538 Quiz 2

Lennox Garay

2023-11-17

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(egg)
```

```
## Warning: package 'egg' was built under R version 4.3.2
```

```
## Loading required package: gridExtra
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

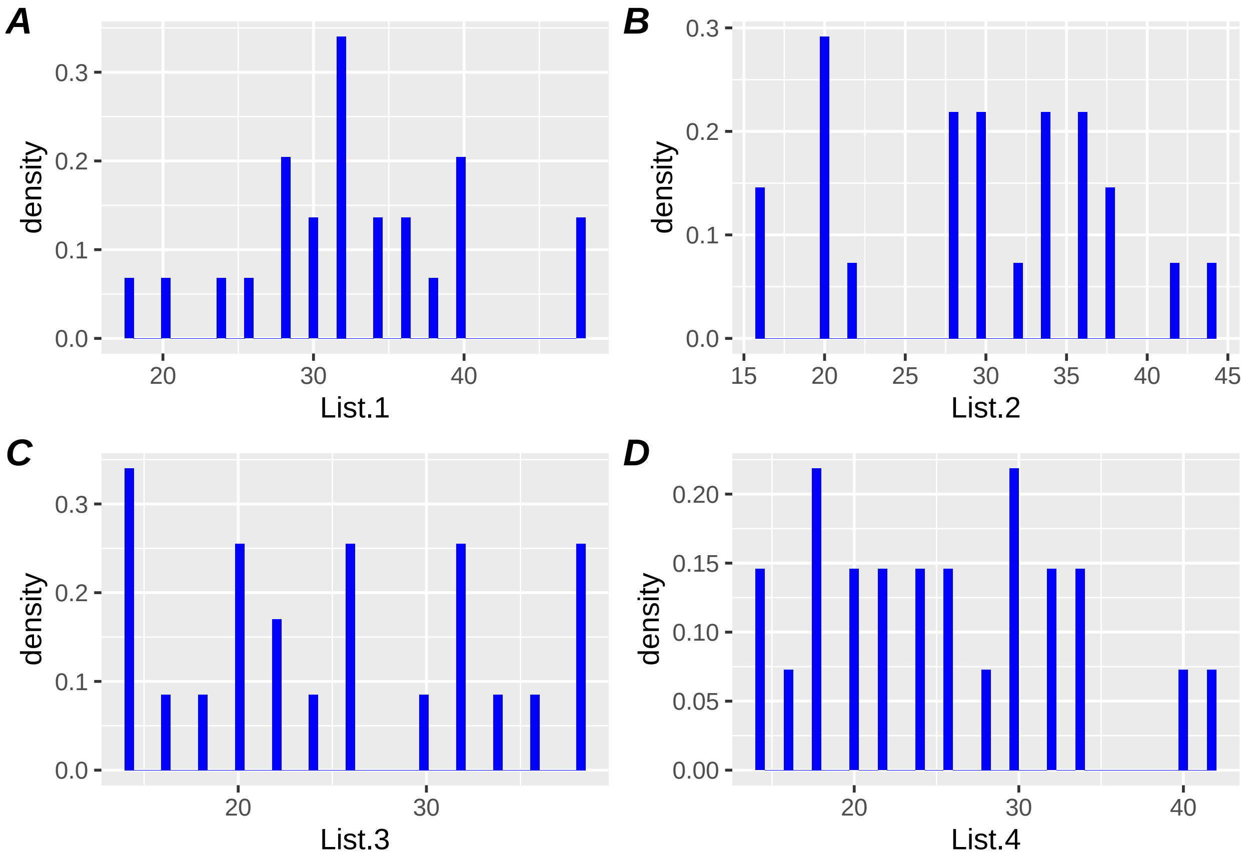
```
library(mvtnorm)
library(invgamma)
```

##a The data

```
data = read.csv('hearing.txt', sep='\t')
df = data.frame(data)
attach(df)
```

```
l1 = ggplot(df, aes(x=List.1)) +
  geom_histogram(aes(x=List.1, y= after_stat(density)), fill='blue', bins = 50)

l2 = ggplot(df, aes(x=List.2)) +
  geom_histogram(aes(x=List.2, y= after_stat(density)), fill='blue', bins = 50)

l3 = ggplot(df, aes(x=List.3)) +
  geom_histogram(aes(x=List.3, y= after_stat(density)), fill='blue', bins = 50)

l4 = ggplot(df, aes(x=List.4)) +
  geom_histogram(aes(x=List.4, y= after_stat(density)), fill='blue', bins = 50)
```

```r
ggarrange(l1,l2,l3,l4, labels = c("A","B","C","D"), nrow = 2)
```




```r
summary(df)
```

```
##      List.1          List.2          List.3          List.4     
##  Min.   :18.00   Min.   :16.00   Min.   :14.00   Min.   :14.00  
##  1st Qu.:28.00   1st Qu.:21.50   1st Qu.:19.50   1st Qu.:19.50  
##  Median :32.00   Median :30.00   Median :25.00   Median :25.00  
##  Mean   :32.75   Mean   :29.67   Mean   :25.25   Mean   :25.58  
##  3rd Qu.:36.50   3rd Qu.:36.00   3rd Qu.:32.00   3rd Qu.:30.50  
##  Max.   :48.00   Max.   :44.00   Max.   :38.00   Max.   :42.00
```

```r
ok = numeric()
for(i in 1:ncol(data)){
  ok = sd(data[,i])
  print(ok)
}
```

```
## [1] 7.408867
## [1] 8.057762
## [1] 8.315779
## [1] 7.779106
```

We get a list standard deviation of 7.40, 8.05, 8.31, 7.7 for lists 1,2,3,and 4, respectively. Lists 1 and 2 have much higher means than lists 3 and 4. I find this difference between lists interesting. Perhaps there is a list effect. I think its hard to say if theres a student effect with just this data.

#b-d:

So, I actually tried to derive all this by hand and it took me hours and got me nowhere. So, the forms for my posteriors are copied from the lecture notes. I don't have any work to "show."

At any rate, our models is as follows:

$y_{ij}|\theta_j, \sigma^2 \sim N(\theta_j, \sigma^2)$
$\theta_j|\mu, \sigma^2 \sim N(\mu, \sigma^2)$
$\mu \sim N(30, 1)$
$\sigma^2 \sim \Gamma^{-1}(2, 10)$
$f(y_{ij}|\theta, \sigma^2) \sim \prod_j \prod_i \frac{1}{\sqrt{2\pi\sigma^2}} \exp[-\frac{1}{2}(\frac{(y_{ij}-\theta_j)^2}{\sigma^2}]$
$f(\theta, \mu, \sigma^2|y) = f(y_{ij}|\theta_j, \sigma^2) * f(\theta_j|\mu, \sigma^2) * f(\mu) * f(\sigma^2)$
$= (\sigma^2)^{\frac{-n}{2}} \exp(-\frac{1}{\sigma^2}(\sum_j \sum_i \frac{(y_{ij}-\theta_j)^2}{2})) * (\sigma^2)^{\frac{-n}{2}} \exp(-\frac{1}{\sigma^2} \sum_j \frac{(\theta_j-\mu)^2}{2}) * \exp(-\frac{1}{2}(\mu-30)^2) * (\sigma^2)^{-2-1} \exp(-\frac{10}{\sigma^2})$
$f(\theta_j|\mu, \sigma^2, y) \sim N(\frac{\frac{\bar{y}}{\sigma_0^2} + \frac{\mu}{\sigma^2}}{\frac{1}{\sigma_0^2} + \frac{1}{\sigma^2}}, [\frac{1}{\sigma_0^2} + \frac{1}{\sigma^2}]^{-1})$
$f(\mu|\theta, \sigma^2, y) \sim N(\frac{\frac{J\bar{\theta}}{\sigma^2} + \frac{\mu_0}{\sigma_0^2}}{\frac{J}{\sigma^2} + \frac{1}{\sigma^2}}, [\frac{J}{\sigma^2} + \frac{1}{\sigma^2}]^{-1})$
$f(\sigma^2|\theta, \mu, y) \sim \Gamma^{-1}(\frac{2+J}{2}, 1 + \sum_j^J \frac{(\theta_j-\mu)^2}{2})$

Upon simplifying this would be an Inverse gamma on sigma^2, but this is a joint posterior conditioned on y with no known distribution.

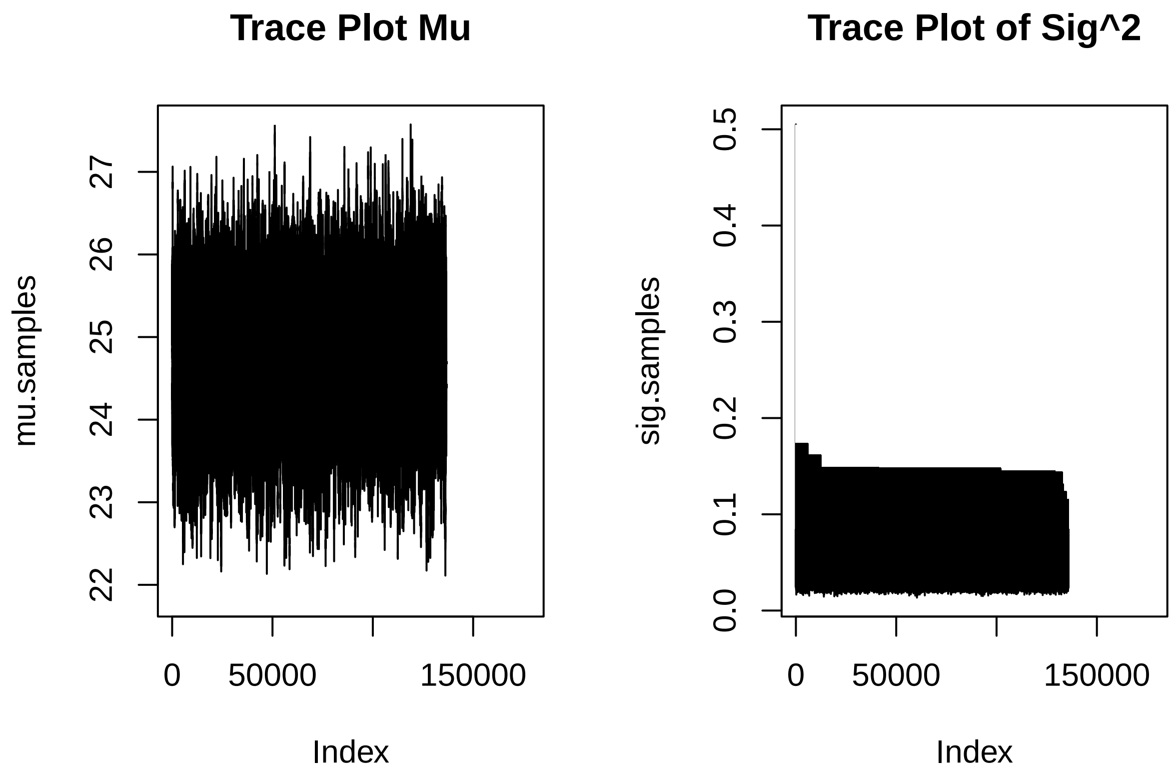
f:

```
#prelim functions

theta.post = function(mu, y.bar, tau2, sig0=1){ ##sig0 = sig^2_h, tau = sig^2 (density)
  n = length(y.bar)
  mean.num = (y.bar/sig0) + (mu/tau2)
  mean.denom = (1/sig0) + (1/tau2)
  mean1 = mean.num/mean.denom
  sig = 1/((1/sig0) + (1/tau2))
  dn = rnorm(n, mean1, sqrt(sig))
  return(dn)
}

mu.post = function(J=24, theta, mu0=30, sig0=1, tau2){
  theta.bar = mean(theta)
  n = length(theta.bar)
  mean.num = (J*theta.bar/tau2) + (mu0/sig0)
  mean.denom = (J/tau2) + (1/sig0)
  mean1 = mean.num / mean.denom
  sig = 1/((J/tau2) + (1/sig0))
  dn = rnorm(n, mean = mean1, sd = sqrt(sig))
  return(dn)
}

tau.post = function(J, theta, mu){ ## n=1 because we are vecotrizing the sum.
  p1 = (2+J)/2
  p2 = (1 + (sum((theta-mu)^2)))/2
  dn = rinvgamma(1, p1, p2)
}
```

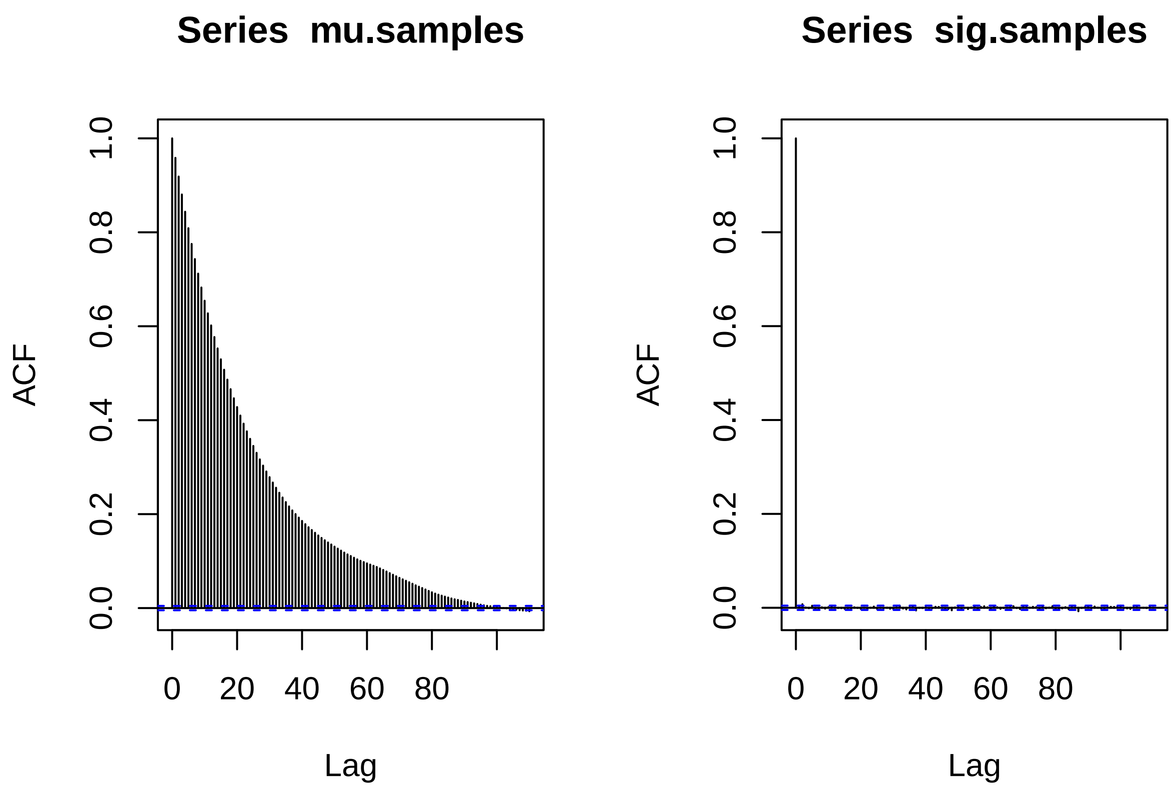
```
set.seed(538)
y.bar = apply(data, 1, mean) ## applying here so i dont have to do it in the for loop.
```

```r
mu0 = 30
sig0 = 1
n = 4
sig.dn = apply(data,1, sd)^2 #sig0
B = 178000
J = length(y.bar)

theta.samples = matrix(NA, nrow = B+1, ncol = J) #each row will be a sample
mu.samples = numeric() # mu density
sig.samples = numeric() ## sig^2 density, tau for our fn

theta.samples[1,] = y.bar
sig.samples[1] = rinvgamma(1, shape = 1, rate = 1)

for(i in 2:(B+1)) {

  #mu|others
  mu.samples[i-1] <- mu.post(J = J,
                             theta = as.numeric(theta.samples[i-1]),
                             mu0 = mu0, sig0 = sig0, tau2 = sig.samples[i-1])

  #theta | others
  theta.samples[i,] <- theta.post(mu = mu.samples[i-1], y.bar = y.bar, tau2 = sig.samples[i-1], sig0 =

  #sigma.sq|others
  sig.samples[i] <- tau.post(J=24, theta = theta.samples[i-1], mu = mu.samples[i-1])

}
```

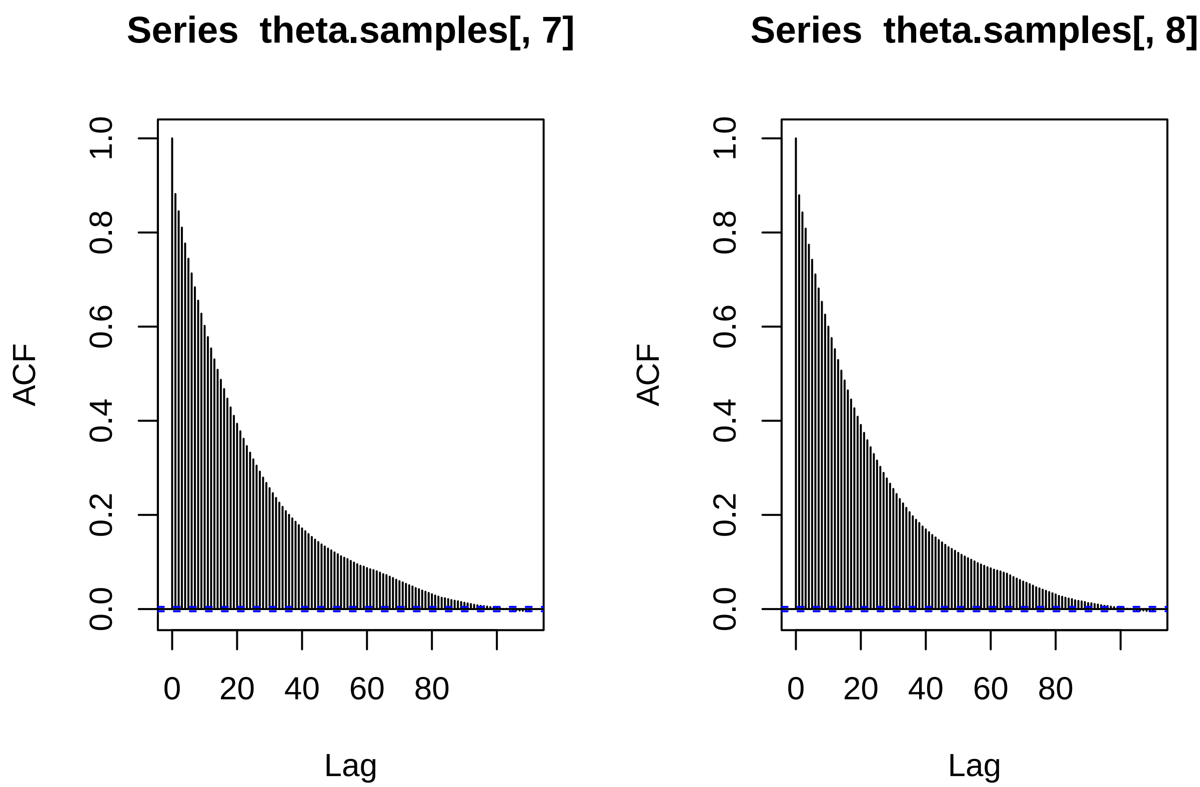
```r
par(mfrow = c(1,2))
plot(mu.samples, type = 'l', main = 'Trace Plot Mu')
plot(sig.samples, type = 'l', main = 'Trace Plot of Sig^2')
```



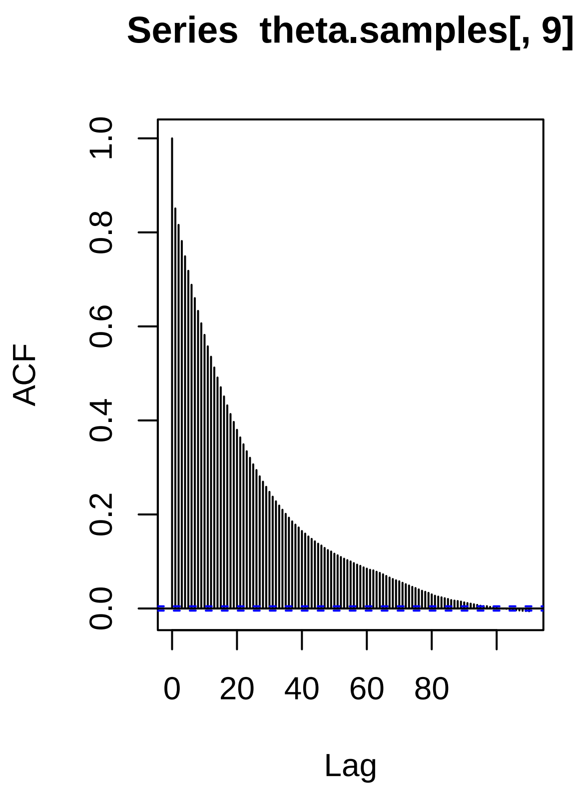

```r
acf(mu.samples, lag.max = 110)
acf(sig.samples, lag.max = 110)
```




```
acf(theta.samples[,7], lag.max = 110)
acf(theta.samples[,8], lag.max = 110)
```

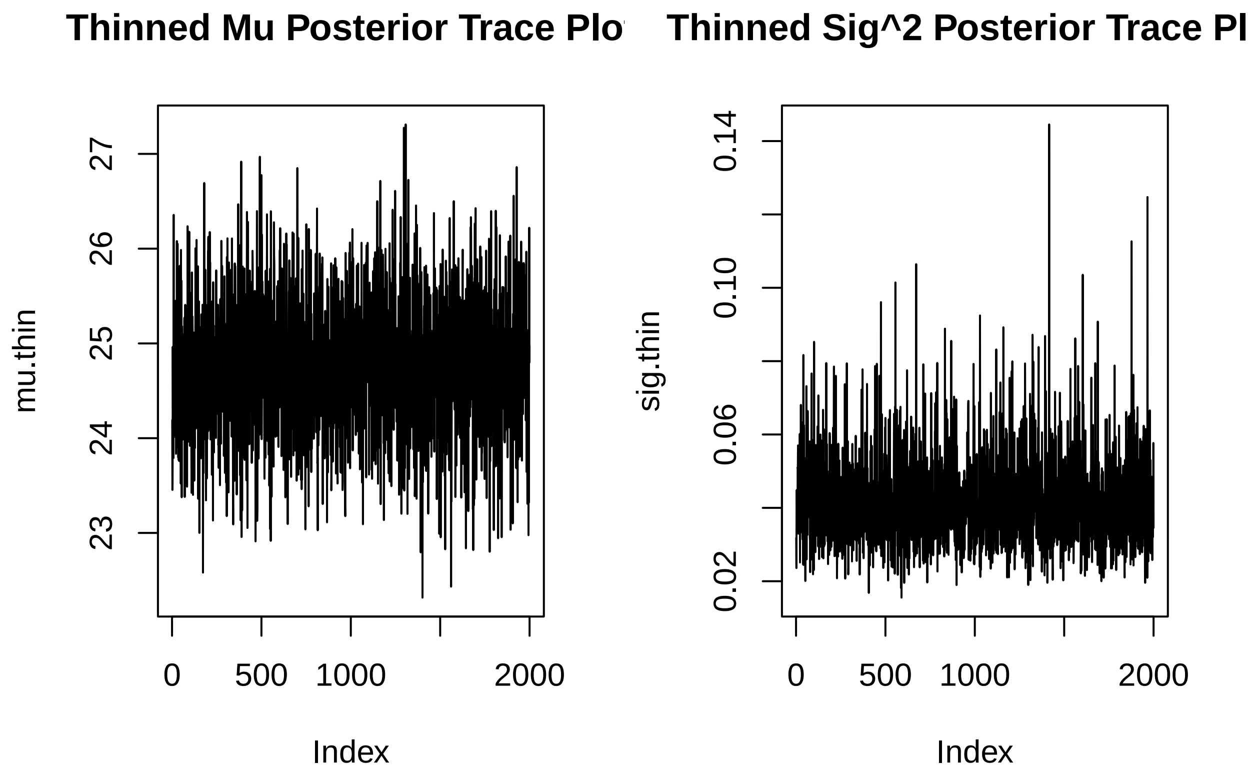



```
acf(theta.samples[,9], lag.max = 110) ## 84
```

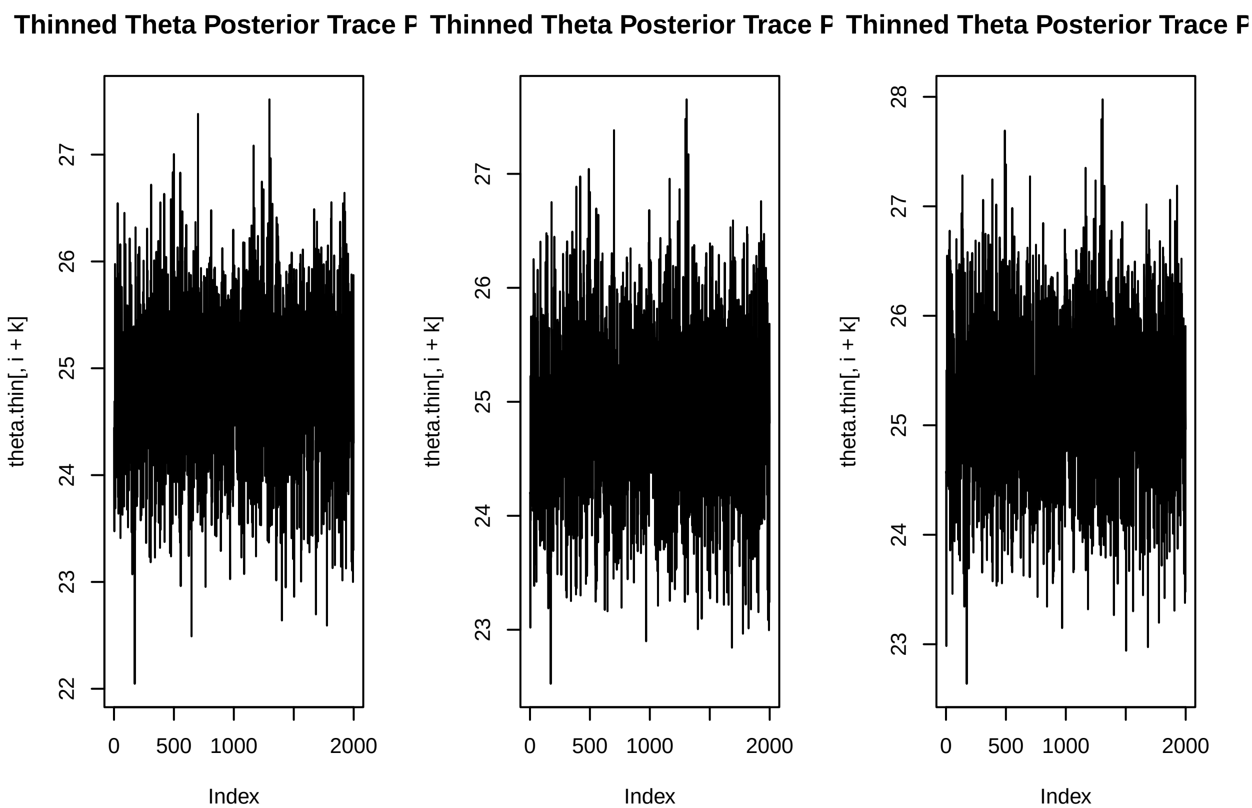



```
burnin = 10000
thin = 84
Eff.samp = floor((B - burnin)/thin)

mu.thin = numeric(); sig.thin = numeric()
theta.thin = matrix(NA, nrow = 2000, ncol = J)
for(i in 1:Eff.samp) {
  mu.thin[i] = mu.samples[(burnin+1+(thin*(i-1)))]
  sig.thin[i] = sig.samples[(burnin + 1 + (thin*(i-1)))]
  theta.thin[i,] = theta.samples[(burnin + 1 + (thin*(i-1))),]
}
```

```
par(mfrow=c(1,2))
plot(mu.thin, type='l', main = 'Thinned Mu Posterior Trace Plot')
plot(sig.thin, type='l',main = 'Thinned Sig^2 Posterior Trace Plot')
```




```r
## this plots the 7th, 8th, and 9th theta posterior.
s=3
k = 2*s
par(mfrow=c(1,3))
for(i in 1:3){
  plot(theta.thin[,i+k], type='l', xlim=c(0,2000), main ='Thinned Theta Posterior Trace Plot')
}
```




```
nrow(theta.thin); length(mu.thin); length(sig.thin)
```

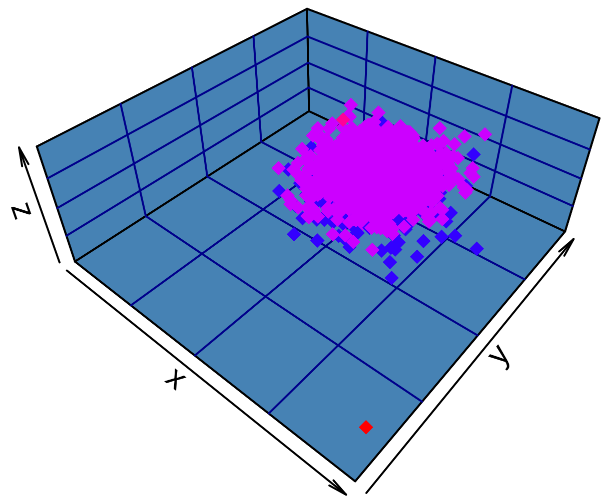
```
## [1] 2000
## [1] 2000
## [1] 2000
```

```
joint.post = matrix(c(mu.thin,sig.thin,theta.thin[,7]), nrow = 3) # each column is a posterior sample.
```

```
library(plot3D)
```

```
mu.3d = joint.post[1,]
sig.3d = joint.post[2,]
theta.3d = joint.post[3,]

scatter3D(mu.3d, sig.3d, theta.3d, pch = 18, bty = "u", colkey = FALSE,
   main ="Sample Joint Posterior f(mu,sig^2,theta|y)", col.panel ="steelblue", expand =0.4,
   col.grid = "darkblue", xlim = c(20,28), ylim = c(20,28), zlim = c(20,28), col=rainbow(10))
```

Sample Joint Posterior f(mu,sig^2,theta|y)



g:

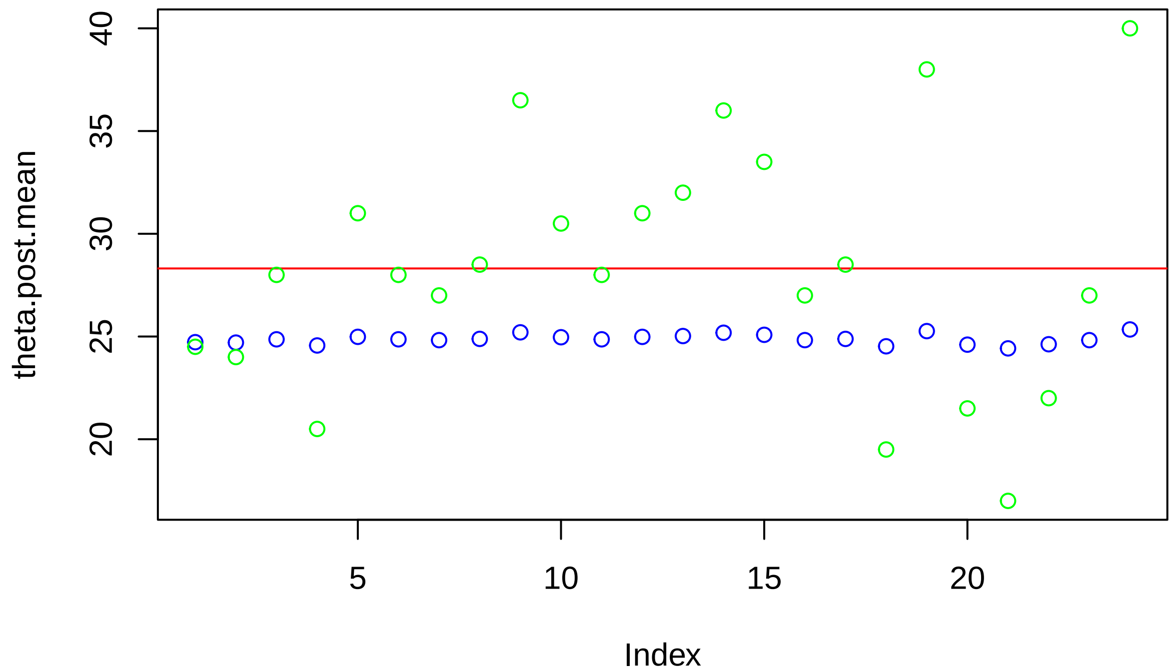
Each column of theta.thin is a distribution for that student (posterior point and interval estimates of theta)

```
for (j in 1:ncol(theta.thin)) {
temp = quantile(theta.thin[j,], probs = c(0.025,0.5, 0.975))
print(temp)
}
```

```
##     2.5%      50%    97.5%
## 23.72642 24.26663 24.69842
##     2.5%      50%    97.5%
## 23.33361 24.06769 24.55036
##     2.5%      50%    97.5%
## 22.44587 23.20009 23.66618
##     2.5%      50%    97.5%
## 24.51749 25.04460 25.62082
##     2.5%      50%    97.5%
## 23.90507 24.74765 25.42983
##     2.5%      50%    97.5%
## 23.77563 24.32180 25.17905
##     2.5%      50%    97.5%
## 23.77728 24.16499 24.66837
##     2.5%      50%    97.5%
## 23.56113 24.01308 24.82679
##     2.5%      50%    97.5%
## 25.50644 26.23482 26.58374
```

```
##     2.5%      50%    97.5%
## 23.68721 23.94745 24.55860
##     2.5%      50%    97.5%
## 24.03844 24.66077 25.20685
##     2.5%      50%    97.5%
## 24.88155 25.43928 25.79554
##     2.5%      50%    97.5%
## 25.04910 25.50006 25.92050
##     2.5%      50%    97.5%
## 23.95507 24.31919 24.81410
##     2.5%      50%    97.5%
## 23.82243 24.35367 24.90591
##     2.5%      50%    97.5%
## 24.66175 25.29764 25.66320
##     2.5%      50%    97.5%
## 23.87258 24.43565 25.05232
##     2.5%      50%    97.5%
## 24.25866 24.99028 25.89710
##     2.5%      50%    97.5%
## 24.90453 25.41466 25.76317
##     2.5%      50%    97.5%
## 24.27081 24.78058 25.31661
##     2.5%      50%    97.5%
## 24.11783 24.83628 25.51486
##     2.5%      50%    97.5%
## 23.39586 24.07077 24.67054
##     2.5%      50%    97.5%
## 23.55188 24.42872 25.47122
##     2.5%      50%    97.5%
## 23.57540 24.10525 24.88462
```

```r
theta.post.mean = apply(theta.samples, 2, mean)
mle = y.bar


plot(theta.post.mean, col = 'blue', ylim= c(min(mle), max(mle)))
abline(h=mean(y.bar), col = 'red')
points(mle, col = 'green')
```




```
cbind(mean(abs(theta.post.mean - mean(y.bar))), mean(abs(y.bar - mean(y.bar))))
```

```
##          [,1]     [,2]
## [1,] 3.436674 4.505208
```

It appears that the posterior mean is more stable around 25 (blue), compared to the sample mean (green). The posterior mean is closer to the overall observed mean score (mean(y.bar)).

The posterior mean isn't super close to the line that forms the observed overall mean score, but its somewhat close. The absolute difference is smaller in the posterior mean when compared to the MLE (ybar for a normal distribution).

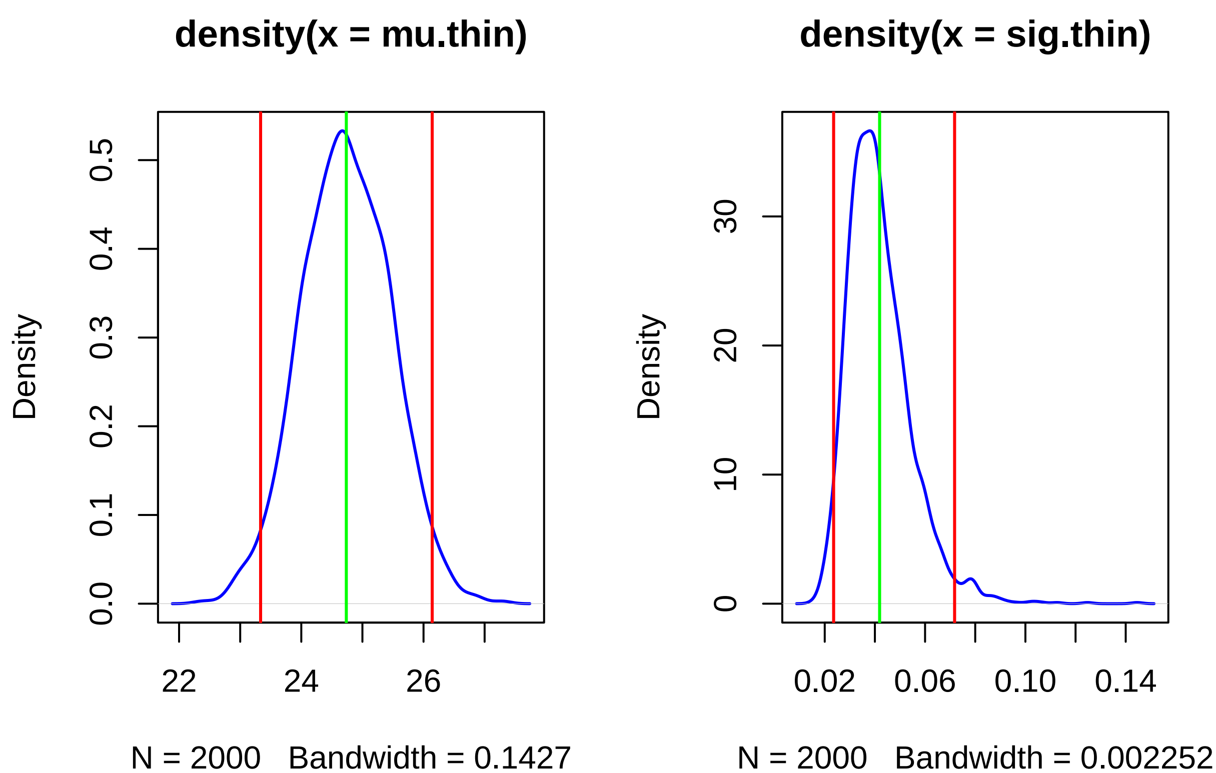
The trace plots for mu and sigma are not amazing. I'm not convinced I have good mixing after thinning the values. Sigma^2 did not need thinning, but mu and theta posteriors did. The thinning for theta looks good, but im not super convinced by mu. I think I could've done better in thinning it.

In the context of the data, it doesnt seem that variation across students is not significant. We can see this in the form of the the blue dots (and the for-loop printing the theta post CI). They are mostly around the same value. Theta posterior allows us to infer on the variation across students; if there are no significant differences in values within the theta samples, then we can say there is no significant student effect.

h

```
mu.low = quantile(mu.thin, 0.025)
mu.up = quantile(mu.thin, 0.975)
sig.lo = quantile(sig.thin, 0.025)
sig.up = quantile(sig.thin, 0.975)
mu.mle = mean(mu.thin)
```

```r
sig.mle = mean(sig.thin)

par(mfrow=c(1,2))
plot(density(mu.thin), lwd = 1.5, col = 'blue')
abline(v = c(mu.low, mu.up), lwd = 1.5, col = 'red')
abline(v = mu.mle, col = 'green', lwd = 1.5)

plot(density(sig.thin), lwd = 1.5, col = 'blue')
abline(v = c(sig.lo, sig.up), lwd = 1.5, col = 'red')
abline(v = sig.mle, col = 'green', lwd = 1.5)
```



Here we see that the posterior means for mu and sig are 24.7 and 0.04, respectively. The posterior mean for sigma tells me that there is very little variation for the posterior of theta. That means that the hyperprior of sigma on theta, shows that there is little student effect on the observed scores.

Mu tells us that the score of the jth student. We see that the mean score in the posterior of mu tells us that this number is 24.7. The observed sample mean is 28.31.

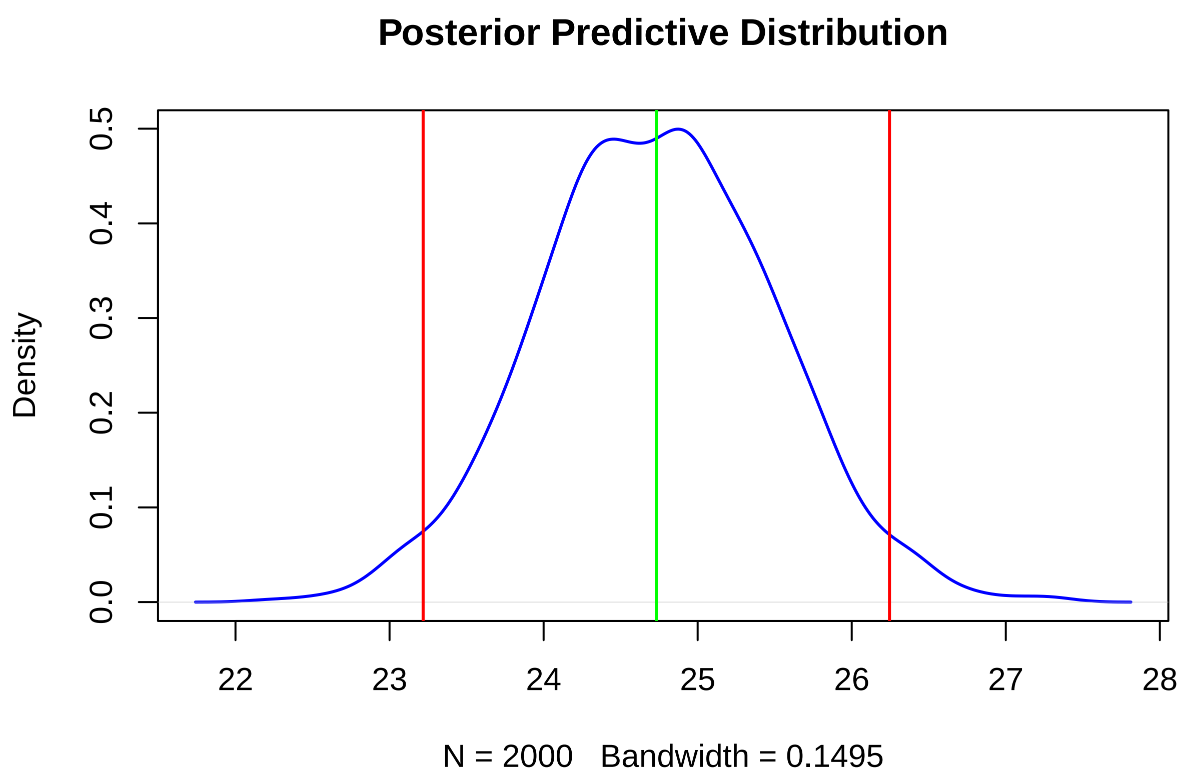
i

Posterior predictive

we have $y_{ij}|\theta_j, \sigma^2 \sim N(\theta_j, \sigma^2)$

so our predictive distribution will replace theta_j and sigma^2 with the posterior samples we calculated earlier.

```r
set.seed(538)
post.pred = rnorm(2000, mean = theta.thin, sd = sqrt(sig.thin))

pp.lo = quantile(post.pred, 0.025)
pp.up = quantile(post.pred, 0.975)
pp.mean = mean(post.pred)


plot(density(post.pred), lwd = 1.5, col = 'blue', main = 'Posterior Predictive Distribution')
abline(v=c(pp.lo, pp.up), lwd = 1.5, col='red')
abline(v=pp.mean, lwd = 1.5, col = 'green')
```



Here we see that 95% of new scores will be within 23.21 and 26.24. The average new score will be 24.73. I think that these results aren't super great. As mentioned before, the posterior mean was quite far from the observed sample mean. This tells us that the MCMC didnt get super close to the true distribution of theta and thus the joint posterior. I'm not sure if this was due to bad mixing, or a typo in my code, or the efficacy of the model as whole. At this point its hard to say, but overall this model could be a little better.

Question 2:

#a joint likelihood: $f(y|\theta, \phi, \sigma^2) = \prod_j \prod_h \frac{1}{\sqrt{2\pi\sigma^2}} \exp[-\frac{1}{2}(y_{ij} - \theta_j - \phi_h)^2/\sigma^2]$ ## b full posterior conditional for θ_j $f(\theta_j|\phi_h, \mu, \sigma^2, y) \sim N(\frac{y_{ij}/\sigma_j^2+\mu/\sigma^2}{1/\sigma_j^2+1/\sigma^2}, [1/\sigma_j^2 + 1/\sigma^2]^{-1})$ ## c full posterior conditional for ϕ_h $f(\phi_h|\theta, \mu, \sigma^2, y) \sim N(\frac{\bar{y}/\sigma_j^2}{1/\sigma_j^2+4/\sigma^2}, [1/\sigma_j^2 + 4/\sigma^2]^{-1})$ ## d conditional posteriors on hyperparameters $f(\mu|\theta, \phi, \sigma^2, y) \sim N(\frac{24\bar{\theta}/\sigma^2+270/\sigma^2}{24/\sigma^2+9/\sigma^2}, [1/\sigma_j^2 + 4/\sigma^2]^{-1})$

$f(\sigma^2|\theta,\phi,\mu,y) \sim \Gamma^{-1}(13, 5 + \sum_j \frac{(\theta_j-\mu)^2}{2})$

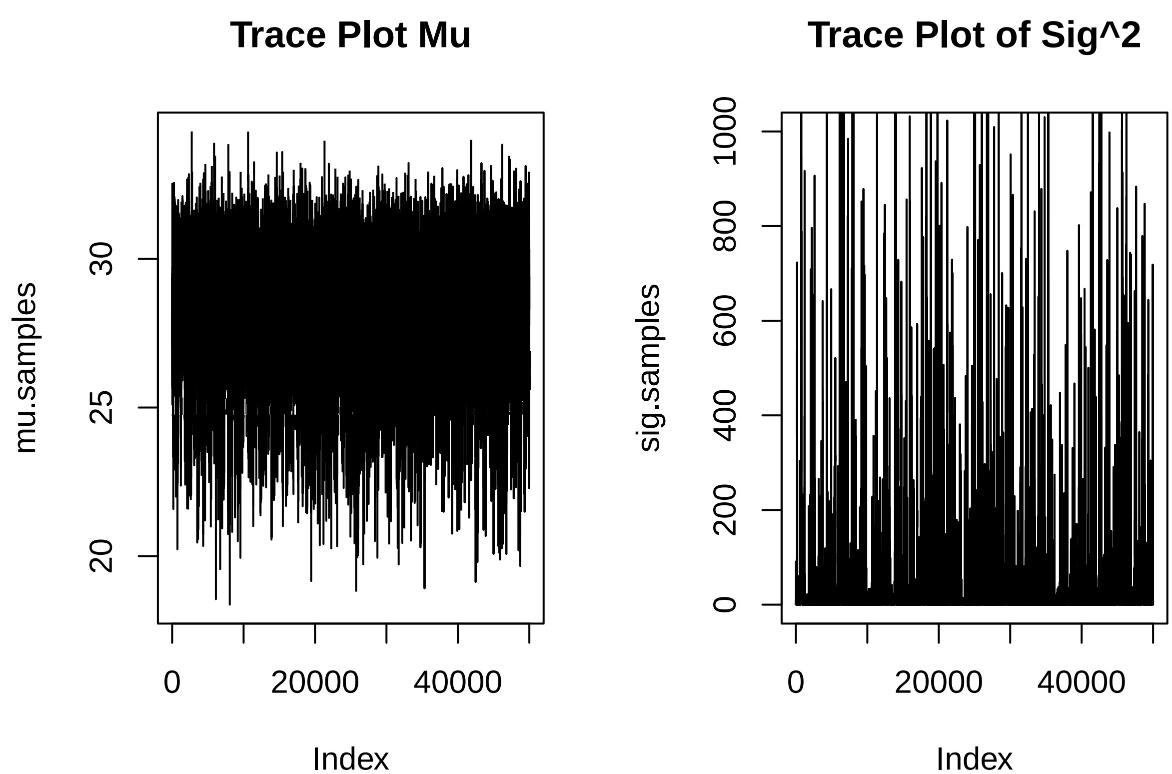
e

```
post.theta = function(y.bar, sig.j, sig, mu){
  j = length(y.bar)
  p1.num = (y.bar/sig.j) + (mu/sig)
  p1.den = (1/sig.j) + (1/sig)
  p1 = p1.num/p1.den
  p2 = 1/((1/sig.j) + (1/sig))
  dn = rnorm(j, mean = p1, sd = sqrt(p2))
  return(dn)
}

post.phi = function(y.bar, sig.j, sig){
  p1.num = (y.bar/sig.j)
  p1.den = ((1/sig.j) + (4/sig))
  p1 = p1.num/p1.den
  p2 = 1/p1
  dn = rnorm(4, mean = p1, sd = sqrt(p2))
}

post.mu = function(theta, sig){
  theta.bar = mean(theta)
  p1.num = (24*theta.bar/sig) + (270/sig)
  p2.num = (24/sig) + (9/sig)
  p1 = p1.num/p2.num
  p2 = 1/pi
  dn = rnorm(1, mean = p1, sd = sqrt(p2))
}


post.sig = function(theta, mu){
  p1 = 13
  p2 = 5 + (sum(theta-mu)^2)/2
  dn = rinvgamma(1, shape = p1, rate = p2)
  return(dn)
}
```

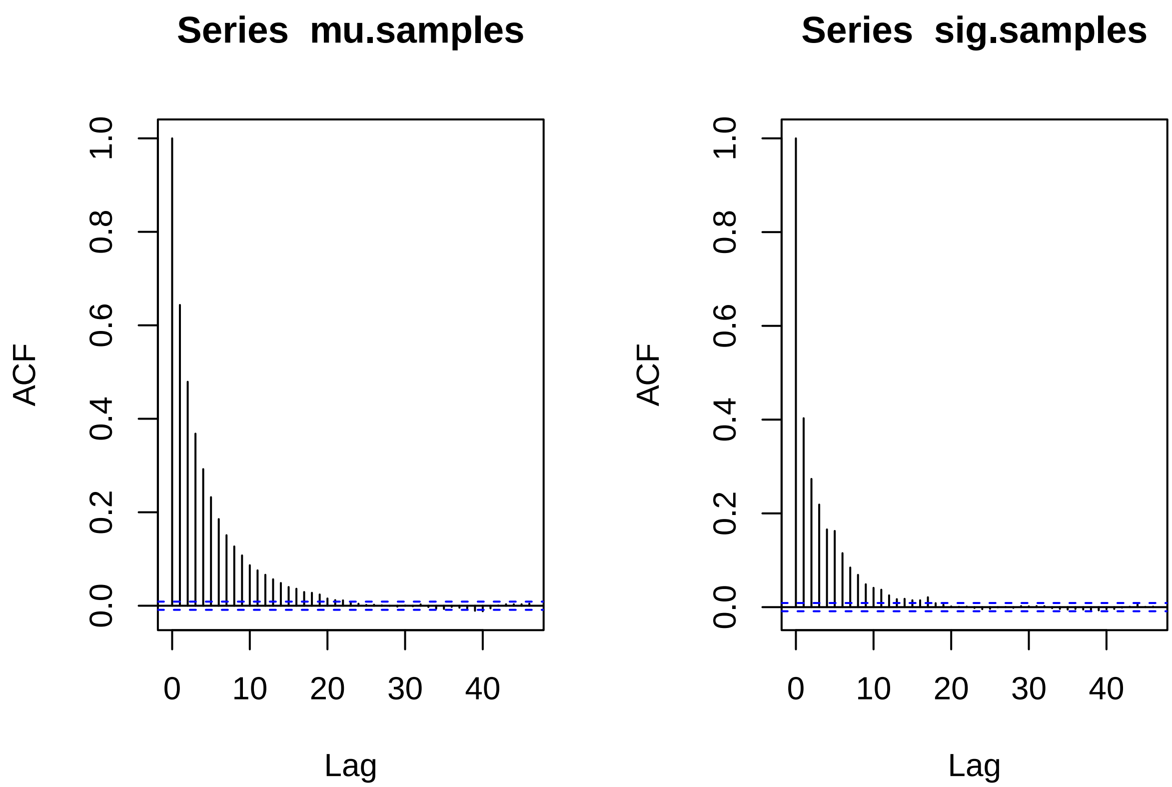
```
set.seed(538)
B = 50000 ## updated B for thin
J = length(y.bar)
y.bar = apply(data, 1, mean) # applying here so i dont have to do it in the for loop fro the samples
sig.j =  apply(data, 1, sd)^2


theta.samples <- matrix(NA, nrow = B+1, ncol = J) #each row is a sample
mu.samples <- numeric()
sig.samples <- numeric()
phi.samples <- matrix(NA, nrow = B+1, ncol = ncol(df)) ## same
```

```
set.seed(11)
theta.samples[1,] <- y.bar
sig.samples[1] <- rinvgamma(1, shape = 1, rate = 1)


for(i in 2:(B+1)) {

  mu.samples[i-1] <- post.mu(as.numeric(theta.samples[i-1]), sig.samples[i-1])

  phi.samples[i-1,] <- post.phi(y.bar, sig.j, sig.samples[i-1])

  theta.samples[i,] <- post.theta(y.bar, sig.j, sig.samples[i-1], mu.samples[i-1])

  sig.samples[i] <- post.sig(as.numeric(theta.samples[i,]), mu.samples[i-1])

}
```

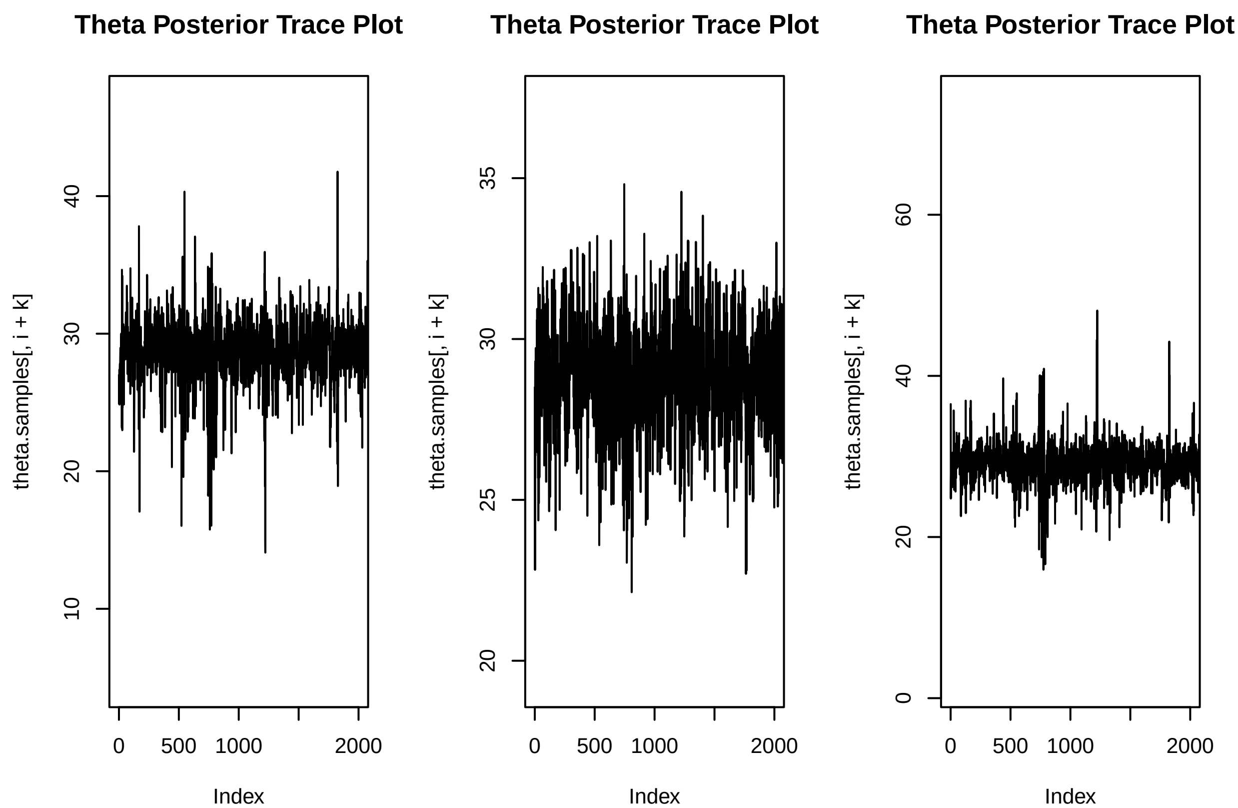
```
par(mfrow=c(1,2))
plot(mu.samples, type = 'l', main = 'Trace Plot Mu')
plot(sig.samples, type = 'l', main = 'Trace Plot of Sig^2', ylim = c(0,1000))
```



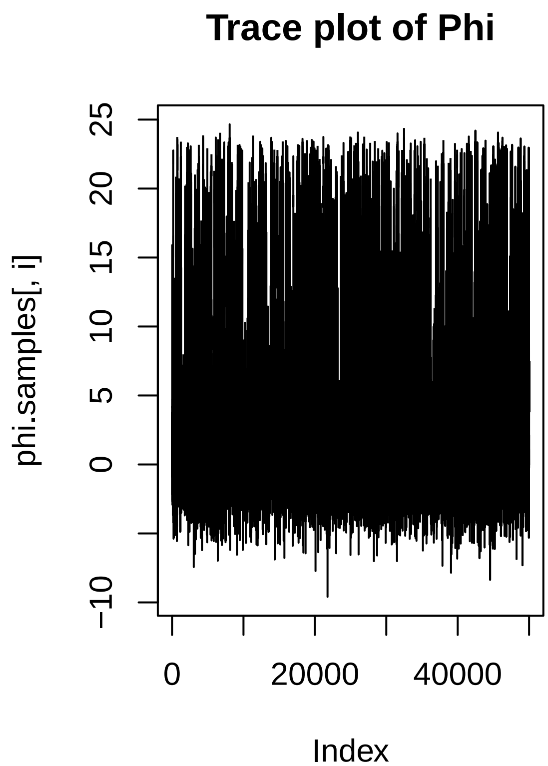

```
acf(mu.samples); acf(sig.samples) ## mu 20, sig 20
```

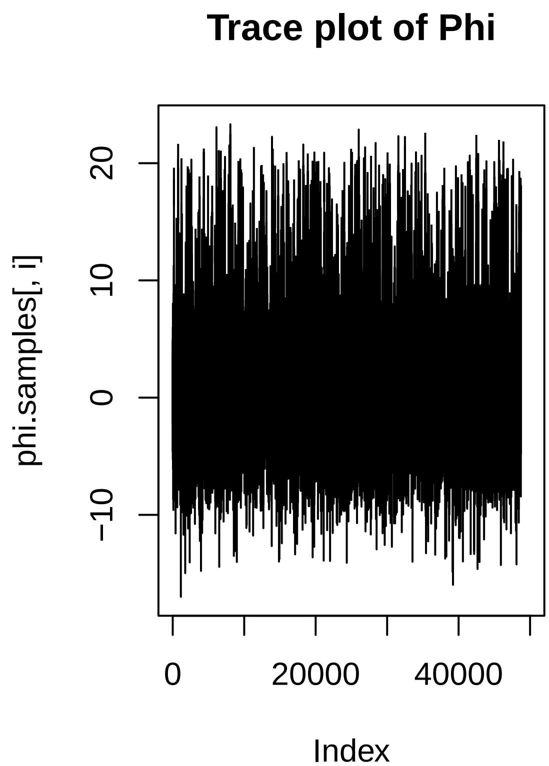



```r
s=3
k = 2*s
par(mfrow=c(1,3))
for(i in 1:3){
  plot(theta.samples[,i+k], type='l', xlim=c(0,2000), main ='Theta Posterior Trace Plot')
}
```

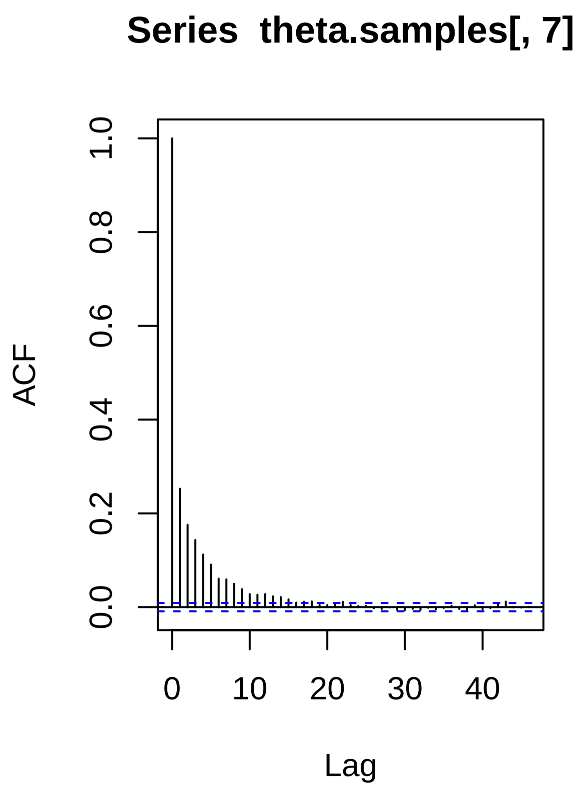


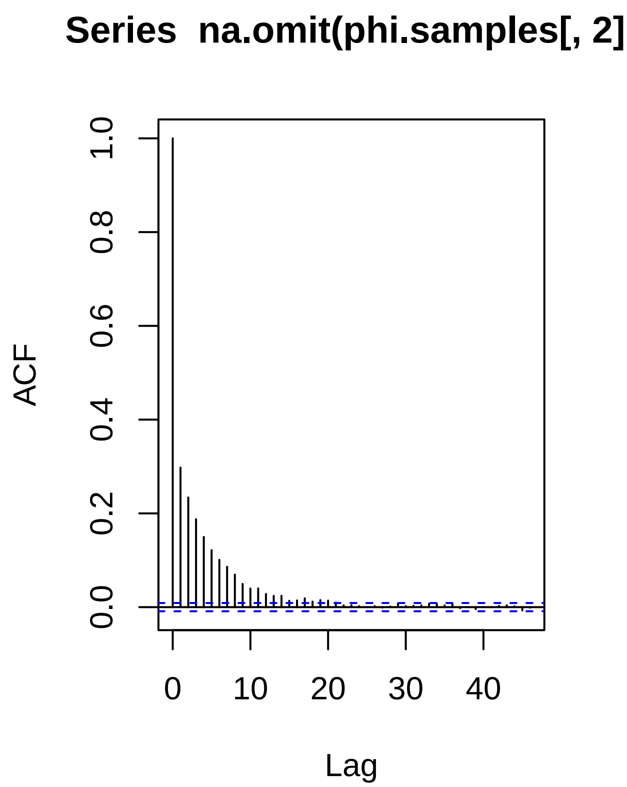

```r
par(mfrow=c(1,2))
for(i in 1:2){
  plot(phi.samples[,i], type='l', main = 'Trace plot of Phi')
}
```





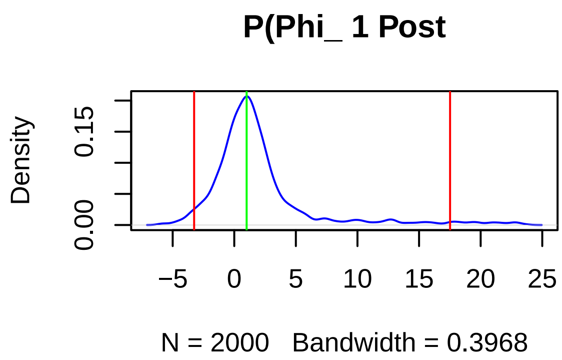

```
acf(theta.samples[,7]); acf(na.omit(phi.samples[,2])) ## theta 15,phi 20
```

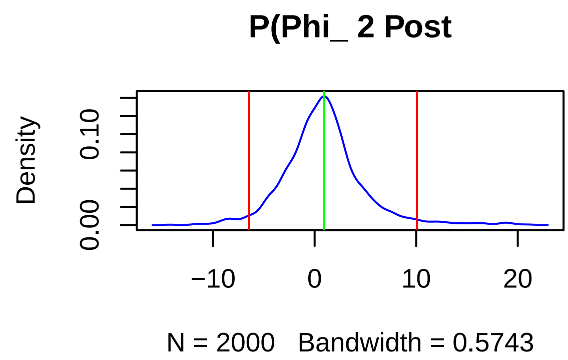


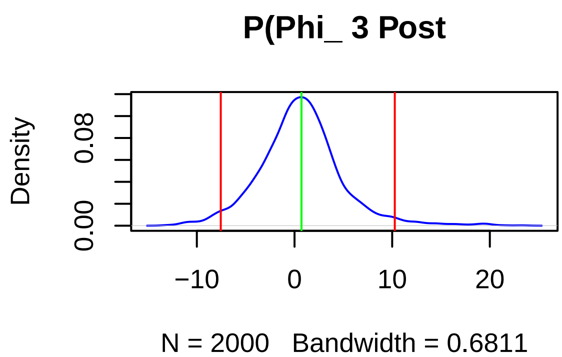


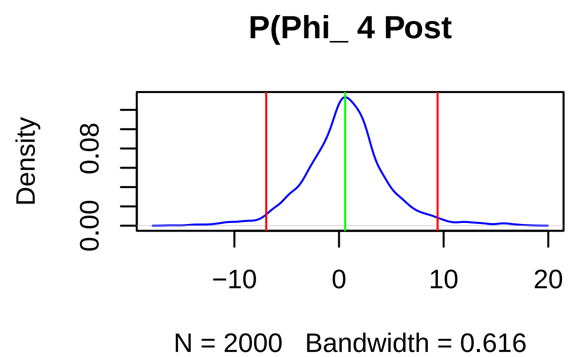

```
burnin = 10000
thin = 20
Eff.samp = floor((B - burnin)/thin)

mu.thin = numeric(); sig.thin = numeric()
theta.thin = matrix(NA, nrow = 2000, ncol = J)
phi.thin = matrix(NA, nrow = 2000, ncol = 4)
for(i in 1:Eff.samp){
  mu.thin[i] = mu.samples[(burnin+1+(thin*(i-1)))]
  sig.thin[i] = sig.samples[(burnin + 1 + (thin*(i-1)))]
  theta.thin[i,] = theta.samples[(burnin + 1 + (thin*(i-1))),]
  phi.thin[i,] = phi.samples[(burnin + 1 + (thin*(i-1))),]
}
```

f










```
##      lower.vec upper.vec
## [1,] -3.258364 17.516999
## [2,] -6.462748 10.065687
## [3,] -7.551174 10.267203
## [4,] -6.960117  9.423583
```

```
##      2.5%       50%     97.5%
## -3.258364  1.103521 17.516999
##       2.5%        50%      97.5%
## -6.4627477  0.6862646 10.0656873
##       2.5%        50%      97.5%
## -7.5511740  0.5271109 10.2672030
##       2.5%        50%      97.5%
## -6.9601171  0.6437991  9.4235829
```

Above is the 95% CI for phi_h posterior in ascending order. We see that the MAP for Phi is about 0 for every single list. There does seem to be a some-what significant difference in the CI between lists. Specifically, list 1 has about a 7 point score difference than the rest of the lists. There is no list effect between lists 2,3, and 4. List 1 seems to have higher scores, however. This suggests that list 1 was easier than the rest.

In the context of the whole data, we can say that despite the lists being designed to be of similar difficulty, its probable that there is a list effect, assuming that the noisy environment is truly consistent across all recordings for each list. The list effect in question suggests that list 1 was easier to score on than the other lists.

I'm not confident on commenting what the specific values of phi represent in the context of the data. I'm guessing it has to do with the range in addition to the average of the score given each list. Say the average in list 1 is 24 for student j1, then the total possible score per list is anywhere between [-3 + 24, 24 + 17] 95% of the time for that student.