

Bayesian Inference 206B Final Exam

Jordan Berninger

3/7/2020

Question 1:

- (a) Metropolis within Gibbs to sample from $p(\nu, \theta | \mathbf{y})$. Specifically, we are using Metropolis Hastings to perform a random walk proposal on $\log(\nu)$ and we are using a Gibbs Sampler on θ , because we know the Gamma distribution has conjugacy in its rate parameter (here, θ is the rate parameter).

First, we will simulate the data from $\Gamma(2, 3)$.

```
set.seed(112358)
n <- 500
x.observed <- rgamma(n, 2, 3)
```

Now we code the log-likelihood of our parameters given the simulated data, because that is easier to handle than the regular likelihood function. We have the following likelihood functions:

$$\begin{aligned} L(\theta, \nu | \mathbf{y}) &\propto f(\mathbf{y} | \theta, \nu) \\ &\propto \left(\frac{\theta^\nu}{\Gamma(\nu)} \right)^n \exp(-\theta \sum y_i) (\Pi y_i^{\nu-1}) \theta \exp(-2\theta) \nu^2 \exp(-\nu) \\ l(\theta, \nu | \mathbf{y}) &\propto n \log\left(\frac{\theta^\nu}{\Gamma(\nu)}\right) - \theta \sum y_i + \log \Pi y_i^{\nu-1} + \log \theta - 2\theta + 2 \log \nu - \nu \end{aligned}$$

We now code the log-likelihood function, run the Metropolis within Gibbs sampler and confirm that our acceptance proportion is in the desired range.

```
##### Mteropolis within Gibbs #####
## function to evaluate log posterior

log_lik <- function(theta.tilde){
  nu <- exp(theta.tilde[1])

  log_lik <- n*log(theta[2]^nu/gamma(nu)) + sum(log(x.observed^(nu-1))) -
    theta[2]*sum(x.observed) -nu -2*theta[2] + log(theta[2])

  return(log_lik)
}

niter <- 10000
theta.store <- matrix(NA,niter,2)
theta <- c(1, 1) ## initial value
var.tuning <- 0.01
accept <- 0

for(i in 1:niter){
  prop <- rnorm(1, log(theta[1]), sqrt(var.tuning))
  ## proposal parameters needs nu and theta - are both randomly drawn?
  ## I added the prior for theta to the above
```

```

acceptance.prob <- min(exp(log_lik(prop)-log_lik(log(theta))),1)
u <- runif(1)
if(u<acceptance.prob){
  theta[1] <- exp(prop[1])
  accept <- accept+1
}else{
  theta[1] <- theta[1]
  accept <- accept
}
theta[2] <- rgamma(1, n*theta[1] + 2, 2+sum(x.observations))
theta.store[i,] <- theta
}

accept/niter ## should be between 0.3 to 0.5

```

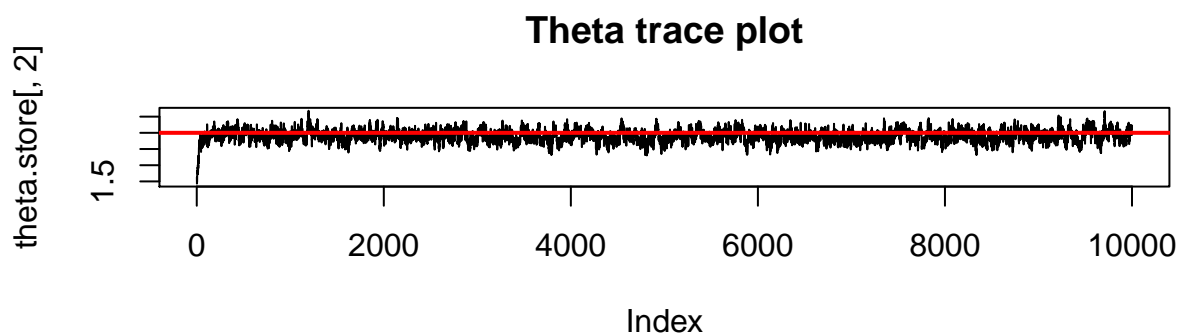
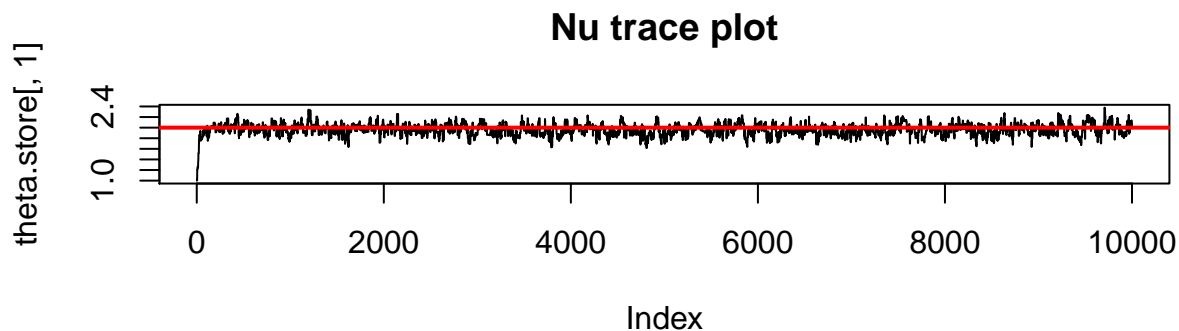
```
## [1] 0.3221
```

The trace plots for our two parameters and confirm that they converge to the true parameter values.

```

# ok this seemed to work out?
par(mfrow = c(2,1))
plot(theta.store[,1],type="l", main = "Nu trace plot")
abline(2,0,lwd=2,col="red")
plot(theta.store[,2],type="l", main = "Theta trace plot")
abline(3,0,lwd=2,col="red")

```



Now we look at the autocorrelation plots and note that there is significant autocorrelation for up to lag 20. In order to remove the autocorrelation from our Markov Chain, we will simply sample every 20th value before we make inference.

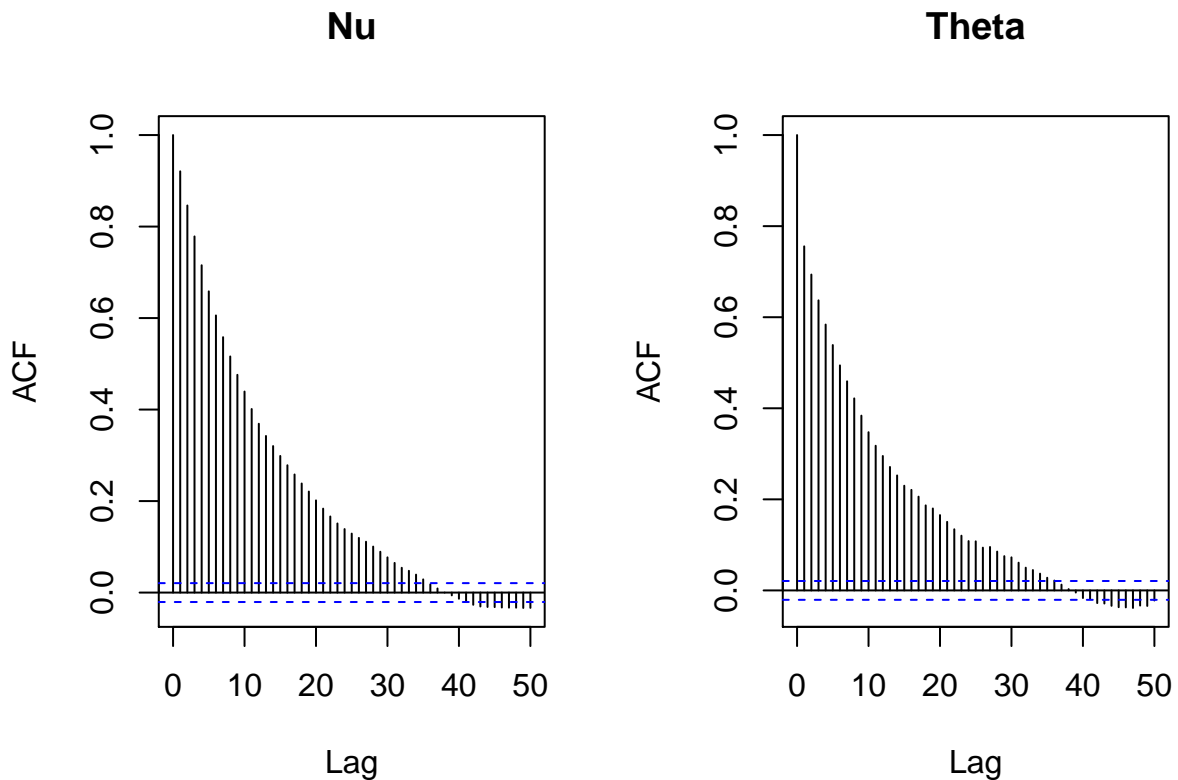
```
library(coda)
burn.in <- 1000
effectiveSize(theta.store[(burn.in+1):niter,1])
```

```
##      var1
## 371.2626
```

```
effectiveSize(theta.store[(burn.in+1):niter,2])
```

```
##      var1
## 471.4465
```

```
par(mfrow=c(1,2))
acf(theta.store[(burn.in+1):niter,1],lag=50, main = "Nu")
acf(theta.store[(burn.in+1):niter,2],lag=50, main = "Theta")
```



After filtering out the autocorrelated values, we confirm the posterior means are close to the true parameter values.

```
posterior.keep <- theta.store[seq(from=(burn.in+1), to=niter,by=20),]
colMeans(posterior.keep)
```

```
## [1] 1.961718 2.901115
```

And finally, we return the 95% credible intervals for ν and θ

```
# nu
quantile(posterior.keep[,1], probs = c(0.025, 0.975))
```

```
##      2.5%    97.5%
## 1.748398 2.185264
```

```
# theta
quantile(posterior.keep[,2], probs = c(0.025, 0.975))
```

```
##      2.5%    97.5%
## 2.558256 3.288178
```

This Metropolis within Gibbs Markov Chain appears to recover the true parameters quite well. To recap, the trace plots show that the Markov Chains converge approximately to a distribution centered on the true parameter mean. The effective sample size shows that the entire markov chain is not victim to extreme autocorrelation (we have enough data for posterior inference even after removing autocorrelated values). The 95% credible interval contains the true values of the parameters from our simulated data.

- (b) Double Metropolis that jointly proposes $\log(\nu)$ and $\log(\theta)$ using a Gaussian random walk centered on the current value of the parameters. To do this, we will use a normal random sampler, whose with variances generated from previous exploration (and was tweaked so the acceptance ratio is in the desired range). We run the loop and confirm that we have the correct acceptance rate.

```
### function to evaluate log posterior

log_lik <- function(theta.tilde){
  nu <- exp(theta.tilde[1])
  theta <- exp(theta.tilde[2])

  log_lik <- (nu*n + 1)*log(theta) -n*log(gamma(nu)) + log(product(x.observations^(nu-1))) -
    theta*sum(x.observations) -nu -2*theta + log(4/(gamma(3)*gamma(2)))

  return(log_lik)
}

niter <- 10000
theta.store <- matrix(NA,niter,2)
theta <- c(1, 1) ## initial value
var.tuning.nu <- 0.005
var.tuning.theta <- 0.005
accept <- 0

var.tuning <- matrix(c(0.005, 0.00001, 0.00001, 0.005), ncol = 2)
# these var.tuning values were derived from playing with the data

## now we substitute in the multivariate normal sampler for the proposal
for(i in 1:niter){
  prop <- mvrnorm(n = 1, mu = log(theta), Sigma = var.tuning)
  # proposal sample is from a mvnormal dist with covariance matrix defined above
  acceptance.prob <- min(exp(log_lik(prop)-log_lik(log(theta))),1)
  u <- runif(1)
  if(u<acceptance.prob){
```

```

    theta <- exp(prop)
    accept <- accept+1
  }else{
    theta <- theta
    accept <- accept
  }
  theta.store[i,] <- theta
}
accept/niter

```

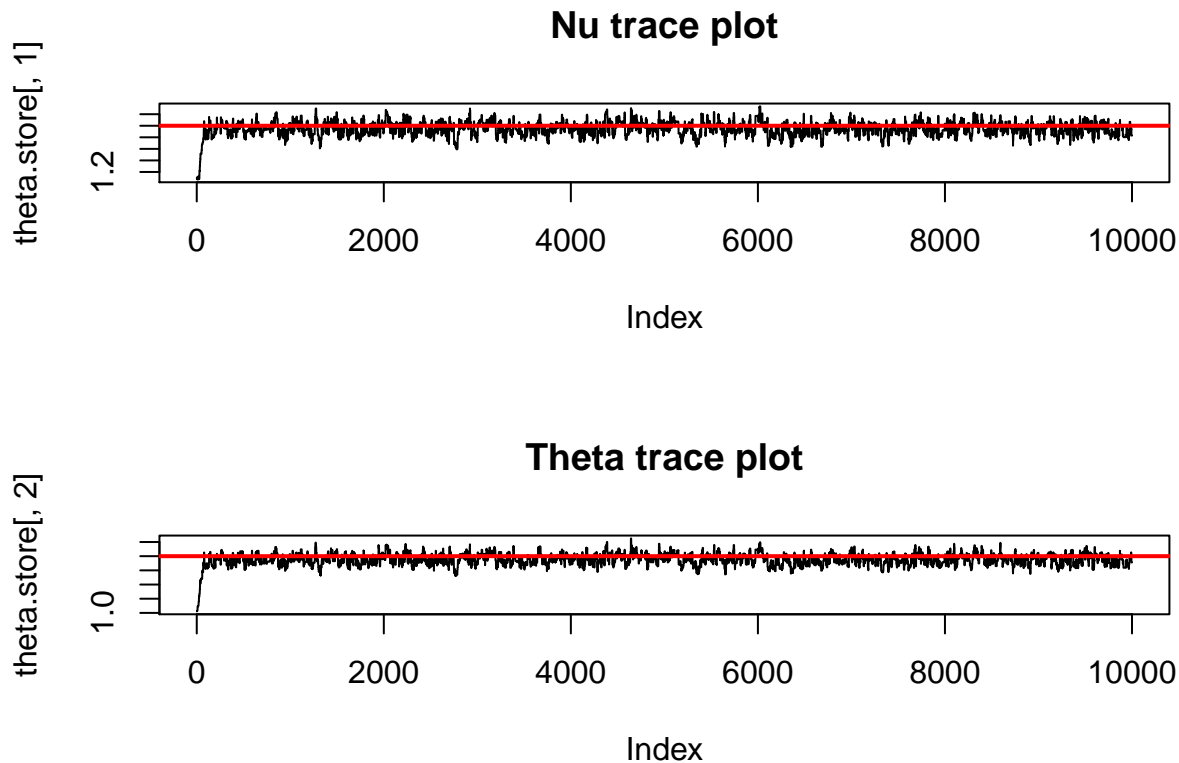
```
## [1] 0.3
```

The trace plots for our two parameters and confirm that they converge to distributions that appear centered on the true parameter values.

```

par(mfrow = c(2,1))
plot(theta.store[,1],type="l", main = "Nu trace plot")
abline(2,0,lwd=2,col="red")
plot(theta.store[,2],type="l", main = "Theta trace plot")
abline(3,0,lwd=2,col="red")

```



Now we look at the autocorrelation plots and note that there is significant autocorrelation for up to lag 20, once again. In order to remove the autocorrelation from our Markov Chain, we will simply sample every 20th value before we make inference.

```

library(coda)
burn.in <- 1000
effectiveSize(theta.store[(burn.in+1):niter,1])

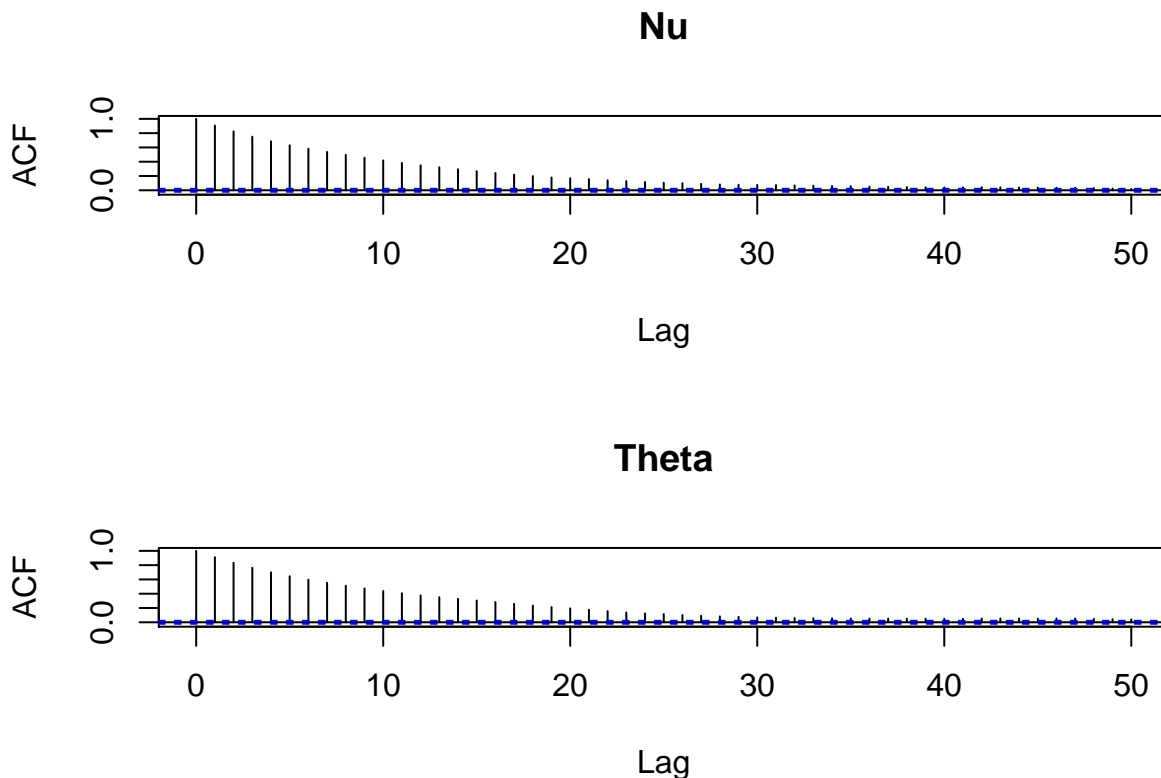
```

```
##      var1
## 380.3162
```

```
effectiveSize(theta.store[(burn.in+1):niter,2])
```

```
##      var1
## 401.2923
```

```
par(mfrow=c(2,1))
acf(theta.store[(burn.in+1):niter,1],lag=50, main = "Nu")
acf(theta.store[(burn.in+1):niter,2],lag=50, main = "Theta")
```



After filtering out the autocorrelated values, we confirm the posterior means are close to the true parameter values.

```
posterior.keep <- theta.store[seq(from=(burn.in+1), to=niter,by=20),]
colMeans(posterior.keep)
```

```
## [1] 1.952362 2.889270
```

And finally, we return the 95% credible intervals for ν and θ , noting that these are similar to, but slightly tighter than, the credible intervals from Metropolis within Gibbs.

```
# nu
quantile(posterior.keep[,1], probs = c(0.025, 0.975))
```

```
##      2.5%      97.5%
## 1.725815 2.190581
```

```
# theta
quantile(posterior.keep[,2], probs = c(0.025, 0.975))
```

```
##      2.5%      97.5%
## 2.508801 3.306980
```

Overall, we see very similar results when we run Metropolis within Gibbs vs Double Metropolis on this data model. Both Markov Chains converge to the true values of the both parameters. The posterior means are very close to the true values, the degree of autocorrelation is comparable and the 95% credible intervals are also quite similar for these two methods.

Question 2:

We want to sample from the posterior distribution $\pi(\theta, \phi, m|y_1, \dots, y_n)$ with a Gibbs Sampler. To do this, we need a data model and prior distributions. Then we state the form of the joint posterior distribution and from there determine the full conditional posterior distributions of our parameters - $\pi(\theta|\phi, m, \mathbf{y})$, $\pi(\phi|\theta, m, \mathbf{y})$ and $\pi(m|\theta, \phi, \mathbf{y})$. Once we have these full conditional distributions, we can implement them in code and sample from our joint posterior distribution, which is a powerful inferential tool.

We begin by stating our data model. We are given a split Poisson distribution, so $f(y_i|\theta) = \frac{\theta^{y_i} \exp(-\theta)}{y_i!}$ for $i = 1, \dots, m$ and $f(y_i|\phi) = \frac{\phi^{y_i} \exp(-\phi)}{y_i!}$ for $i = m+1, \dots, n$.

We are given a uniform prior on m , so $\pi(m) = \frac{1}{n}$, for $i = 1, \dots, n$. We are also told that $\theta \sim \Gamma(a, b)$ and $\phi \sim \Gamma(\gamma, \delta)$.

Now, we want to use our joint posterior distribution to determine the full conditional posterior distributions. To make things easier to read, let $s_k = \sum_{i=1}^k y_i$. Now, we have

$$\begin{aligned} \pi(\theta, \phi, m|\mathbf{y}) &\propto f(\mathbf{y}|\theta, \phi, m) \pi(\theta) \pi(\phi) \pi(m) \\ &\propto \left[\frac{\theta^{s_m} \exp(-m\theta)}{\prod_{i=1}^m y_i} \right] \left[\frac{\phi^{s_n-s_m} \exp(-(n-m)\phi)}{\prod_{i=m+1}^n y_i} \right] \left[\frac{b^a}{\Gamma(a)} \theta^{a-1} \exp(-b\theta) \right] \left[\frac{\delta^\gamma}{\Gamma(\gamma)} \phi^{\gamma-1} \exp(-\delta\phi) \right] \left[\frac{1}{n} \right] \\ &\propto \left[\theta^{s_m} \exp(-m\theta) \right] \left[\phi^{s_n-s_m} \exp(-(n-m)\phi) \right] \left[\theta^{a-1} \exp(-b\theta) \right] \left[\phi^{\gamma-1} \exp(-\delta\phi) \right] \\ &\propto \left[\theta^{a+s_m-1} \exp(-\theta(b+m)) \right] \left[\phi^{\gamma+s_n-s_m-1} \exp(-\phi(\delta+n-m)) \right] \end{aligned}$$

Since the joint posterior factors nicely, we have $\pi(\theta|m, \mathbf{y}) \sim \Gamma(a+s_m, b+m)$, $\pi(\phi|m, \mathbf{y}) \sim \Gamma(\gamma+s_n-s_m, \delta+n-m)$, and $\pi(m|\theta, \phi, m, \mathbf{y}) \propto \frac{\Gamma(a+s_m)\Gamma(\gamma+s_n-s_m)}{(m+b)^{a+s_m}(\delta+n-m)^{\gamma+s_n-s_m}}$.

These are the conditional distributions required for the Gibbs Sampler. Now, we import the data, code the conditional distributions, run the algorithm, check the diagnostics and visualize some samples from the posterior distribution.

```
coal <- data.frame(year = c(1851:1962), count = c(4,
5, 4, 1, 0, 4, 3, 4, 0, 6, 3, 3, 4, 0, 2, 6, 3, 3, 5, 4, 5, 3,
1, 4, 4, 1, 5, 5, 3, 4, 2, 5, 2, 2, 3, 4, 2, 1, 3, 2, 2, 1, 1,
1, 1, 3, 0, 0, 1, 0, 1, 1, 0, 0, 3, 1, 0, 3, 2, 2, 0, 1, 1, 1,
0, 1, 0, 1, 0, 0, 0, 2, 1, 0, 0, 0, 1, 1, 0, 2, 3, 3, 1, 1, 2,
1, 1, 1, 1, 2, 4, 2, 0, 0, 0, 1, 4, 0, 0, 0, 1, 0, 0, 0, 0, 0,
1, 0, 0, 1, 0, 1))
# the full conditional distribution
```

```

full = function(m, theta, phi, y, n, alpha, beta, gamma, delta){
  theta^(alpha-1+ifelse(m>1,sum(y[1:m]),0))*exp(-(beta+m)*theta)*phi^(gamma-1+ifelse(m<n,sum(y[(m+1):n])
}

# Initial values for the hyperparameters
alpha <- 0.01
beta <- 0.01
delta <- 0.01
gamma <- 0.01
m <- 40

set.seed(112358)
y <- coal$count
n <- length(y)

# MCMC set up
n0 <- 10000 # Burn-in
n1 <- 10000 # posterior draws
niter <- n0+n1
store <- matrix(0,niter,3)

### Gibbs Sampler ###
for (iter in 1:niter){
  theta = rgamma(1, ifelse(m>1,sum(y[1:m]),0)+alpha, m+beta)
  phi = rgamma(1, ifelse(m<n,sum(y[(m+1):n]),0)+gamma, n-m+delta)
  fulls = NULL
  for (j in 1:n)
    fulls = c(fulls, full(j, theta, phi, y, n, alpha, beta, gamma, delta))
  fulls = fulls/sum(fulls)
  m = sample(1:n, size=1, prob=fulls)
  store[iter,] = c(theta,phi,m)
}

```

After running the Gibbs Sampler, we need to filter out the autocorrelated data points (we keep every 10th data point) and confirm the resulting data does not have significant autocorrelation.

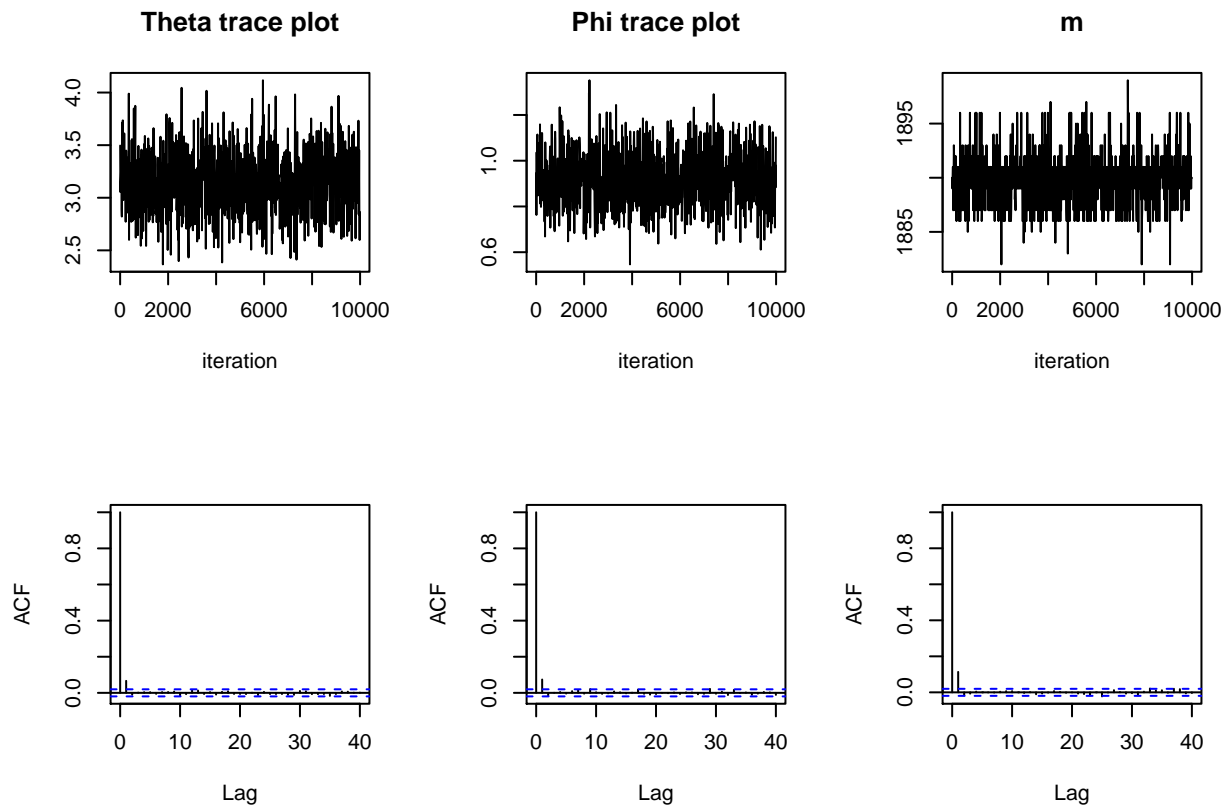
```

store = store[(n0+1):niter,]

ind = seq(1,n1,by=n1/1000)

par(mfrow=c(2,3))
plot(ind,store[ind,1],xlab="iteration",ylab="",main="Theta trace plot",type="l")
plot(ind,store[ind,2],xlab="iteration",ylab="",main="Phi trace plot",type="l")
plot(ind,1850+store[ind,3],xlab="iteration",ylab="",main="m",type="l")
acf(store[,1],main="")
acf(store[,2],main="")
acf(store[,3],main="")

```

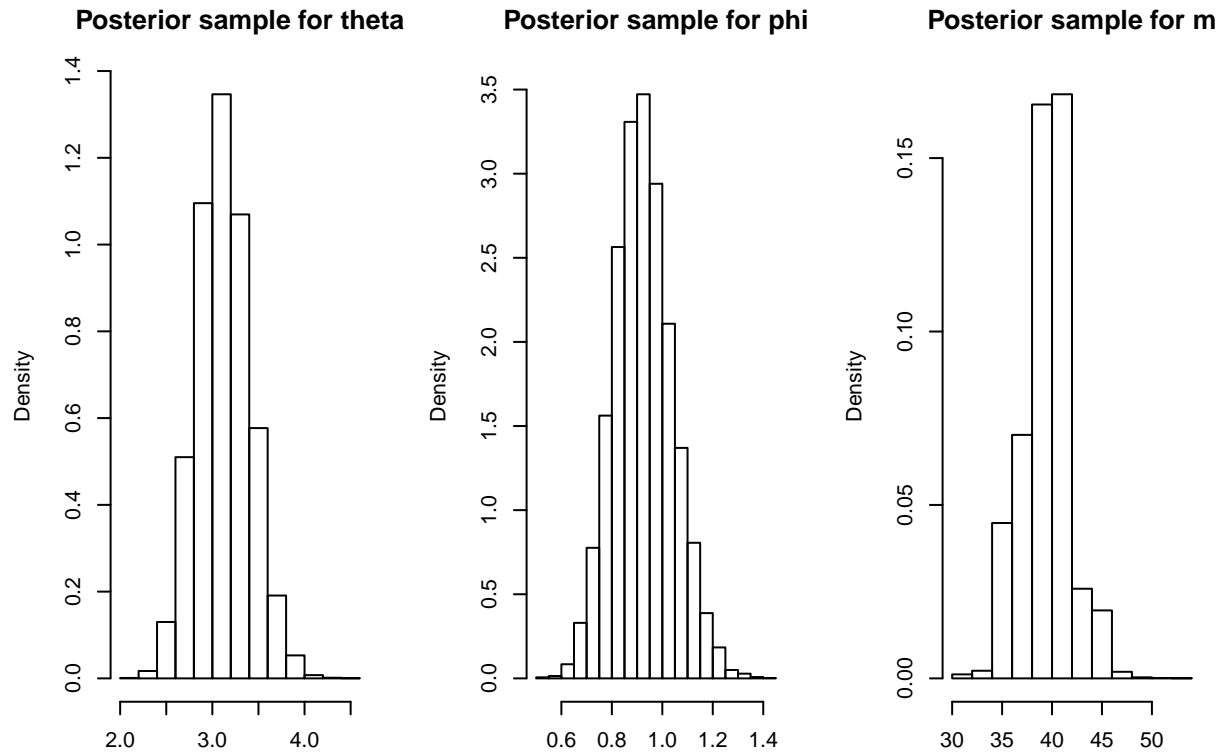



Now we check out the posterior means for our parameters, the 95% credible intervals and the histograms of the posterior samples.

```
# theta, phi and m posterior means
colMeans(store[ind,])
```

```
## [1] 3.1288181 0.9229747 39.8540000
```

```
par(mfrow=c(1,3))
hist(store[,1],xlab="",main="Posterior sample for theta",prob=TRUE)
hist(store[,2],xlab="",main="Posterior sample for phi",prob=TRUE)
hist(store[,3],xlab="",main="Posterior sample for m",prob=TRUE)
```



Lastly, we are interested in the 95% credible intervals for our 3 parameters.

```
# theta
quantile(store[ind,1], probs = c(0.025, 0.975))
```

```
##      2.5%      97.5%
## 2.584544 3.714572
```

```
# phi
quantile(store[ind,2], probs = c(0.025, 0.975))
```

```
##      2.5%      97.5%
## 0.6950245 1.1575094
```

```
# m
quantile(store[ind,3], probs = c(0.025, 0.975))
```

```
##      2.5% 97.5%
##      36    46
```

Question 3

We begin by reparameterizing the model to make things simpler. Note that the following model is equivalent to what we are given in the question. Let $y_{i,j} \sim N(\mu_i, \tau^2)$ and $\mu_i \sim N(\beta, \sigma^2)$. We keep the same prior distribution on β, σ^2 and τ^2 . This reparameterization gives us the following joint posterior distribution:

$$\begin{aligned}
p(\beta, \sigma^2, \mu_i, \tau^2 | \mathbf{y}) &\propto f(\mathbf{y} | \mu_i, \tau^2) f(\mu_i | \beta, \sigma^2) \pi(\sigma^2) \pi(\tau^2) \pi(\beta) \\
&\propto \Pi_i \Pi_j (2\pi\tau^2)^{-1/2} \exp\left(\frac{-1}{2\tau^2} (y_{i,j} - \mu_i)^2\right) \\
&\quad * \Pi_i (2\pi\sigma^2)^{-1/2} \exp\left(\frac{-1}{2\sigma^2} (\mu_i - \beta)^2\right) \\
&\quad * (\sigma^2)^{-1-1} \exp(-1/\sigma^2) * (\tau^2)^{-2-1} \exp(-2/\tau^2) * (1) \\
&\propto (\tau^2)^{-20/2} \exp\left(\sum_i \sum_j \frac{-1}{2\tau^2} (y_{i,j} - \mu_i)^2\right) \\
&\quad * (\sigma^2)^{-5/2} \exp\left(\frac{-1}{2\sigma^2} \sum_i (\mu_i - \beta)^2\right) \\
&\quad * (\sigma^2)^{-2} \exp(-1/\sigma^2 - 2/\tau^2) * (\tau^2)^{-3}
\end{aligned}$$

We can derive the full conditionals from this joint posterior distribution. For each of our parameters, we will treat the other parameters as constants and use proportionality to simplify the joint posterior. This gives us the following conditional distributions.

First, we solve for β .

$$\begin{aligned}
p(\beta) &\propto f(\mu_i | \beta, \sigma^2) \pi(\beta) \\
&\propto \Pi_i^5 (2\pi\sigma^2)^{-1/2} \exp\left(\frac{-1}{2\sigma^2} (\mu_i - \beta)^2\right) * (1) \\
&\propto \exp\left(\sum_{i=1}^5 \frac{-1}{2\sigma^2} (\mu_i - \beta)^2\right) \\
&\propto \exp\left(\sum_{i=1}^5 \frac{-1}{2\sigma^2} (\mu_i^2 - 2\mu_i\beta + \beta^2)\right) \\
&\propto \exp\left(\frac{-1}{2\sigma^2} (-2n\bar{\mu}\beta + \beta^2)\right) \\
&\propto \exp\left(\frac{-1}{2\sigma^2} (\beta^2 - 2n\bar{\mu}\beta + 4n^2\bar{\mu}^2)\right) \\
&\propto \exp\left(\frac{-1}{2\sigma^2} (\beta - 2n\bar{\mu})^2\right) \\
&\sim N(2n\bar{\mu}, \sigma^2)
\end{aligned}$$

Next, we do this for μ_i .

$$\begin{aligned}
p(\mu_i | \cdot) &\propto f(\mathbf{y} | \mu_i, \tau^2) f(\mu_i | \beta, \sigma^2) \\
&\propto \Pi_j^5 \left((2\pi\tau^2)^{-1/2} \exp\left(\frac{-1}{\tau^2} (y_{i,j} - \mu_i)^2\right) \right) \left((2\pi\sigma^2)^{-1/2} \exp\left(\frac{-1}{\sigma^2} (\mu_i - \beta)^2\right) \right) \\
&\propto \exp\left(\frac{-1}{2} \left(\frac{\mu_i^2 (2\sigma^2 + \tau^2) - 2/mu_i (\sigma^2 \sum_j^5 y_{i,j} + \tau^2 \beta)}{\sigma^2 \tau^2} \right)\right) \\
&\sim N\left(\frac{\sigma^2 \sum_{j=1}^5 y_{i,j} + \tau^2 \beta}{2\sigma^2 + \tau^2}, \frac{\tau^2 \sigma^2}{2\sigma^2 + \tau^2}\right)
\end{aligned}$$

Now, we do it for τ^2 .

$$\begin{aligned}
p(\tau^2 | \cdot) &\propto f(\mathbf{y} | \mu_i, \tau^2) \pi(\tau^2) \\
&\propto \prod_i \prod_j (2\pi\tau^2)^{-1/2} \exp\left(\frac{-1}{2\tau^2} (y_{i,j} - \mu_i)^2\right) (\tau^2)^{-2-1} \exp(-2/\tau^2) \\
&\propto (\tau^2)^{-3-(20/2)} \exp\left(\frac{-1}{2\tau^2} \left(\sum_i \sum_j (y_{i,j} - \mu_i)^2 - 4\right)\right) \\
&\sim \text{Inv} - \Gamma\left(11, \frac{\sum_i \sum_j (y_{i,j} - \mu_i)^2 - 4}{2}\right)
\end{aligned}$$

And we follow the same process for σ^2 as we did for τ^2 .

$$\begin{aligned}
p(\sigma^2 | \cdot) &\propto f(\mu_i | \beta, \sigma^2) \pi(\sigma^2) \\
&\propto \prod_i^5 (2\pi\sigma^2)^{-1/2} \exp\left(\frac{-1}{2\sigma^2} (\mu_i - \beta)^2\right) (\sigma^2)^{-1-1} \exp(-1/\sigma^2) \\
&\propto (\sigma^2)^{-2-5/2} \exp\left(\frac{-1}{2\sigma^2} \left(\sum_i^5 (\mu_i - \beta)^2 - 2\right)\right) \\
&\sim \text{Inv} - \Gamma\left(4.5, \frac{\sum_i^5 (\mu_i - \beta)^2 - 2}{2}\right)
\end{aligned}$$

Now we have all the full conditional posterior distributions that we wanted. We note that there is very nice conjugacy in this model structure. Our variance parameters, σ^2 and τ^2 both have inverse gamma prior and posterior distribution. Also, our center variables, β and μ_i have normal posterior distributions, which is nice. Since we are able to derive the full conditional distributions for this model, we can implement a Gibbs Sampler to draw inference on the joint posterior distribution.

Question 4

- (a) It is very straightforward to simulate predictors from a standard normal and responses for $B_o = 0.5$ and $B_1 = 2$ given the model. We also note that this is a Bayesian Probit model, very similar to an example from class.

```

set.seed(112358)
# Set our sample size and sample from the standard normal variable
N <- 500
x <- rnorm(N)

# Create n x D design matrix, D is number of parameters
D <- 2
# We pad our X observations with a column of 1's to facilitate matrix multiplication
X <- matrix(c(rep(1, N), x), ncol = D)

# True values of regression coefficients beta
true_beta <- c(0.5, 2)

# Obtain the vector with probabilities of success p using the probit link
p <- pnorm(X %*% true_beta)

# Generate binary observation data y
y <- rbinom(N, 1, p)

```

Now that we have the simulated data, but we also need a prior distribution on β before we can implement the Gibbs Sampler. This prior distribution was not stated explicitly in the problem, but from class notes, we know a multivariate normal prior on β has nice conjugacy properties.

```
# Conjugate prior on the coefficients beta
beta_0 <- rep(0, D)
Q_0 <- diag(10, D)
```

Now, we are ready to implement our Gibbs Sampler. However, we cannot derive the full conditional posterior distributions for β_0 and β_1 yet. In class, we learned how data augmentation can be used in a situation like this to derive the full posterior conditionals. Accordingly, we create z_i , for $i = 1, \dots, n$ such that $z_i > 0$ if $y_i = 1$ and $z_i < 0$ if $y_i = 0$ and $Z \sim \text{Truncated-Normal}(x'\beta, \mathbf{1})$.

Structuring the model as above allows us to derive the full conditional distribution for β and in class we derived the following: $\beta | \cdot \sim \text{MVN}((x'x + \Sigma^{-1})^{-1}(x'z + \Sigma^{-1}\mu), (x'x + \Sigma^{-1})^{-1})$ where μ and Σ are the parameters from the normal prior distribution of β . We are now ready to code the Gibbs Sampler.

```
# initialize values for variables and matrices to store output
beta <- rep(0, D)
z <- rep(0, N)
niter <- 10000
burn_in <- 5000
# Empty matrix to store chain of betas
beta_store <- matrix(0, nrow = niter, ncol = D)

# Compute posterior variance of theta
prec_0 <- solve(Q_0)
V <- solve(prec_0 + crossprod(X, X))

for (t in 2:niter) {
  mu_z <- X %*% beta
  # Sample the augmented variables
  z[y == 0] <- rtruncnorm(N - sum(y), mean = mu_z[y == 0], sd = 1, a = -Inf, b = 0)
  z[y == 1] <- rtruncnorm(sum(y), mean = mu_z[y == 1], sd = 1, a = 0, b = Inf)
  M <- V %*% (prec_0 %*% beta_0 + crossprod(X, z))
  beta <- c(rmvnorm(1, M, V))
  beta_store[t, ] <- beta
}

post_beta <- colMeans(beta_store[-(1:burn_in), ])
```

We now return the desired 95% credible intervals for β_0 and β_1 and note that the true values are both captured in these intervals.

```
# 95% credible interval for b0
quantile(probs = c(0.025, 0.975), beta_store[-(1:burn_in), 1])
```

```
##      2.5%      97.5%
## 0.3378334 0.7061263
```

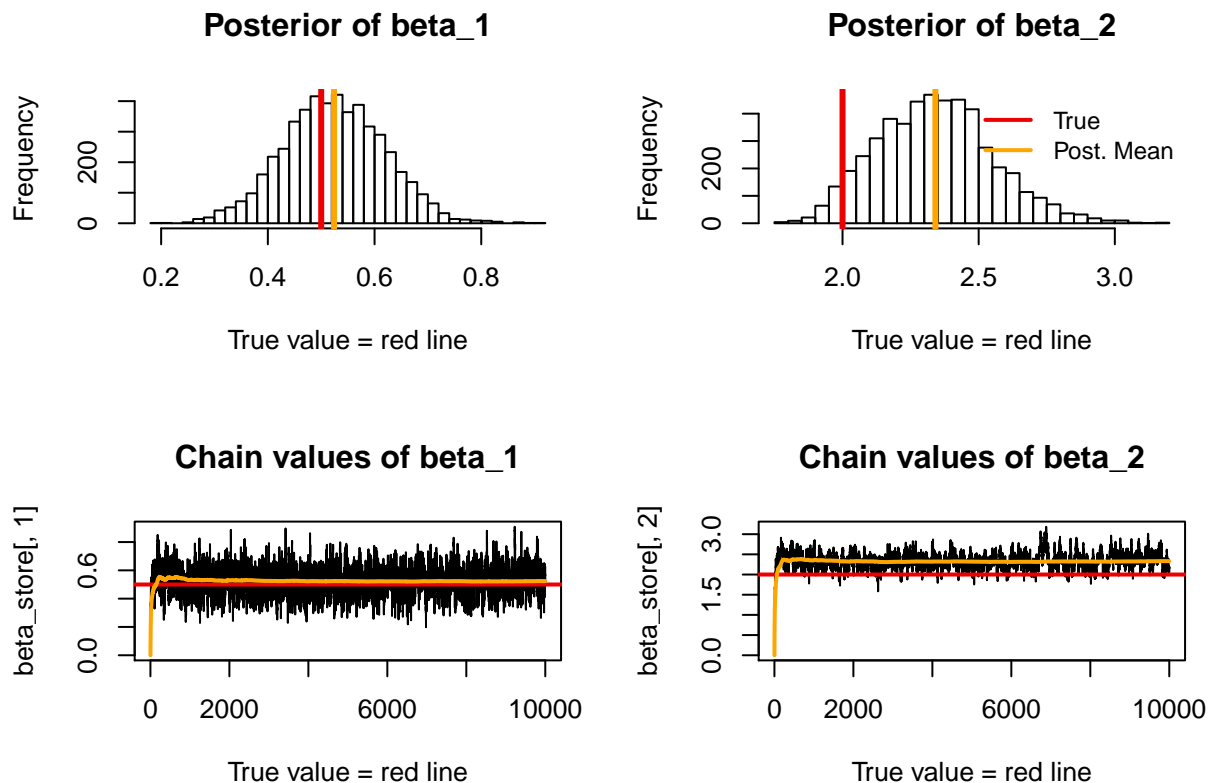
```
# 95% credible interval for b1
quantile(probs = c(0.025, 0.975), beta_store[-(1:burn_in), 2])
```

```
##      2.5%      97.5%
## 1.962259 2.787188
```

We also visualize the histogram of our posterior sample and the our parameter chain to confirm that it converges.

```
par(mfrow = c(2,2))
hist(beta_store[-(1:burn_in),1],breaks=30, main="Posterior of beta_1",
     xlab="True value = red line")
abline(v = post_beta[1], col="orange", lwd=3)
abline(v = true_beta[1], col="red2", lwd=3)
#abline(v = mle_beta[1], col="darkolivegreen", lwd=3)
hist(beta_store[-(1:burn_in),2], breaks=30, main="Posterior of beta_2",
     xlab="True value = red line")
abline(v = post_beta[2], col="orange", lwd=3)
abline(v = true_beta[2], col="red2", lwd=3)
#abline(v = mle_beta[2], col="darkolivegreen", lwd=3)
legend("topright", c("True", "Post. Mean"), lty=1, lwd=2,
     col=c("red2","orange"), bty='n', cex=.95)

plot(beta_store[, 1], type = "l", xlab="True value = red line" ,
     main = "Chain values of beta_1")
abline(h = true_beta[1], col="red2", lwd=2)
lines(cumsum(beta_store[, 1])/(1:niter), col="orange", lwd=2)
plot(beta_store[, 2], type = "l", xlab="True value = red line" ,
     main = "Chain values of beta_2")
abline(h = true_beta[2], col="red2", lwd=2)
lines(cumsum(beta_store[, 2])/(1:niter), col="orange", lwd=2)
```



Question 5 (Theoretical Question 1)

We are given that $y_i = (\Phi x_i)' \beta + \epsilon$, where ϕ is a $m \times p$ matrix with known values, $\epsilon \sim N(0, \sigma^2)$, and $\pi(\sigma^2) \propto 1/\sigma^2$.

We can look at this and see that it is a simple linear regression model, but instead of using \vec{X} as our predictors, we are using ϕx as our predictors. Essentially, we are using ϕ to filter out variables as a method of variable reduction. How ϕ is chosen is not of concern to us, but we note that it allows us to have an m -dimensional vector for β instead of a p -dimensional vector. We also note that since $\epsilon \sim N(0, \sigma^2)$, we know that $y_i \sim N((\Phi x_i)' \beta, \sigma^2)$.

Now, since we want $p(\beta|y_1, \dots, y_n)$ and $p(\sigma^2|y_1, \dots, y_n)$, we will start by deriving the full joint posterior distribution $p(\beta, \sigma^2|y_1, \dots, y_n)$ and we will integrate out β and σ^2 one at a time. First the joint posterior distribution is given by:

$$\begin{aligned} p(\beta, \sigma^2|\phi, y_1, \dots, y_n, x_1, \dots, x_n) &\propto \left(\prod^n f(y_i|\phi, \beta, \sigma^2) \right) \pi(\beta|\sigma^2) \pi(\sigma^2) \\ &\propto (\sigma^2)^{-n/2} \exp\left(\frac{-1}{2\sigma^2} \sum^n (y_i - (\Phi x_i)' \beta)^2\right) (\sigma^2)^{-1/2} \exp\left(\frac{-1}{2\sigma^2} (\beta - 0)^2\right) \left(\frac{1}{\sigma^2}\right) \\ &\propto (\sigma^2)^{-(n+3)/2} \exp\left(\frac{-1}{2\sigma^2} (\beta^2 + \sum^n (y_i - (\Phi x_i)' \beta)^2)\right) \end{aligned}$$

To get $p(\beta|y_1, \dots, y_n)$, we will integrate out σ^2 from the joint posterior, using the variable substitution: $\nu = 1/\sigma^2$

$$\begin{aligned} p(\beta|y_1, \dots, y_n) &\propto \int_{-\infty}^{\infty} p(\beta, \sigma^2|\phi, y_1, \dots, y_n) d\sigma^2 \\ &\propto \int_0^{\infty} (\sigma^2)^{-(n+3)/2} \exp\left(\frac{-1}{2\sigma^2} (\beta^2 + \sum^n (y_i - (\Phi x_i)' \beta)^2)\right) d\sigma^2 \\ &\propto \int_0^{\infty} (\nu)^{(n+3)/2} \exp\left(\frac{-\nu}{2} (\beta^2 + \sum^n (y_i - (\Phi x_i)' \beta)^2)\right) d\nu \\ &\propto \left(\frac{(\beta^2 + \sum^n (y_i - (\Phi x_i)' \beta)^2)}{2} \right)^{\frac{-(n-2)+1}{2}} \int_0^{\infty} \eta^{(n-3)/2} \exp(-\eta) d\eta \\ &\propto \left(1 + \frac{(\beta^2 + \sum^n (y_i - (\Phi x_i)' \beta)^2)}{2} \right)^{\frac{-(n-2)+1}{2}} \\ &\propto \left(1 + \frac{(\beta - \hat{\beta})^2}{2j} \right)^{\frac{-(n-2)+1}{2}} \\ &\sim t_{n-2}(\hat{\beta}, j) \end{aligned}$$

For, some constant j that can be derived through some nasty algebra. Now, to get $p(\sigma^2|y_1, \dots, y_n)$, we will integrate out β from the joint posterior. So we have,

$$\begin{aligned}
p(\sigma^2|y_1, \dots, y_n) &\propto \int_{-\infty}^{\infty} p(\beta, \sigma^2|\phi, y_1, \dots, y_n) d\beta \\
&\propto \int_{-\infty}^{\infty} (\sigma^2)^{-(n+3)/2} \exp\left(\frac{-1}{2\sigma^2}(\beta^2 + \sum_{i=1}^n (y_i - (\Phi x_i)' \beta)^2)\right) d\beta \\
&\propto (\sigma^2)^{-(n+3)/2} \int_{-\infty}^{\infty} \exp\left(\frac{-1}{2\sigma^2}(\beta^2 + \sum_{i=1}^n (y_i - (\Phi x_i)' \beta)^2)\right) d\beta \\
&\propto (\sigma^2)^{-(n+3)/2} \exp\left(\frac{-\sum_{i=1}^n (y_i^2)}{2\sigma^2}\right) \int_{-\infty}^{\infty} \exp\left(\frac{-1}{2\sigma^2}(\beta^2 - 2(\Phi x_i)' \beta \sum_{i=1}^n (y_i) - ((\Phi x_i)' \beta)^2)\right) d\beta \\
&\propto (\sigma^2)^{-(n+3)/2} \exp\left(\frac{-\sum_{i=1}^n (y_i^2)}{2\sigma^2}\right) \int_{-\infty}^{\infty} \exp\left(\frac{-1}{2\sigma^2 k}(\beta - \hat{\beta})^2\right) d\beta \\
&\propto (\sigma^2)^{-(n+3)/2} \exp\left(\frac{-\sum_{i=1}^n (y_i^2)}{2\sigma^2}\right) \\
&\sim \text{Inv} - \Gamma(n+1, \frac{-\sum_{i=1}^n (y_i^2)}{2})
\end{aligned}$$

Now, we want an expression for $f(y_1, \dots, y_n|x_1, \dots, x_n)$. This can be thought of as the joint posterior predictive distribution. To get this, we need to integrate out our parameters. We have,

$$\begin{aligned}
f(y_1, \dots, y_n|x_1, \dots, x_n) &\propto \int_{\beta} \int_{\sigma^2} f(y_1, \dots, y_n, \beta, \sigma^2|x_1, \dots, x_n) d\sigma^2 d\beta \\
&\propto \int_{\beta} \int_{\sigma^2} f(y_1, \dots, y_n|\beta, \sigma^2) \pi(\beta|x_1, \dots, x_n) \pi(\sigma^2|x_1, \dots, x_n) d\sigma^2 d\beta \\
&\propto \int_{\beta} \int_{\sigma^2} (\sigma^2)^{-n/2} \exp\left(\frac{-1}{2\sigma^2} \sum_{i=1}^n (y_i - (\Phi x_i)' \beta)^2\right) \\
&\quad * \left(1 + \frac{(\beta - \hat{\beta})^2}{2j}\right)^{\frac{-(n-2)+1}{2}} \\
&\quad * (\sigma^2)^{-(n+3)/2} \exp\left(\frac{-\sum_{i=1}^n (y_i^2)}{2\sigma^2}\right) d\sigma^2 d\beta
\end{aligned}$$

This seems incredibly hard to derive into a closed form and we should instead use a MCMC model to sample from and thus estimate this distribution.

Problem 6 (Theoretical Question 2)

We are given the following regression model: $y_i = x_i' \beta + \epsilon_i$, with $\epsilon_i \sim N(0, \sigma^2)$ and the following prior distributions: $\beta_i|\sigma^2 \sim \frac{\lambda}{2\sigma} \exp(\frac{-\lambda|\beta_i|}{\sigma})$ and $\pi(\sigma^2) \propto 1/\sigma^2$. This means that $y_i \sim N(x_i \beta, \sigma^2 \mathbf{I})$. In other words, each β_i conditioned on σ^2 has double exponential distribution / Laplace distribution prior and σ^2 has an improper, uninformative prior. Now, we want to derive the form of our joint posterior distribution.

$$\begin{aligned}
p(\beta, \sigma^2|y_1, \dots, y_n) &\propto f(y_1, \dots, y_n|\beta, \sigma^2) \pi(\beta|\sigma^2) \pi(\sigma^2) \\
&\propto (2\pi\sigma^2)^{-n/2} \exp\left(\frac{-1}{2\sigma^2} (y - x\beta)'(y - x\beta)\right) \left(\prod_{i=1}^p \frac{\lambda}{2\sigma} \exp\left(\frac{-\lambda|\beta_i|}{\sigma}\right)\right) (1/\sigma^2) \\
&\propto (\sigma^2)^{-(n+p+2)/2} \lambda^p \exp\left(\frac{-1}{2\sigma^2} ((y - x\beta)'(y - x\beta) - \sigma \sum_{i=1}^p |\beta_i|)\right) \\
&\propto (\sigma^2)^{-(n+p+2)/2} \exp\left(\frac{-1}{2\sigma^2} ((y - x\beta)'(y - x\beta) - \sigma \sum_{i=1}^p |\beta_i|)\right)
\end{aligned}$$

Now, we notice that it will be very difficult to derive full conditional distributions for β and σ^2 because of the $\sigma \sum_i^p |\beta_i|$ term in the exponentiation. If this term was not present, then it would be clear that σ^2 follows an inverse-gamma distribution and that we could drop constants and complete the square to see that β follows a normal distribution. However, we cannot ignore the $\sigma \sum_i^p |\beta_i|$ term.

To find the full conditional distributions for β and σ^2 , we need to use data augmentation. We note that the Laplace distribution can be thought of as a hierarchical normal-exponential distribution, specifically, $\frac{a}{2} \exp(-a|z|) = \int (2\pi\sigma)^{-1/2} \exp(\frac{-z^2}{2\sigma^2}) \frac{a^2}{2} \exp(\frac{-a^2 s}{2}) ds$. In this case, we have $\beta|\sigma^2, \gamma_i^2 \sim N(0, \sigma^2 \gamma_i^2)$ and γ_i^2 follows an exponential $\lambda^2/2$ distribution. Now, we substitute this into our joint posterior distribution and the full conditional distributions can be derived.

$$\begin{aligned} p(\beta, \sigma^2 | y_i, \dots, y_n) &\propto (2\pi\sigma^2)^{-n/2} \exp\left(\frac{-1}{2\sigma^2} (y - x\beta)'(y - x\beta)\right) \left[\prod_j^p f(\beta|\sigma^2, \gamma_i^2) \pi(\gamma_i) \right] \pi(\sigma^2) \\ &\propto (2\pi\sigma^2)^{-n/2} \exp\left(\frac{-1}{2\sigma^2} (y - x\beta)'(y - x\beta)\right) (1/\sigma^2) \left[\prod_j^p (2\pi\sigma^2 \gamma_i^2)^{-1/2} \exp\left(\frac{\beta_j^2}{2\sigma^2 \gamma_i^2}\right) \frac{\lambda^2}{2} \exp(-\lambda^2 \gamma_i^2 / 2) \right] \end{aligned}$$

Now, we are able to derive our full conditionals. Let D be a p -dimensional diagonal matrix with $D_{i,i} = \gamma_i$ and let $M = X'X + D^{-1}$. Now, we derive the full conditional distribution for β

$$\begin{aligned} p(\beta | \cdot) &\propto (2\pi\sigma^2)^{-n/2} \exp\left(\frac{-1}{2\sigma^2} (y - x\beta)'(y - x\beta)\right) (1/\sigma^2) \left[\prod_j^p (2\pi\sigma^2 \gamma_i^2)^{-1/2} \exp\left(\frac{\beta_j^2}{2\sigma^2 \gamma_i^2}\right) \frac{\lambda^2}{2} \exp(-\lambda^2 \gamma_i^2 / 2) \right] \\ &\propto \exp\left(\frac{-1}{2\sigma^2} \left((y - x\beta)'(y - x\beta) + \beta' D^{-1} \beta \right) \right) \\ &\propto \exp\left(\frac{-1}{2\sigma^2} \left(y'y - 2y'X\beta + \beta'M\beta \right) \right) \\ &\propto \exp\left(\frac{-1}{2\sigma^2} \left((\beta - M^{-1}X'Y)'M(\beta - M^{-1}X'Y) + Y'(\mathbf{I} - XM^{-1}X') \right) \right) \\ &\propto \exp\left(\frac{-1}{2\sigma^2} \left((\beta - M^{-1}X'Y)'M(\beta - M^{-1}X'Y) \right) \right) \\ &\sim MVN(M^{-1}X'Y, \sigma^2 M^{-1}) \end{aligned}$$

Now, we derive the full conditional distribution for σ^2 .

$$\begin{aligned} p(\sigma^2 | \cdot) &\propto (\sigma^2)^{-n/2} (\sigma^2)^{-1} (\sigma^2)^{-p/2} \exp\left(\frac{-1}{2\sigma^2} \left((y - x\beta)'(y - x\beta) + \beta' D^{-1} \beta \right) \right) \\ &\propto (\sigma^2)^{-(n+p+2)/2} \exp\left(\frac{-1}{2\sigma^2} \left((y - x\beta)'(y - x\beta) + \beta' D^{-1} \beta \right) \right) \\ &\sim \text{Inv} - \Gamma\left(\frac{n+p}{2}, \left((y - x\beta)'(y - x\beta) + \beta' D^{-1} \beta \right) / 2\right) \end{aligned}$$

Finally, we derive the full conditional distribution for γ_i^2 . Defining $\eta^2 = 1/\gamma_i^2$ will make things easier.

$$\begin{aligned} p(\gamma_i^2) &\propto (\gamma_i^2)^{-1/2} \exp\left(\frac{\beta_i^2}{2\sigma^2 \gamma_i^2}\right) \exp\left(\frac{-\lambda^2 \gamma_i^2}{2}\right) \\ &\propto (\eta_i^2)^{-3/2} \exp\left(\frac{-1}{2} \left(\frac{\beta_i^2 \eta_i^2}{\sigma^2} + \frac{\lambda^2}{\eta_i^2} \right) \right) \\ &\propto (\eta_i^2)^{-3/2} \exp\left(\frac{-1}{2} \left(\frac{\beta_i^2 (\eta_i^2 - (\sigma\lambda/\beta_i))}{2\sigma^2 \eta_i^2} \right) \right) \\ &\sim \text{Inv} - \text{Normal}(\lambda\sigma/\beta_i, \lambda^2) \end{aligned}$$