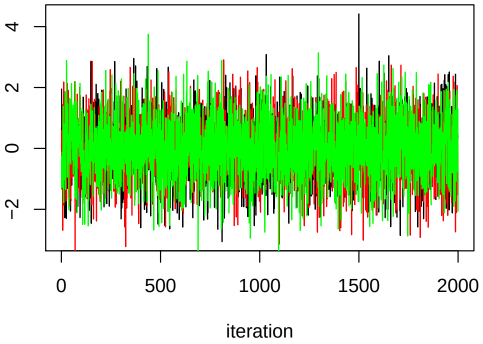
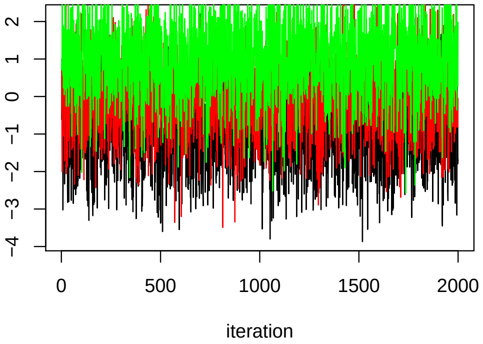
# Bayes: Homework 8
Leslie Gains-Germain

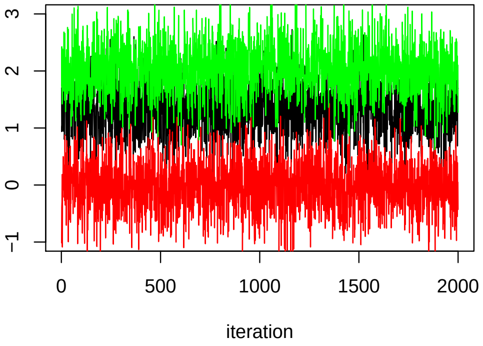
1. (a) i. Scenario 1: three chains from $N(0,1)$ white noise. I actually ran 2000 iterations in each chain so that I could use Raftery and Lewis's diagnostic in part (e).



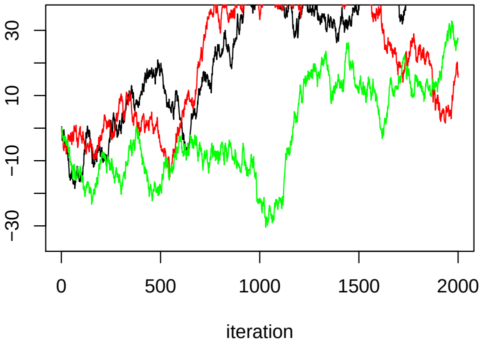
ii. Scenario 2: one chain from $N(-1,1)$, one chain from $N(0,1)$, and one chain from $N(1,1)$.



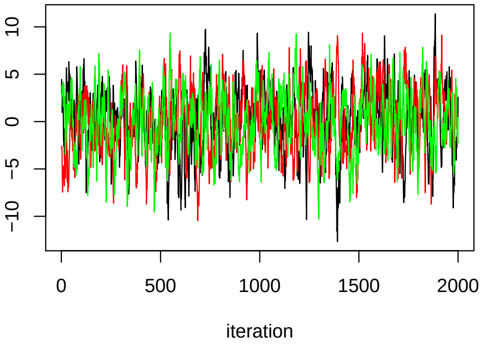
iii. Scenario 3: three from $MVN(0,\Sigma)$ with common $\sigma^2 = 1$ and $\rho = 0.8$.



iv. Scenario 4: three from non-stationary correlated chains using `diffinv(rnorm(999))`.



v. Scenario 5: three from stationary chains with correlation using `filter(rnorm(1000), filter=rep(1, 10), circular=TRUE)`.



(b) The effective sample size is calculated differently in the newest version of the `coda` package than described in the textbook. In the textbook, the effective sample size is dividing the number of iterations (over all chains) by a formula that measures autocorrelation, so that chains with higher autocorrelation have lower effective sample sizes. In the newest version of the `coda` package, however, the number of iterations is divided by the spectral density. Details are on page 286 of the BDA3 text and the Effective Sample size section of the convergence diagnostic document.

In the coda package, $\hat{R}$ is calculated in the same way as the described in Chapter 11 in the BDA3 textbook.

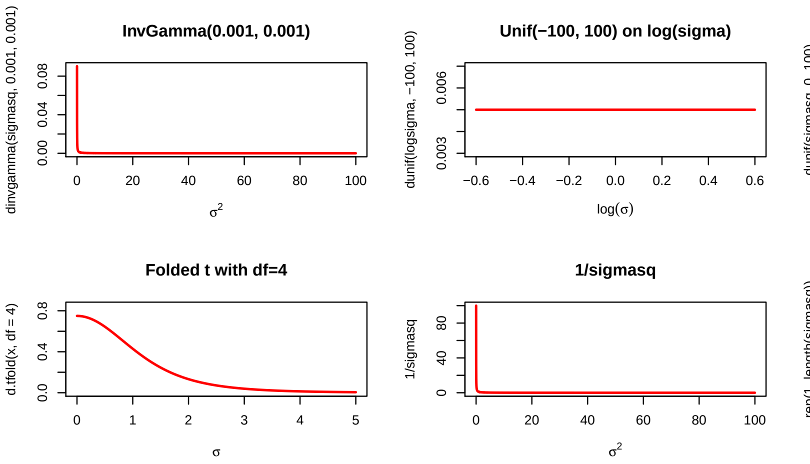
(c) The below table shows the $\hat{R}$ and effective sample sizes for each scenario.

| scenario | neff | rhat |
|---|---:|---:|
| 1 | 5674.27 | 1.00 |
| 2 | 6291.69 | 1.74 |
| 3 | 5526.62 | 3.61 |
| 4 | 7.34 | 2.22 |
| 5 | 540.03 | 1.00 |

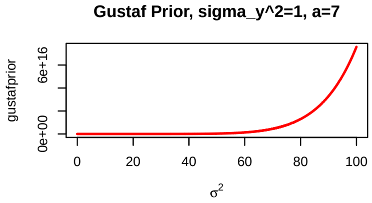
(d) The table below shows the z-statistics for all three chains from Geweke's diagnostic. The last three columns show the results from Raftery and Lewis's diagnostic (burn in and iterations required as well as the dependence factor).

| scenario | zstats1 | zstats2 | zstats3 | burn.in | n.iter | dependence.factor |
|---|---:|---:|---:|---:|---:|---:|
| 1 | 0.00 | -1.87 | -0.60 | 2.00 | 3660.00 | 0.98 |
| 2 | -0.17 | -1.17 | 0.12 | 3.00 | 4255.00 | 1.14 |
| 3 | -2.01 | 0.16 | -0.74 | 3.00 | 4197.00 | 1.12 |
| 4 | -11.17 | -3.51 | -2.35 | 198.00 | 215594.00 | 57.60 |
| 5 | 0.15 | -2.19 | -0.09 | 20.00 | 21264.00 | 5.68 |

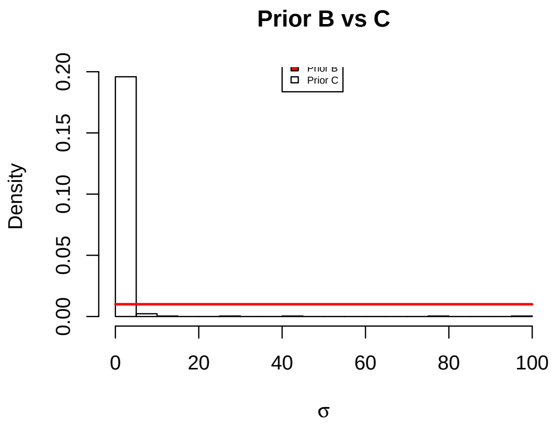
2. (a) $\sigma_\alpha^2$ is $Inv - Gamma(0.001, 0.001)$ which is the same as a scaled inverse chi squared distribution with parameters $\nu = 0.002$ and $s^2 = 1$.



(b) The plots of all the priors are shown below.

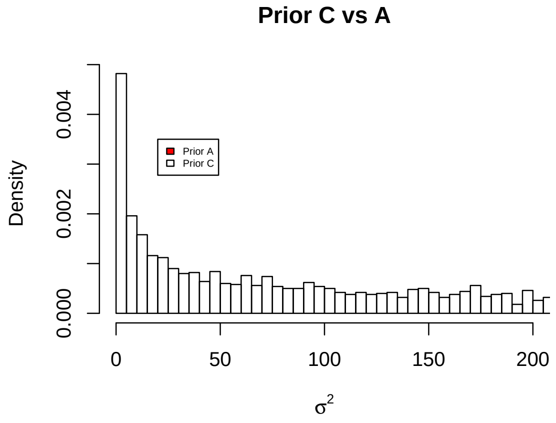


(c) Next, I took random draws from prior B and I exponentiated these draws to compare to prior C. Clearly, a prior of $log(\sigma)$ is very different from a uniform prior on $\sigma$. The plot is hard to see, but you can tell that after drawing from a $Unif(-100, 100)$ prior on $log(\sigma)$, you get many very large values for $\sigma$ (not shown), and you get many very very small values for $\sigma$. This is why the $log(\sigma)$ prior blows up at zero. Priors B and C generate very different values for $\sigma$. It makes sense why Gelman recommends a uniform prior on the standard deviation rather than a uniform prior on the log standard deviation.



(d) I took 10000 random draws from a $Unif(0,100)$ distribution of $\sigma_\theta$, and then I exponentiated these draws to compare to the Inverse Gamma prior on $\sigma_\theta^2$. It actually looks like the $Uniform(0,100)$ prior on $\sigma_\theta$ is similar in shape to a $Inv-Gamma(0.001,0.001)$ prior on $\sigma_\theta^2$, but the uniform prior on the standard deviation looks better because although there are many $\sigma_\theta^2$ values near 0, the prior does not blow up at zero like the Inverse Gamma prior does.

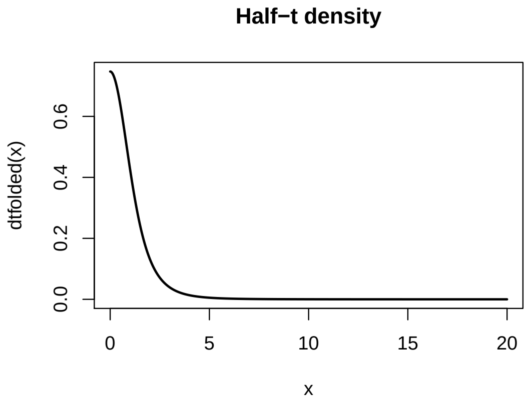
```
## Error in xy.coords(x, y):  could not find function "dinvgamma"
```



(e) A half-Cauchy distribution is a half t-distribution with 1 df.

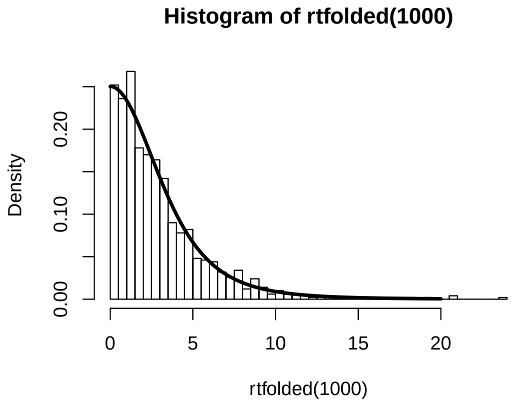
3. (a) i. The `dtfolded` function below calculates the density of a folded half t distribution with 4 degrees of freedom and a scale of 1.

```r
# function to calculate density of a folded non-central t-distribution
dtfolded <- function (x, df=4, A=1) {
  p <- (1+1/df*(x/A)^2)^(-(df+1)/2)
  num.int <- sum(p*(x[2]-x[1]))
  dt <- p/num.int
  return(dt)
}

x <- seq(0, 20, by=0.01)

plot(x, dtfolded(x), type="l", main="Half-t density", lwd=2)
```



ii. The `rtfolded` function below generates a random draw from the folded t distribution with 4 degrees of freedom and scale parameter of 3.

```r
rtfolded <- function(nsim, df=4, mu=0, sigma=1, A=3){
  z <- abs(rnorm(nsim, mu, sigma))
  x <- rgamma(nsim, df/2, (1/2*A^2))
  theta <- mu+sigma*z*sqrt(df/x)
  return(theta)
}

set.seed(31)
hist(rtfolded(1000), freq=FALSE, nclass=40)
lines(x, dtfolded(x, A=3), lwd=3)
```



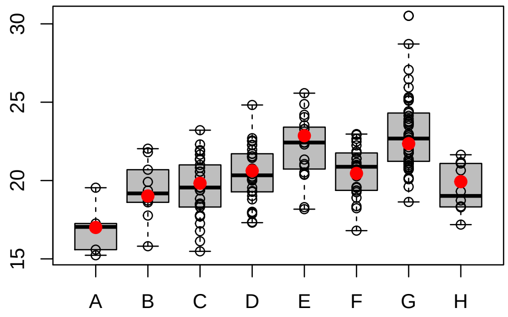
(b) The basic hierarchical model is:

$$y_{ij} \sim N(\mu + \alpha_j, \sigma_y^2)$$
$$\alpha_j \sim N(0, \sigma_\alpha^2)$$

(c) I used the code below to simulate $y_{ij}$. I first simulated the group effects, and then I simulated individual effects within each group. The code is adapted from your code in `SchoolsFakeData_CompareMethods_F13.R`. The boxplots showing the simulated data are shown below.

```
 n.j <- c(5, 10, 30, 30, 20, 25, 50, 10)
 n.tot <- sum(n.j)

 J <- 8
 sigma.y <- 2

 mu <- 20
 sigma.alpha <- sqrt(2)

alpha <- numeric(J)
for(j in 1:J){
  alpha[j] <- rnorm(1, 0, sd=sigma.alpha)
}

 set.seed(2567)
 y.ij <- numeric(n.tot)
 school <- rep(NA,n.tot)
 n.index <- c(0, cumsum(n.j)) +1
 for (j in 1:J) {
   y.ij[n.index[j]:(n.index[j+1]-1)] <- rnorm(n.j[j], mean=mu+alpha[j], sd=sigma.y)
   school[n.index[j]:(n.index[j+1]-1)] <- rep(LETTERS[j], n.j[j])
 }
 school <- factor(school)
```

```
### Plot the raw fake data ####

boxplot(y.ij ~ school, col="gray", var.width=TRUE)

  points(school, y.ij)

  points(1:8, mu+alpha, col="red", cex=2, pch=20)
```



(d) Priors:

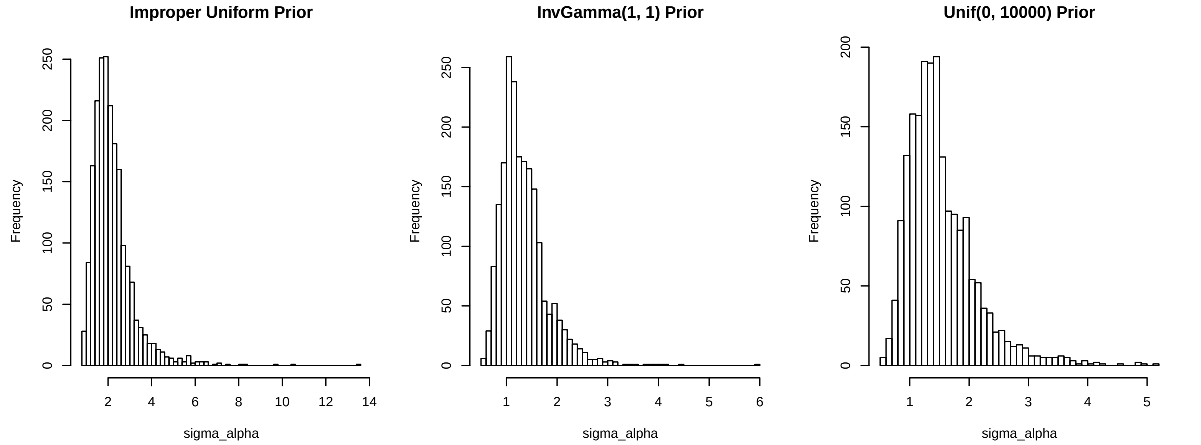
$$\mu \propto 1$$

(e) Fit the model in STAN

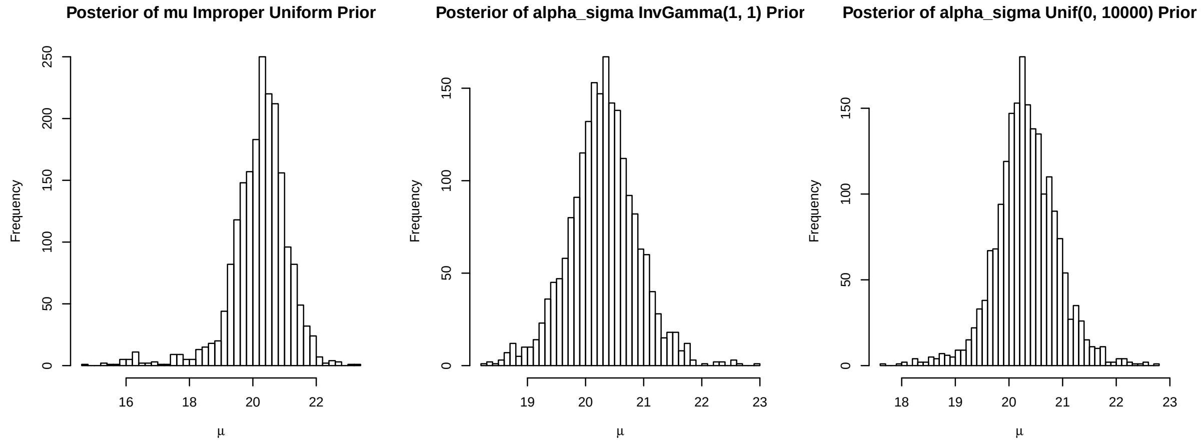
```
## The following numerical problems occured the indicated number of times on chain 1

## If this warning occurs sporadically, such as for highly constrained variable types like

covariance matrices, then the sampler is fine,but if this warning occurs often then your

model may be either severely ill-conditioned or misspecified.

## The following numerical problems occured the indicated number of times on chain 2

## If this warning occurs sporadically, such as for highly constrained variable types like

covariance matrices, then the sampler is fine,but if this warning occurs often then your

model may be either severely ill-conditioned or misspecified.
```

```
## The following numerical problems occured the indicated number of times on chain 1
## If this warning occurs sporadically, such as for highly constrained variable types like
covariance matrices, then the sampler is fine,but if this warning occurs often then your
model may be either severely ill-conditioned or misspecified.
```

(f)



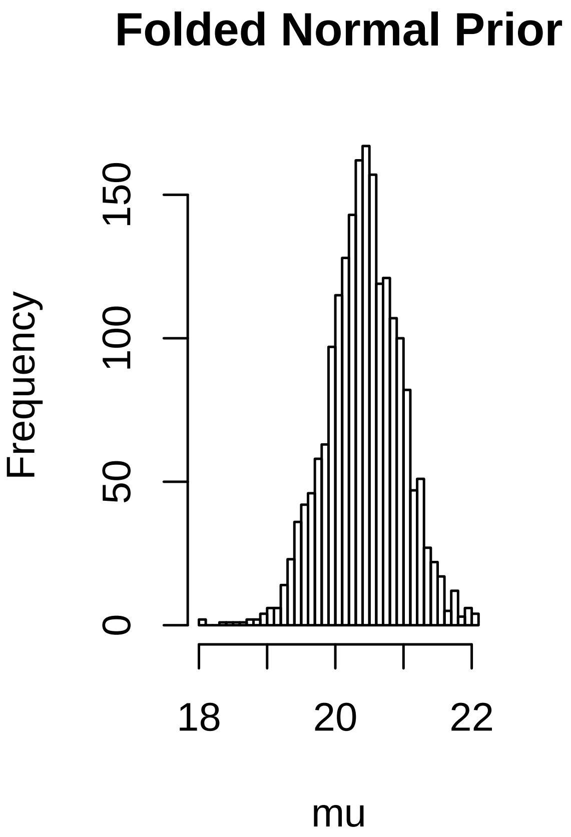
(g)



(h)

```
## The following numerical problems occured the indicated number of times on chain 1
## If this warning occurs sporadically, such as for highly constrained variable types like
covariance matrices, then the sampler is fine,but if this warning occurs often then your
model may be either severely ill-conditioned or misspecified.
```

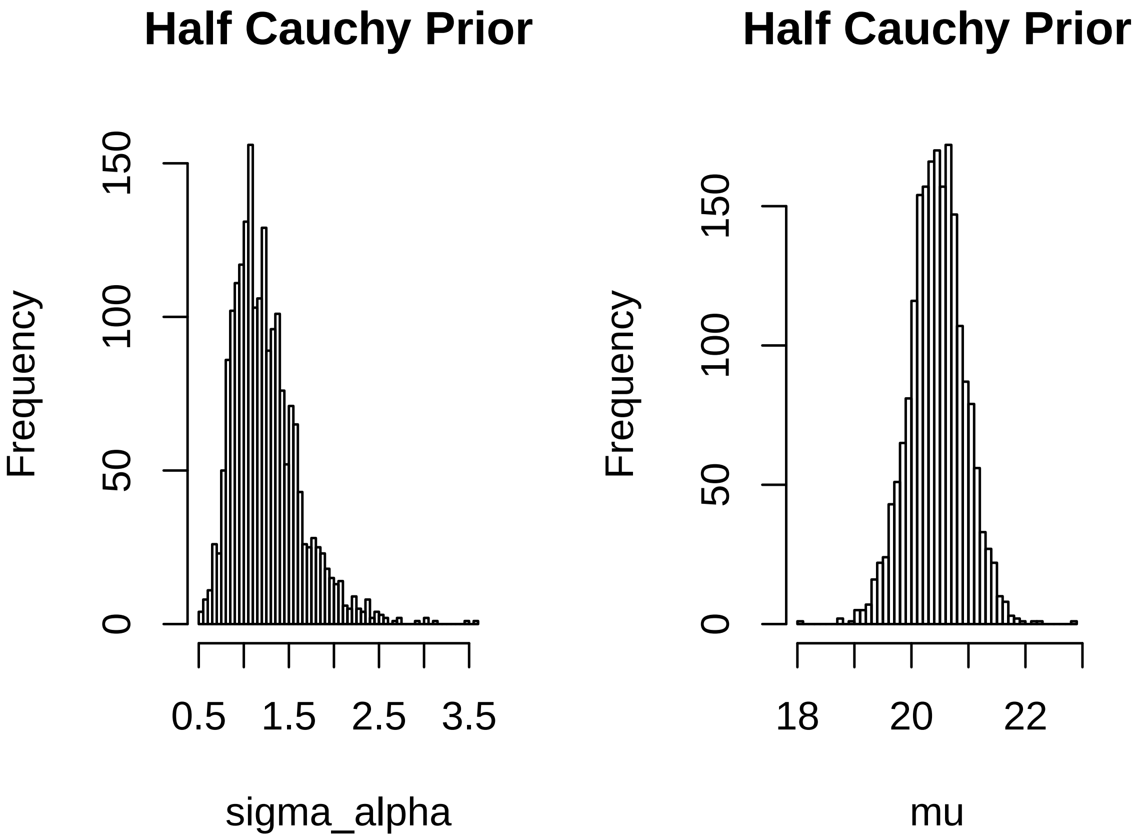
```
## Error in hist(extract(reparam.fit)$psi * extract(reparam.fit)$sigma_eta, :  dims [product
2000] do not match the length of object [0]
```



(i)

```
## The following numerical problems occured the indicated number of times on chain 1
## If this warning occurs sporadically, such as for highly constrained variable types like
covariance matrices, then the sampler is fine,but if this warning occurs often then your
model may be either severely ill-conditioned or misspecified.
## The following numerical problems occured the indicated number of times on chain 2
## If this warning occurs sporadically, such as for highly constrained variable types like
covariance matrices, then the sampler is fine,but if this warning occurs often then your
model may be either severely ill-conditioned or misspecified.
## The following numerical problems occured the indicated number of times on chain 3
## If this warning occurs sporadically, such as for highly constrained variable types like
covariance matrices, then the sampler is fine,but if this warning occurs often then your
model may be either severely ill-conditioned or misspecified.
## The following numerical problems occured the indicated number of times on chain 4
## If this warning occurs sporadically, such as for highly constrained variable types like
```

```
covariance matrices, then the sampler is fine,but if this warning occurs often then your
model may be either severely ill-conditioned or misspecified.
```


Half Cauchy Prior
Frequency
sigma_alpha
Half Cauchy Prior
Frequency
mu