2}\left[\frac{(\kappa\tau^2 + ns^2(y)) - 1}{1-\theta} + \frac{\mu^2 + \bar{y}^2}{\theta}\right]\right\}$$

From which we see the parameters are updated according to the rule $(\mu^2, \kappa, \kappa\tau^2) \to (\mu^2 + \bar{y}^2, \kappa + n, \kappa\tau^2 + (n-1)s^2(y))$.

To find Jeffrey's prior, $p(\theta) \propto I^{1/2}(\theta)$ where $I(\theta) = -\mathbb{E}\left[\frac{d^2\ell(\theta)}{d\theta^2}\right]$ where $\ell(\theta)$ is the log-likelihood. We calculate these quantities below using the fact $\mathbb{E}_{y|\theta}[\bar{y}^2] = \theta$ and $\mathbb{E}_{y|\theta}[s^2(y)] = \theta(1-\theta)$.

$$\begin{aligned}
\ell(\theta) &= -\frac{1}{2}\log(\theta) - \frac{n-1}{2}\log(1-\theta) - \frac{1}{2}\left[\frac{(n-1)s^2(y)}{1-\theta} + \frac{\bar{y}^2}{\theta}\right] \\
\ell'(\theta) &= -\frac{1}{2\theta} + \frac{n-1}{2(1-\theta)} - \frac{(n-1)s^2(y)}{(1-\theta)^2} + \frac{\bar{y}^2}{\theta^2} \\
\ell''(\theta) &= \frac{1}{2\theta^2} + \frac{n-1}{2(1-\theta)^2} - 2\frac{(n-1)s^2(y)}{(1-\theta)^3} - 2\frac{\bar{y}^2}{\theta^3} \\
-\mathbb{E}_{y|\theta}[\ell''(\theta)] &= -\frac{1}{2\theta^2} - \frac{n-1}{2(1-\theta)^2} + 2\frac{(n-1)\mathbb{E}_{y|\theta}[s^2(y)]}{(1-\theta)^3} + 2\frac{\mathbb{E}_{y|\theta}[\bar{y}^2]}{\theta^3} \\
&= -\frac{1}{2\theta^2} + \frac{n-1}{2(1-\theta)^2} - 2\frac{(n-1)(1-\theta)}{(1-\theta)^3} + 2\frac{\theta}{\theta^3} \\
&= \frac{3}{2}\left(\frac{1}{\theta^2} + \frac{(n-1)}{(1-\theta)^2}\right)
\end{aligned}$$

Therefore the square root of the information, and hence Jeffrey's prior, is given by

$$p(\theta) \propto I^{1/2}(\theta) = \left[\frac{3}{2}\left(\frac{1}{\theta^2} + \frac{(n-1)}{(1-\theta)^2}\right)\right]^{1/2} \propto \left[\frac{1}{\theta^2} + \frac{(n-1)}{(1-\theta)^2}\right]^{1/2}$$

Exercise 2

Suppose we have J subjects and we wish to regression $y_j \in \mathbb{R}^{n_j}$ onto the predictor X_j for $j = 1, 2, \ldots, J$. Moreover, suppose that the observations within each y_j are correlated through the equivariance model so that

$$y_j|\beta, \sigma^2, \rho \overset{ind.}{\sim} N(X_j\beta, \sigma^2 R(\rho))$$

for $j = 1, 2, \ldots, J$.

(a)

Suppose that we assume the model parameters have priors $\beta \sim N(\beta_0, \Sigma_0)$, $\sigma^2 \sim \text{Inv-}\chi^2(\nu, \tau^2)$, and Jeffrey's prior on ρ. Using the fact that we observe y_j independently, we can write the full posterior as

$$\begin{aligned}
p(\beta, \sigma^2, \rho|y) &\propto p(y|\beta, \sigma^2, \rho)p(\beta)p(\sigma^2)p(\rho) \\
&\propto \left(\prod_{j=1}^{J} p(y_j|\beta, \sigma^2, \rho)\right) p(\beta)p(\sigma^2)p(\rho) \\
&\propto \prod_{j=1}^{J} \left[(2\pi\sigma^2)^{-n_j/2}|R_j(\rho)|^{-1/2} \exp\left\{-\frac{1}{2}\left[(y_j - X_j\beta)^T \frac{R_j^{-1}(\rho)}{\sigma^2}(y_j - X_j\beta)\right]\right\}\right] \\
&\times (2\pi)^{-p/2}|\Sigma_0|^{-1/2} \exp\left\{-\frac{1}{2}\left[(\beta - \beta_0)^T\Sigma_0^{-1}(\beta - \beta_0)\right]\right\} \\
&\times (\sigma^2)^{-\frac{\nu}{2}-1} \exp\left\{-\frac{\nu\tau^2}{2\sigma^2}\right\} \\
&\times p(\rho)
\end{aligned}$$

where $R_j(\rho)$ is the $n_j \times n_j$ equi-correlation matrix and n_j is the length of y_j. Moreover, $p(\rho)$ is Jeffrey's prior for ρ. If we adopt Jeffrey's prior we developed in the first exercise of this assignment (this is equivalent to the model $\beta_0 = 0, \Sigma_0 = 0, \sigma^2 = 1$), we can rewrite this prior for θ in terms of ρ. As Jeffrey's prior is invariant to reparameterization, we can write Jeffrey's prior for ρ as follows

$$p(\rho) \propto p_\theta\left(\theta(\rho)\right)\left|\frac{d}{d\rho}\theta(\rho)\right| = p_\theta\left(\frac{p(n-1)+1}{n}\right)\left|\frac{n-1}{n}\right|$$

where $p_\theta(\cdot)$ is Jeffrey's prior as given in exercise 1 part (c). As our goal is to develop a full Gibbs sampler, we can first we write the full posterior conditionals for β, σ^2, and ρ as follows.

$$\begin{aligned}
p(\beta|y, \sigma^2, \rho) &\propto \exp\left\{-\frac{1}{2}\left[\sum_{j=1}^{J}(y_j - X_j\beta)^T(\sigma^2 R_j(\rho))^{-1}(y_j - X_j\beta) + (\beta - \beta_0)^T\Sigma_0^{-1}(\beta - \beta_0)\right]\right\} \\
p(\sigma^2|y, \beta, \rho) &\propto (\sigma^2)^{-\frac{n+\nu}{2}-1} \exp\left\{-\frac{\sum_{j=1}^{J}(y_j - X_j\beta)^T R_j^{-1}(\rho)(y_j - X_j\beta) + \nu\tau^2}{2\sigma^2}\right\} \\
p(\rho|y, \beta, \sigma^2) &\propto \left(\prod_{j=1}^{J}|R_j(\rho)|^{-1/2}\right) \exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{j=1}^{J}(y_j - X_j\beta)^T R_j^{-1}(\rho)(y_j - X_j\beta)\right]\right\} p_\theta\left(\frac{\rho(n-1)+1}{n}\right)
\end{aligned}$$

where $n = \sum_{j=1}^{J} n_j$. Recall this model formulation is very similar to of that we discussed in class on 11/26. In this case, however, we set $\Sigma_j = \sigma^2 R_j(\rho)$, $\beta_j = \beta$ for all j, and the priors on β_j are a common $N(\beta_0, \Sigma_0)$. Notice, by defining $R = \oplus_{j=1}^{J} R_j(\rho) \in \mathbb{R}^{n\times n}$ is an invertible matrix since each $R_j(\rho)$ is invertible. Therefore, $R^{-1/2} = U_R S_R^{-1/2}$ exists from its eigendecomposition. Moreover, by construction $R^{-1/2}y|\beta, \sigma, \rho \sim N(R^{-1/2}X\beta, \sigma^2 I_{n\times n})$ where $X = [X_1^T \quad X_2^T \ldots X_J^T]^T \in \mathbb{R}^{n\times p}$. Therefore, by regressing $R^{-1/2}Y$ onto $R^{-1/2}X$ we can use the conditional posterior updates from regular Bayesian linear modeling to write the closed form of $\beta|y, \sigma^2, \rho$. In addition, we can directly see the conditional posterior of σ^2 from the expression above. The conditional posteriors have the following form

$$\beta|y,\sigma^2,\rho \sim N\left(\Sigma_\beta\left(\frac{1}{\sigma^2}X^TR^{-1}y+\Sigma_0^{-1}\beta_0\right),\Sigma_\beta=\left(\frac{1}{\sigma^2}X^TR^{-1}X+\Sigma_0^{-1}\right)^{-1}\right)$$
$$\sigma^2|y,\beta,\rho \sim \text{Inv-}\chi^2\left(n+\nu,\frac{\sum_{j=1}^J(y_j-X_j\beta)^TR_j^{-1}(\rho)(y_j-X_j\beta)+\nu\tau^2}{n+\nu}\right)$$

Moreover, notice since R is block diagonal that $R^{-1}=\oplus_{j=1}^J R_j^{-1}(\rho)$ which allows us to utilize the closed form expression for $R_j^{-1}(\rho)$ we developed in exercise 1 part a. It is not immediately clear if $p(\rho|y,\beta,\sigma^2)$ has a closed form. For this reason, we will use a grid search over possible values of ρ to identify which point maximizes this conditional posterior with given values of β and σ^2. Indeed, we grid, the unit interval $[0,1]$, and for each θ value in the grid calculate the corresponding ρ value, and evaluate this conditional posterior. We formalize this idea in the full Gibbs sampler.

The full Gibbs sampler is summarized below.

```
0. Input starting values of parameters $\beta^{(0)}, (\sigma^2)^{(0)}, \rho^{(0)}$, the grid granularity $G$, and number of samples max.iters.
1. for $t = 1, 2, \ldots,$ max.iters do:
    a. Sample $\beta^{(t)}|y, (\sigma^2)^{(t-1)}, \rho^{(t-1)}$ according to the normal distribution above
    b. Sample $(\sigma^2)^{(t)}|y, \beta^{(t)}, \rho^{(t-1)}$ according to the inverse-Chi square distribution above
    c. for $g = 1, 2, \ldots, G$ do:
        i. Calculate $\rho_g = \rho(g/G)$
        ii. Set the conditional posterior value $vals[g] = p(\rho_g|y, \beta^{(t)}, (\sigma^2)^{(t)})$
    d. Set $\rho^{(t)} = \rho(g_*/G) : g_* = \text{argmax}_{g\in[G]} vals[g]$
2. Output: Samples of parameter values $(\beta, \sigma^2, \rho)$ from the joint posterior $p(\beta, \sigma^2, \rho|y)$
```

(b)

In this exercise we implement the Gibbs sampler introduced above to analyze the Stroke's dataset. We assume noninformative priors;$\beta \sim N(0, 10I_n)$, $\sigma^2 \sim \text{Inv-}\chi^2(0,0)$, and the Jeffrey's prior for ρ discussed above.

[Aside: In what follows, I have several issues that arise from poor sampling of the ρ parameter. I believe the issue arises in the grid search implementation. Provided more time, perhaps a Metropolis-within-Gibbs sampler for ρ could lead to better sampling performance.]

```r
# load libraries
pacman::p_load(bayesplot, Matrix, MASS, rstan)

# source a few of Luis's helpful functions
rinvchisq <- function(ns, nu, nu_tau2) 1/rgamma(ns,
    nu/2, nu_tau2/2)
mcmc_array <- function(ns, nchains = 1, params) {
    nparams <- length(params)
    array(dim = c(ns, nchains, nparams), dimnames = list(iterations = NULL,
        chains = paste0("chain", 1:nchains), parameters = params))
}

# read in data
setwd("~/Documents/Work/github/Courses/MA 578/final_exam/")
strokes <- within(read.csv("stroke.csv"), subject <- factor(subject))

# look at head
head(strokes)
```

```
##   subject group week score
## 1       1     A    1    45
## 2       1     A    2    45
## 3       1     A    3    45
## 4       1     A    4    45
## 5       1     A    5    80
## 6       1     A    6    80
```

```r
# set useful paramters
y <- matrix(strokes$score, ncol = 1)
X <- model.matrix(~week, data = strokes)
n_vec <- unname(table(strokes$subject))
n <- sum(n_vec)
J <- length(unique(strokes$subject))
r_inv <- function(p, nj) 1/(1 - p) * (diag(nj) -
    p/(1 + (nj - 1) * p) * matrix(1, ncol = nj,
        nrow = nj))
R_inv <- function(p, n) as.matrix(bdiag(lapply(n,
    function(x) r_inv(p, x))))
r_det <- function(p, nj) (1 - p)^(nj - 1) * (1 +
    p * (nj - 1))

# set sampler parameters
max.iters <- 5000
warmup <- floor(max.iters/2)
grid.size <- 100
grid <- seq(0, 1, length.out = grid.size + 2)[-c(1,
    102)]

# calculate rho.grid and allocate memory
rho.grid <- sapply(grid, function(x) (n * x -
    1)/(n - 1))
grid.vals <- numeric(grid.size)
R_inv_grid <- lapply(rho.grid, function(x) R_inv(x,
    n_vec))
log_p_theta <- function(x, n) 0.5 * log(1/x^2 +
    (n - 1)/(1 - x)^2)
log_p_rho <- function(p, beta, sigma2, X, y, n) {
    -0.5 * sum(sapply(n_vec, function(x) log(r_det(p,
        x)))) - 0.5 * (1/sigma2 * crossprod((y -
        X %*% beta), R_inv_grid[[which(p == rho.grid)]] %*%
        (y - X %*% beta))) + log_p_theta((p *
        (n - 1) + 1)/n, n)

}

# set up storage
nchains <- 1
params <- c(colnames(X), "sigma2", "rho", "lp__")
sims <- mcmc_array(max.iters - warmup, nchains,
    params)

# set initial parameters
ols_est <- coef(lm(score ~ week, data = strokes))
```

```
beta <- matrix(unname(ols_est), ncol = 1)
sigma2 <- (summary(lm(score ~ week, data = strokes))$sigma)^2
rho <- 0.1

# set priors
beta_prior <- list(c(0, 0), 10 * diag(2))
sigma2_prior <- list(0, 0)


# Gibbs sampler
for (t in 1:max.iters) {

    # cache R^{-1}
    R_inv_here <- R_inv(rho, n_vec)

    # sample beta
    Sigma_beta <- solve(1/sigma2 * crossprod(X,
        R_inv_here %*% X) + solve(beta_prior[[2]]))
    Mean_beta <- Sigma_beta %*% (1/sigma2 * crossprod(X,
        R_inv_here %*% y) + solve(beta_prior[[2]]) %*%
        beta_prior[[1]])
    beta <- matrix(mvrnorm(1, Mean_beta, Sigma_beta),
        ncol = 1)

    # sample sigma2
    sigma2 <- rinvchisq(1, n + sigma2_prior[[1]],
        crossprod((y - X %*% beta), R_inv_here %*%
            (y - X %*% beta)) + sum(unlist(sigma2_prior)))

    # sample rho
    for (g in 1:grid.size) {
        # calculate log p(p|y, beta, sigma2) for
        # stability
        grid.vals[g] <- log_p_rho(rho.grid[g],
            beta, sigma2, X, y, n)
    }
    rho <- rho.grid[which.max(grid.vals)]

    # calculate log posterior
    lp.here <- sum(sapply(n_vec, function(x) -0.5 *
        x * log(sigma2))) - 0.5 * sum(sapply(n_vec,
        function(x) log(r_det(rho, x)))) - 0.5 *
        crossprod((y - X %*% beta), 1/sigma2 *
            R_inv(rho, n_vec) %*% (y - X %*% beta)) -
        0.5 * crossprod(beta - beta_prior[[1]],
            solve(beta_prior[[2]]) %*% (beta -
                beta_prior[[1]])) - (sigma2_prior[[1]]/2 +
        1) * log(sigma2) - (sigma2_prior[[1]] *
        sigma2_prior[[2]])/(2 * sigma2) + log_p_theta(rho,
        n)

    # store if applicable
    if (t > warmup) {
        sims[t - warmup, 1, ] <- c(beta, sigma2,
            rho, lp.here)
    }

    # print updates
    if (t%%1000 == 0)
        message(paste(t/max.iters * 100), "% finished")

}
```

```
## 20% finished
```

```
## 40% finished
```

```
## 60% finished
```

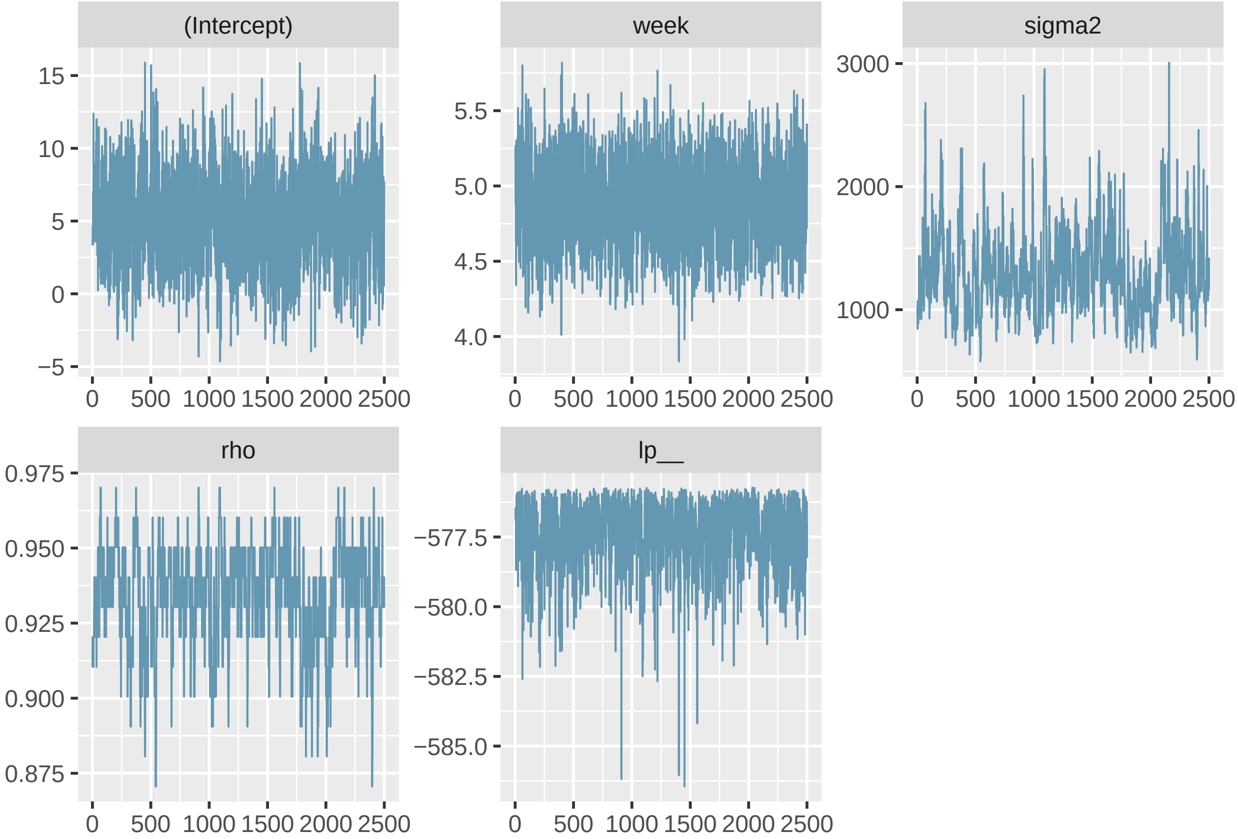
```
## 80% finished
```

```
## 100% finished
```

```
# visualize results
monitor(sims)
```

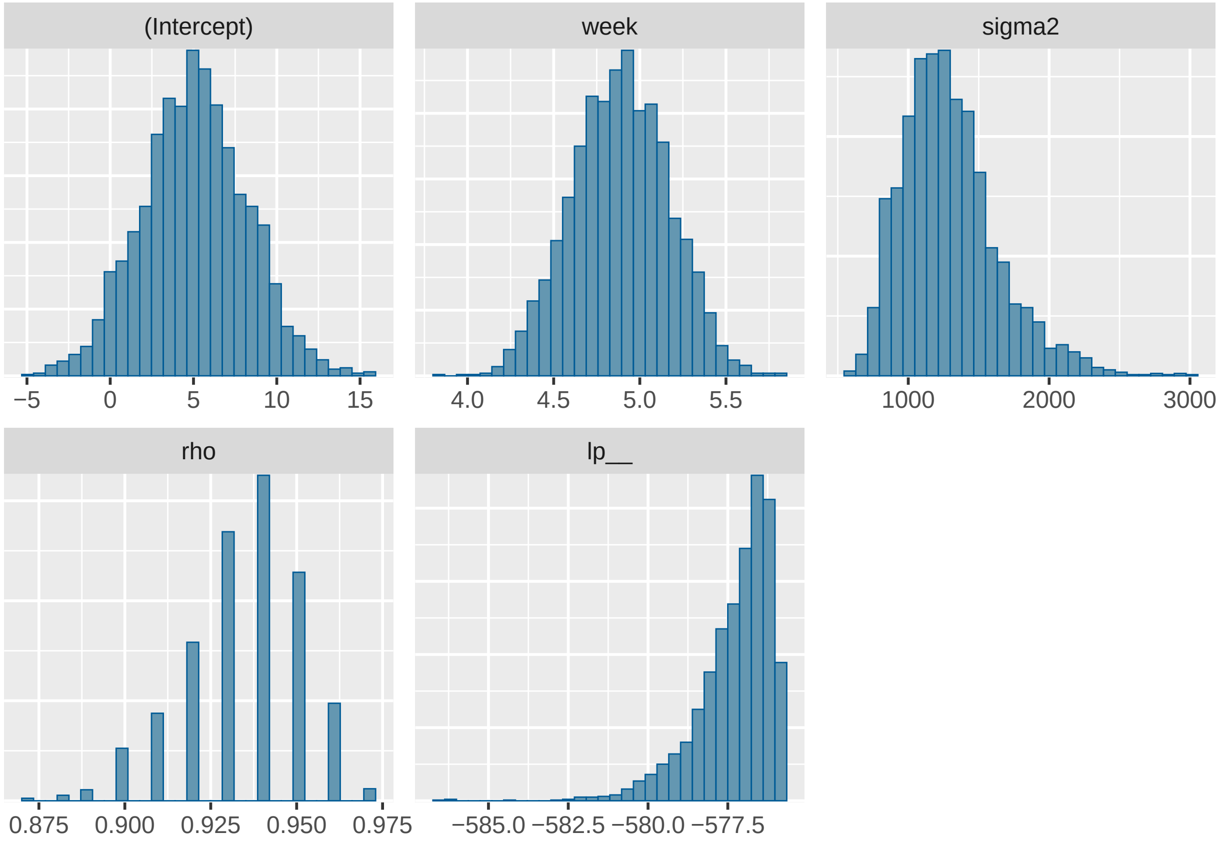
```
## Inference for the input samples (1 chains: each with iter = 2500; warmup = 1250):
##
##                 Q5    Q50    Q95   Mean    SD  Rhat Bulk_ESS Tail_ESS
## (Intercept)   -0.3    4.9    9.8    4.9   3.2  1.00      411      606
## week           4.4    4.9    5.3    4.9   0.3  1.00     1092     1104
## sigma2       820.4 1255.8 1907.7 1299.5 333.7  1.00       58      192
## rho            0.9    0.9    1.0    0.9   0.0  1.01       59      180
## lp__        -579.5 -577.0 -576.0 -577.3   1.2  1.00      266      741
##
## For each parameter, Bulk_ESS and Tail_ESS are crude measures of
## effective sample size for bulk and tail quantities respectively (an ESS > 100
## per chain is considered good), and Rhat is the potential scale reduction
## factor on rank normalized split chains (at convergence, Rhat <= 1.05).
```

```
mcmc_trace(sims)
```

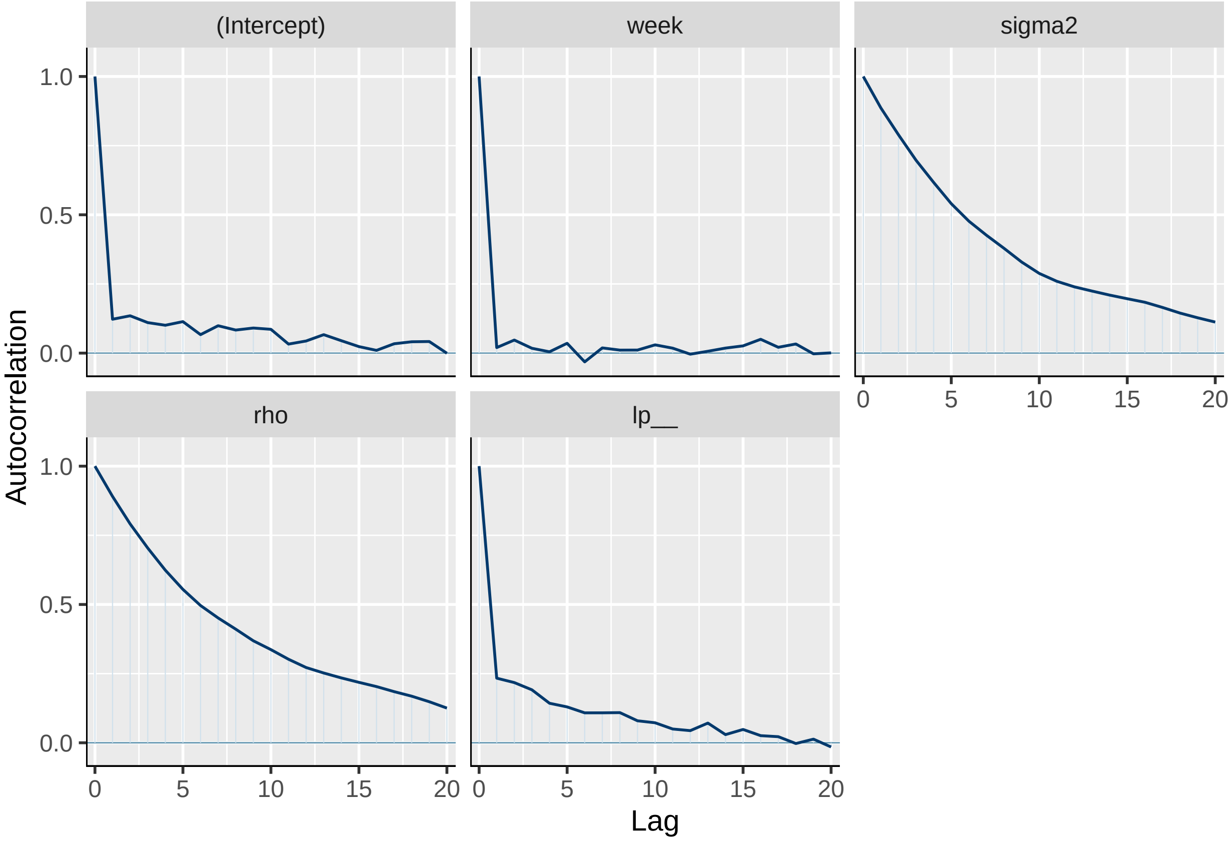


```r
mcmc_hist(sims)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
mcmc_acf(sims)
```



It appears that our Markov Chain from our implementation of the Gibbs sampler is mixing well for the β coefficients but is not for the σ^2 and ρ values. In particular, it appears that the variance is very heigh and ρ centers around a correlation values of .94 - an extremely high value. This point is captured sufficiently by the heavy autocorrelation found in both the σ^2 and ρ plots. To improve this model, one way consider adopting a Metropolis-within-Gibbs approach where we accept and reject certain values of ρ instead of using a grid of possible values.

We compare our results to that of a frequentist based approach by fitting this model in lme4. Under a frequentist guise, our model takes the form $y \sim N(X\beta, \oplus_{j=1}^{J} \sigma^2 R_j(\rho))$. Written another way,

$$y_j \sim X_j^T \beta + \sigma^2 \delta_j \quad \delta_j \sim N(0, R(\rho))$$

From this we see that our fixed effects are given by the model matrix $X_j\beta$ while the random effects $\sigma^2\delta_j$ have a random intercept term that changes from subject to subject (i.e. (1|subject)). We fit this model below.

```
pacman::p_load(lme4)
m1 <- lmer(score ~ week + (1 | subject), data = strokes)
summary(m1)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: score ~ week + (1 | subject)
##    Data: strokes
##
## REML criterion at convergence: 1468.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1794 -0.6118  0.0208  0.6453  3.0440
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  subject  (Intercept) 392.71   19.817
##  Residual              79.67    8.926
## Number of obs: 192, groups:  subject, 24
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  31.0342     4.2870   7.239
## week          4.7297     0.2811  16.824
##
## Correlation of Fixed Effects:
##      (Intr)
## week -0.295
```

```
vcov(m1)
```

```
## 2 x 2 Matrix of class "dpoMatrix"
##             (Intercept)        week
## (Intercept)   18.378240 -0.35565798
## week          -0.355658  0.07903511
```

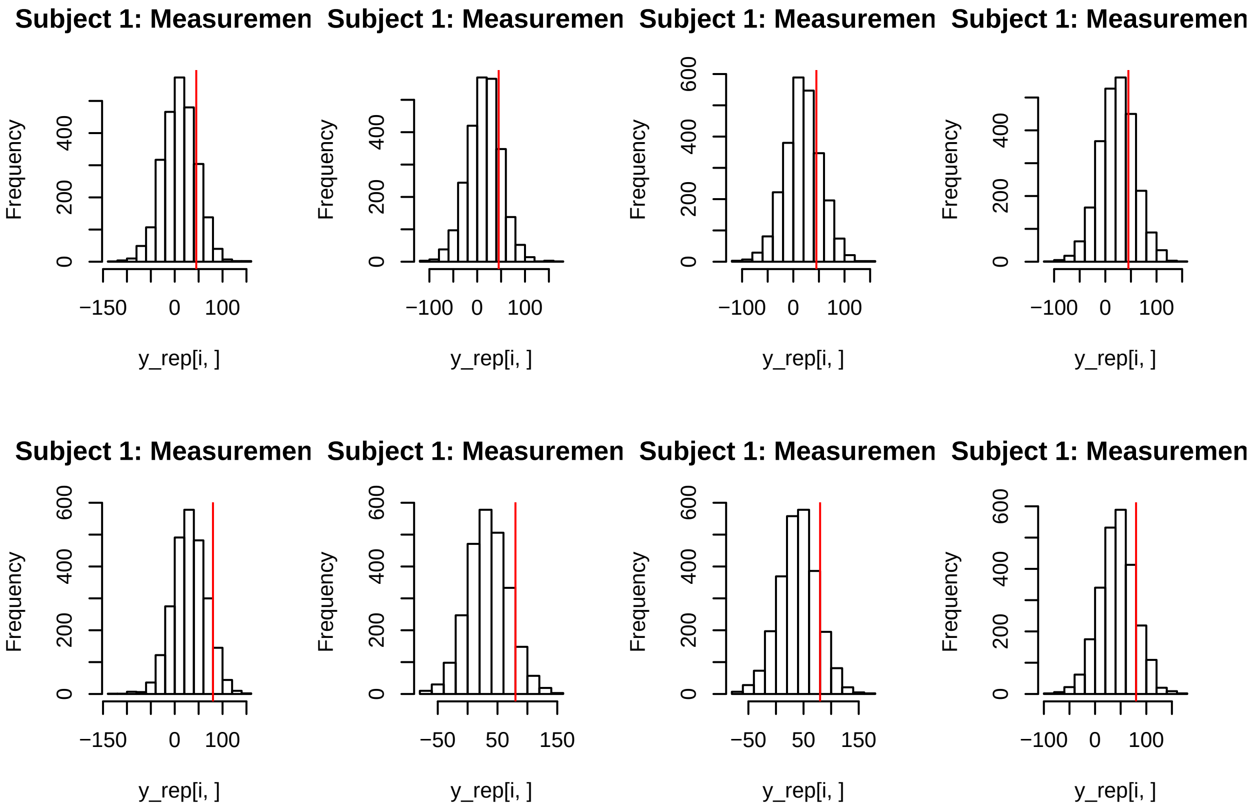
From these results we see that estimates for $\hat{\beta} = (31.0342, 4.7297)$. While the effect on week matches with our Gibbs sampler, the intercept is roughly 20 points higher than that of results produced by our Bayesian methodology. Moreover, the residual variance estimate is given by 79.67 which is significantly lower than the mean of the σ^2 posterior distribution given above. Lastly, the estimated value for subject level correlation is given by $\hat{\rho} = \frac{392.71}{392.71+79.67} = 0.8313$ which is smaller than that of the estimated value above.

These results suggest that our Gibbs is not fitting well. This could be due to a misspecification in the prior or some bug I haven't been able to sniff out. In either case, these results are largely not in agreement with the results of the Gibbs sampler above.

(c)

With our samples from the full posterior we can create a few posterior predictive checks. Namely, we can generate $y^{rep}|\beta, \sigma^2, \rho$ based on the MCMC samples drawn above. As $y \in \mathbb{R}^{192}$ we focus on the first subject's measurement and the model's ability to capture this variation.

```
# set up paramater values
betas <- sims[, 1, 1:2]
sigma2s <- sims[, 1, 3]
rhos <- sims[, 1, 4]

# set up R matrix
r <- function(n, s2, p) {
    mat <- matrix(rep(p, n^2), ncol = n, nrow = n)
    diag(mat) <- rep(s2 * p, n)
    return(mat)
}
R <- function(n_vec, s2, p) as.matrix(bdiag(lapply(n_vec,
    function(x) r(x, s2, p))))
# sample y_rep
vars <- lapply(1:length(rhos), function(x) R(n_vec,
    sigma2s[x], rhos[x]))
y_rep <- matrix(NA, ncol = length(rhos), nrow = length(y))
for (i in 1:length(rhos)) {
    y_rep[, i] <- mvrnorm(1, X %*% matrix(betas[i,
        ], ncol = 1), vars[[i]])
}

# visualize posterior prdictive focus on
# subject 1
par(mfrow = c(2, 4))
for (i in 1:8) {
    hist(y_rep[i, ], main = paste("Subject 1: Measurement",
        i))
    abline(v = y[i], col = "red")
}
```



From this figure it is clear that the posterior predictive is not capturing changes in the Subject's measurements well. Indeed, as we saw above the variances are very high and this is shown through the high dispersion of each posterior predictive. In addition, we see that these distributions may not be centering over the true value. This could be due to the high dispersion causing flawed updates for our β parameters centering the shape of these distributions.

Next we turn to an outlier analysis using uncorrelated residuals. To derive the uncorrelated residuals, we first center the y vector by $X\beta$ to arrive at our residual distribution

$$Y - X\beta|\sigma^2, Y \sim N(Y - X\hat{\beta}, \sigma^2 X\Sigma_\beta X^T)$$

where $\hat{\beta} = \Sigma_\beta(\frac{1}{\sigma^2}X^T R^{-1}y + \Sigma_0^{-1}\beta_0)$. Assuming noninformative priors these forms result to the weighted least squares parameters $\Sigma_\beta = \frac{1}{\sigma^2}(X^T R^{-1} X)^{-1}$, $\hat{\beta} = (X^T R^{-1} X)^{-1} X^T R^{-1} y$. Therefore, as in the standard regression case, by letting $R^{-1/2}X = QR'$ be the QR decomposition of our weighted design matrix, notice

$$Q^T(y - X\beta)|y, \sigma^2, \rho \sim N\left(Q^T y - Q^T X\hat{\beta}, \sigma^2 Q^T X \Sigma_\beta X^T Q\right)$$

But notice that

$$\begin{aligned}
Q^T y - Q^T X\hat{\beta} &= Q^T y - Q^T R^{1/2}(R^{-1/2}X)[(R^{-1/2}X)^T(R^{-1/2}X)]^{-1}(R^{-1/2}X)^T R^{-1/2} y \\
&= Q^T y - Q^T R^{1/2}(QR')[(QR')^T(QR')]^{-1}(QR')^T R^{-1/2} y \\
&= Q^T y - Q^T R^{1/2} QR'[R'^T R']^{-1} R'^T Q^T R^{-1/2} y \\
&= Q^T y - Q^T R^{1/2} Q Q^T R^{-1/2} y \\
&= Q^T y - Q^T y \\
&= 0
\end{aligned}$$

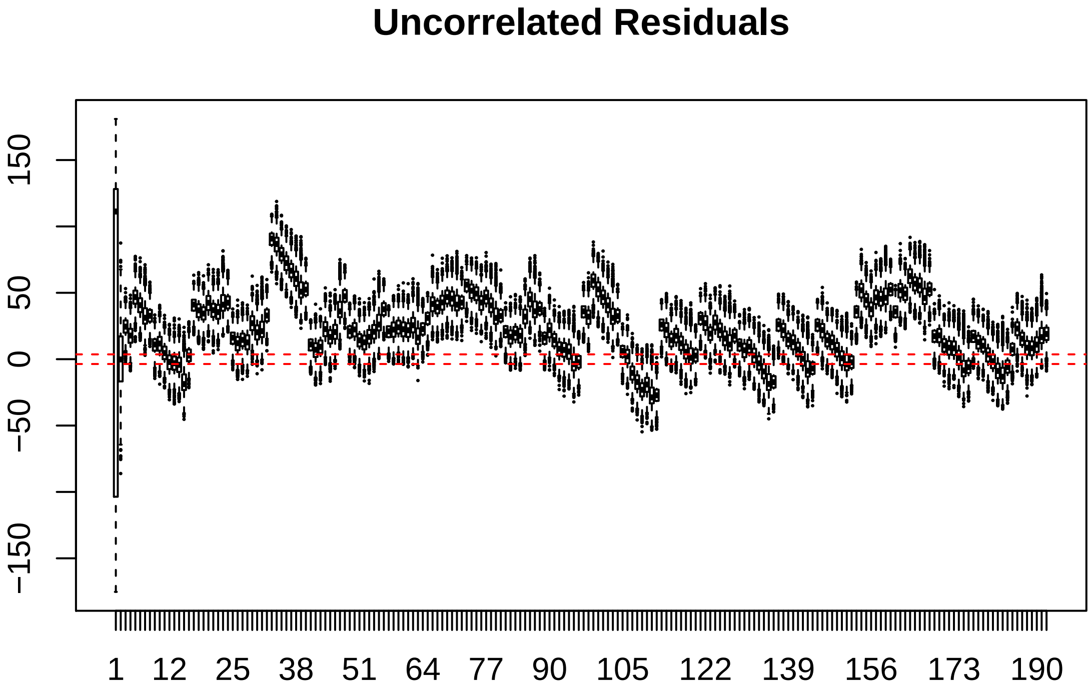
Moreover, notice that we can similarly reduce the variance as follows

$$\begin{aligned}
\sigma^2 Q^T X \Sigma_\beta X^T Q &= \sigma^2 Q^T R^{1/2}(R^{-1/2}X)\frac{[(R^{-1/2}X)^T(R^{-1/2}X)]^{-1}}{\sigma^2}(R^{-1/2}X)^T R^{1/2} Q \\
&= Q^T R^{1/2}(QR')[(QR')^T(QR')]^{-1}(QR')^T R^{1/2} Q \\
&= Q^T R^{1/2} QR'[R'^T R']^{-1} R'^T Q^T R^{1/2} Q \\
&= I
\end{aligned}$$

Therefore, we see by standardizing by the QR decomposition of $R^{-1/2}X = QR'$ we can use standard residual diagnostic techniques, such as the QQ-plot and standard normal critical values for residual analysis. We focus on an use a outlier threshold corrected for multiple testing from these uncorrelated residuals. Indeed as this QR decomposition relies on knowledge of model parameters ρ, σ^2, we will also have a full distribution of each residuals. We complete this analysis below.

```r
# get standardization matrices
mat_sqrt <- function(mat) {
    tmp <- eigen(mat)
    tmp$vectors %*% diag(sqrt(tmp$values))
}
mat_neg_sqrt <- function(mat) {
    tmp <- eigen(mat)
    tmp$vectors %*% diag(1/sqrt(tmp$values))
}

Qs <- lapply(1:length(rhos), function(x) qr.Q(qr(mat_sqrt(R(n_vec,
    s2 = sigma2s[x], p = rhos[x])) %*% X), complete = TRUE))

# get uncorrelated residuals
uncorr_residuals <- do.call("cbind", lapply(1:length(rhos),
    function(x) crossprod(Qs[[x]], y - X %*% matrix(betas[x,
        ], ncol = 1))))
```

```
# plot uncorrelated residuals
outlier_threshold <- function(ntests = 1, alpha = 0.05) -qnorm(0.5 *
    (1 - (1 - alpha)^(1/ntests)))
k <- outlier_threshold(length(y))
boxplot(t(uncorr_residuals), main = "Uncorrelated Residuals",
    cex = 0.1)
abline(h = k, col = "red", lty = 2)
abline(h = -k, col = "red", lty = 2)
```



The above plot shows the boxplot of the uncorrelated residuals plotted against their index. In red we have plotted a critical threshold adjusted for $n = 192$ hypothesis tests. From this chart we seem there are several issues. Firstly, there is clear trend throughout the residuals. Particularly, there is clear trend within each subject suggesting the model is failing to capture this within subject variation. This, again, can be attributed to the poor sampling of the ρ and σ^2 parameter. Moreover, we see from the above that nearly all the data falls outside the critical values thus making the majority of points "outliers". Again, however, these transformed residuals were functions of the unknown model parameters ρ and σ^2. Improving our estimation of these parameters will change this transformation and hence our outleir analysis as well.

(d)

To analyze if the inclusion of each interaction term we adopt the same modeling approach as above but instead include more terms in the modeling equation. That is we include an effect for group and week as well as an interaction. We run this Gibbs sampler for 5000 iterations with a 2500 warmup period resulting in 2500 draws from the target distribution.

```
#----------------------------------
# Interaction Model
#----------------------------------

# set up model matrix
X <- model.matrix(~group + group:week, data = strokes)
```

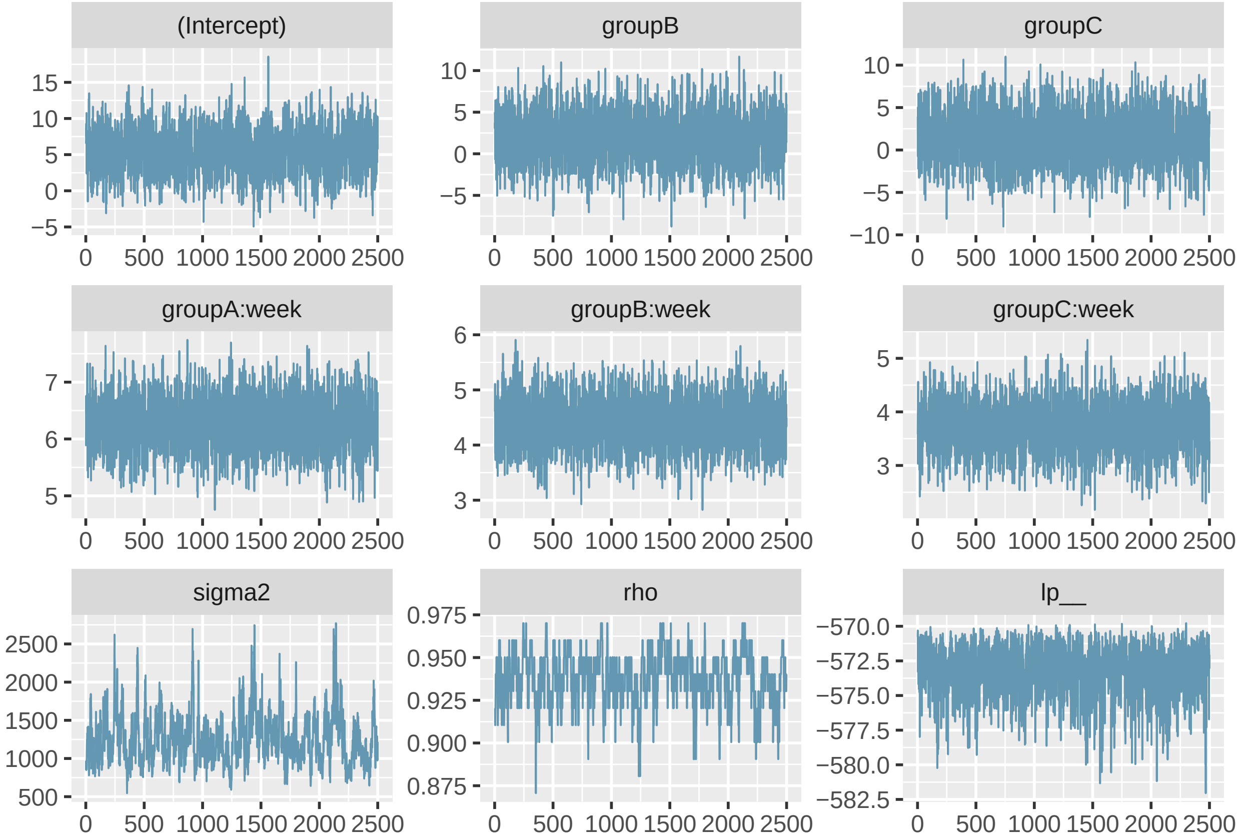
```r
# set sampler parameters
max.iters <- 5000
warmup <- floor(max.iters/2)

# set up storage
nchains <- 1
params <- c(colnames(X), "sigma2", "rho", "lp__")
sims.int <- mcmc_array(max.iters - warmup, nchains,
    params)

# set initial parameters
ols_est <- coef(lm(score ~ week * group, data = strokes))
beta <- matrix(unname(ols_est), ncol = 1)
sigma2 <- (summary(lm(score ~ week * group, data = strokes))$sigma)^2
rho <- 0.1

# set priors
beta_prior <- list(rep(0, 6), 10 * diag(6))
sigma2_prior <- list(0, 0)

# Gibbs sampler
for (t in 1:max.iters) {

    # cache R^{-1}
    R_inv_here <- R_inv(rho, n_vec)

    # sample beta
    Sigma_beta <- solve(1/sigma2 * crossprod(X,
        R_inv_here %*% X) + solve(beta_prior[[2]]))
    Mean_beta <- Sigma_beta %*% (1/sigma2 * crossprod(X,
        R_inv_here %*% y) + solve(beta_prior[[2]]) %*%
        beta_prior[[1]])
    beta <- matrix(mvrnorm(1, Mean_beta, Sigma_beta),
        ncol = 1)

    # sample sigma2
    sigma2 <- rinvchisq(1, n + sigma2_prior[[1]],
        crossprod((y - X %*% beta), R_inv_here %*%
            (y - X %*% beta)) + sum(unlist(sigma2_prior)))

    # sample rho
    for (g in 1:grid.size) {
        # calculate log p(p|y, beta, sigma2) for
        # stability
        grid.vals[g] <- log_p_rho(rho.grid[g],
            beta, sigma2, X, y, n)
    }
    rho <- rho.grid[which.max(grid.vals)]

    # calculate log posterior
    lp.here <- sum(sapply(n_vec, function(x) -0.5 *
        x * log(sigma2))) - 0.5 * sum(sapply(n_vec,
        function(x) log(r_det(rho, x)))) - 0.5 *
        crossprod((y - X %*% beta), 1/sigma2 *
            R_inv(rho, n_vec) %*% (y - X %*% beta)) -
        0.5 * crossprod(beta - beta_prior[[1]],
            solve(beta_prior[[2]]) %*% (beta -
                beta_prior[[1]])) - (sigma2_prior[[1]]/2 +
        1) * log(sigma2) - (sigma2_prior[[1]] *
        sigma2_prior[[2]])/(2 * sigma2) + log_p_theta(rho,
        n)

    # store if applicable
    if (t > warmup) {
        sims.int[t - warmup, 1, ] <- c(beta, sigma2,
            rho, lp.here)
    }

    # print updates
    if (t%%1000 == 0)
        message(paste(t/max.iters * 100), "% finished")
}
```

```
## 20% finished

## 40% finished

## 60% finished

## 80% finished

## 100% finished
```

```
# visualize results
monitor(sims.int)
```

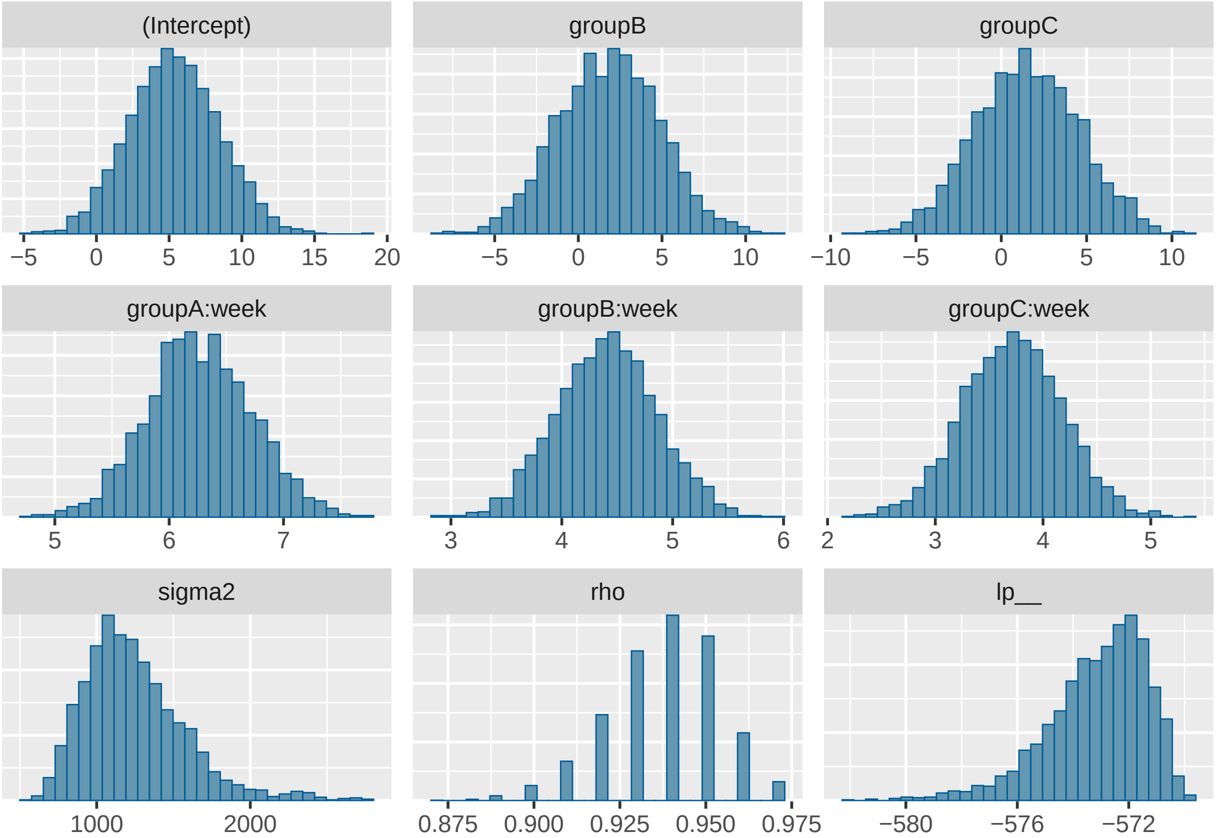
```
## Inference for the input samples (1 chains: each with iter = 2500; warmup = 1250):
##
##                 Q5    Q50    Q95   Mean    SD  Rhat Bulk_ESS Tail_ESS
## (Intercept)    0.3    5.3   10.6    5.3   3.1  1.00      363      899
## groupB        -3.3    1.9    6.7    1.8   3.0  1.00     1205     1079
## groupC        -3.3    1.5    6.4    1.5   3.0  1.00     1202     1203
## groupA:week    5.5    6.2    7.1    6.3   0.5  1.00     1232     1306
## groupB:week    3.7    4.4    5.1    4.4   0.4  1.00     1164     1182
## groupC:week    3.0    3.7    4.5    3.7   0.5  1.00     1189     1222
## sigma2       802.4 1220.0 1912.1 1274.5 350.0  1.01       50       84
## rho            0.9    0.9    1.0    0.9   0.0  1.01       50       77
## lp__        -576.8 -572.8 -570.8 -573.2   1.9  1.00      241      808
##
## For each parameter, Bulk_ESS and Tail_ESS are crude measures of
## effective sample size for bulk and tail quantities respectively (an ESS > 100
## per chain is considered good), and Rhat is the potential scale reduction
## factor on rank normalized split chains (at convergence, Rhat <= 1.05).
```

```r
mcmc_trace(sims.int)
```

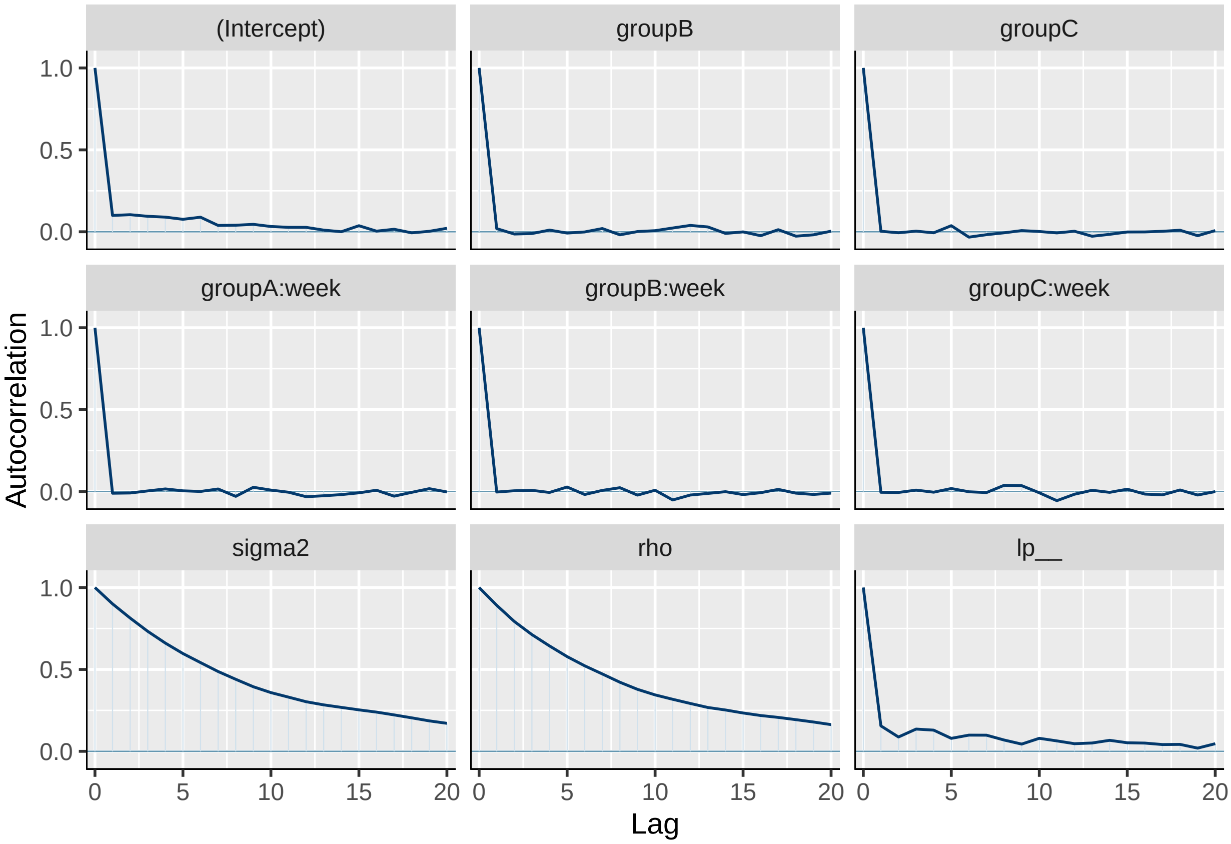


```r
mcmc_hist(sims.int)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```r
mcmc_acf(sims.int)
```



```
# calculate 95\% probability regions for
# parameters of interest
quantile(sims.int[, 1, 4], probs = c(0.0275, 0.975))
```

```
##    2.75%    97.5%
## 5.416406 7.172072
```

```
quantile(sims.int[, 1, 5], probs = c(0.0275, 0.975))
```

```
##    2.75%    97.5%
## 3.566310 5.292378
```

```
quantile(sims.int[, 1, 6], probs = c(0.0275, 0.975))
```

```
##    2.75%    97.5%
## 2.826998 4.610384
```

To informally test the hypothesis H_0: week:group = 0 against H_1: week:group$\neq$ 0 we consider the posterior distributions of both of these parameters. From this histograms, it is clear that each of these distributions are far from zero indicating that each interaction term plays a role in the model. Moreover, we construct 95% credibility intervals for these parameters. As each of these intervals do not contain 0, this provides informal evidence that suggest the interaction term is in H_1.