

## Bayesian Computation with R - Chapter 4 Exercises

```
library(LearnBayes)
library(ggplot2) # not part of this book
```

### Question 1 - Inference about a normal population

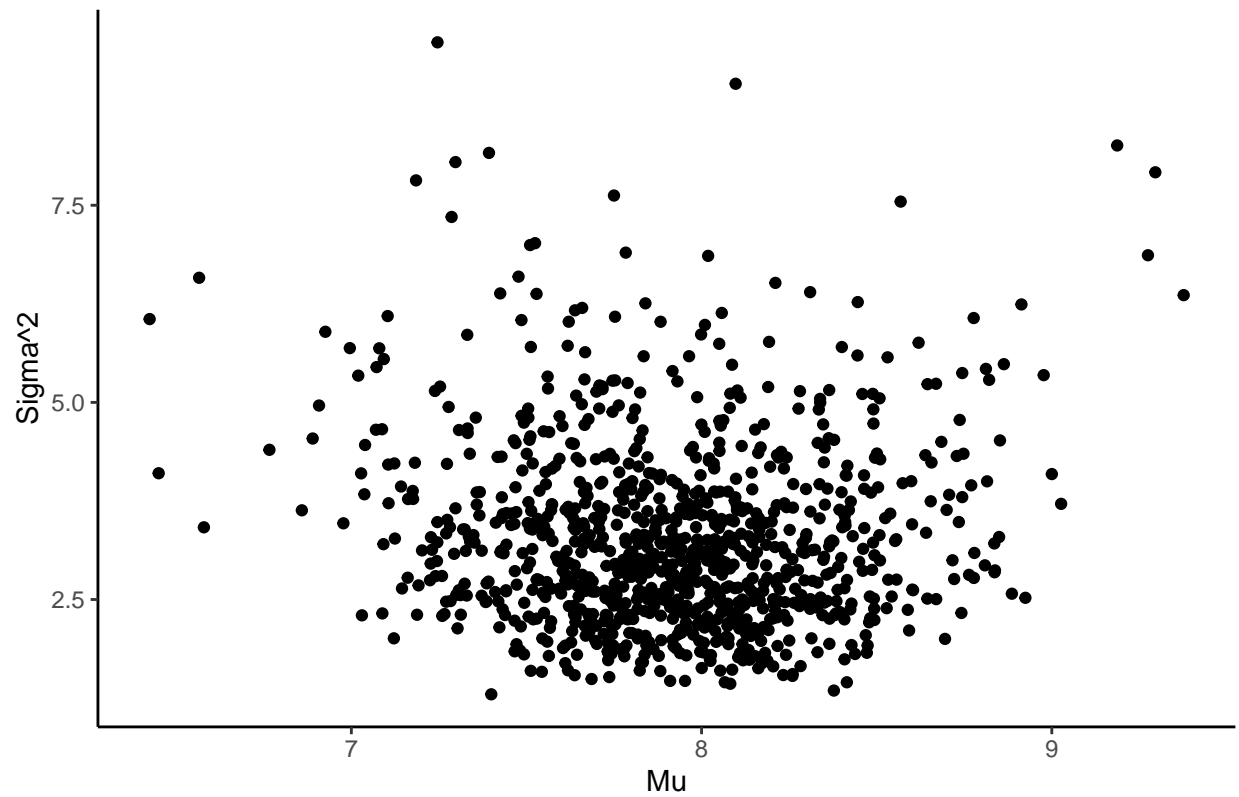
```
ys <- c(9,8.5,7,8.5,6,12.5,6,9,8.5,7.5,8,6,9,8,7,10,9,7.5,5,6.5)

## 1
## Emulate section 4.2
S <- sum((ys - mean(ys))^2)
n <- length(ys)
sigma2s <- S/rchisq(1000,n-1)
mus <- rnorm(1000, mean=mean(ys), sd=sqrt(sigma2s)/sqrt(n))

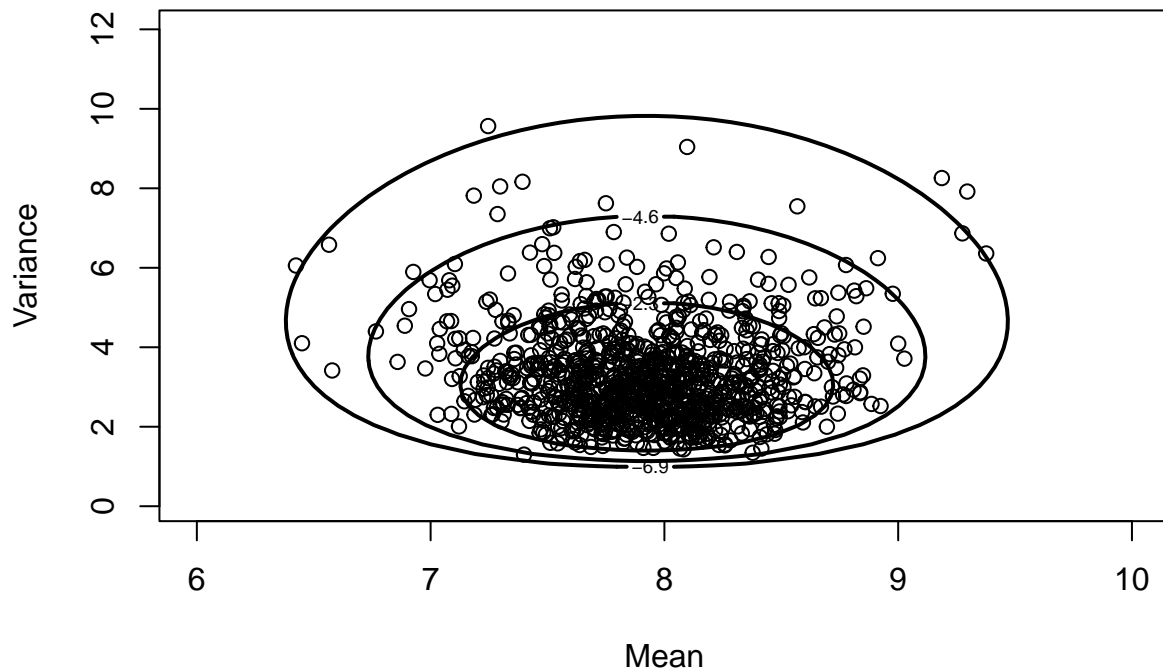
## Plot the simulations
plot_df <- cbind.data.frame(sigma2s,mus)

ggplot(plot_df,aes(mus,sigma2s)) +
  geom_point() +
  ggtitle("The simulated value pairs for variance and mean") +
  xlab("Mu") +
  (ylab("Sigma^2")) +
  theme_classic()
```

The simulated value pairs for variance and mean



```
## Plot using book's functions and contours  
d <- mycontour(normchi2post,c(6,10,0.1,12),ys,xlab="Mean",ylab="Variance")  
points(mus,sigma2s)
```



```
## 2
ps <- c(0.05,0.95)

## 90% CI for Mu
quantile(mus,ps)

##      5%      95%
## 7.252931 8.598823

## 90% CI for Sigma
quantile(sqrt(sigma2s),ps)

##      5%      95%
## 1.348997 2.363270

## 3
## Just do a brute force estimation on our sims
p75s <- mus + 0.674 * sqrt(sigma2s)

## mean and sd
mean(p75s)

## [1] 9.121894

sd(p75s)

## [1] 0.4506371
```

## Question 2 - The Behrens-Fisher problem

```
## 1
## This is a pen-and-paper exercise
## Generally, just repeat the equations in section 4.2
## To show independence, we just show that the joint distribution is equal to the independent distribut.

## 2
## our answer follows from part 1
## Because the distributions are independent, we can simply simulate values from the two distributions

## 3
yms <- c(120,107,110,116,114,111,113,117,114,112)
Sm <- sum((yms - mean(yms))^2)
nm <- length(yms)
sigma2ms <- Sm/rchisq(1000,nm-1)
mums <- rnorm(1000, mean=mean(yms), sd=sqrt(sigma2ms)/sqrt(nm))

yfs <- c(110,111,107,108,110,105,107,106,111,111)
Sf <- sum((yfs - mean(yfs))^2)
nf <- length(yfs)
sigma2fs <- Sf/rchisq(1000,nf-1)
mufs <- rnorm(1000, mean=mean(yfs), sd=sqrt(sigma2fs)/sqrt(nf))

mu_diffs <- mums - mufs

## compute an 90% CI on the mu differences, if this interval excludes 0, then we think the means are di.
ps <- c(0.05,0.95)
quantile(mu_diffs,ps)

##          5%          95%
## 2.298773 7.412856

## We compute a 90% credible interval that male golden jackals are 2.23-7.14 mm longer than females
```

## Question 3 - Comparing 2 proportions

```
N <- c(1601,162527)
S <- c(510,412368)

## 1
## Another pen and paper exercise that I won't include in this Rmarkdown
## Assume this is proven for the rest of the exercise

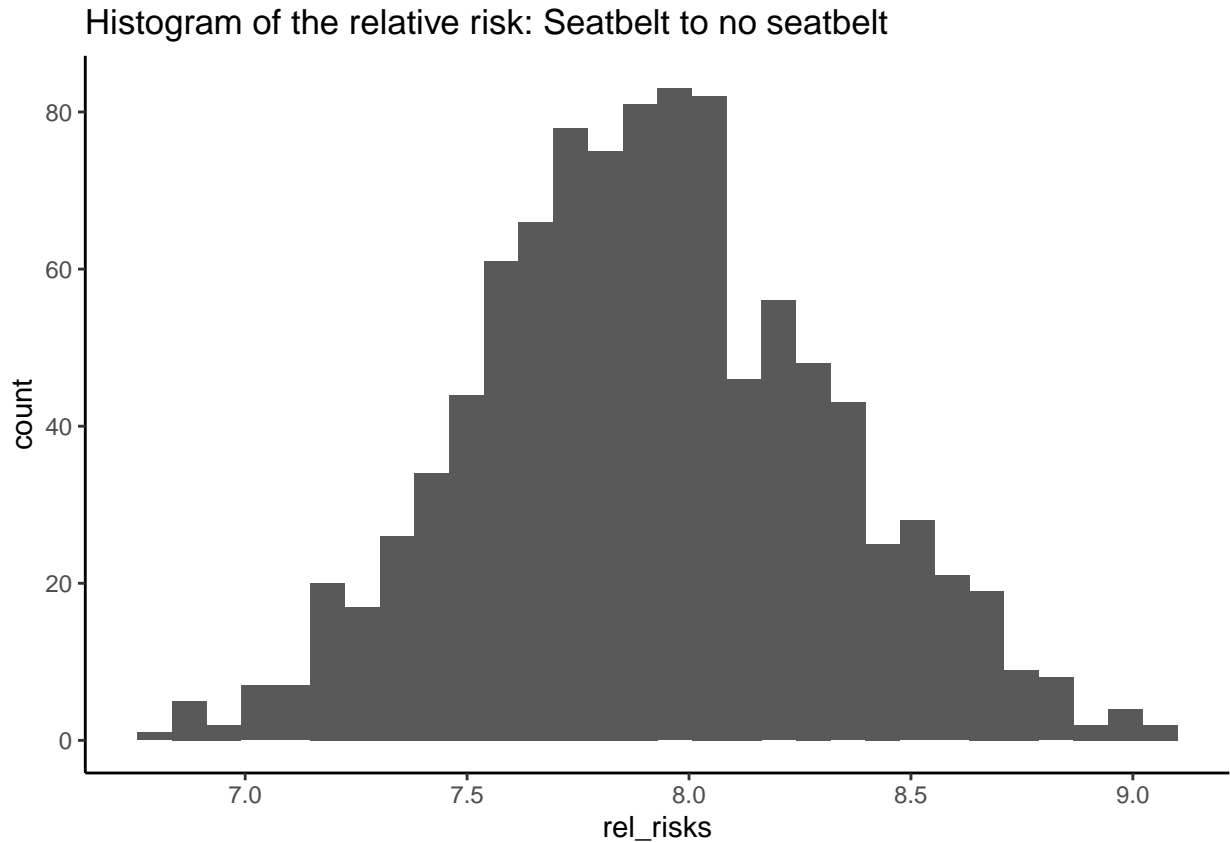
## 2
## The choice of a uniform prior is equivalent to a beta(1,1) prior
## Thus, we can use the beta(1,1) as a conjugate prior
## and the independent posterior distributions are given by beta(1+y,1+n-y)

pNs<- rbeta(1000,1+N[1],1+N[2])
pSs<- rbeta(1000,1+S[1],1+S[2])

rel_risks <- pNs / pSs
```

```
plot_df <- cbind.data.frame(rel_risks)
ggplot(plot_df,aes(rel_risks)) +
  geom_histogram() +
  ggtitle("Histogram of the relative risk: Seatbelt to no seatbelt") +
  theme_classic()
```

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



## Based on the histogram, people are ~8x more likely to die in an accident if they are not wearing a seatbelt

```
## use the quantile function to produce the interval
quantile(rel_risks,c(0.025,0.975))
```

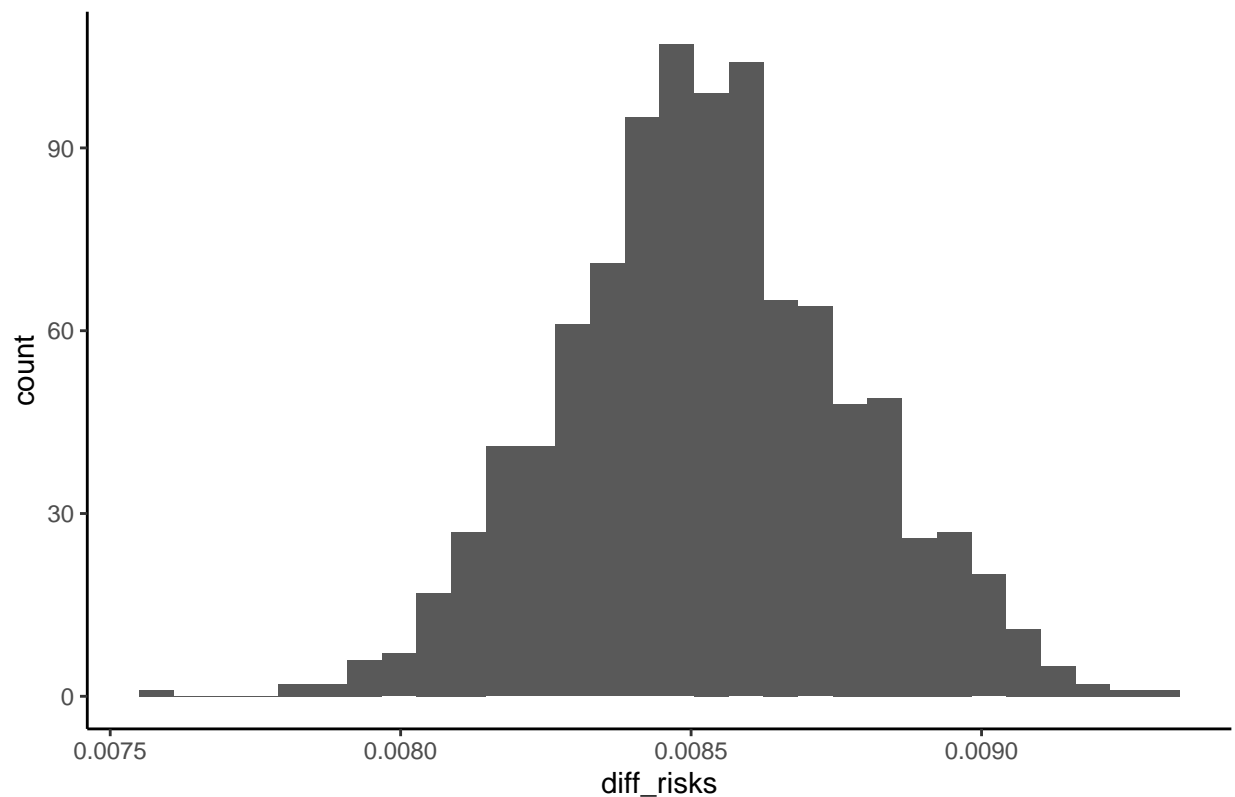
```
##      2.5%      97.5%
## 7.160231 8.705309
```

```
## 4
diff_risks <- pNs - pSs

plot_df <- cbind.data.frame(diff_risks)
ggplot(plot_df,aes(diff_risks)) +
  geom_histogram() +
  ggtitle("Histogram of the difference in risk: Seatbelt to no seatbelt") +
  theme_classic()
```

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

Histogram of the difference in risk: Seatbelt to no seatbelt



```
## 5
length(diff_risks > 0) / length(diff_risks)

## [1] 1
## All of our simulated Ps (probability of death in an accident) are greater for no seatbelt than for s
## our answer here is %100
```

#### Question 4 - Learning from rounded data

```
ys <- c(10,11,12,11,9)

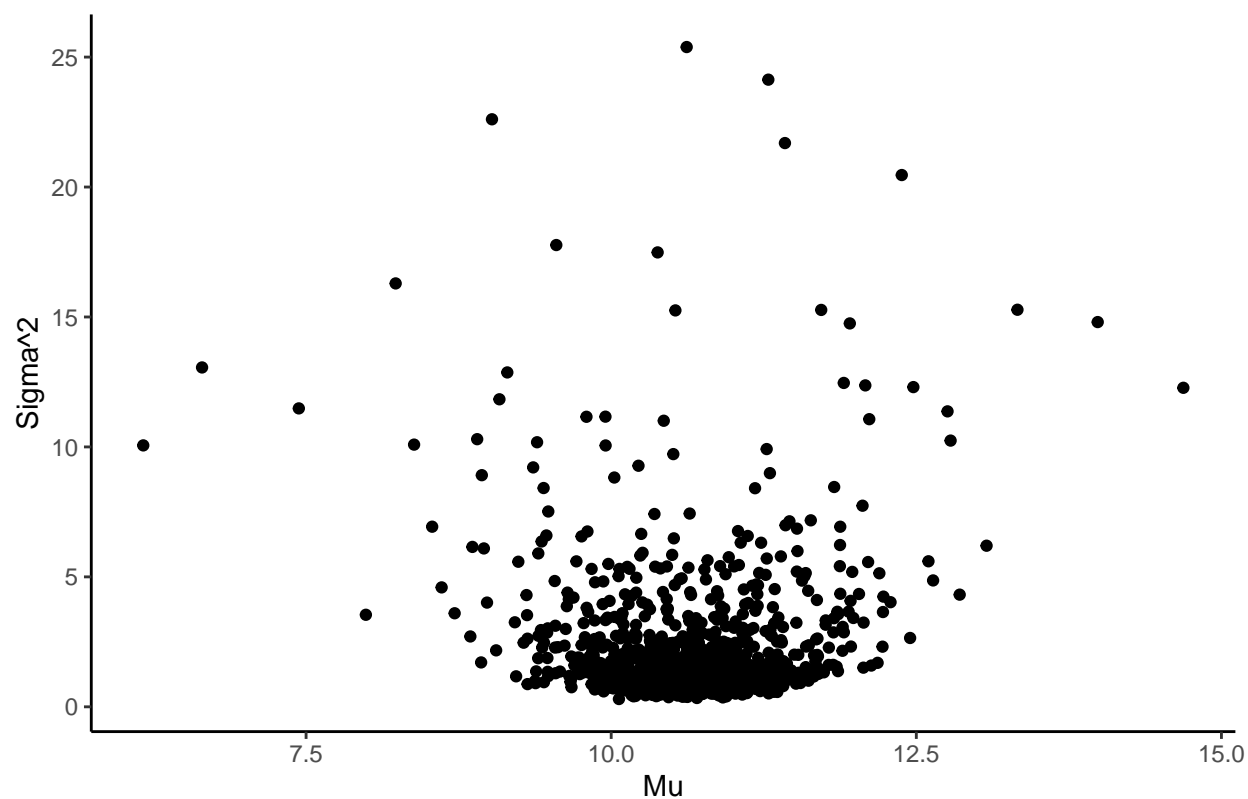
## Just copy our code from exercise 1
S <- sum((ys - mean(ys))^2)
n <- length(ys)
sigma2s <- S/rchisq(1000,n-1)
mus <- rnorm(1000, mean=mean(ys), sd=sqrt(sigma2s)/sqrt(n))

## Plot the simulations
plot_df <- cbind.data.frame(sigma2s,mus)

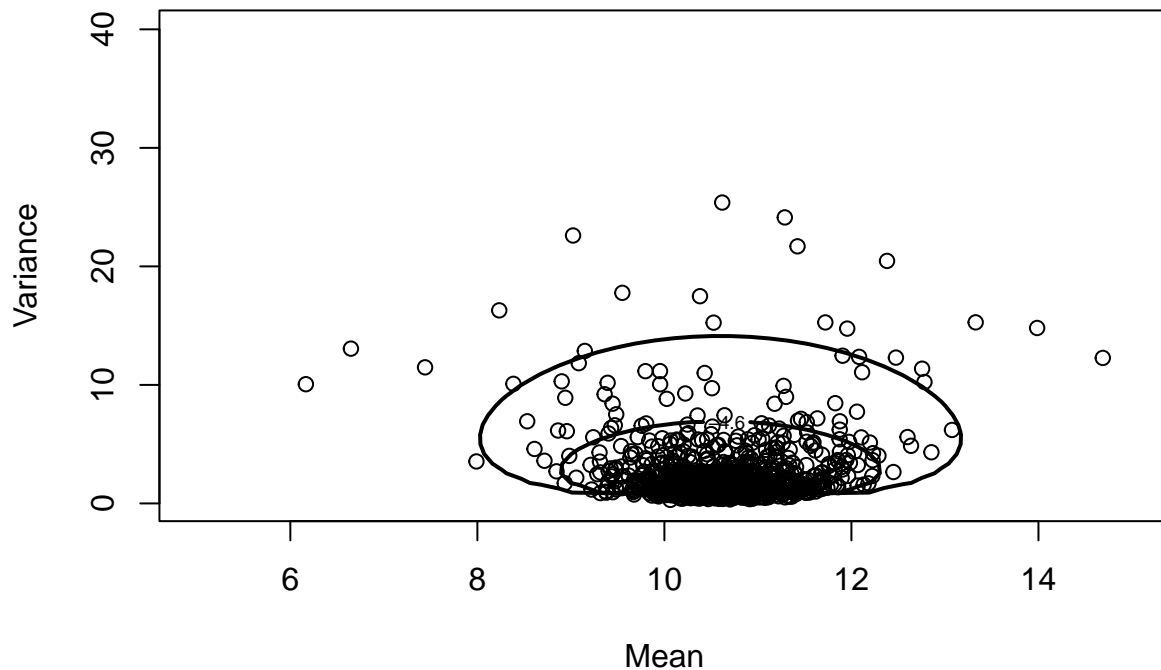
ggplot(plot_df,aes(mus,sigma2s)) +
  geom_point() +
  ggtitle("The simulated value pairs for variance and mean") +
  xlab("Mu") +
```

```
(ylab("Sigma^2")) +  
theme_classic()
```

The simulated value pairs for variance and mean



```
## Plot using book's functions and contours  
d <- mycontour(normchi2post,c(5,15,0.1,40),ys,xlab="Mean",ylab="Variance")  
points(mus,sigma2s)
```



*## 2 Another pen-and-paper exercise, I should try to add these using images from Latec or something*

*## 3*

*## From part 1, we have a box that approximately covers the posterior distribution  
## Use the function simcontour to similar data*

*## The posterior distribution doesn't have a nice closed form,  
## but it is easy to compute explicitly for a given value  
## We use our grid from part 1 to compute posterior probabilities of parameter pairs*  
grid <- c(5,15,0.1,40)

*## First let's write a function that computes the pdf for given data and parameters*

```
pnorm_rounded <- function(x,mu,sigma2) {
  return(ifelse(trunc(x)==x,pnorm(x+0.5,mu,sqrt(sigma2)) - pnorm(x-0.5,mu,sqrt(sigma2)),0))
}
```

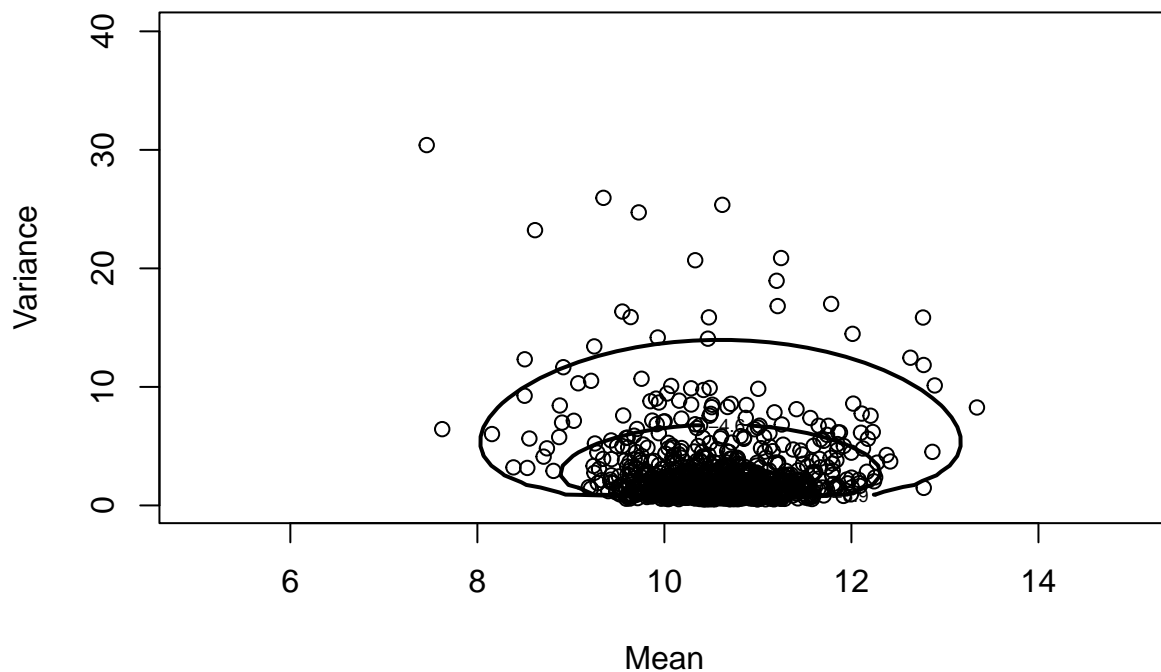
*## Now write a function that uses the previous to compute log of posterior probability for given data a*

```
lognorm_rounded_post <- function(theta,ys) {
  return(log( (1/theta[2]) * prod(pnorm_rounded(ys,theta[1],theta[2])) ))
}
```

*## Plot the contours of the distribution*

```
mycontour(lognorm_rounded_post,grid,ys,xlab="Mean",ylab="Variance")
sims <- simcontour(lognorm_rounded_post,grid,ys,1000)
points(sims$x,sims$y)
```





```
## 4
```

```
## compute the mean/sd for the standard sims
mean(mus)
```

```
## [1] 10.616
```

```
sd(mus)
```

```
## [1] 0.7059032
```

```
## compute the mean/sd for the sims from rounded density
mean(sims$x)
```

```
## [1] 10.59252
```

```
sd(sims$x)
```

```
## [1] 0.6747269
```

```
## The posterior distributions for Mu are effectively the same
```

```
## Intuitively, this shouldnt surprise us as rounding the data points should not impact the mean
```

```
## For example
```

```
sims <- rnorm(1000,mean=10,sd=sqrt(10))
rounded_sims <- round(sims)
mean(sims)
```

```
## [1] 9.974618
```

```
sd(sims)

## [1] 3.178327
mean(rounded_sims)

## [1] 9.985
sd(rounded_sims)

## [1] 3.177242
## I think the point of this exercise is to demonstrate how bayesian inference can be used in cases where
## How would we have proven that rounding does not impact the mean in traditional statistics?
```

## Question 5 - Estimating the parameters of a poisson/gamma density

```
ys <- c(2,5,0,2,3,1,3,4,3,0,3,
        2,1,1,0,6,0,0,3,0,1,1,
        5,0,1,2,0,0,2,1,1,1,0)

## This will be very similar to question 4, where we make some big posterior function,
## and then use mycontour and simcontour to answer the question

gamma_func <- function(y,a,b) {
  return( gamma(y+a) *
          (b ^ a) *
          (1/( gamma(a) * factorial(y) )) *
          (1/( (b+1) ^ (y+a) ))
        )
}

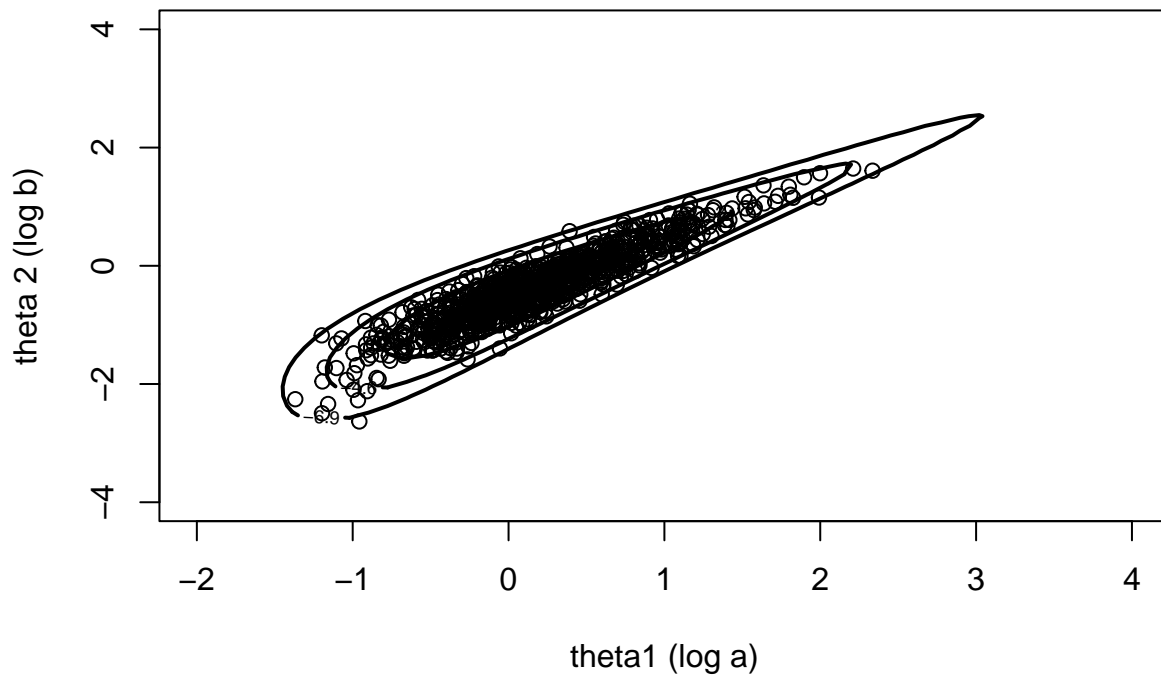
poisson_gamma_post <- function(theta,ys) {
  a <- exp(theta[1])
  b <- exp(theta[2])

  return(log( (1/(a*b)) * prod(gamma_func(ys,a,b)) ))
}

## Now that we've defined the computation of the posterior for a given parameter pair
## We have to find the grid

## It took me a lot of painful guessing to find this appropriate grid
## (Because the posterior probabilities of a and b are not independent)
## Theres probably a better way
grid <- c(-2,4,-4,4)

mycontour(poisson_gamma_post,grid,ys,xlab="theta1 (log a)", ylab="theta 2 (log b)")
sims <- simcontour(poisson_gamma_post,grid,ys,1000)
points(sims$x,sims$y)
```



```
as <- exp(sims$x)
bs <- exp(sims$y)
qs <- c(.05,.95)
quantile(as,qs)
```

```
##          5%          95%
## 0.510362 3.050683
```

```
quantile(bs,qs)
```

```
##          5%          95%
## 0.2699284 1.9520736
```

## Question 6 - Comparison of two Poisson rates

```
## 1
## A pen and paper exercise
```

```
## 2
## From part 1, because the posteriors are independent, we can simply simulate draws from the two marginal posteriors.
## Also note that the Gamma distribution is a conjugate prior for poisson parameter
## This means that the posterior is an easy computation update to the prior
```

```
g1 <- c(144,2.4)
g2 <- c(100,2.5)
data1 <- c(260,4)
data2 <- c(165,4)
```

```

post1 <- g1+data1
post2 <- g2+data2

lambda1s <- rgamma(1000,post1[1],post1[2])
lambda2s <- rgamma(1000,post2[1],post2[2])

## 3
## posterior probability from sims
length(lambda1s[lambda1s>=1.5*lambda2s]) / length(lambda1s)

## [1] 0.686

```

## Question 7 - Fitting a gamma density

```

# I am lost on this question
# I think that:
# The Gamma distribution has two equivalent forms:
# 1) Scale form: expressed here with parameters alpha, lambda
# 2) Rate form: used exclusively throughout this book (prior to this question) with parameters alpha,
# The densities look similar and I assume can be proven to be mathematically equivalent

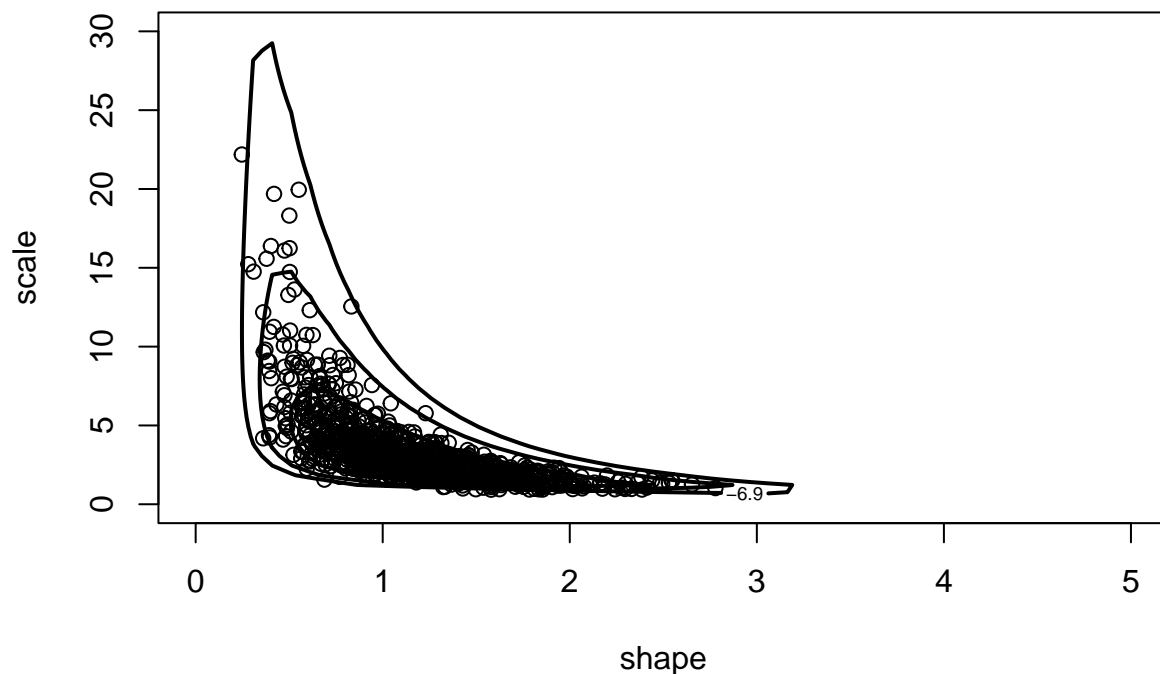
gamma_sampling_post <- function(theta,y) {
  sum(dgamma(y,shape=theta[1],scale=theta[2],log=TRUE))
}

ys <- c(12.2,0.9,0.8,5.3,2,1.2,1.2,1,.3,1.8,3.1,2.8)

## 1
grid <- c(0.001,5,0.001,30)

mycontour(gamma_sampling_post,grid,ys,xlab="shape", ylab="scale")
sims <- simcontour(gamma_sampling_post,grid,ys,1000)
points(sims$x,sims$y)

```



```
## 2
# How I think the jacobian part works here
# If  $\theta = (\alpha, \lambda)$  and  $\theta_1 = (\alpha, \beta)$ 
# then  $\theta_1$  is a 2d transformation of  $\theta$  given by  $h(\theta[\alpha, \lambda]) = \theta_1[\alpha = \alpha, \beta = 1/\lambda]$ 
# and the inverse of  $h$  is  $h^{-1}(\theta_1[\alpha, \beta]) = \theta[\alpha = \alpha, \lambda = 1/\beta]$ 
# per Wikipedia: https://en.wikipedia.org/wiki/Integration\_by\_substitution#Application\_in\_probability

# Based on the above, the function to compute the posterior of  $\theta_1 = \{\alpha, \beta\}$ 
# should be given by  $g(h^{-1}(\theta_1)) * |\text{Jacobian Determinant of } h^{-1}(\theta_1)|$ 

# The jacobian matrix of  $h_1$  is
# ( 1, 0
#   0,  $1/\beta^2$ )

# and the jacobian term is  $-1/\beta^2$ 

# I'm confused because the jacobian doesnt make sense
# If i simply substitute  $1/\beta$  for  $\lambda$  in the density function given for scale form, then I obtain
# Multiplying by  $|-1/\beta^2|$  results in an extra term

mus <- sims$x*sims$y
qs <- c(0.05, 0.95)
quantile(mus, qs)
```

```
##          5%          95%
## 1.899098 5.107671

## 3
thetas <- cbind(sims$x,sims$y)

## Take a theta = (alpha,lambda) and transform it into (f1(alpha,lambda),f2(alpha,lambda))
## f1(alpha,lambda) = alpha
## f2(alpha,lambda) = 1/lambda

## the jacobian matrix is [1,0,0,-1/lambda^2] and the determinant is -1/lambda^2
```