Assignment #6

Elliot Smith

10/1/2018

Problem 1

Part i

$p(\beta|y, \lambda_1, \lambda_2, k) \propto p(y|\lambda_1, \lambda_2, k, \beta) \times p(\lambda_1|y, \lambda_2, k, \beta) \times p(\lambda_2|y, \lambda_1, k, \beta) \times p(k|y, \lambda_1, \lambda_2, \beta) \times p(\beta)$

Since $p(y|\lambda_1, \lambda_2, k, \beta)$ and $p(k|y, \lambda_1, \lambda_2, \beta)$ don't depend on β, we may ignore them and our equation becomes:

$p(\beta|y, \lambda_1, \lambda_2, k) \propto p(\lambda_1|y, \lambda_2, k, \beta) \times p(\lambda_2|y, \lambda_1, k, \beta) \times p(\beta)$

Where:

$$\begin{aligned} p(\lambda_1|y, \lambda_2, k, \beta) &= p(y|\lambda_1, \lambda_2, k)p(\lambda_1) \\ &\sim Gamma\left(\alpha + \sum_{t=1}^{k} y_t, \beta + k\right) \end{aligned}$$

$$\begin{aligned} p(\lambda_2|y, \lambda_1, k, \beta) &= p(y|\lambda_1, \lambda_2, k)p(\lambda_2) \\ &\sim Gamma\left(\alpha + \sum_{t=k+1}^{n} y_t, \beta + (n-k)\right) \end{aligned}$$

So:

$$\begin{aligned} p(\beta|y, \lambda_1, \lambda_2, k) &\propto p(\lambda_1|y, \lambda_2, k, \beta) \times p(\lambda_2|y, \lambda_1, k, \beta) \times p(\beta) \\ &\propto \beta^{\alpha} \lambda_1^{\alpha-1+\sum_{t=1}^{k} y_t} e^{-(\beta+k)\lambda_1} \beta^{\alpha} \lambda_2^{\alpha-1+\sum_{t=k+1}^{n} y_t} e^{-(\beta+(n-k))\lambda_2} \beta^{a_\beta - 1} e^{-b_\beta \beta} \\ &\propto \beta^{2\alpha} e^{-(\beta+k)\lambda_1 - (\beta+(n-k))\lambda_2} \beta^{a_\beta - 1} e^{-b_\beta \beta} \\ &\propto \beta^{2\alpha + a_\beta - 1} e^{-\beta\lambda_1 - \beta\lambda_2 - b_\beta \beta} \\ &\propto \beta^{2\alpha + a_\beta - 1} e^{-(\lambda_1 + \lambda_2 + b_\beta)\beta} \\ &\sim Gamma(2\alpha + a_\beta, \lambda_1 + \lambda_2 + b_\beta) \end{aligned}$$

Part ii

Please refer to the Code Appendix section for the implementation of the Gibbs sampler.

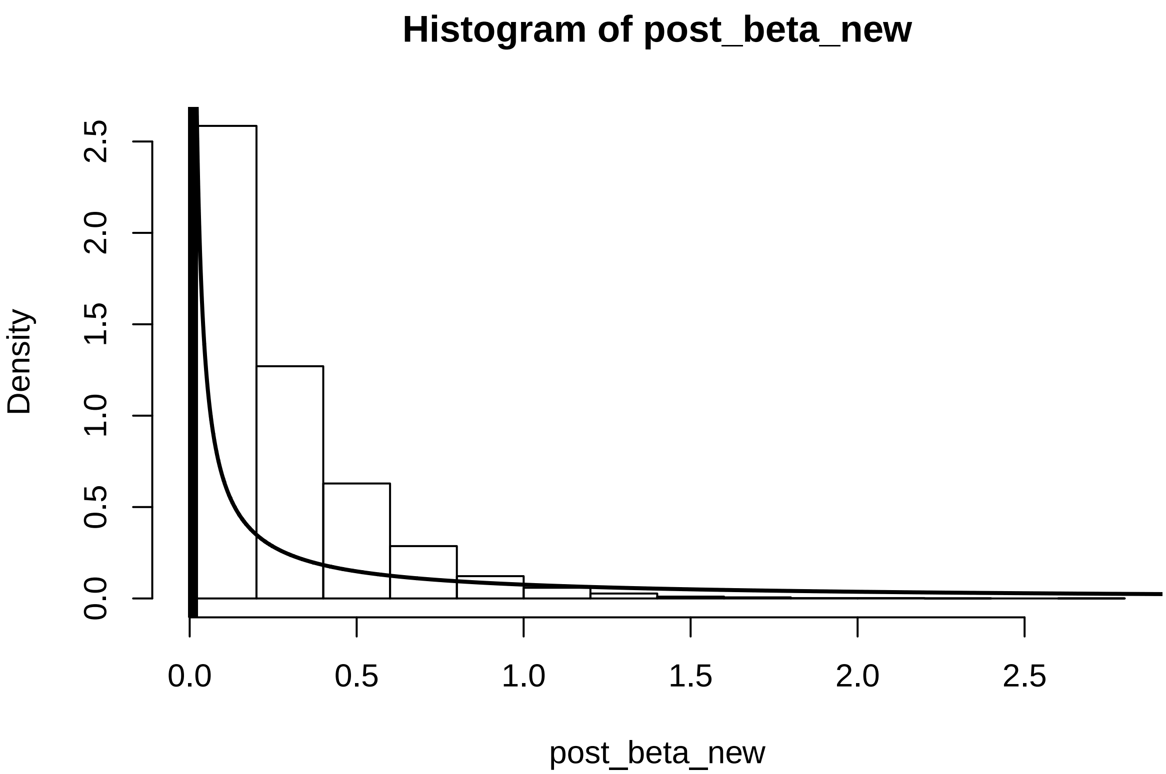
Part iii

```
## HPD Interval Lambda_1 with Beta = 0.01:  ( 2.579477 , 3.700989 )
## HPD Interval Lambda_2 with Beta = 0.01:  ( 0.7055055 , 1.160724 )
## HPD Interval k + 1850 with Beta = 0.01:  ( 1885 , 1894 )
```

```
## HPD Interval Lambda_1 with Beta Hyperprior:  ( 2.53931 , 3.689825 )
## HPD Interval Lambda_2 with Beta Hyperprior:  ( 0.6980847 , 1.155413 )
## HPD Interval k + 1850 with Beta Hyperprior:  ( 1885 , 1894 )
```

There does not appear to be substantial difference between the posterior credible intervals of the two models.

Part iv



No. According to the results and the density representation there is not evidence to suggest the the posterior distribution $[\beta|y]$ differs substantially from our previous choice of $\beta = 0.01$

Problem 2

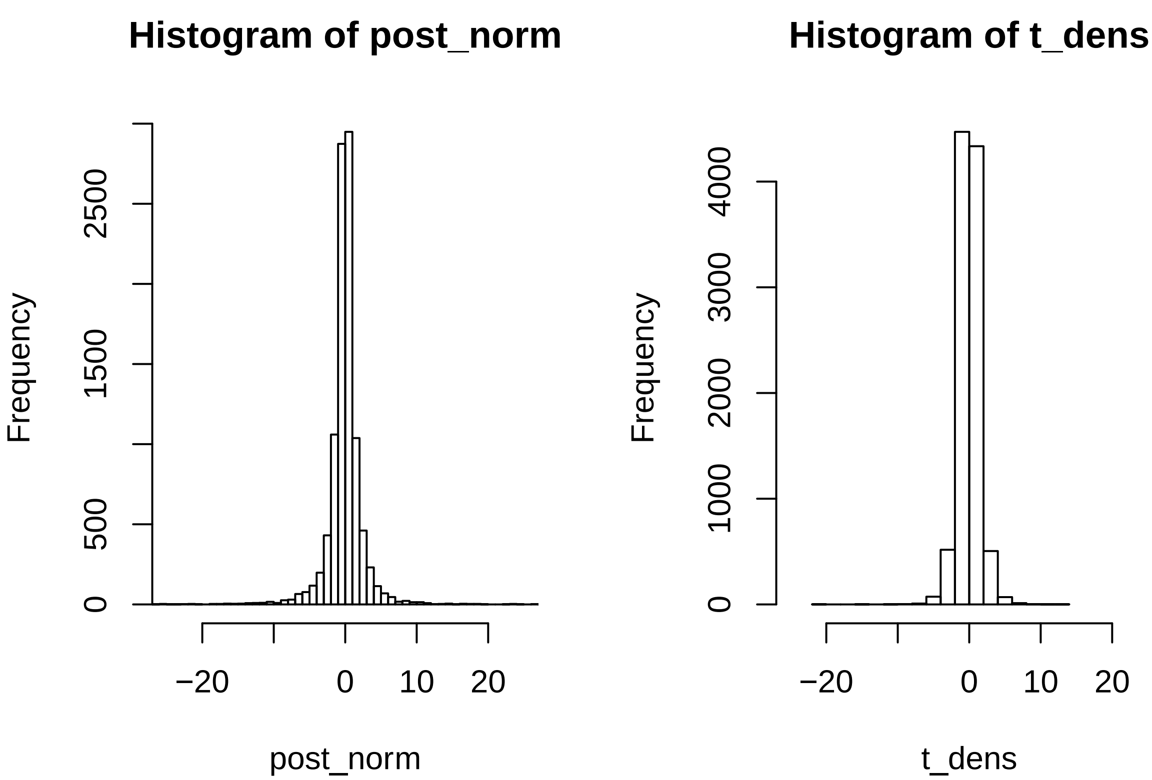
Part i

$$
\begin{aligned}
p(y|\mu,\alpha,\beta) &= \int \frac{\beta^\alpha}{\Gamma(\alpha)}\tau^{\alpha-1}e^{-\beta\tau}\left(\frac{\tau}{2\pi}\right)^{\frac{1}{2}} e^{-\frac{\tau}{2}(y-\mu)^2}d\tau \\
&= \frac{\beta^\alpha}{\Gamma(\alpha)}\frac{1}{(2\pi)^{\frac{1}{2}}}\int \tau^{\alpha-\frac{1}{2}}e^{-\beta\tau}e^{-\frac{\tau}{2}(y-\mu)^2}d\tau \\
&= \frac{\beta^\alpha}{\Gamma(\alpha)}\frac{1}{(2\pi)^{\frac{1}{2}}}\int \tau^{\alpha-\frac{1}{2}}e^{-(\beta+\frac{1}{2}(y-\mu)^2)\tau}d\tau \\
&= \frac{\beta^\alpha}{\Gamma(\alpha)}\frac{1}{(2\pi)^{\frac{1}{2}}}\frac{\Gamma(\alpha+\frac{1}{2})}{(\beta+\frac{1}{2}(y-\mu)^2)^{\alpha+\frac{1}{2}}} \\
&= \frac{\Gamma(\alpha+\frac{1}{2})}{\Gamma(\alpha)}\frac{1}{(2\pi\beta)^{\frac{1}{2}}}\frac{1}{(1+\frac{1}{2\beta}(y-\mu)^2)^{\alpha+\frac{1}{2}}}
\end{aligned}
$$

And now substituting in for z_i and our Gamma parameters for α and β we get the following result:

$p(z_i) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\pi\nu}\Gamma(\frac{\nu}{2})}\frac{1}{\left(1+\frac{z_i^2}{\nu}\right)^{\frac{\nu+1}{2}}}$

Part ii



As we can see from our result, we get similar density when comparing our results from Part i with the $t_\nu(0,1)$ distribution.

Part iii

Yes, this result does make sense, because whenever the variance of a normally distributed random variable is unknown and a conjugate prior placed over it that follows a Gamma distribution, the resulting marginal distribution of the variable will follow a Student's t-distribution.

Problem 3

Part i

$$
\begin{aligned}
p(\theta|y, \tau_y, \xi_i) &\propto p(\theta)p(y|\theta, \tau_y, \xi_i) \\
&\propto 1 \times \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\pi\nu}\sigma\Gamma(\frac{\nu}{2})} \frac{1}{\left(1 + \frac{1}{\nu}(\frac{y_i-\mu}{\sigma})^2\right)^{\frac{\nu+1}{2}}}
\end{aligned}
$$

$$
\begin{aligned}
p(\tau_y|y, \theta, \xi_i) &\propto p(\tau_y)p(y|\theta, \tau_y, \xi_i) \\
&\propto \frac{\beta^\alpha}{\Gamma(\alpha)} \tau_y^{0.01-1} e^{-0.01\tau_y} \times \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\pi\nu}\sigma\Gamma(\frac{\nu}{2})} \frac{1}{\left(1 + \frac{1}{\nu}(\frac{y_i-\mu}{\sigma})^2\right)^{\frac{\nu+1}{2}}}
\end{aligned}
$$

$$
\begin{aligned}
p(\xi_i|y, \theta, \tau_y) &\propto p(\xi_i)p(y|\theta, \tau_y, \xi_i) \\
&\propto \frac{\beta^\alpha}{\Gamma(\alpha)} \xi_i^{2-1} e^{-2\xi_i} \times \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\pi\nu}\sigma\Gamma(\frac{\nu}{2})} \frac{1}{\left(1 + \frac{1}{\nu}(\frac{y_i-\mu}{\sigma})^2\right)^{\frac{\nu+1}{2}}}
\end{aligned}
$$

I was not sure how to do these calculations, so I stopped at this point.

Part ii

For Part ii, until the end of Problem 3, I randomly chose distributions to use. Since I was unable to get the correct result from Part i, I wanted to display that I knew how to perform the rest of the calculations in an effort to get some partial credit.

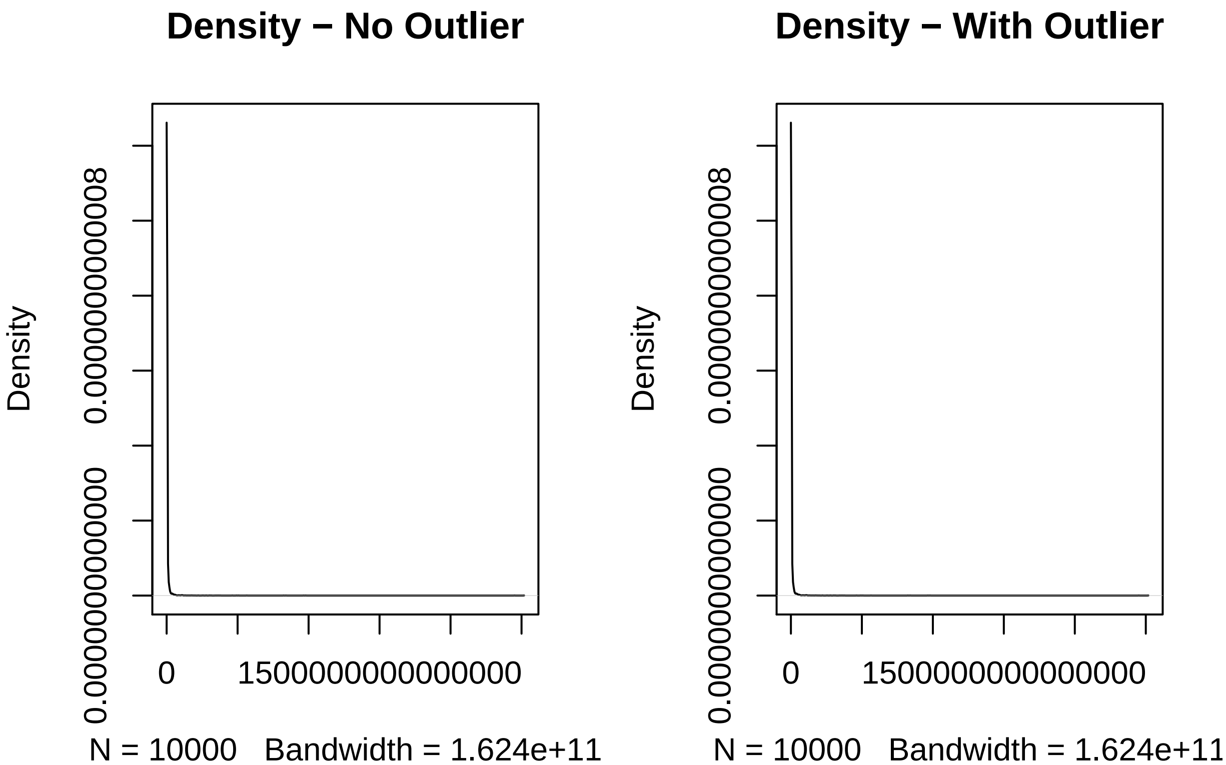
Please see the Code Appendix section for the sampling method.

Part iii

```
## HPD Interval:  ( 575832694102 , 11682933879912 )
```

As expected, my interval is very far off from the output from the previous Assignment with the Gaussian errors, due to the fact that I completely randomized the distributions used to calculate the parameters.

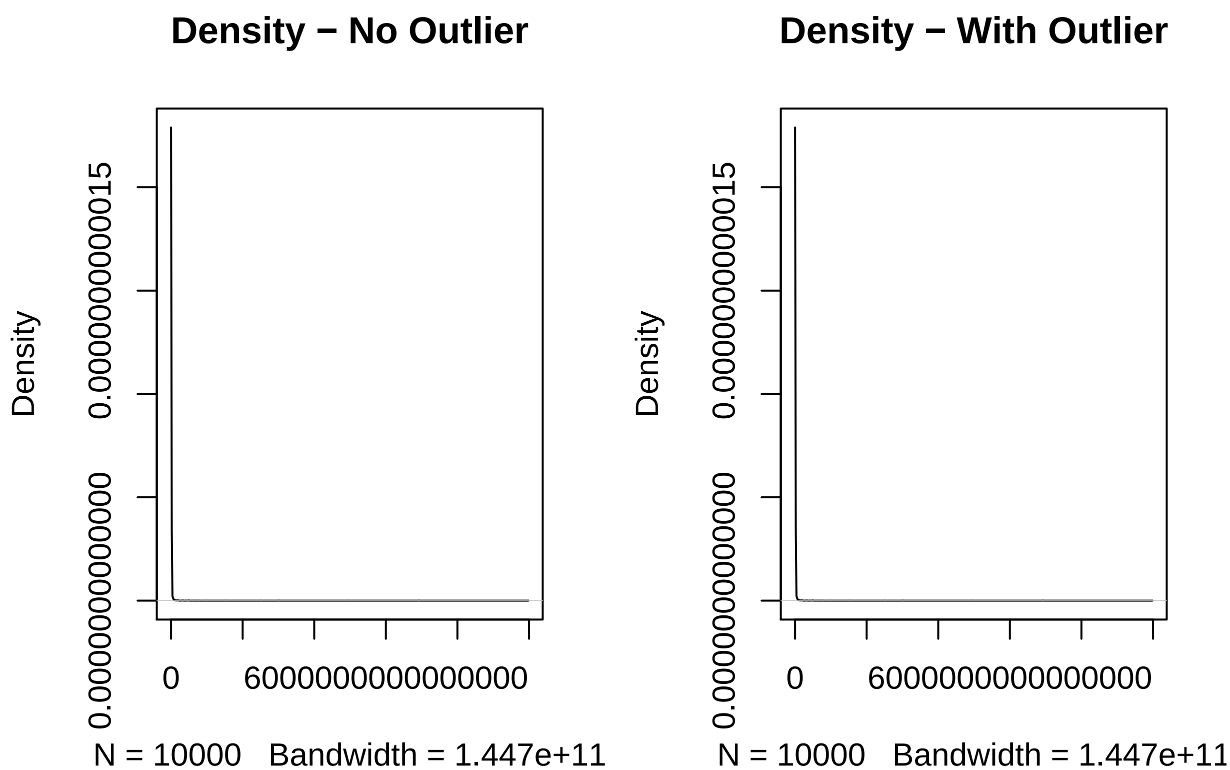
Part iv



Again its impossible for me to tell whether or not the outlier made a substantial change since my distributions from Part i are wrong.

Part v

```
## HPD Interval:  ( 526043092375 , 10335974695510 )
```



Again, due to my distributions being incorrect, it is impossible for me to tell if and by how much the results compare.

Code Appendix

```
########## Workspace Prep ##########



## Turn scientific notation off

options(scipen = 999)

## Load in the necessary packages

library(coda)
library(gamair)



########## Problem 1 ##########



##### Part i #####
```

```r
##### Part ii #####



### Unchanged code from Lecture



year = (1851:1962)
y = 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)

n = length(y)

# Hyperparameters:
alpha = 0.5; beta = 0.01

# Pick some initial values:
k = ceiling(n/2) # midpoint is CP
lambda_1 = mean(y[1:k])
lambda_2 = mean(y[(k+1):n])

# Sample from the posterior distribution:
S = 10^4

# Storage:
post_lambda_1 = array(0, c(S, 1))
post_lambda_2 = array(0, c(S, 1))
post_k = array(0, c(S, 1));

for(s in 1:S){
    # Sample lambda_1:
    lambda_1 = rgamma(n = 1,
                      shape = alpha + sum(y[1:k]),
                      rate = beta + k)
    # Sample lambda_2:
    # NOTE: assuming k < n throughout!
    lambda_2 = rgamma(n = 1,
                      shape = alpha + sum(y[(k+1):n]),
                      rate = beta + (n-k))
    # Sample k:
    log_g = cumsum(y)*log(lambda_1/lambda_2) + (1:n)*(lambda_2 - lambda_1)
    #pm<-exp(log_g)/sum(exp(log_g)); k<-min((1:n)[runif(1)<cumsum(pm)])
    k = sample(1:n, 1, prob = exp(log_g))

    # Store:
    post_lambda_1[s] = lambda_1
    post_lambda_2[s] = lambda_2
    post_k[s] = k
```

```r
}

### Edited code from lecture, factors in beta prior

# Hyperparameters:
alpha = 0.5; beta = 0.01

## Set the beta params

b_alpha <- 0.1
b_beta <- 0.1

# Pick some initial values:
k = ceiling(n/2) # midpoint is CP
lambda_1 = mean(y[1:k])
lambda_2 = mean(y[(k+1):n])

# Sample from the posterior distribution:
S = 10^4

# Storage:
post_lambda_1_new = array(0, c(S, 1))
post_lambda_2_new = array(0, c(S, 1))
post_k_new = array(0, c(S, 1))
post_beta_new = array(0, c(S, 1))

for(s in 1:S){

    # Sample lambda_1:
    lambda_1 = rgamma(n = 1,
                      shape = alpha + sum(y[1:k]),
                      rate = beta + k)
    # Sample lambda_2:
    # NOTE: assuming k < n throughout!
    lambda_2 = rgamma(n = 1,
                      shape = alpha + sum(y[(k+1):n]),
                      rate = beta + (n-k))
    # Sample k:
    log_g = cumsum(y)*log(lambda_1/lambda_2) + (1:n)*(lambda_2 - lambda_1)
    #pm<-exp(log_g)/sum(exp(log_g)); k<-min((1:n)[runif(1)<cumsum(pm)])
    k = sample(1:n, 1, prob = exp(log_g))

    # Sample beta:

    beta <- rgamma(n = 1,
                   shape = (2 * alpha) + b_alpha,
                   rate = lambda_1 + lambda_2 + b_beta)

    # Store:
```

```
    post_lambda_1_new[s] = lambda_1
    post_lambda_2_new[s] = lambda_2
    post_k_new[s] = k
    post_beta_new[s] = beta
}



##### Part iii #####



## Compute the HPD intervals for beta = 0.01

ci_lambda_1 <- as.vector(HPDinterval(as.mcmc(post_lambda_1), prob = 0.95)[1, 1:2])
ci_lambda_2 <- as.vector(HPDinterval(as.mcmc(post_lambda_2), prob = 0.95)[1, 1:2])
ci_k = as.vector(HPDinterval(as.mcmc(post_k + 1850), prob = 0.95)[1, 1:2])

cat("HPD Interval Lambda_1 with Beta = 0.01: ", "(", ci_lambda_1[1], ",", ci_lambda_1[2], ")")
cat("HPD Interval Lambda_2 with Beta = 0.01: ", "(", ci_lambda_2[1], ",", ci_lambda_2[2], ")")
cat("HPD Interval k + 1850 with Beta = 0.01: ", "(", ci_k[1], ",", ci_k[2], ")")

## Compute the HPD intervals for beta hyperprior

ci_lambda_1_new <- as.vector(HPDinterval(as.mcmc(post_lambda_1_new), prob = 0.95)[1, 1:2])
ci_lambda_2_new <- as.vector(HPDinterval(as.mcmc(post_lambda_2_new), prob = 0.95)[1, 1:2])
ci_k_new = as.vector(HPDinterval(as.mcmc(post_k_new + 1850), prob = 0.95)[1, 1:2])

cat("HPD Interval Lambda_1 with Beta Hyperprior: ", "(", ci_lambda_1_new[1], ",", ci_lambda_1_new[2], "
cat("HPD Interval Lambda_2 with Beta Hyperprior: ", "(", ci_lambda_2_new[1], ",", ci_lambda_2_new[2], "
cat("HPD Interval k + 1850 with Beta Hyperprior: ", "(", ci_k_new[1], ",", ci_k_new[2], ")")



##### Part iv #####



par(mfrow = c(1, 1))

hist(post_beta_new, freq = FALSE)
x <- rgamma(n = S, shape = b_alpha, rate = b_beta)
lines(sort(x), y = dgamma(sort(x), shape = b_alpha, rate = b_beta), lwd = 2)
abline(v = 0.01, lwd = 5)



########## Problem 2 ##########



##### Part i #####
```

```r
##### Part ii #####



nu <- 4
S <- 10000

post_gamma <- rgamma(n = S, shape = (nu / 2), rate = (nu / 2))

post_norm <- rnorm(n = S, mean = 0, sd = 1 / post_gamma)

t_dens <- rt(n = S, df = nu)

par(mfrow = c(1, 2))

hist(post_norm, xlim = c(-25, 25), breaks = 200)
hist(t_dens, xlim = c(-25, 25), breaks = 20)



##### Part iii #####



########## Problem 3 ##########



## Load in the data

data(hubble)



##### Part i #####



##### Part ii #####



## Perform the Gibbs sampler

S <- 10000

post_xi <- array(0, c(S, 1))
post_tau <- array(0, c(S, 1))
post_theta <- array(0, c(S, 1))

for (s in 1:S) {
```

```r
    xi <- rt(n = 1,
             df = 4)

    tau <- rgamma(n = 1,
                  shape = 10,
                  rate = 10 + post_xi)

    theta <- runif(n = 1,
                   min = 0,
                   max = 1 + post_tau)

    post_xi[s] <- xi
    post_tau[s] <- tau
    post_theta[s] <- theta

}



##### Part iii #####



## Generate the HPD interval

post_hub <- (1 / post_theta) * 3.09e19/(60^2*24*365)

ci_post_hub <- round(as.vector(HPDinterval(as.mcmc(post_hub), prob = 0.95)[1, 1:2]), 8)

cat("HPD Interval: ", "(", ci_post_hub[1], ",", ci_post_hub[2], ")")



##### Part iv #####



## Set the plots side-by-side

par(mfrow = c(1, 2))

## Plot the density

plot(density(post_hub), main = "Density - No Outlier")

## Now create an outlier and re-run the Gibbs sampler and plot again for comparison

hubble[11,2] <- 1

S <- 10000

post_xi <- array(0, c(S, 1))
post_tau <- array(0, c(S, 1))
```

```r
post_theta <- array(0, c(S, 1))

for (s in 1:S) {

    xi <- rt(n = 1,
             df = 4)

    tau <- rgamma(n = 1,
                  shape = 10,
                  rate = 10 + post_xi)

    theta <- runif(n = 1,
                   min = 0,
                   max = 1 + post_tau)

    post_xi[s] <- xi
    post_tau[s] <- tau
    post_theta[s] <- theta

}

## Plot the density

plot(density(post_hub), main = "Density - With Outlier")



##### Part v #####



## Fix the outlier for the hubble data

hubble[11,2] <- 1

## Perform the Gibbs sampler

S <- 10000

post_xi <- array(0, c(S, 1))
post_tau <- array(0, c(S, 1))
post_theta <- array(0, c(S, 1))

for (s in 1:S) {

    xi <- rt(n = 1,
             df = 400)

    tau <- rgamma(n = 1,
                  shape = 10,
                  rate = 10 + post_xi)

    theta <- runif(n = 1,
```

```r
                   min = 0,
                   max = 1 + post_tau)

    post_xi[s] <- xi
    post_tau[s] <- tau
    post_theta[s] <- theta

}

## Generate the HPD interval

post_hub <- (1 / post_theta) * 3.09e19/(60^2*24*365)

ci_post_hub <- round(as.vector(HPDinterval(as.mcmc(post_hub), prob = 0.95)[1, 1:2]), 8)

cat("HPD Interval: ", "(", ci_post_hub[1], ",", ci_post_hub[2], ")")

## Set the plots side-by-side

par(mfrow = c(1, 2))

## Plot the density

plot(density(post_hub), main = "Density - No Outlier")

## Now create an outlier and re-run the Gibbs sampler and plot again for comparison

hubble[11,2] <- 1

S <- 10000

post_xi <- array(0, c(S, 1))
post_tau <- array(0, c(S, 1))
post_theta <- array(0, c(S, 1))

for (s in 1:S) {

    xi <- rt(n = 1,
             df = 400)

    tau <- rgamma(n = 1,
                  shape = 10,
                  rate = 10 + post_xi)

    theta <- runif(n = 1,
                   min = 0,
                   max = 1 + post_tau)

    post_xi[s] <- xi
    post_tau[s] <- tau
    post_theta[s] <- theta

}
```

```r
## Plot the density

plot(density(post_hub), main = "Density - With Outlier")
```