Bayesian Project

James Spillane(18342921)

Luke Finn(18391376)

## Data

**Let’s firstly take a look at the data checking for any irregularities and anything that may help us overall in displaying our results.**

#Import data and print head of the data  
dat <- read.csv('Bayesian Data.csv')  
head(dat)

## What.college.do.you.go.to. What.year.are.you.in. What.gender.are.you.  
## 1 NUIG 4th Year Male  
## 2 NUIG 4th Year Male  
## 3 NUIG 4th Year Male  
## 4 NUIG 4th Year Male  
## 5 NUIG 4th Year Male  
## 6 NUIG 4th Year Male  
## How.much.do.you.think.the.average.student.in.NUIG.spends.on.a.night.out.  
## 1 40  
## 2 50  
## 3 40  
## 4 70  
## 5 60  
## 6 70  
## What.year.do.you.expect.spends.the.most.money.  
## 1 1st Year  
## 2 1st Year  
## 3 1st Year  
## 4 2nd Year  
## 5 2nd Year  
## 6 2nd Year  
## How.much.have.you.spent.on.your.last.2.4.nights.out.on.alcohol....night.one...night.two......  
## 1 55  
## 2 41  
## 3 60  
## 4 50  
## 5 20  
## 6 70

**As we can see our columns seem to be in a format that makes it tedious to call in the future, let’s firstly rename these columns.**

#Let's firstly rename the column 'What college do you go to?':  
names(dat)[1] <- 'College'  
#Now let's rename the column 'what year are you in?':  
names(dat)[2] <- 'Year'  
#Rename the column 'What gender are you?':  
names(dat)[3] <- 'Gender'  
#Rename the column 'How much do you think the average student in NUIG spends on a night out?'  
names(dat)[4] <- 'Expected\_Spending'  
#Rename the column 'What year do you expect spends the most money?':  
names(dat)[5] <- 'Expected\_Year'  
#Rename the column 'How much have you spent on your last 2-4 nights out on alcohol?:  
names(dat)[6] <- 'Spending'  
head(dat)

## College Year Gender Expected\_Spending Expected\_Year Spending  
## 1 NUIG 4th Year Male 40 1st Year 55  
## 2 NUIG 4th Year Male 50 1st Year 41  
## 3 NUIG 4th Year Male 40 1st Year 60  
## 4 NUIG 4th Year Male 70 2nd Year 50  
## 5 NUIG 4th Year Male 60 2nd Year 20  
## 6 NUIG 4th Year Male 70 2nd Year 70

**As we can see our columns are now much easier to call. Let’s take a look at some of our columns which have numerical values and see if there are any outliers.**

dat$Expected\_Spending

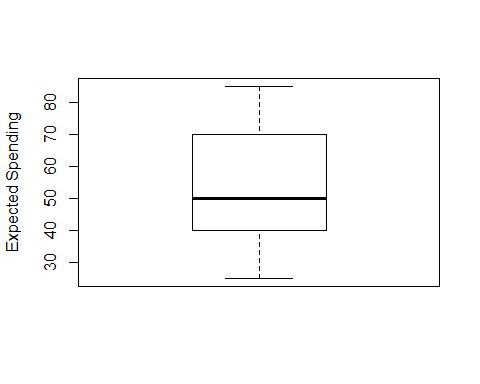
## [1] 40 50 40 70 60 70 70 70 55 50 60 55 50 25 30 40 50 30 80 70 85 50 60 40 70  
## [26] 65 60 50 55 50 40 35 45 30 70 40 75 50 70 50 70 30 50 50 25 40 80 40 50 40  
## [51] 70 60 70 50 60 40

dat$Spending

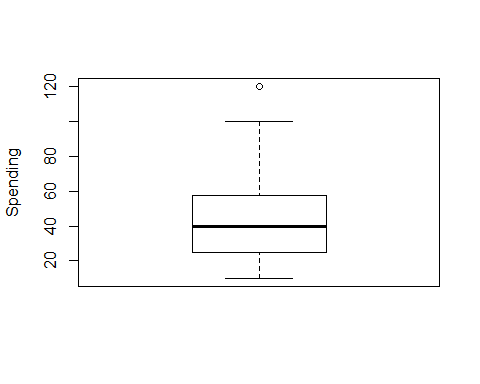
## [1] 55 41 60 50 20 70 80 80 75 40 65 40 52 25 21 25 40 30 40  
## [20] 30 30 40 35 40 40 45 20 20 20 55 40 50 65 10 100 43 60 25  
## [39] 80 25 70 20 40 90 18 35 60 30 30 20 30 20 120 15 50 30

**As we can see there doesn’t seem to be any crazy outliers in our data, to double check let’s plot these using a boxplot.**

boxplot(dat$Expected\_Spending,  
 ylab = "Expected Spending"  
)



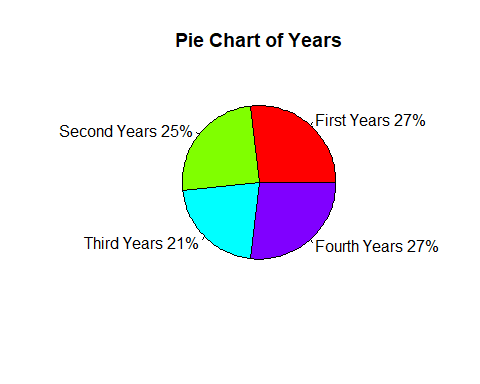
boxplot(dat$Spending,  
 ylab = "Spending"  
)



**As we can see from the boxplot for Expected Spending seems good i.e. seems to be no outliers however our boxplot for actual spending seems to have an outlier when spending=120. I don’t feel this is an outlier in our case as I have seen people spend that much on a night out and therefore will leave this alone.**

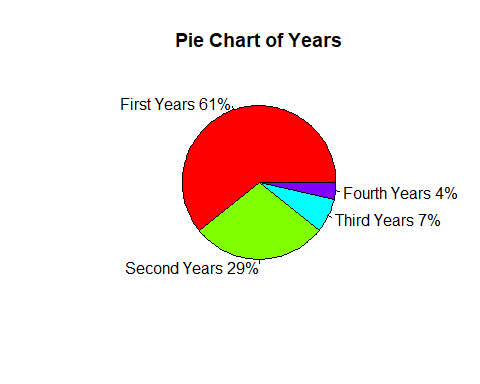
**Now let’s take a look to see if our data would be a good representation for each year i.e. is there a similar amount of data for each year.**

firstyr\_pct = sum(dat$Year=='1st Year')/length(dat$Year)  
secondyr\_pct = sum(dat$Year=='2nd Year')/length(dat$Year)  
thirdyr\_pct = sum(dat$Year=='3rd Year')/length(dat$Year)  
fourthyr\_pct = sum(dat$Year=='4th Year')/length(dat$Year)  
  
slices <- c(firstyr\_pct, secondyr\_pct, thirdyr\_pct, fourthyr\_pct)  
lbls <- c('First Years', 'Second Years', 'Third Years', 'Fourth Years')  
pct <- round(slices/sum(slices)\*100)  
lbls <- paste(lbls, pct) # add percents to labels  
lbls <- paste(lbls,"%",sep="") # ad % to labels  
pie(slices,labels = lbls, col=rainbow(length(lbls)),  
 main="Pie Chart of Years")



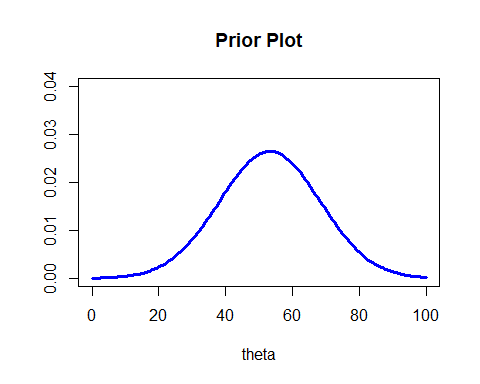
**There seems to be a good distribution of the data by Year group and therefore we can be happy that our findings wont be effected by lack of data for specific years. Now let’s take a look at what year the participants of the survey thought would spend the most on drink.**

firstyr\_pct1 = sum(dat$Expected\_Year=='1st Year')/length(dat$Expected\_Year)  
secondyr\_pct1 = sum(dat$Expected\_Year=='2nd Year')/length(dat$Expected\_Year)  
thirdyr\_pct1 = sum(dat$Expected\_Year=='3rd Year')/length(dat$Expected\_Year)  
fourthyr\_pct1 = sum(dat$Expected\_Year=='4th Year')/length(dat$Expected\_Year)  
  
slices <- c(firstyr\_pct1, secondyr\_pct1, thirdyr\_pct1, fourthyr\_pct1)  
lbls <- c('First Years', 'Second Years', 'Third Years', 'Fourth Years')  
pct <- round(slices/sum(slices)\*100)  
lbls <- paste(lbls, pct) # add percents to labels  
lbls <- paste(lbls,"%",sep="") # ad % to labels  
pie(slices,labels = lbls, col=rainbow(length(lbls)),  
 main="Pie Chart of Years")



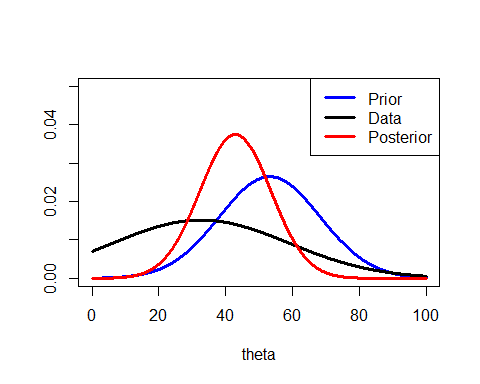
**To no surprise it is believed that first years spend the most on nights out according to our data. Let’s check to see if this is true. We will do this by comparing point estimates for posteriors for 1st years, 2nd years, 3rd years and 4th years.**

#Plot prior  
m0 <- mean(dat$Expected\_Spending)  
s0 <- sd(dat$Expected\_Spending)  
  
x <- seq(0, 100, length=100)  
Prior <- dnorm(x, mean=m0, sd=s0)  
plot(x, Prior, type='l',lwd=3,xlim = c(0,100), ylim=c(0,0.04),col = 'blue', main = 'Prior Plot', xlab = 'theta', ylab = '')



**As we can see our prior distribution is centred around 55 and follows a normal distribution**

#Plot prior, data and posterior  
firstyr\_spending <- dat$Spending[dat$Year=='1st Year']  
xbar\_1stYrs <- mean(firstyr\_spending)  
se\_1stYrs <- var(firstyr\_spending)/sqrt(length(firstyr\_spending))  
  
Prior <- c(m0, s0)  
Data\_1stYrs <- c(xbar\_1stYrs, s0)  
Posterior\_1stYrs <- round(normal\_update(Prior, Data\_1stYrs),3)  
x <- seq(0,100, length=1000)  
priorx\_1stYrs <- dnorm(x, mean=m0, sd=s0)  
datax\_1stYrs <- dnorm(x, mean=xbar\_1stYrs, sd=se\_1stYrs)  
postx\_1stYrs <- dnorm(x, mean=Posterior\_1stYrs[1], sd=Posterior\_1stYrs[2])  
plot(x, priorx\_1stYrs, type='l',lwd=3,xlim = c(0,100), ylim=c(0,0.05),col = 'blue', main = '', xlab = 'theta', ylab = '')  
lines(x, datax\_1stYrs,col='black',lwd=3)  
lines(x, postx\_1stYrs,col='red',lwd=3)  
legend("topright", c("Prior","Data","Posterior"), lty = 1, lwd= 3, col = c('blue','black','red'))

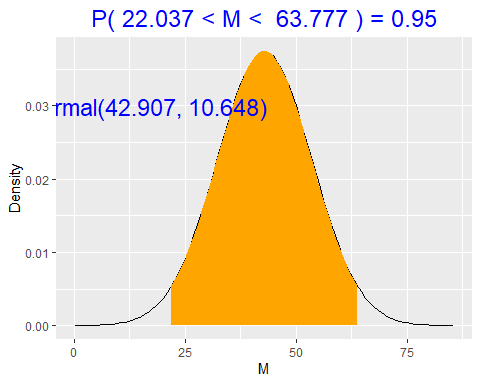


**As we can see our posterior has shifted(headed toward our data) to the left which indicates that 1st years may spend less than the average student. Let’s take a look at a point and interval estimate for theta.**

mysims\_1stYrs <- rnorm(10000, mean=Posterior\_1stYrs[1], sd=Posterior\_1stYrs[2])  
median(mysims\_1stYrs)

## [1] 42.91827

normal\_interval(0.95, Posterior\_1stYrs)

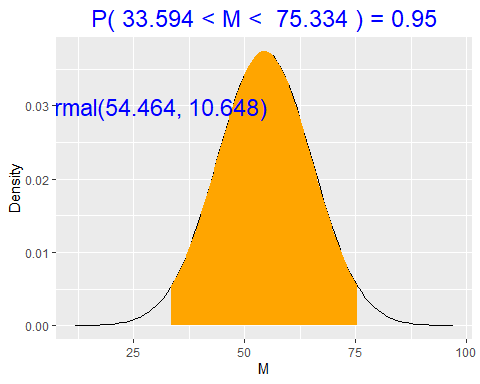


**As this is a normal distribution our mode=median=mean and will all be roughly around 42.75(for first years). With 95% probability 1st years spend between 22.04 and 63.78. We will take 42.75 to be our figure to compare to other years. Let’s repeat the steps above for various years and see which year does spend the most.**

#Second Years  
secondyr\_spending <- dat$Spending[dat$Year=='2nd Year']  
xbar\_2ndYrs <- mean(secondyr\_spending)  
se\_2ndYrs <- var(secondyr\_spending)/sqrt(length(secondyr\_spending))  
Data\_2ndYrs <- c(xbar\_2ndYrs, s0)  
Posterior\_2ndYrs <- round(normal\_update(Prior, Data\_2ndYrs),3)  
mysims\_2ndYrs <- rnorm(10000, mean=Posterior\_2ndYrs[1], sd=Posterior\_2ndYrs[2])  
median(mysims\_2ndYrs)

## [1] 54.37043

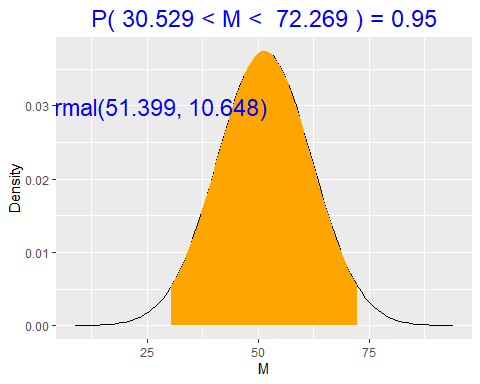
normal\_interval(0.95, Posterior\_2ndYrs)



#Third Years  
thirdyr\_spending <- dat$Spending[dat$Year=='3rd Year']  
xbar\_3rdYrs <- mean(thirdyr\_spending)  
se\_3rdYrs <- var(thirdyr\_spending)/sqrt(length(thirdyr\_spending))  
Data\_3rdYrs <- c(xbar\_3rdYrs, s0)  
Posterior\_3rdYrs <- round(normal\_update(Prior, Data\_3rdYrs),3)  
mysims\_3rdYrs <- rnorm(10000, mean=Posterior\_3rdYrs[1], sd=Posterior\_3rdYrs[2])  
median(mysims\_3rdYrs)

## [1] 51.14425

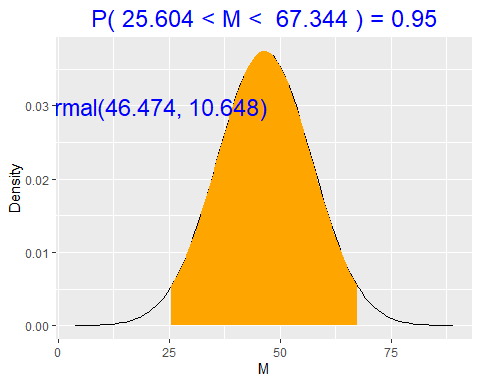
normal\_interval(0.95, Posterior\_3rdYrs)



#Fourth Years  
fourthyr\_spending <- dat$Spending[dat$Year=='4th Year']  
xbar\_4thYrs <- mean(fourthyr\_spending)  
se\_4thYrs <- var(fourthyr\_spending)/sqrt(length(fourthyr\_spending))  
Data\_4thYrs <- c(xbar\_4thYrs, s0)  
Posterior\_4thYrs <- round(normal\_update(Prior, Data\_4thYrs),3)  
mysims\_4thYrs <- rnorm(10000, mean=Posterior\_4thYrs[1], sd=Posterior\_4thYrs[2])  
median(mysims\_4thYrs)

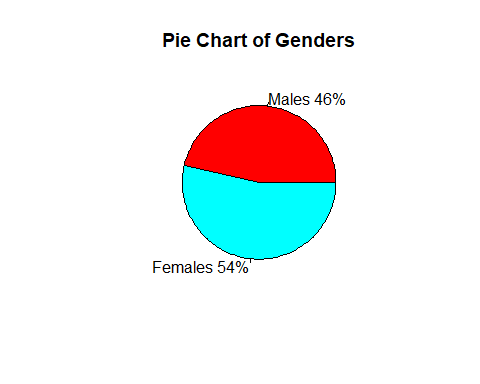
## [1] 46.58509

normal\_interval(0.95, Posterior\_4thYrs)



**As we can see First Years spend the least followed by fourth years then third years and finally second years spending the most on average. Is this surprising? I feel the survey users may have presumed that first year students spend more as they go out more but it makes sense that first years spend the least as they do usually go out more and therefore have less to spend. Let’s repeat the same steps above to see whether Males or Females spend more on nights out on average. Let’s firstly take a look at how many males and females we have data from.**

males = sum(dat$Gender=='Male')/length(dat$Gender)  
females = 1-males  
  
  
slices <- c(males, females)  
lbls <- c('Males', 'Females')  
pct <- round(slices/sum(slices)\*100)  
lbls <- paste(lbls, pct) # add percents to labels  
lbls <- paste(lbls,"%",sep="") # ad % to labels  
pie(slices,labels = lbls, col=rainbow(length(lbls)),  
 main="Pie Chart of Genders")

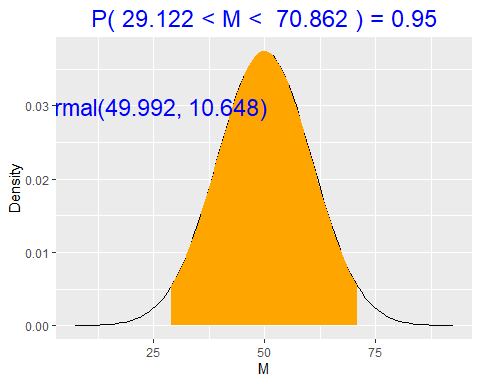


**As we can see we have a good split in the data of males to females and can therefore be more confident in our results. Let’s now create posteriors for both males and females and computer point estimates and 95% confidence intervals to analyse which gender spends more on average.**

#Males  
male\_spending <- dat$Spending[dat$Gender=='Male']  
xbar\_males <- mean(male\_spending)  
se\_males <- var(male\_spending)/sqrt(length(male\_spending))  
Data\_males <- c(xbar\_males, s0)  
Posterior\_males <- round(normal\_update(Prior, Data\_males),3)  
mysims\_males <- rnorm(10000, mean=Posterior\_males[1], sd=Posterior\_males[2])  
median(mysims\_males)

## [1] 49.96246

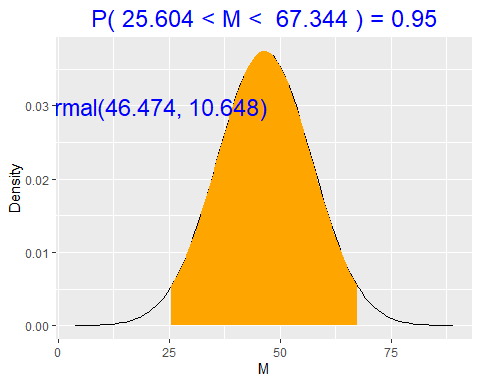
normal\_interval(0.95, Posterior\_males)



#Females  
female\_spending <- dat$Spending[dat$Gender=='Female']  
xbar\_females <- mean(female\_spending)  
se\_females <- var(female\_spending)/sqrt(length(female\_spending))  
Data\_females <- c(xbar\_females, s0)  
Posterior\_females <- round(normal\_update(Prior, Data\_4thYrs),3)  
mysims\_females <- rnorm(10000, mean=Posterior\_females[1], sd=Posterior\_females[2])  
median(mysims\_females)

## [1] 46.36452

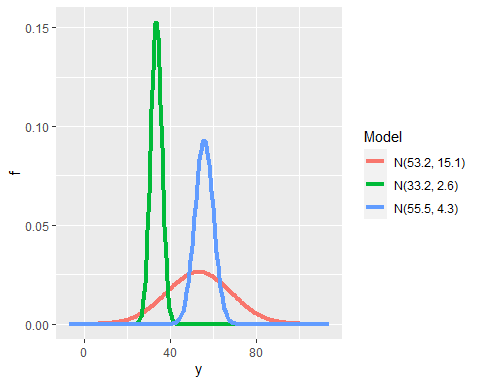
normal\_interval(0.95, Posterior\_females)



**From our data we can see that on average males spend more than females. With 95% probability men spend between 29.12 and 70.86, with 95% probability females spend between 25.60 and 67.34.**

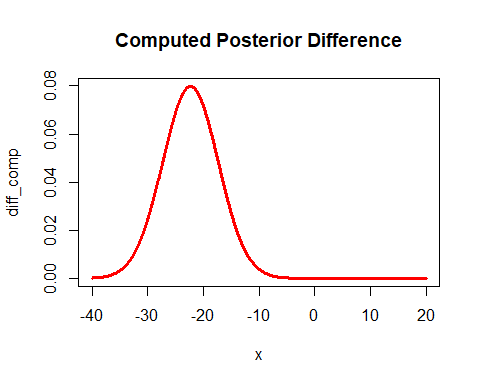
**A lecturer has taken a look at our findings and is interested that first years spend the least and second years spend the most. The lecturer asks you to investigate the posterior mean difference between first year students spending and second year students spending. Does a difference exist? If so, what is the difference?**

xbar1 <- mean(firstyr\_spending)  
xbar2 <- mean(secondyr\_spending)  
Prior <- c(m0, s0)  
  
se1 <- sd(firstyr\_spending)/sqrt(length(firstyr\_spending))  
se2 <- sd(secondyr\_spending)/sqrt(length(secondyr\_spending))  
DataFirstYears <- c(xbar1, se1)  
DataSecondYears <- c(xbar2, se2)  
Posterior1 <- normal\_update(Prior, DataFirstYears)  
Posterior2 <- normal\_update(Prior, DataSecondYears)  
many\_normal\_plots(list(Prior, Posterior1, Posterior2))

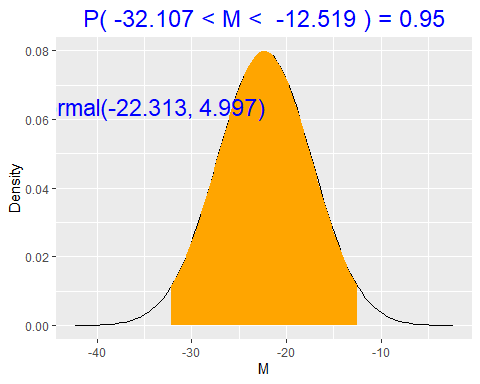


**As we can see in red our Prior data seems to be centred around 53 and is quite spread. We can see our posterior for first years represented by the green curve as we can see this seems to be centred around 33 and is much more precise than our prior. Finally we can see our posterior represented by the blue curve which seems to be less precise than our first year posterior but seems to be centred around 55 instead of 30 which suggests that second years do spend more money than first years. To check this lets calculate the difference between the posteriors**

PosteriorDiff <- c(Posterior1[1]-Posterior2[1], sqrt(Posterior1[2]^2+Posterior2[2]^2))  
x <- seq(-40,20, length=1000)  
diff\_comp <- dnorm(x, mean=PosteriorDiff[1], sd=PosteriorDiff[2])  
plot(x, diff\_comp, type='l', col='red',xlim = c(-40,20), lwd=3, main = 'Computed Posterior Difference')



normal\_interval(0.95, round(PosteriorDiff, 3))



**The posterior mean difference is between -32.107 and -12.519 euros with probability 0.95. Since this is strictly negative, we can see there is a high probability that first years do spend less than second years.With 95% probability first years spend between 12.52 and 32.11 less than second years. As 0 is not in the interval we can suggest that a difference does exist.**

**You report your findings to your lecturer and he informs you that he has asked some first years and they seem to be spending much more than the second years he has been talking to, he would like you to check against your model if these students may be outliers?**

1-pnorm(0, mean = PosteriorDiff[1], sd=PosteriorDiff[2])

## [1] 4.005258e-06

**The probability that the difference is greater than 0 is 0.0000004005258 i.e. the probability that first years spend more than second years is 0.0000004005258 and therefore we can conclude that these students are outliers.**

**We present the findings to the lecturer and he realises he had an error in his data with his updated data he has found that first years spend 13% less than second years. He wants us to use our bayesian analysis to test this claim.**

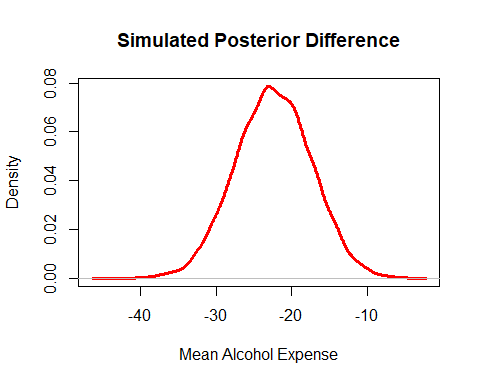
prop1stYears <- Posterior2[1]\*0.87  
pnorm(prop1stYears, mean=Posterior1[1], sd=Posterior1[2])

## [1] 1

**The probability that first years spend 13% less than second years is 100%.**

**Let’s take a look at doing this by simulation. Simulation is often used for more complex issue let’s see if it has any effect on our findings.**

firstyearsims <- rnorm(10000, mean=Posterior1[1], sd=Posterior1[2])  
finalyearsims <- rnorm(10000, mean=Posterior2[1], sd=Posterior2[2])  
diff\_sims <- firstyearsims - finalyearsims  
  
plot(density(diff\_sims), col='red', lwd=3, main='Simulated Posterior Difference', xlab= 'Mean Alcohol Expense')



**As we can see we have similar results when using simulated data, the simulated posterior difference distribution seems to be more bellshaped than our previous posterior difference but also seems to be centred around -25.**

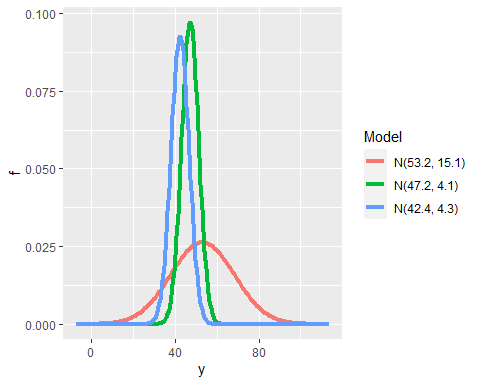
quantile(diff\_sims, probs= c(0.025, 0.975))

## 2.5% 97.5%   
## -32.26945 -12.57533

**As we can see our confidence intervals are almost identical and we can therefore be confident that both our ways were done correctly.**

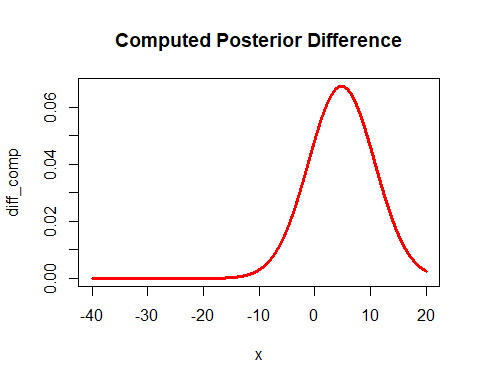
**Now let’s take a look to see if a difference does exist between males and females.**

xbar1 <- mean(male\_spending)  
xbar2 <- mean(female\_spending)  
  
se1 <- sd(male\_spending)/sqrt(length(male\_spending))  
se2 <- sd(female\_spending)/sqrt(length(female\_spending))  
DataMales <- c(xbar1, se1)  
DataFemales <- c(xbar2, se2)  
Posterior1 <- normal\_update(Prior, DataMales)  
Posterior2 <- normal\_update(Prior, DataFemales)  
many\_normal\_plots(list(Prior, Posterior1, Posterior2))

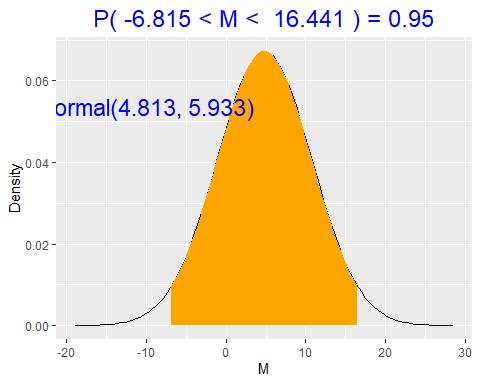


**The posterior for males is centred around 47.2, the posterior for females is centred around 42.4. We can see that both posteriors seem to be very close and because of this I feel like a difference does not exist, let’s take a look to see if this is true.**

PosteriorDiff <- c(Posterior1[1]-Posterior2[1], sqrt(Posterior1[2]^2+Posterior2[2]^2))  
x <- seq(-40,20, length=1000)  
diff\_comp <- dnorm(x, mean=PosteriorDiff[1], sd=PosteriorDiff[2])  
plot(x, diff\_comp, type='l', col='red',xlim = c(-40,20), lwd=3, main = 'Computed Posterior Difference')



normal\_interval(0.95, round(PosteriorDiff, 3))

 **With 95% probability the posterior mean difference in spending for males and females is between -6.82 and 16.44. As 0 is included in the interval we cannot suggest that a difference exists.**

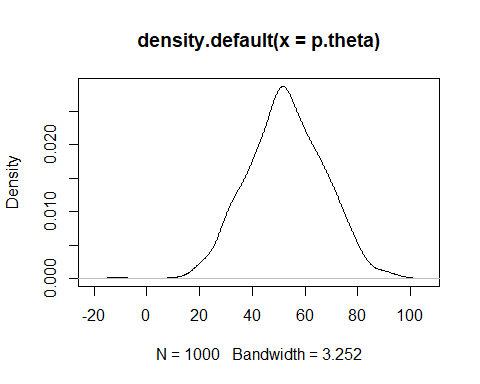
##RJags

**Thus far, we have worked with prior distributions and likelihoods of a sufficiently convenient form to simplify the construction of the posterior. In reality, we need to be able to work with much more complex models which relate to real life problems. RJags allows us to specify prior distributions that accuractly reflect the prior information rather than use a prior of a mathematically convenient form. i.e. allows us to create a much more accurate prior. One thing you need to be careful with when using RJags on a complex model with hundreds of parameters is the curse of dimensionality, as our data is quite simple we do not need to worry about this.RJags uses the Gibbs Sampling algorithm which is another way of constructing a transition kernel to produce a Markov chain with the desired target output. The aim of the Gibbs sampler is to make sampling from a high-dimensional distribution more tractable by sampling from a collection of more manageable smaller dimensional distributions.**

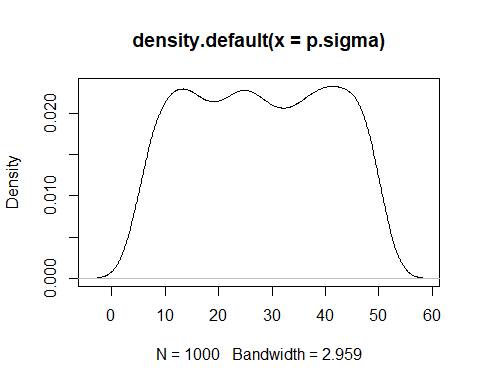
#Hyperior distribution  
p.theta <- rnorm(1000, m0, s0)  
p.sigma <- runif(1000, 5, 50)  
#Store samples in a dataframe  
samples <- data.frame(p.theta, p.sigma)  
head(samples)

## p.theta p.sigma  
## 1 79.50983 17.49345  
## 2 53.68902 47.14461  
## 3 32.28390 17.05029  
## 4 74.25984 46.43724  
## 5 62.15381 47.04037  
## 6 74.11116 44.31514

#Plot density plots of hyperprior values  
plot(density(p.theta))



plot(density(p.sigma))



**I specified the hyperiors for theta following a normal distribution with our mean and standard deviation and the hyperpriors for sigma following a uniform distribution with min=5 and max=50. We can see our simulated values for theta and sigma in our dataframe and they look good. Let’s now define and compile our model.**

library(rjags)

## Warning: package 'rjags' was built under R version 3.6.3

## Loading required package: coda

## Warning: package 'coda' was built under R version 3.6.3

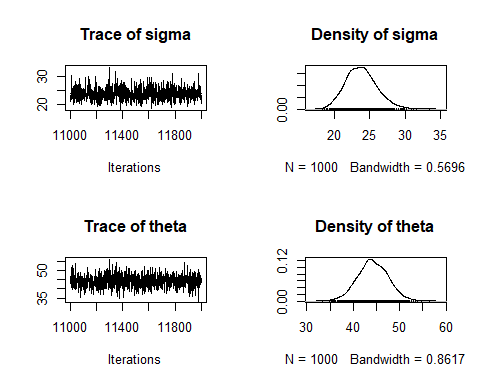
## Linked to JAGS 4.3.0

## Loaded modules: basemod,bugs

# DEFINE the model  
spending\_model <- "model{  
 # Likelihood model for X[i]  
 for(i in 1:length(X)) {  
 X[i] ~ dnorm(theta,sigma^(-2))  
 }  
   
 # Prior models for theta and sigma  
 theta ~ dnorm(53.21, 15.06^(-2))  
 sigma ~ dunif(5, 50)  
 }"  
  
#Compile the model  
spending\_jags <- jags.model(textConnection(spending\_model), data = list(X = dat$Spending),  
 inits = list(.RNG.name = "base::Wichmann-Hill", .RNG.seed = 1989))

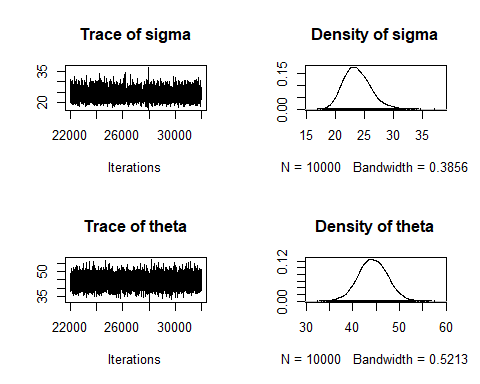
## Compiling model graph  
## Resolving undeclared variables  
## Allocating nodes  
## Graph information:  
## Observed stochastic nodes: 56  
## Unobserved stochastic nodes: 2  
## Total graph size: 66  
##   
## Initializing model

#Burn in 10000 iterations  
spending\_sim <- update(spending\_jags, n.iter = 10000)  
  
#Simulate the posterior  
spending\_sim <- coda.samples(model=spending\_jags, variable.names=c("theta","sigma"), n.iter=1000)  
plot(spending\_sim)



**As we can see our trace plots are not well mixed with an obvious pattern emerging therefore let’s increase the number of simulations from the the posterior to see if this helps.**

spending\_sim <- update(spending\_jags, n.iter = 10000)  
spending\_sim <- coda.samples(model=spending\_jags, variable.names=c("theta","sigma"), n.iter=10000)  
plot(spending\_sim)



#summarise the simulations  
summary(spending\_sim)

##   
## Iterations = 22001:32000  
## Thinning interval = 1   
## Number of chains = 1   
## Sample size per chain = 10000   
##   
## 1. Empirical mean and standard deviation for each variable,  
## plus standard error of the mean:  
##   
## Mean SD Naive SE Time-series SE  
## sigma 23.74 2.302 0.02302 0.03077  
## theta 44.33 3.103 0.03103 0.03118  
##   
## 2. Quantiles for each variable:  
##   
## 2.5% 25% 50% 75% 97.5%  
## sigma 19.72 22.12 23.54 25.19 28.81  
## theta 38.19 42.23 44.28 46.42 50.44

#Store the chains in a data frame  
spending\_chains <- data.frame(sim=1:10000, spending\_sim[[1]])  
#Check out the head of spending\_chains  
head(spending\_chains)

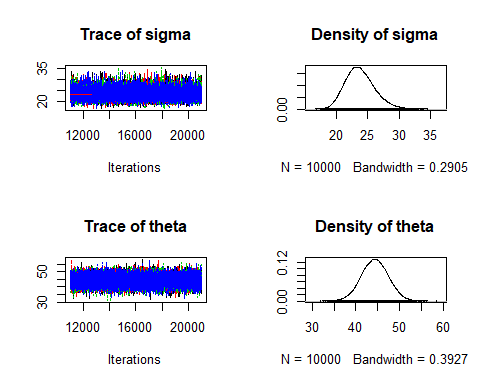
## sim sigma theta  
## 1 1 24.79417 43.20901  
## 2 2 24.09294 48.69577  
## 3 3 23.90030 43.62516  
## 4 4 23.80544 42.82647  
## 5 5 21.87210 43.73327  
## 6 6 24.56996 43.86349

**Our traceplots are now very well mixed, the density of sigma is right skewed and the density of theta is symmetrical. We can also see that the mean standard error ratio is large suggesting that the chain has converged. To get a better idea of how the model is behaving we will repeat the above steps n times where n is the number of chains. This allows us initialize with n random starting points and if all the chains converge to the same distribution then that tells us that we have ended up at that unique stationary distribution that is our posterior of interest. This can be easily done in RJags by specifying n.chains=n when compiling the model.**

#Compile the model using m=4 chains  
spending\_jags\_multi <- jags.model(textConnection(spending\_model), data=list(X=dat$Spending), n.chains=4)

## Compiling model graph  
## Resolving undeclared variables  
## Allocating nodes  
## Graph information:  
## Observed stochastic nodes: 56  
## Unobserved stochastic nodes: 2  
## Total graph size: 66  
##   
## Initializing model

#Burn in 10000 iterations  
spending\_sim\_multi <- update(spending\_jags\_multi, n.iter=10000)  
  
#Simulate the posterior  
spending\_sim\_multi <- coda.samples(model=spending\_jags\_multi, variable.names=c("theta","sigma"), n.iter=10000)  
#Plot results  
plot(spending\_sim\_multi)



**We cannot interpret much from the above trace plots therefore let’s present these trace plots in an easy and interpretable way.**

library(dplyr)

## Warning: package 'dplyr' was built under R version 3.6.3

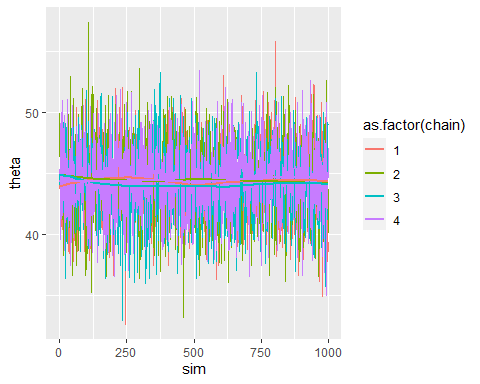
##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

#Initialize dataframe  
spending\_chains\_multi <- data.frame(sim=rep(1:10000,4),  
 chain=c(rep(1,10000),  
 rep(2,10000),  
 rep(3,10000),  
 rep(4,10000)),  
 rbind(spending\_sim\_multi[[1]],  
 spending\_sim\_multi[[2]],  
 spending\_sim\_multi[[3]],  
 spending\_sim\_multi[[4]]))  
#Plot multiple chain trace plots in ggplot  
spending\_chains\_multi %>% filter(sim<1000) %>%  
ggplot(aes(x=sim,y=theta,color=as.factor(chain))) + geom\_line() +  
 geom\_smooth(aes(color=as.factor(chain)),se=FALSE)

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

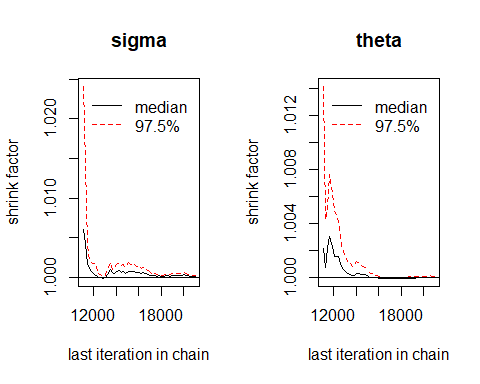


**Taking a look at the mean line there are some slight discrepancies but there are no vast differences between the chains. Note: this is only for the first 1000 simulations but it would follow a similar pattern for all of the 10000 simulations. This can be quite tedious and sometimes difficult to analyse therefore instead we analyse the Gelman-Ruben statistic which measures the between chain variability and then with each of the chains look at the variability within that chain. It then gets a weighted sum of the between chain and within chain variances from which we can calculate our Gelman-Ruben Statistic. Let’s calculate the Gelman-Ruben statistic for our model.**

#Calculate Gelman-Ruben statistic  
gelman.diag(spending\_sim\_multi)

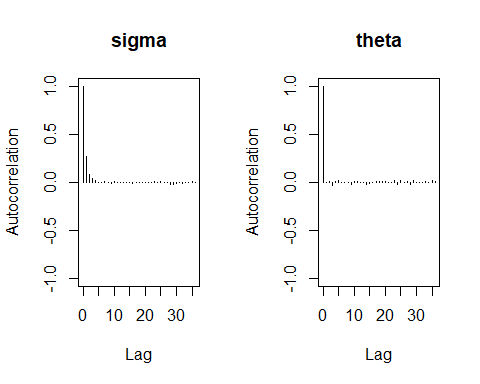
## Potential scale reduction factors:  
##   
## Point est. Upper C.I.  
## sigma 1 1  
## theta 1 1  
##   
## Multivariate psrf  
##   
## 1

#Gelman-Ruben plot  
gelman.plot(spending\_sim\_multi)



**As we can see our Gelman\_Ruben statistic is equal to 1 suggesting that the chains have converged. Also taking a look at our plots we can see that our first plot seems to very quickly converge to 1 over iterations. This is not quite as smooth for our second plot but taking a look at the Y-axis for our second plot we can see that it is very small and therefore we do not need to be worried about this. Finally, we should check for autocorrelation i.e. is there serial correlation in my iterations from simulation to simulation, why is it important to check for autocorrelation? Excessive autocorrelation may indicate issues with the model specification which may be caused by clustering in the data not being accounted for.**

autocorr.plot(spending\_sim)

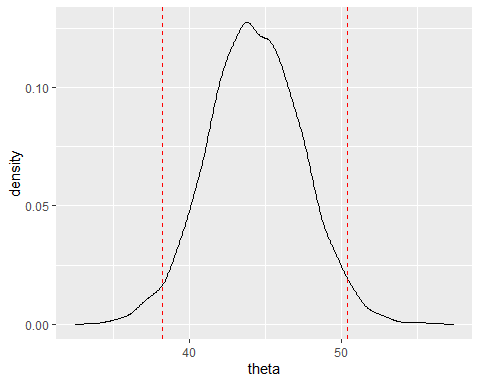


**Looking at our graphs we can see that for both sigma and theta there only seems to be minimal autocorellation and therefore we should be confident that we have correctly specified our model. Let’s now calculate plot and interpret a 95% credible interval for theta the mean university spending**

ci95 <- quantile(spending\_chains$theta, probs=c(0.025, 0.975))  
ci95

## 2.5% 97.5%   
## 38.19439 50.44050

#Plot 95% credible interval  
ggplot(spending\_chains, aes(x=theta)) + geom\_density() +   
geom\_vline(xintercept = ci95, col = 'red', lty = 'dashed')



**The mean spending for students in NUIG on alcohol on a night out is between 38.08 and 50.50 euros with 95% probability. Let’s take a look at what the population thought is the mean spending for students in NUIG on alcohol on a night out and calculate the probability of spending being greater than his figure.**

mean(dat$Expected\_Spending)

## [1] 53.21429

#Calculate the probabiulity that the mean spending for students in NUIG on alcohol on a night out is greater than 53.21429  
sum(spending\_sim[[1]]>53.21429)/length(spending\_sim[[1]])

## [1] 0.00135

**The probability that spending is greater than 53.21 is 0.00135.**

##Bayesian Regression

**In Bayesian Linear Regression we formulate linear regression using probability distributions rather than point estimates. The reponse y is not estimated as a single value but is assumed to be drawn from a probability distribution. We have to firstly convert our qualitative variables to numerical values, let’s do this.**

#We need to change Year, Gender and Expected\_Year to a numerical value (regression cannot interpret qualitative data)  
  
#Duplicating data  
dat2 <- dat  
#Converting column to character  
dat2$Expected\_Year <- as.character(dat2$Expected\_Year)  
dat2$Year <- as.character(dat2$Year)  
dat2$Gender <- as.character(dat2$Gender)  
#Male =0 , Female=1  
dat2[dat2 == 'Male'] <- '1'  
dat2[dat2 == 'Female'] <- '2'  
#4th Year = 4, 3rd Year = 3, etc  
dat2[dat2 == "4th Year"] <- "4"  
dat2[dat2 == "3rd Year"] <- "3"  
dat2[dat2 == "2nd Year"] <- "2"  
dat2[dat2 == "1st Year"] <- "1"  
#Converting column back to factor  
dat2$Year <- as.factor(dat2$Year)  
dat2$Year <- as.factor(dat2$Year)  
dat2$Gender <- as.factor(dat2$Gender)

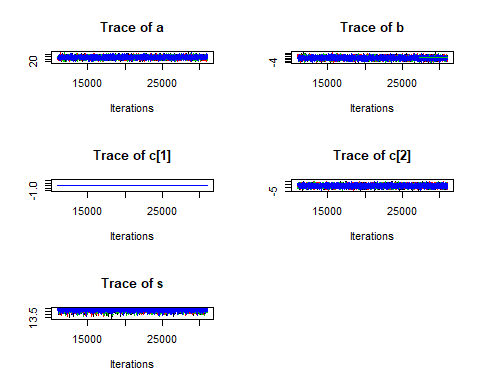
**We now must define our prior distributions for our regression parameters. Let be the intercept, be the coefficient for Year and be the coefficient for gender. Our response y represents spending.**

#Let's define the prior distributions  
#b\_0 represents the average spending for males in 1st Year  
  
prior\_b0 <- rnorm(10000, mean=mean(male\_spending), sd=sd(male\_spending))  
prior\_b1 <- rnorm(10000, mean=5, sd=5)  
prior\_b2 <- rnorm(10000, mean=5, sd=7.5)

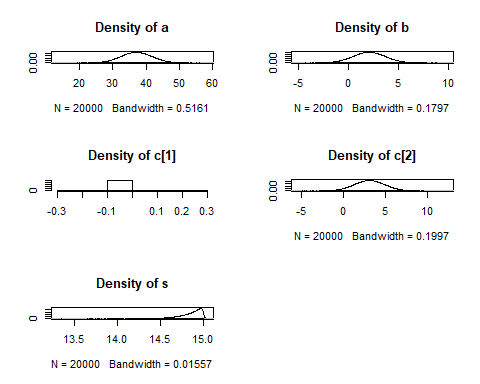
#Define the model  
regress\_model <- "model{  
 # Define model for data Y[i]  
 for(i in 1:length(Y)) {  
 Y[i] ~ dnorm(m[i], s^(-2))  
 m[i] <- a + b\*X1[i] + c[X2[i]]  
 }  
  
 # Define the a, b, c and s priors  
 a ~ dnorm(47, 22^(-2))  
 b ~ dnorm(5, 5^(-2))  
 c[1] <- 0  
 c[2] ~ dnorm(5, 2^(-2))  
 s ~ dunif(0, 15)  
}"   
  
regress\_jags <- jags.model(textConnection(regress\_model),   
 data = list(Y = dat2$Spending, X1 = dat2$Year, X2 = dat2$Gender),  
 n.chains = 4)

## Compiling model graph  
## Resolving undeclared variables  
## Allocating nodes  
## Graph information:  
## Observed stochastic nodes: 56  
## Unobserved stochastic nodes: 4  
## Total graph size: 195  
##   
## Initializing model

#Burn in model  
regress\_sim <- update(regress\_jags, n.iter=10000)  
regress\_sim <- coda.samples(model=regress\_jags, variable.names = c("a","b","c","s"), n.iter=20000)  
plot(regress\_sim, density=FALSE)



plot(regress\_sim, trace=FALSE)

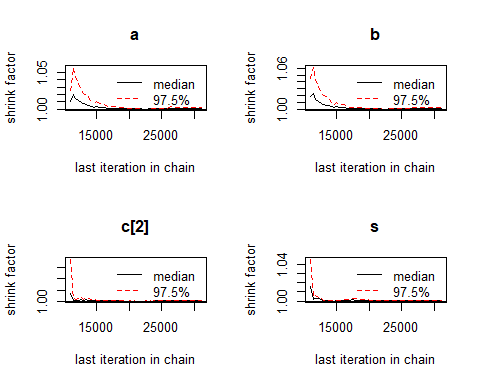


**Taking a look at the density of a it seems to follow a normal distribution centred around 37, the density of b also seems to follow a normal distribution centred around 1.5. The density of c[2] also follows a normal distribution centred around 3. The density of s is negatively skewed. As we have used multiple chains it is difficult to analyse the trace plots for convergence therefore let’s print out our gelman-ruben statistics and also plot them to ensure our models have converged.**

gelman.diag(regress\_sim[,-3])

## Potential scale reduction factors:  
##   
## Point est. Upper C.I.  
## a 1 1  
## b 1 1  
## c[2] 1 1  
## s 1 1  
##   
## Multivariate psrf  
##   
## 1

gelman.plot(regress\_sim[,-3])



**As we can see our gelman-ruben statistic is equal to 1 suggesting that our model has converged. Taking a look at our gelman ruben plots we can see that the model converged quite quickly to 1 and seems to remain at 1 for the majority of the time. Let’s now take a look at making predictions. Let’s now take a look at how each of the independent variables effect our predictor**

summary(regress\_sim)[2]

## $quantiles  
## 2.5% 25% 50% 75% 97.5%  
## a 28.0216073 34.014301 37.106207 40.273841 46.254477  
## b -1.1011513 1.000779 2.095039 3.184810 5.267388  
## c[1] 0.0000000 0.000000 0.000000 0.000000 0.000000  
## c[2] -0.3555516 1.948271 3.155954 4.369317 6.704973  
## s 14.4030312 14.760806 14.878103 14.949055 14.995526

**From the above table we can see that while keeping all other variables constant for one unit increase in Year spending will go up by 1.851158. The 95% credible interval for year is -1.19 and 4.89, as 0 is included in this interval we cannot suggest that Year has an effect on alcohol spending. We can also see that while keeping all other variables constant if the person is a female they would be expected to spend 3.104119 more than a female on a night out.The 95% credible interval for for gender is between -0.42 and 6.63. Now, let’s use our model to make a prediction about a 3rd year male student.**

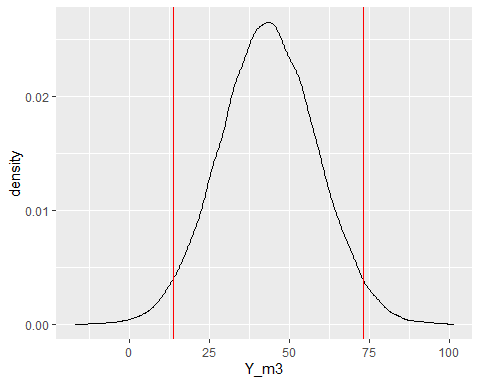
library(dplyr)  
#Save the simulations as a df  
regress\_chains <- data.frame(regress\_sim[[1]])  
#work out mean spending for a 3rd year male student  
regress\_chains <- regress\_chains %>% mutate(m\_m3 = a+ b\*3 + c.1.)  
#Simulate from normal dist   
regress\_chains <- regress\_chains %>% mutate(Y\_m3 = rnorm(n=length(regress\_chains$m\_m3), mean=m\_m3, sd=s))  
#Print first 6 predictions  
head(regress\_chains)

## a b c.1. c.2. s m\_m3 Y\_m3  
## 1 26.78566 5.424817 0 7.1216453 14.71220 43.06011 43.62223  
## 2 29.27091 3.130280 0 3.6610231 14.74791 38.66175 47.57177  
## 3 31.58050 4.665041 0 3.0859427 14.88685 45.57562 47.68745  
## 4 29.44682 4.722303 0 -1.4838731 14.83171 43.61373 45.63764  
## 5 31.17378 5.287257 0 0.6687454 14.84948 47.03555 68.86778  
## 6 33.45118 4.505068 0 1.3897581 14.84639 46.96638 38.82788

#Construct a posterior credible interval for the prediction  
ci <- quantile(regress\_chains$Y\_m3, c(0.025, 0.975))  
ci

## 2.5% 97.5%   
## 13.84365 72.96789

ggplot(regress\_chains, aes(x = Y\_m3)) + geom\_density() +   
 geom\_vline(xintercept = ci, color = "red")



**With 95% probability the spending of a male 3rd year nuig student on a night out would be between 13.94 and 72.91**

**Hierarchical modelling is another modelling technique which constructs prior distributions in a hierarchical fashion. In a hierarchical model, parameters have distributions which are conditional on additional parameters(hyperparameters). Hierarchical models are often used for complex models which have hundreds of variables and therefore I won’t be creating one for my data as I feel it is too simple to highlight the benefit of using these hierarchical models.**