Problem Set 3

Prasanna Parasurama

Due: 3/15/19

```
library(lme4)
library(bayesm)
library(ggplot2)
library(dplyr)
library(estimatr)
library(reshape2)
```

Problem 1 (Coding Exercise)

For this exercise, you will be working with a dataset provided by an R package. Many R packages have standardized datasets that allow you to test code. The specific dataset you will be working with is the "sleepstudy" dataset available from the "lme4" package. When you load the "lme4" package, the dataframe "sleepstudy" will be available for you to use.

The dataset contains results from a sleep study experiment. 18 subjects were deprived of sleep over 9 days, and given reaction time tests on each day. The columns are as follows:

Reaction : reaction time (in seconds)

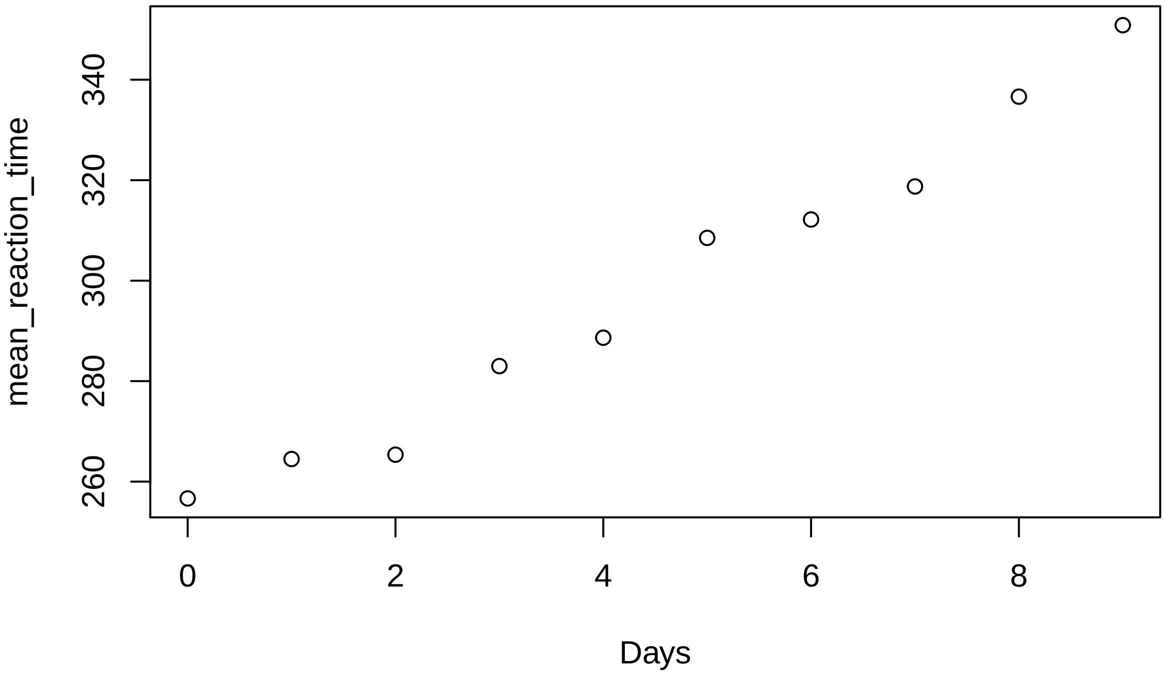
Days : number of days with sleep deprivate (0 is no sleep deprivation)

Subject : unique subject identifier

Before doing any regression analysis, it's always a good idea to get a feel for the data.

a. Plot the mean reaction time as the number of sleep deprivation days increase. What do you see?

```
sleepstudy %>%
  group_by(Days) %>%
  summarise(mean_reaction_time = mean(Reaction)) %>%
  plot()
```



As we would expect, average reaction time increases as sleep deprivation days increase.

At first, we think that the following model is the correct model to explain the impact of sleep deprivation on reaction times:

$$r_{it} = \mu + \beta s_{it} + \epsilon_{it}$$

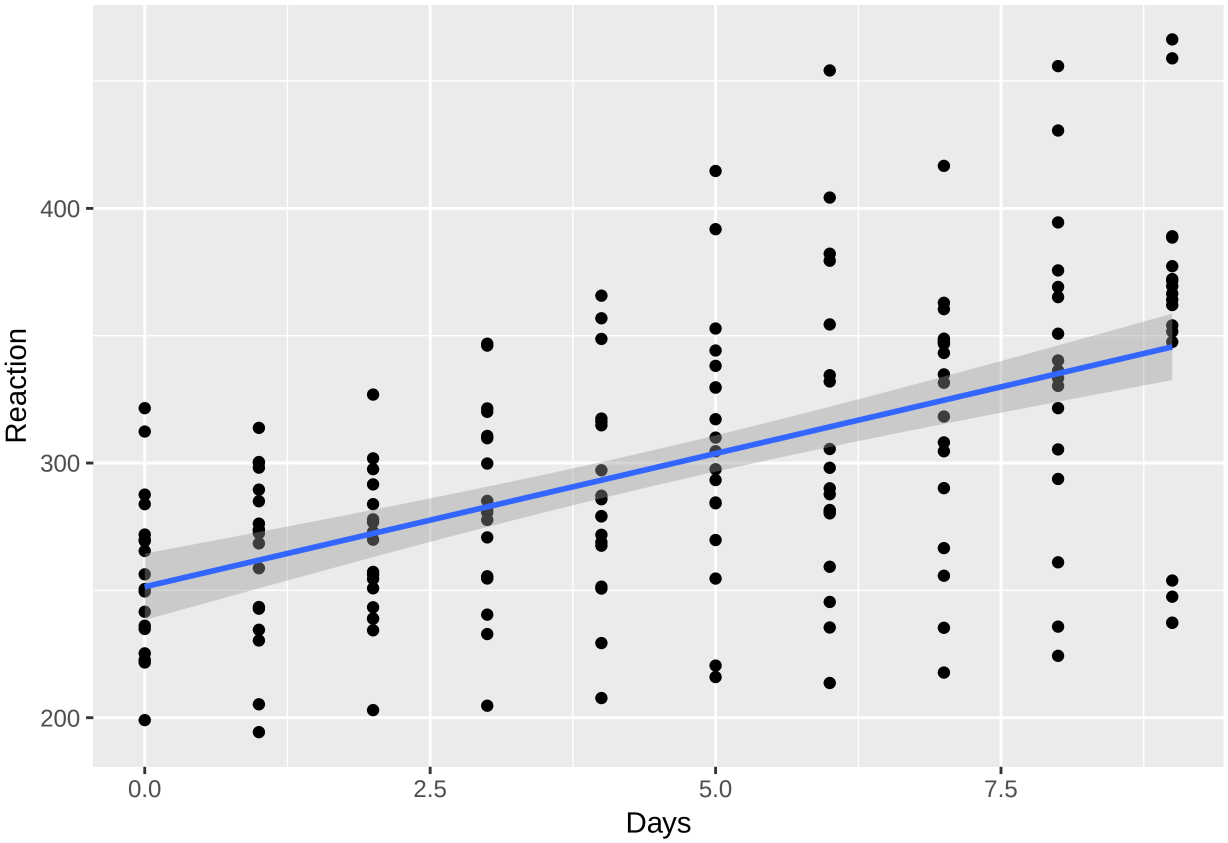
where,

- r_{it} is reaction time of subject i on day t
- s_{it} is the number of days without sleep deprivation for subject i on day t

R has two ways for us to do these simple regressions: (1) using the plot functions or (2) using the regression functions.

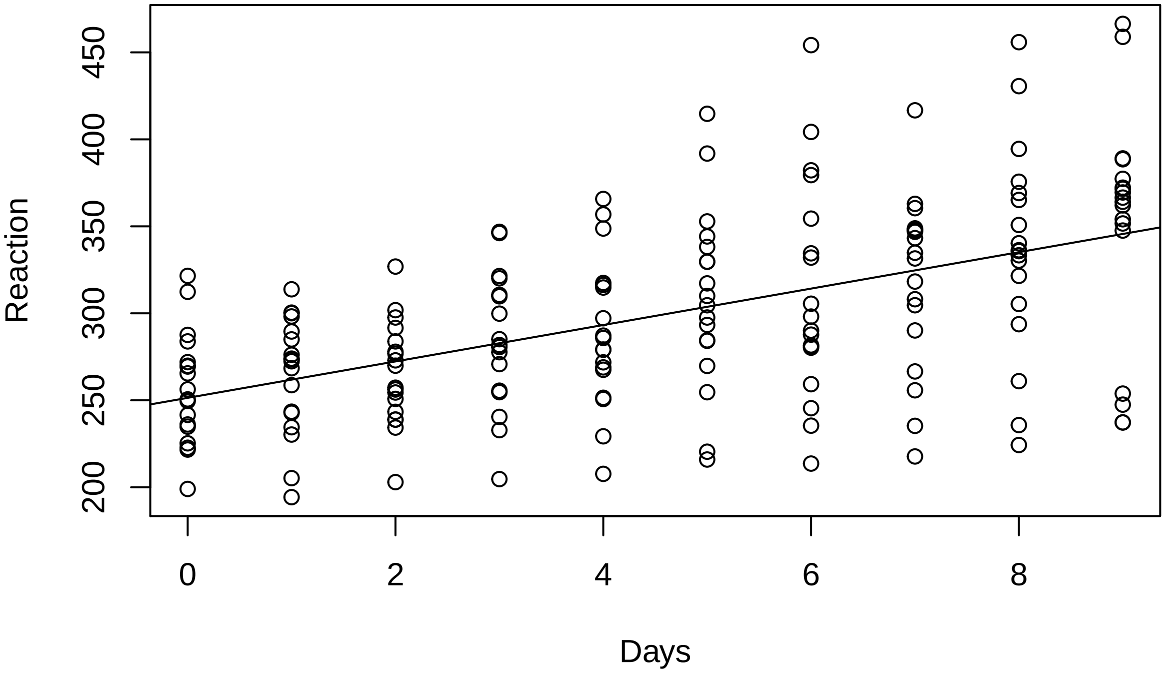
b. Using ggplot and the "stat_smooth" option, plot a scatter plot with an ols line from the above model.}

```
ggplot(data=sleepstudy, aes(x = Days, y=Reaction)) +
  geom_point() +
  stat_smooth(method="lm")
```



c. Using the 'lm_robust' function, run the above regression. Remake the same plot.

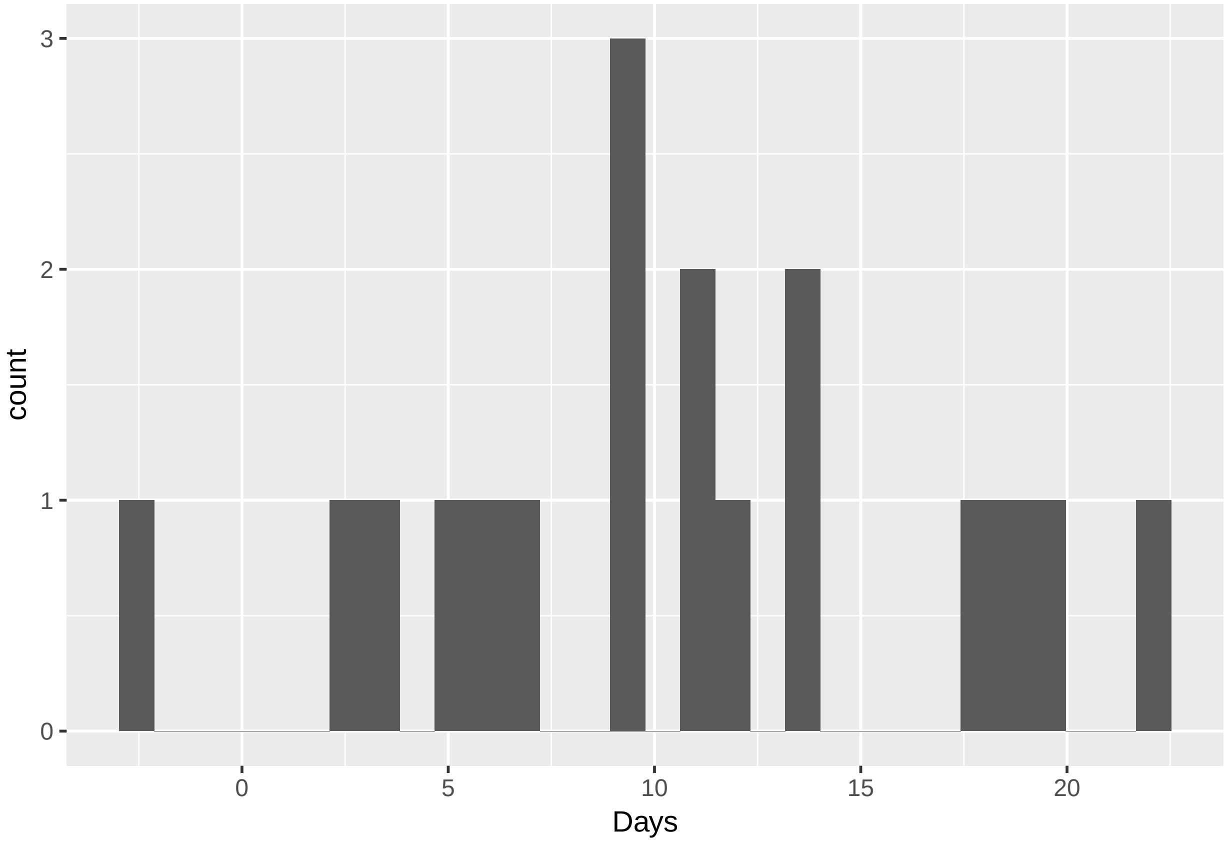
```
base_model = lm_robust(Reaction ~ Days, data=sleepstudy)
attach(sleepstudy)
plot(Days, Reaction)
abline(base_model)
```



d. One issue with the above procedure is it assumes a constant slope β for all subjects. Run a seperate ols regression for each individual, and make a histogram of the estimates. What do you see?

```r
random_coefs = lmList(Reaction~Days | Subject, data=sleepstudy)

ggplot(coef(random_coefs), aes(x=Days)) +
  geom_histogram()
```

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



Note that the estimates vary quite a bit across subjects.

To address the issues with differing responsiveness to sleep deprivation, we instead decide to estimate the following random-slope model:

$$\begin{aligned}
&r_{it} = \mu + \beta s_{it} + \beta_i s_{it} + \epsilon_{it} \\
&\beta_i \sim N(0, \sigma_\beta^2) \\
&\epsilon_{i,t} \sim N(0, \sigma_\epsilon^2) \\
&\beta_i \perp \epsilon_{i,t}
\end{aligned}$$

e. You have seen this model before, what type of model is it? Random slopes model.

f. What is the covariance matrix of the random-slope model? Compare it to the covariance matrix of the previous model? What has changed?

In the random-slopes model, the covariance matrix of β will be $\sigma_\beta^2 I$. This comes directly from $\beta_i \sim N(0, \sigma_\beta^2)$

The model from part d, makes no assumption about β_i coming from the same distribution. So the covariance matrix doesn't have to be diagonal.

g. One useful statistic people calculate when trying to determine if the random-slope model is correct is the intraclass correlation coefficient, ICC, defined as:

$$ICC = \frac{Var(\beta_i)}{Var(\beta_i) + Var(\epsilon_{ij})}$$

Why is this a good measure to think about whether the above model is the correct model? Estimate the ICC for this data, interpret your results. (Note: What are unbiased estimates for each of the components? Try using the law of total variance on the sum of squared errors.)}

ICC tells us the fraction of variance explained by variance in the random slopes. If there is no variance in slopes, then ICC would be 0, so a simple fixed/random effects model would do.

To estimate the above model, there are two different approaches that one can take: (1) using Maximum likelihood or (2) using Bayesian methods. Lucky for us, there are R packages for both of them.

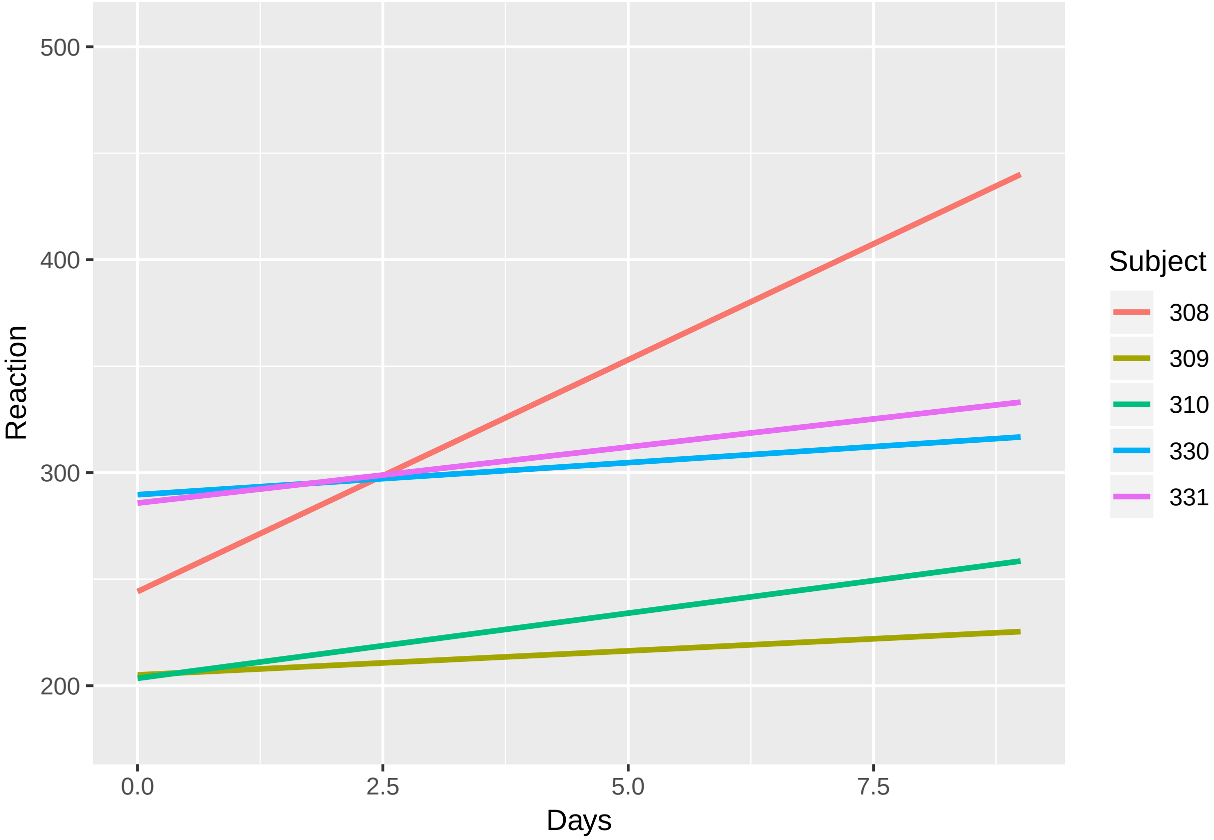
Maximum Likelihood Approach

h. Run a seperate ols regression for each individual, plot the resulting ols estimates across subjects (pick 5 subjects). What do you see?

```
random_coefs
```

```
## Call: lmList(formula = Reaction ~ Days | Subject, data = sleepstudy)
## Coefficients:
##     (Intercept)      Days
## 308    244.1927 21.764702
## 309    205.0549  2.261785
## 310    203.4842  6.114899
## 330    289.6851  3.008073
## 331    285.7390  5.266019
## 332    264.2516  9.566768
## 333    275.0191  9.142045
## 334    240.1629 12.253141
## 335    263.0347 -2.881034
## 337    290.1041 19.025974
## 349    215.1118 13.493933
## 350    225.8346 19.504017
## 351    261.1470  6.433498
## 352    276.3721 13.566549
## 369    254.9681 11.348109
## 370    210.4491 18.056151
## 371    253.6360  9.188445
## 372    267.0448 11.298073
##
## Degrees of freedom: 180 total; 144 residual
## Residual standard error: 25.59182
```

```
ggplot(data=sleepstudy %>% filter(Subject %in% c(308,309,310,330,331)), aes(y=Reaction, x=Days, color=S
```



i. In part (d), we had you run an individual-by-individual ols regression. Why is this not the best estimation strategy?

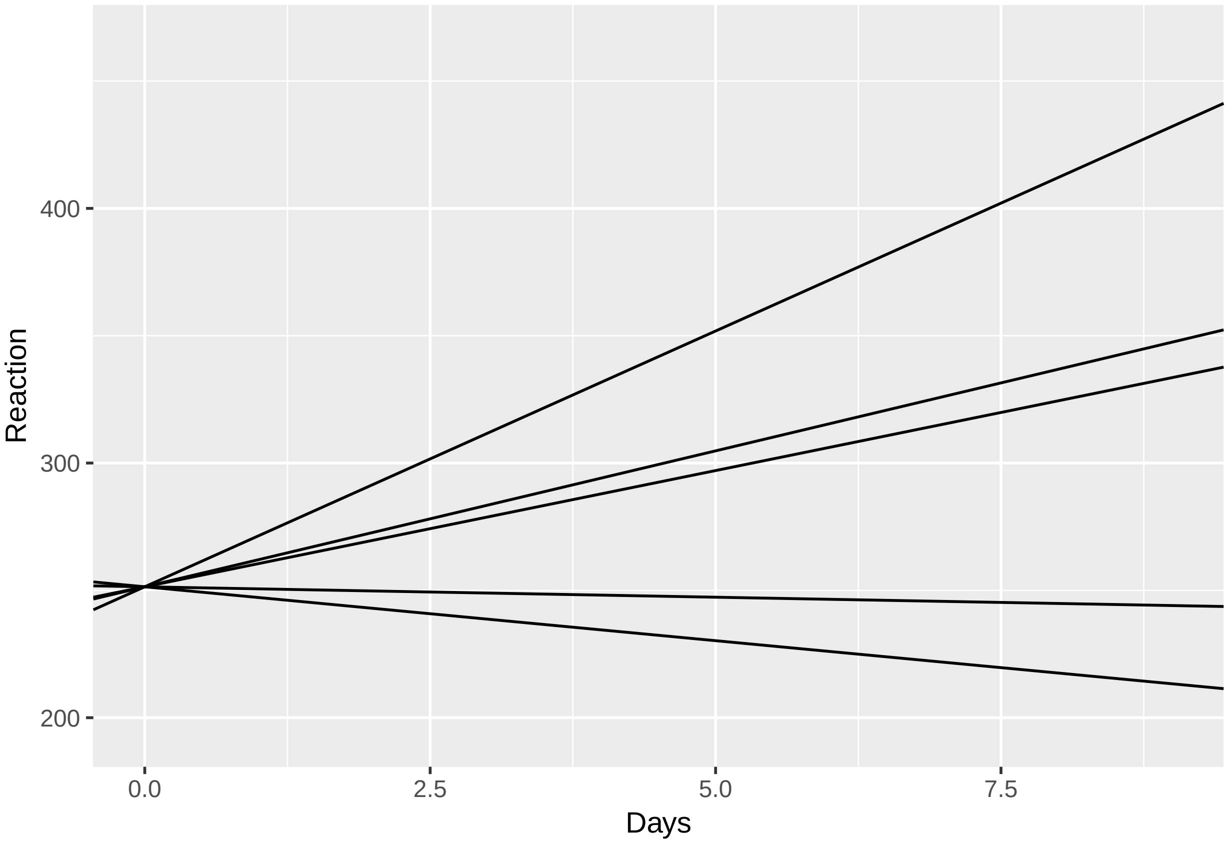
In individual-by-individual OLS regression, the individual slopes could be picking up too much individual idiosyncratic differences/outliers.

There may be some variance across individuals, but we know sleep affects reaction time in more or less the same way across individuals.

j. Using the lme4 package, run the regression in the model specified above. Plot the resulting unit-level regression lines for 5 subjects, how do they compare to the individual-by-individual ols regressions?

```
random_slopes = coef(lmer(Reaction ~ Days + (-1+Days|Subject), sleepstudy))
```

```
estimates = head(random_slopes$Subject,5)

ggplot(data=sleepstudy, aes(x=Days, y=Reaction)) +
  geom_blank() +
  geom_abline(data=estimates, aes(intercept=`(Intercept)`, slope=Days))
```



They all have the same intercept.

The **lme4** package is maximizing a likelihood function, an alternative approach would be to use a bayesian method. The next part of this exercise asks you to understand these differences and run a bayesian method. In order for this model to be bayesian, we now write down the following model:

Bayesian Approach

To help you better understand the main package you will be using, I have provided an example code in the code folder: bayesm_example.R.

Consider the following model:

$$\begin{aligned} r_{it} &= \mu_i + \bar{\beta}_i s_{it} + \epsilon_{ij} \\ \bar{\mu}_i &= \mu + \mu_i \\ \bar{\beta}_i &= \beta + \beta_i \\ \begin{bmatrix} \mu_i \\ \beta_i \end{bmatrix} &\sim N(0, \sigma^2_\beta) \\ \sigma^2_\beta &\sim IW(v_0, V_0) \end{aligned}$$

where,

IW is an inverse-wishart distribution (a common prior distribution used)

k. What is} β {in the model above? In words, no calculations are needed. β is the group mean.

l. What differs in this model from the previous model we estimated? Both the intercept and slope are random for an individual. In the previous model, all the individuals had the same intercept.

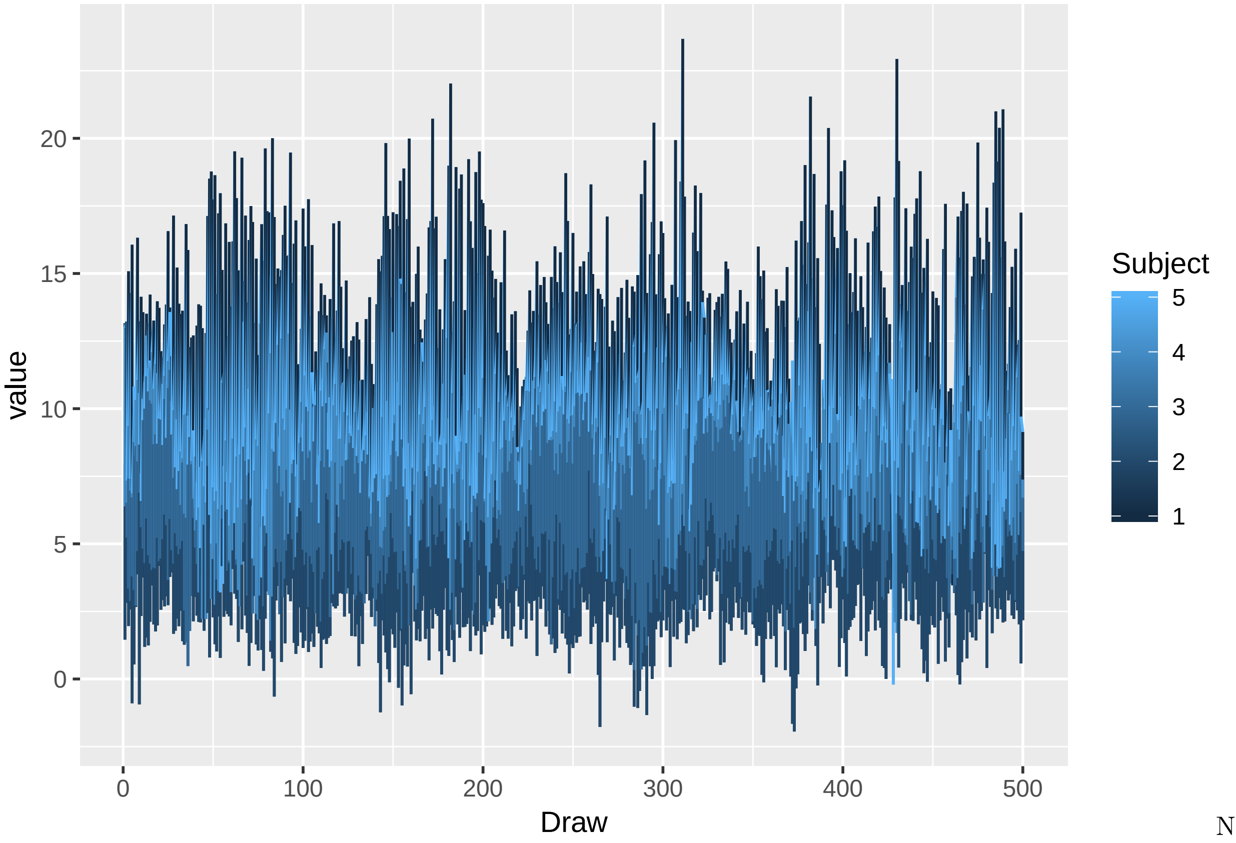
Using the rhierLinearModel function from bayesm (do not change the default settings except the number of MCMC draws), run an MCMC algorithm to estimate the model above, with 2000 draws. Do the following:

```r
# Get subjects
subjects=levels(sleepstudy$Subject)
n_subjects = length(subjects)

# Create matrix of potential covariates for our random effect
# Setting to 1 because our model assumes that there is a common mean across groups
Z=matrix(c(rep(1,n_subjects)), ncol=1)
nz=ncol(Z)

regdata=NULL    #<- This will be a named list to store the data
                # Each element of the list is the observations (both IV and DV) for each i
for(s in subjects){
  y = (sleepstudy %>% filter(Subject == s))$Reaction        # Create LHS variable
  X=cbind(1, (sleepstudy %>% filter(Subject == s))$Days)  # Create constant + RHS variables
  regdata[[which(subjects==s)]] = list(y=y,X=X)                     # For each unit i, add t
}

# Create data for MCMC
Data=list(regdata=regdata,Z=Z)
Mcmc=list(R=2000,keep=1)

# Run linear hierarchical model
out=rhierLinearModel(Data=Data,Mcmc=Mcmc)
```

```
##
## Starting Gibbs Sampler for Linear Hierarchical Model
##     18  Regressions
##     1  Variables in Z (if 1, then only intercept)
##
## Prior Parms:
## Deltabar
##      [,1] [,2]
## [1,]    0    0
## A
##      [,1]
## [1,] 0.01
## nu.e (d.f. parm for regression error variances)=  3
## Vbeta ~ IW(nu,V)
## nu =  5
## V
##      [,1] [,2]
## [1,]    5    0
## [2,]    0    5
##
## MCMC parms:
## R=  2000  keep=  1  nprint=  100
##
##  MCMC Iteration (est time to end - min)
##  100 (0.0)
##  200 (0.0)
##  300 (0.0)
##  400 (0.0)
##  500 (0.0)
##  600 (0.0)
##  700 (0.0)
##  800 (0.0)
##  900 (0.0)
##  1000 (0.0)
##  1100 (0.0)
##  1200 (0.0)
##  1300 (0.0)
##  1400 (0.0)
##  1500 (0.0)
##  1600 (0.0)
##  1700 (0.0)
##  1800 (0.0)
##  1900 (0.0)
##  2000 (0.0)
##  Total Time Elapsed: 0.00
```

m1. Plot the last 500 MCMC draws (these are the betadraw variables) for the slope term for 5 of the subjects, does it seem to converge?

```
last_500_draws = melt(out$betadraw[1:5,2,1501:2000], value.names = "Slope", varnames =  c("Subject", "Di

ggplot(last_500_draws, aes(x=Draw, y=value, color=Subject)) + geom_line()
```

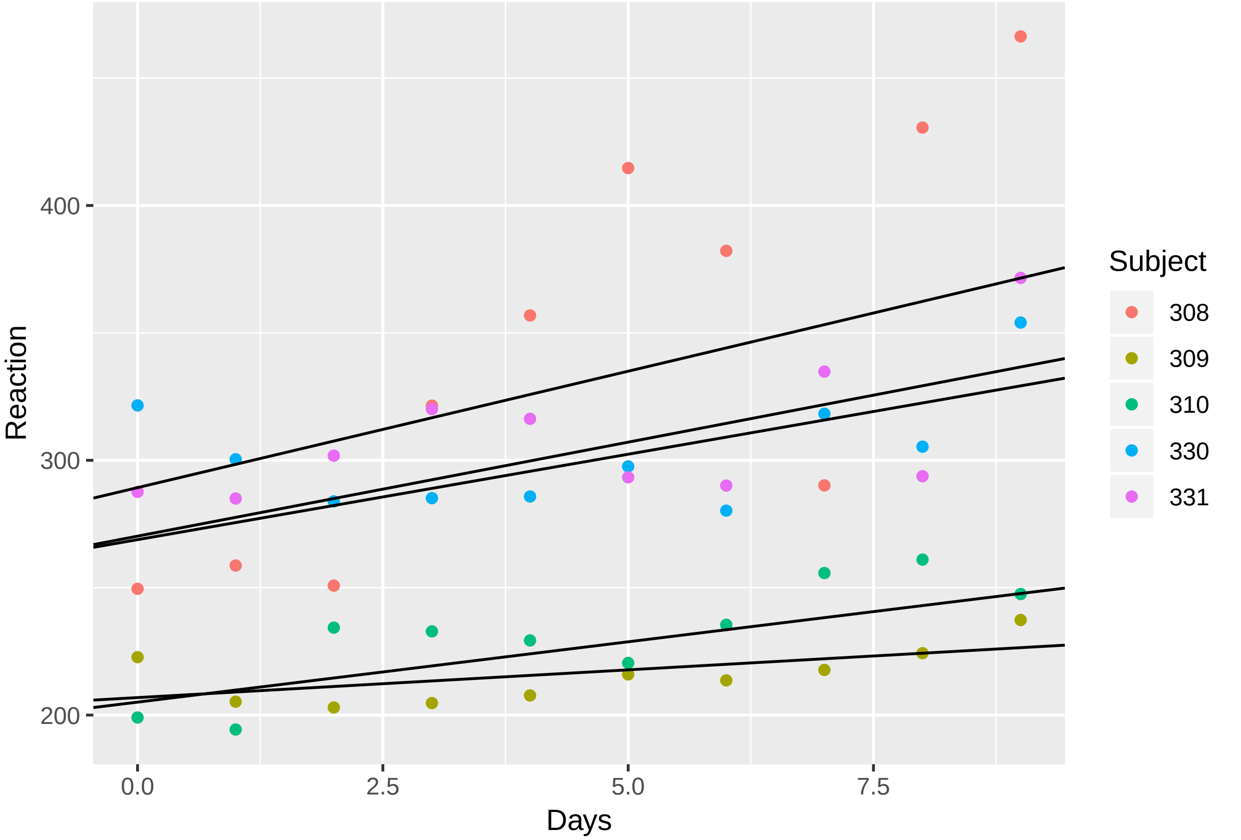


No,

the slopes don't seem to converge.

m2. Using the last draw from the MCMC draws, plot the new unit-level regression line for 5 subjects, i.e. make a scatterplot of the data and plot a regression line using both the new slope and intercept terms (μ_i, β_i) (Note: If your model output is stored in the variable, mcmc, then you can get the last draw using the following mcmc$betadraws[„-1])}.

```
last_draw = data.frame(out$betadraw[1:5,1:2,2000])
ggplot(data=sleepstudy %>% filter(Subject %in% c(308,309,310,330,331))) +
  geom_point(aes(x=Days, y=Reaction, color=Subject)) +
  geom_abline(data=last_draw, aes(intercept=X1, slope=X2))
```



Problem 2 (Coding Exercise)

The goal of this exercise is to predict the average height of a randomly selected person from the population. We have provided a dataset 'height.csv' dataframe with heights for men and women in the population. For now, assume that height, X_i for each individual i, is distributed as $X_i \sim N(\mu, \sigma^2)$.

a. Recast this problem as a regression problem and estimate μ.

$$X_i = \alpha + \epsilon_i$$
$$\epsilon \sim N(0, \sigma^2)$$

```
heights = read.csv("data/height.csv")

height_reg = lm(Height ~ 1, data=heights)
summary(height_reg)
```

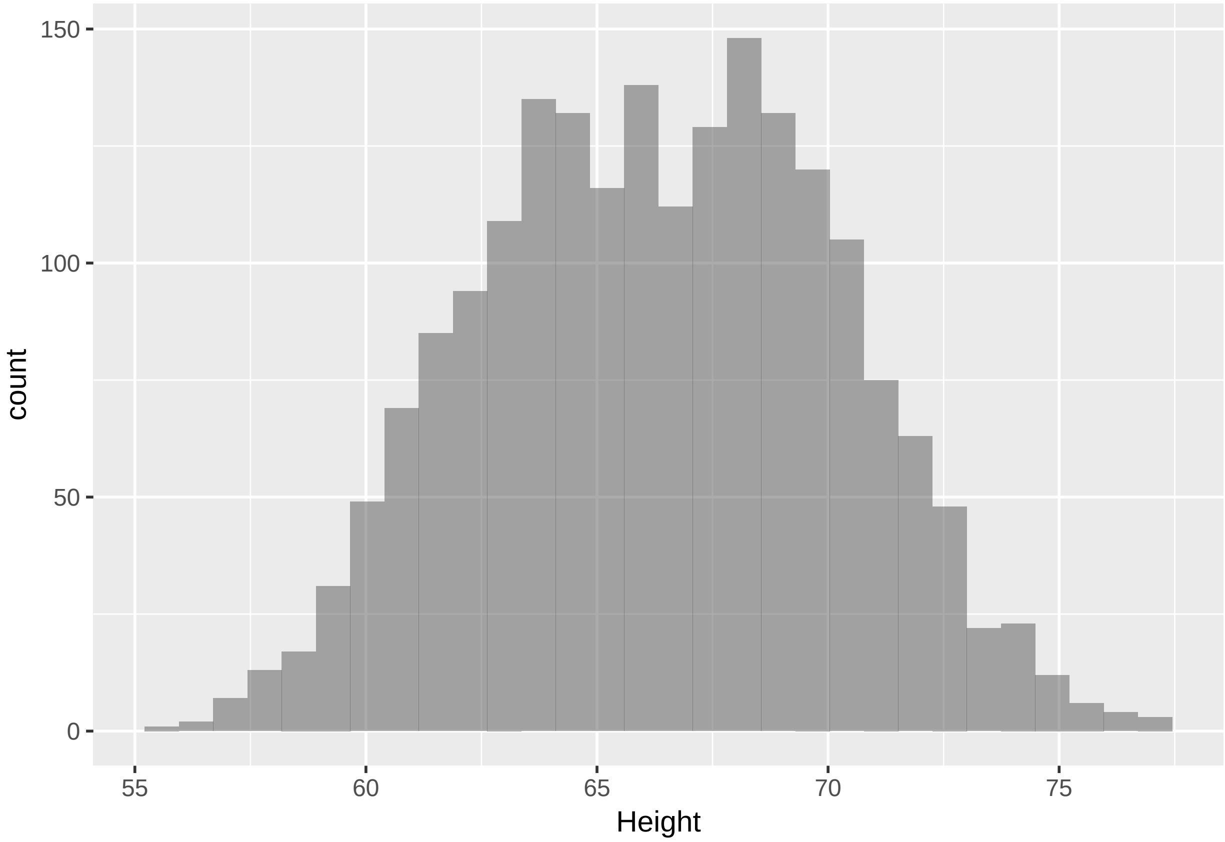
```
##
## Call:
## lm(formula = Height ~ 1, data = heights)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.6775  -2.8807  -0.0081   2.8716  10.8151
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  66.3457     0.0859   772.4   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.841 on 1999 degrees of freedom
```

$\hat{\mu} = \alpha = 66.35$.

b. Going back to the data, plot histograms for the overall population, and then for men and women seperately. What do you notice?

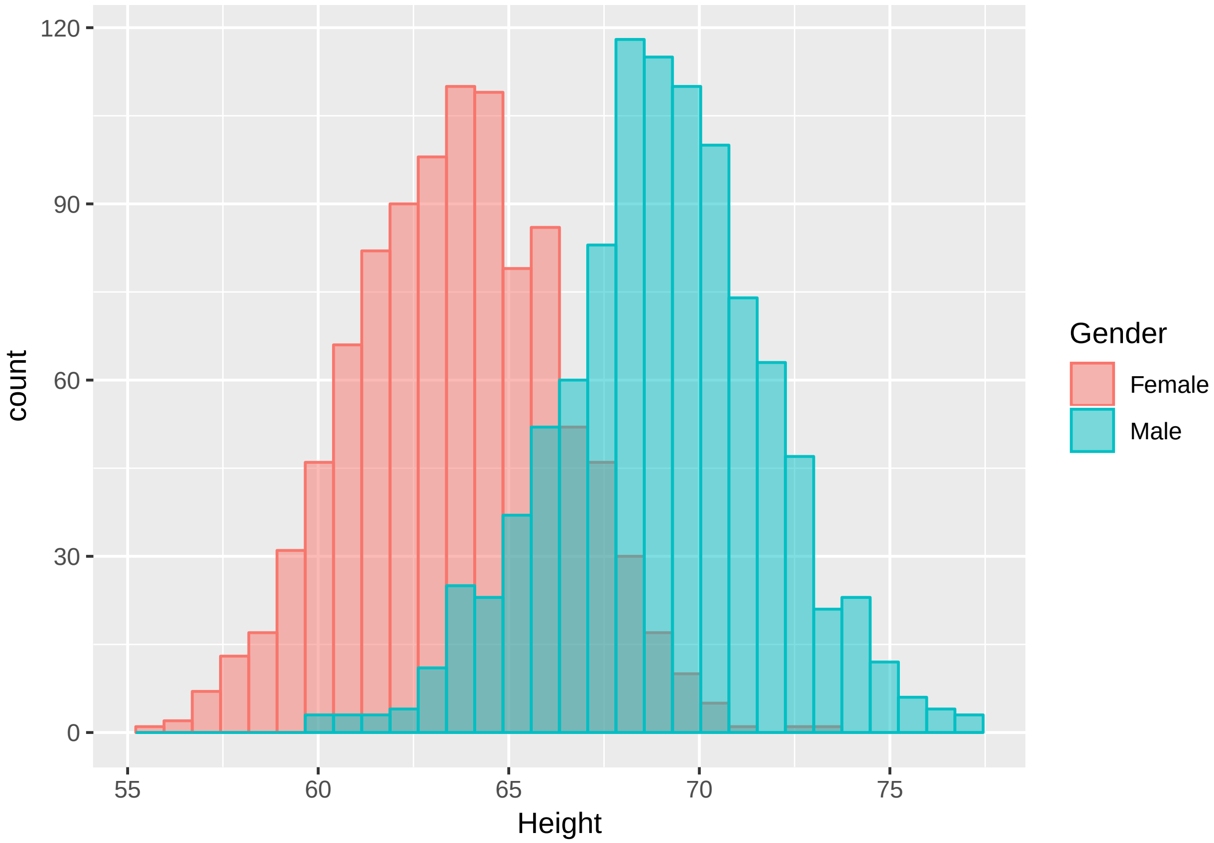
```
ggplot(heights, aes(x=Height)) + geom_histogram(position="identity", alpha=0.5)
```

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



```
ggplot(heights, aes(x=Height, fill=Gender, color=Gender)) + geom_histogram(position="identity", alpha=0
```

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



It looks like male and female heights come from two different distributions.

Given the answer to part (b), one way to think about this problem is that we have some proportion of men and women in the population (ω_m, ω_w) and the height of individual i depends on which subpopulation the person comes from. This new model is:

$$
\begin{aligned}
X_i &= \omega_w X_w + (1-\omega_w) X_m \\
X_w &\sim N(\mu_w, \sigma_w^2) \\
X_m &\sim N(\mu_m, \sigma_m^2) \\
\omega_w &\in [0,1]
\end{aligned}
$$

This is a hard estimation problem to code up (see Expectation Maximization), but luckily for us we can use 'bayesm' again.

Once again, recast this problem in a regression framework.
(Hint: What is the mean and variance of X_i?)}

$$X_i = \alpha + \beta Male_i + \epsilon_i$$

$$\epsilon \sim N(0, Var(X))$$

where, $Var(X) = \omega^2 \sigma_w^2 + (1-\omega)\sigma_m^2$

d. Using your answer to the last part, justify running a linear hierarchical mixture model regression. Use **rhierLinearMixture** to run this model.
(Note: **ncomp** option specifies the number of mixture components)}

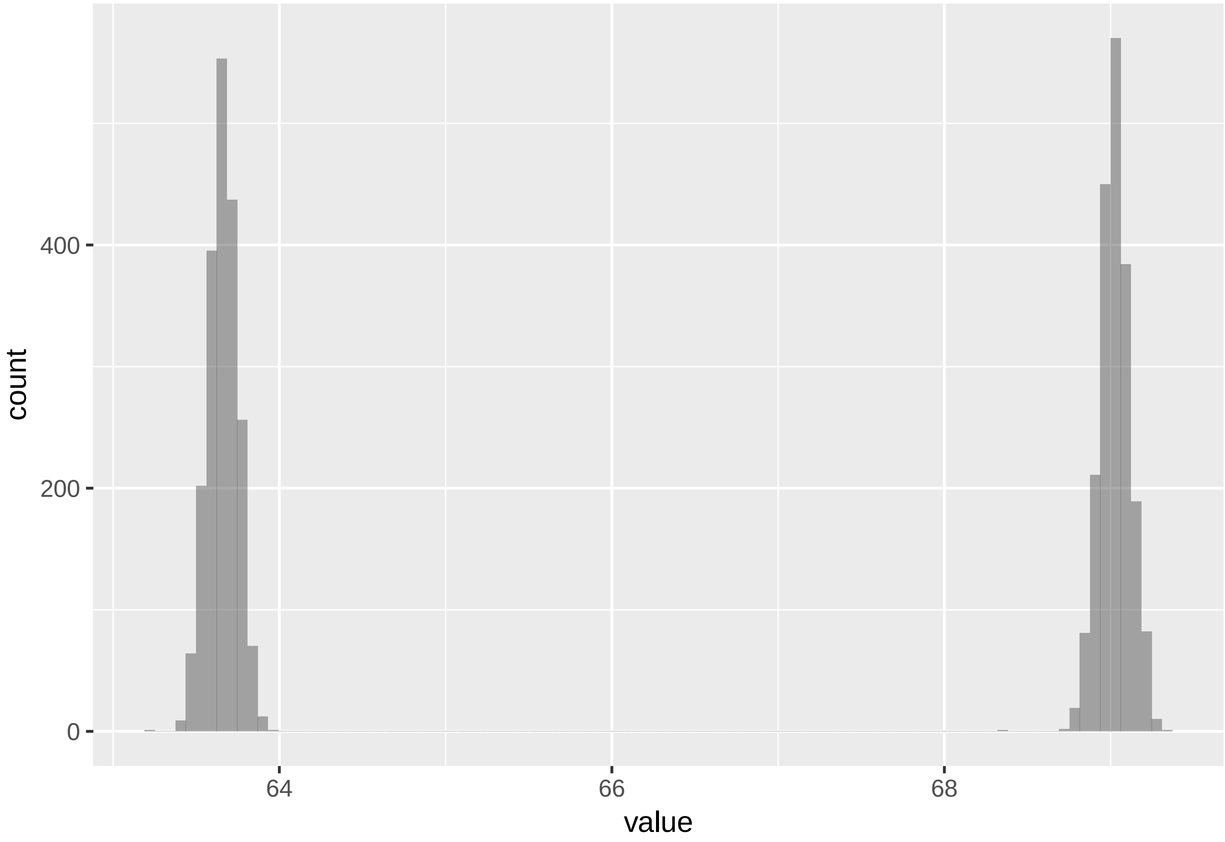
We can think of $Male_i \sim Bernoulli(\omega)$.

```
subjects=levels(heights$Gender)
n_subjects = length(subjects)

# Create matrix of potential covariates for our random effect
# Setting to 1 because our model assumes that there is a common mean across groups
Z=matrix(c(rep(0,n_subjects)), ncol=1)
nz=ncol(Z)

regdata=NULL    #<- This will be a named list to store the data
                # Each element of the list is the observations (both IV and DV) for each i
for(s in subjects){
  y = (heights %>% filter(Gender == s))$Height         # Create LHS variable
  X = matrix(c(rep(1,length(y))))
  regdata[[which(subjects==s)]] = list(y=y,X=X)                                 # For each unit i, add t
}

# Create data for MCMC
Data=list(regdata=regdata,Z=Z)
Mcmc=list(R=2000,keep=1)

# Run linear hierarchical model
out=rhierLinearMixture(Data=Data,Mcmc=Mcmc,Prior = list(ncomp=2))
```

```
##
## Starting MCMC Inference for Hierarchical Linear Model:
##    Normal Mixture with 2 components for first stage prior
##    for  2  cross-sectional units
##
## Prior Parms:
## nu.e = 3
## nu = 4
## V
##      [,1]
## [1,]    4
## mubar
##      [,1]
## [1,]    0
## Amu
##      [,1]
## [1,] 0.01
## a
## [1] 5 5
## deltabar
## [1] 0
## Ad
##      [,1]
## [1,] 0.01
##
## MCMC Parms:
## R=  2000  keep=  1  nprint=  100
##
##  MCMC Iteration (est time to end - min)
##  100 (0.0)
##   200 (0.0)
##   300 (0.0)
##   400 (0.0)
##   500 (0.0)
##   600 (0.0)
##   700 (0.0)
##   800 (0.0)
##   900 (0.0)
##   1000 (0.0)
##   1100 (0.0)
##   1200 (0.0)
##   1300 (0.0)
##   1400 (0.0)
##   1500 (0.0)
##   1600 (0.0)
##   1700 (0.0)
##   1800 (0.0)
##   1900 (0.0)
##   2000 (0.0)
##   Total Time Elapsed: 0.00
```

e. Using the last MCMC draw, plot a histogram of the beta estimates for men and women observations seperately. What do you notice about these new distributions?

```
ggplot(melt(out$betadraw[,,]), aes(x=value, color=Var1)) + geom_histogram(position="identity", alpha=0.5
```



The distributions are well seperated.