lecture_4

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
##################################################################
#### Normal Model with Conjugate and Noninformative Priors      ###
##################################################################

## Reading in Data:
data <- read.table("data/hitters.post1970.txt",header=T,row.names=NULL)
dim(data)
```

```
## [1] 20810    26
```

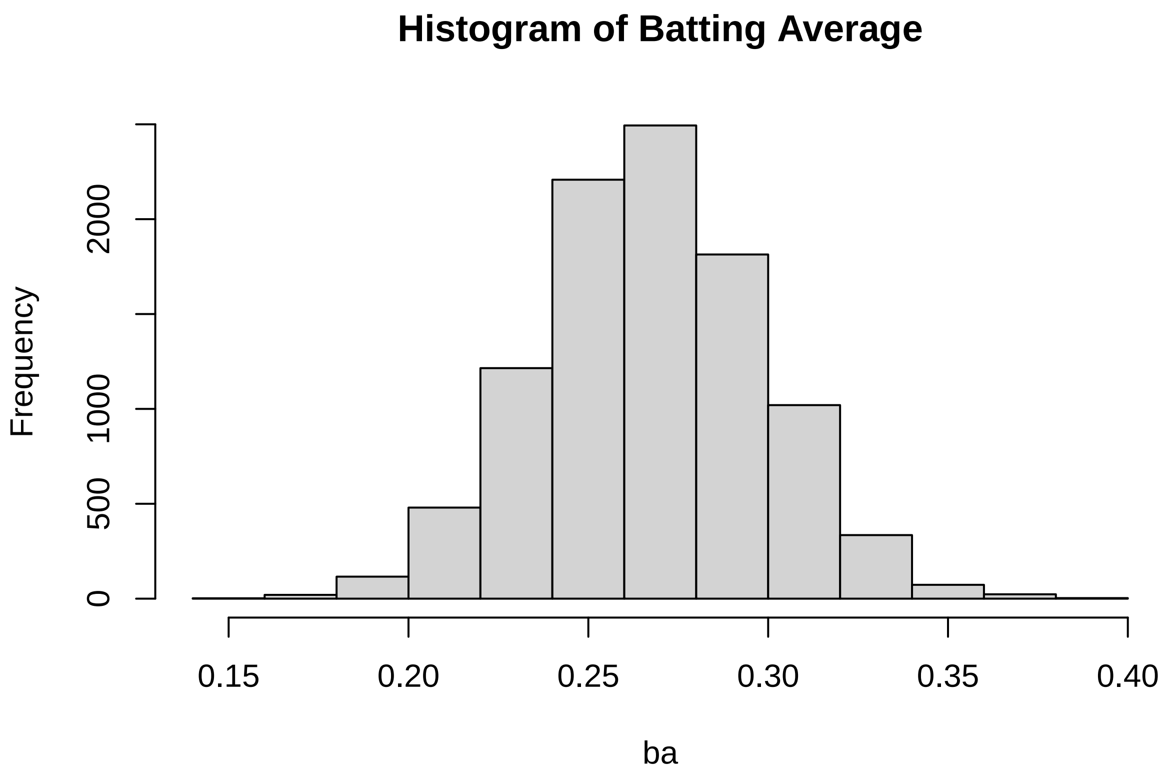
```
data[1:5,]
```

```
##    playerID yearID stint teamID lgID   G  AB   R   H X2B X3B HR RBI SB CS BB SO
## 1 aaronha01   1970     1    ATL   NL 150 516 103 154  26   1 38 118  9  0 74 63
## 2 aaronha01   1971     1    ATL   NL 139 495  95 162  22   3 47 118  1  1 71 58
## 3 aaronha01   1972     1    ATL   NL 129 449  75 119  10   0 34  77  4  0 92 55
## 4 aaronha01   1973     1    ATL   NL 120 392  84 118  12   1 40  96  1  1 68 51
## 5 aaronha01   1974     1    ATL   NL 112 340  47  91  16   0 20  69  1  0 39 29
##   IBB HBP SH SF GIDP pos birth age hand
## 1  15   2  0  6   13  RF  1934  37    R
## 2  21   2  0  5    9  RF  1934  38    R
## 3  15   1  0  2   17  RF  1934  39    R
## 4  13   1  0  4    7  LF  1934  40    R
## 5   6   0  1  2    6  LF  1934  41    R
```

```
## Reducing data to player-seasons where ab >= 200
data <- data[data$AB >= 200,]
dim(data)
```

```
## [1] 9803   26
```

```
## Calculating batting average
ba <- data$H/data$AB
n <- length(ba)
hist(ba,main="Histogram of Batting Average")
```



```
min(ba)
```

```
## [1] 0.1524664
```

```
data[which(ba==min(ba)),]
```

```
##        playerID yearID stint teamID lgID  G  AB  R  H X2B X3B HR RBI SB CS BB
## 11891 masonji01   1975     1    NYA   AL 94 223 17 34   3   2  2  16  0  2 22
##       SO IBB HBP SH SF GIDP pos birth age hand
## 11891 49   0   0  5  1   10  SS  1950  26    L
```

```
max(ba)
```

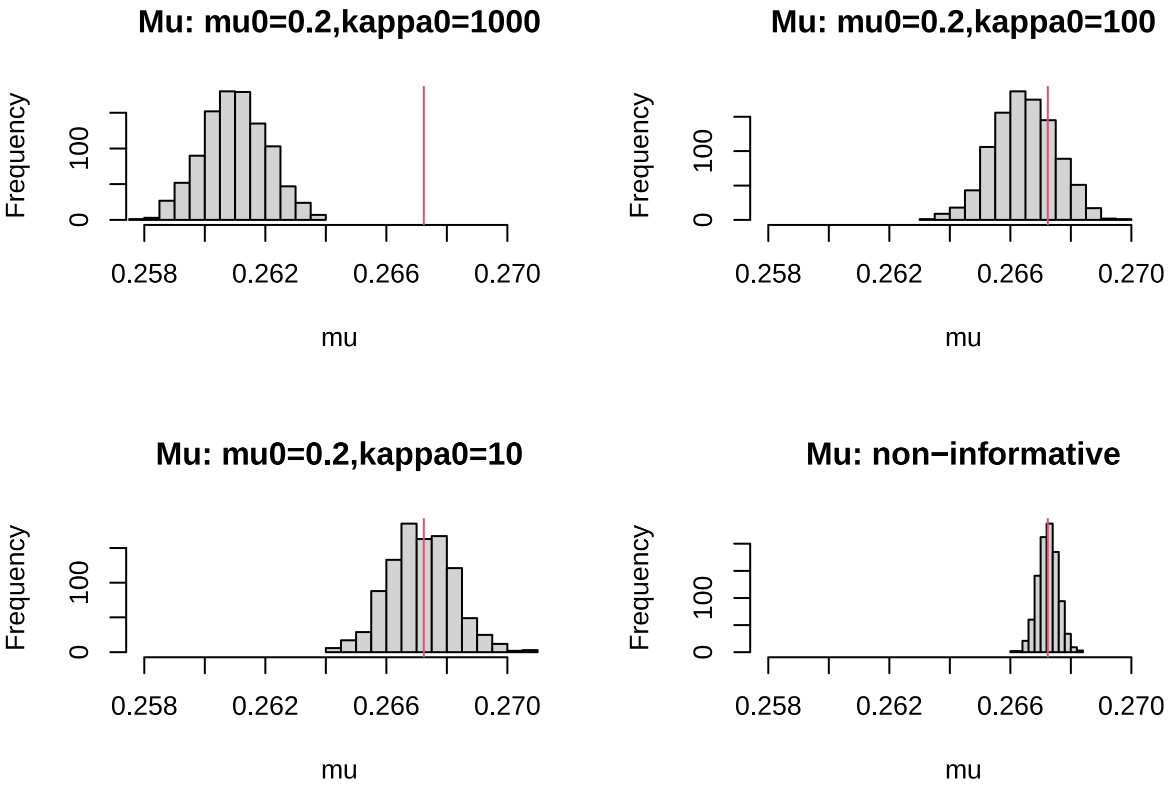
```
## [1] 0.3937947
```

```
data[which(ba==max(ba)),]
```

```
##       playerID yearID stint teamID lgID   G  AB  R   H X2B X3B HR RBI SB CS BB
## 7511 gwynnto01   1994     1    SDN   NL 110 419 79 165  35   1 12  64  5  0 48
##      SO IBB HBP SH SF GIDP pos birth age hand
## 7511 19  16   2  1  5   20  RF  1960  35    L
```

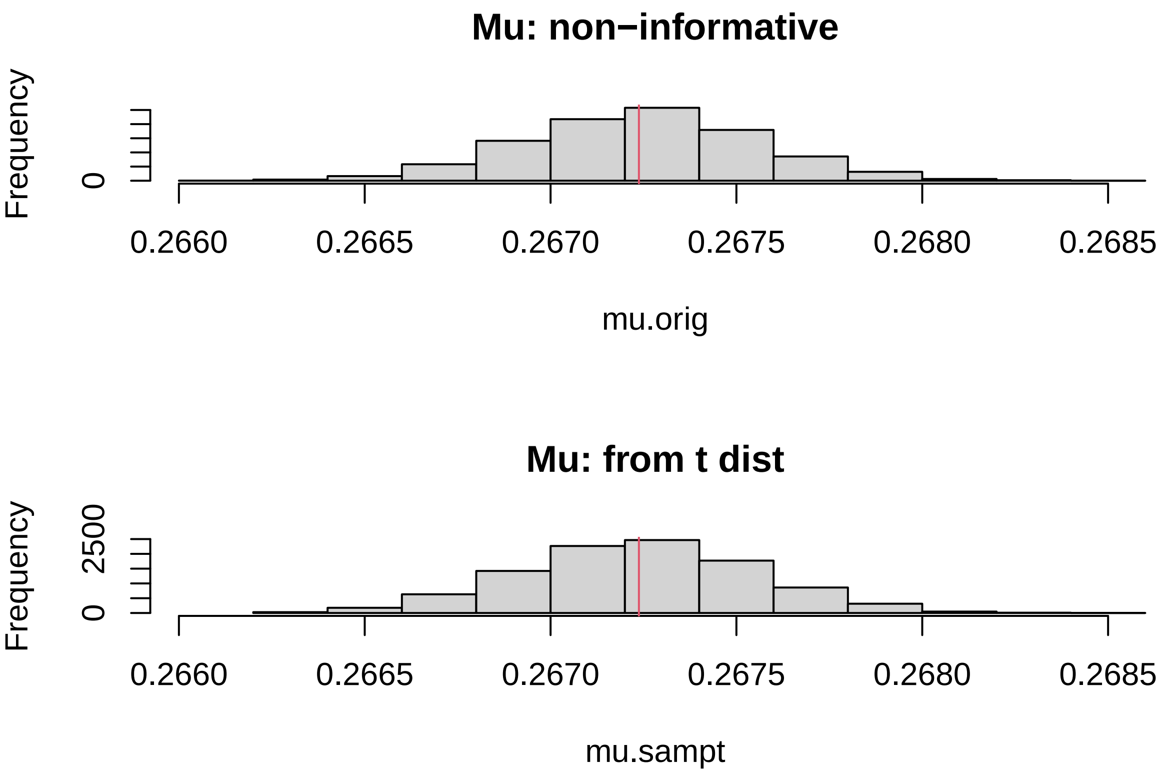
```
sample.norm.conj <- function(y,mu0,kappa0,nu0,sigsq0,numsamp){
    n <- length(y)
    y.mean <- mean(y)
    y.ss <- var(y)*(n-1)
    discrep <- (kappa0*n/(kappa0+n))*(y.mean-mu0)^2
    x <- rgamma(numsamp,shape=(nu0+n)/2,rate=(nu0*sigsq0+y.ss+discrep)/2)
    sigsq.samp <- 1/x
    postvar <- 1/(n/sigsq.samp + kappa0/sigsq.samp)
    postmean <- (n*y.mean/sigsq.samp + kappa0*mu0/sigsq.samp)/(n/sigsq.samp + kappa0/sigsq.samp)
    mu.samp <- rnorm(numsamp,mean=postmean,sd=sqrt(postvar))
    out <- cbind(mu.samp,sigsq.samp)
```

```r
    out
}

## checking the posterior for different conjugate priors and non-informative
##  (different values of mu0,kappa0)

theta1 <- sample.norm.conj(ba,0.2,10,10,10,1000)   # mu0 = 0.2, kappa0 = 10, etc.
theta2 <- sample.norm.conj(ba,0.2,100,10,10,1000)   # mu0 = 0.2, kappa0 = 100, etc.
theta3 <- sample.norm.conj(ba,0.2,1000,10,10,1000)   # mu0 = 0.2, kappa0 = 1000, etc.
theta4 <- sample.norm.conj(ba,0.2,0,0,10,1000)   # non-informative kappa0 = 0, nu0 = 0

par(mfrow=c(2,2))
minmu <- min(theta1[,1],theta2[,1],theta3[,1],theta4[,1],mean(ba))
maxmu <- max(theta1[,1],theta2[,1],theta3[,1],theta4[,1],mean(ba))
hist(theta3[,1],main="Mu: mu0=0.2,kappa0=1000",xlim=c(minmu,maxmu),xlab="mu")
abline(v=mean(ba),col=2)
hist(theta2[,1],main="Mu: mu0=0.2,kappa0=100",xlim=c(minmu,maxmu),xlab="mu")
abline(v=mean(ba),col=2)
hist(theta1[,1],main="Mu: mu0=0.2,kappa0=10",xlim=c(minmu,maxmu),xlab="mu")
abline(v=mean(ba),col=2)
hist(theta4[,1],main="Mu: non-informative",xlim=c(minmu,maxmu),xlab="mu")
abline(v=mean(ba),col=2)
```



```r
## could have also generated mu directly from t distribution:
mu.sampt <- rt(10000,n-1)
mu.sampt <- mu.sampt*sqrt(var(ba)/n)+mean(ba)

## compare to original sampling scheme
theta <- sample.norm.conj(ba,0.2,0,0,10,10000)
```

```
mu.orig <- theta[,1]

xmin <- min(mu.orig,mu.sampt)
xmax <- max(mu.orig,mu.sampt)
par(mfrow=c(2,1))
hist(mu.orig,main="Mu: non-informative",xlim=c(xmin,xmax))
abline(v=mean(ba),col=2)
hist(mu.sampt,main="Mu: from t dist",xlim=c(xmin,xmax))
abline(v=mean(ba),col=2)
```



```
####################################################################
#### Grid Search and Grid Sample: Semi-Conjugate Normal Example ###
####################################################################

## Reading in Data:
data <- read.table("data/hitters.post1970.txt",header=T)

## Reducing data to player-seasons where ab > 200
data <- data[data$AB > 200,]


## Calculating batting average
ba <- data$H/data$AB
n <- length(ba)
hist(ba)
min(ba)
```

```
## [1] 0.1524664
```

```r
data[which(ba==min(ba)),]
```

```
##        playerID yearID stint teamID lgID  G  AB  R  H X2B X3B HR RBI SB CS BB
## 11891 masonji01   1975     1    NYA   AL 94 223 17 34   3   2  2  16  0  2 22
##       SO IBB HBP SH SF GIDP pos birth age hand
## 11891 49   0   0  5  1   10  SS  1950  26    L
```

```r
max(ba)
```

```
## [1] 0.3937947
```

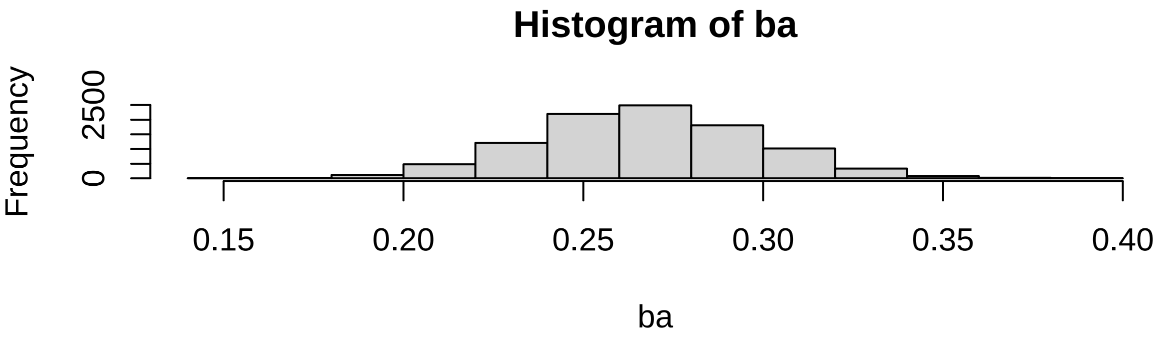
```r
data[which(ba==max(ba)),]
```

```
##       playerID yearID stint teamID lgID   G  AB  R   H X2B X3B HR RBI SB CS BB
## 7511 gwynnto01   1994     1    SDN   NL 110 419 79 165  35   1 12  64  5  0 48
##      SO IBB HBP SH SF GIDP pos birth age hand
## 7511 19  16   2  1  5   20  RF  1960  35    L
```

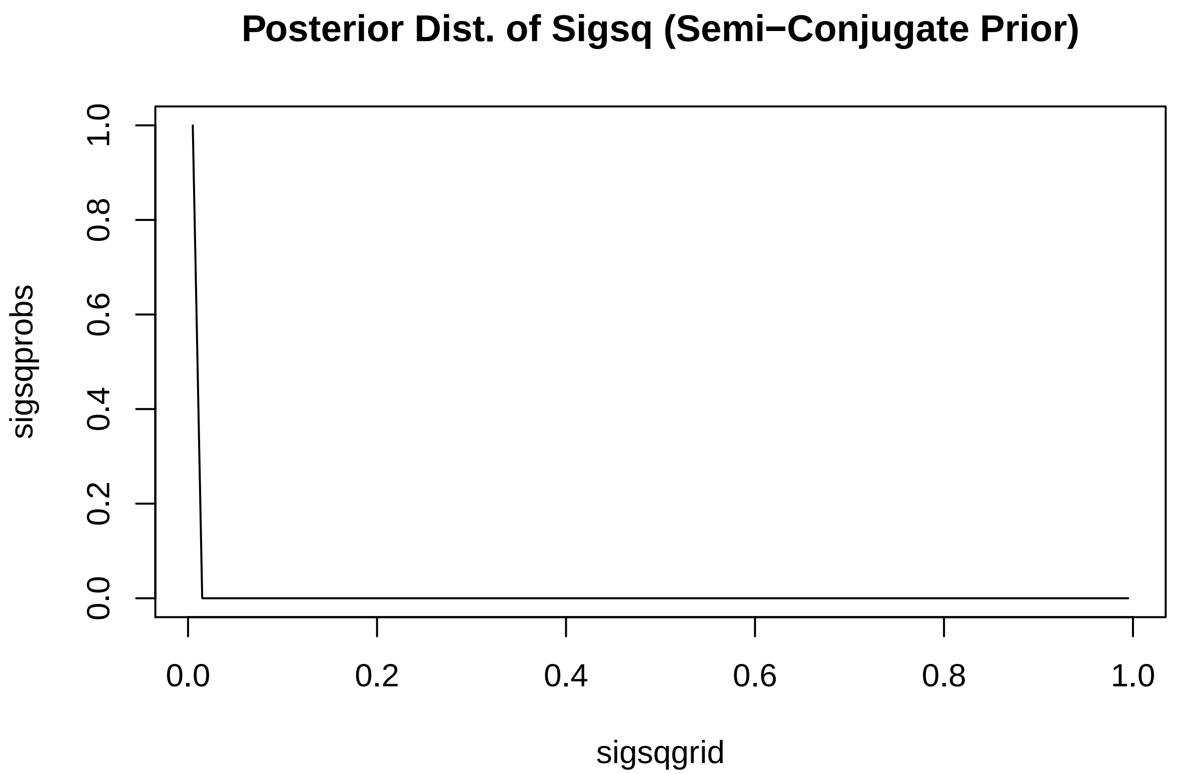
```r
evaluatepostsigsq <- function(sigsqvalues,y,mu0,nu0,tausq0,sigsq0){
    m <- length(sigsqvalues)
    logvals <- rep(0,m)
    n <- length(y)
    for (i in 1:m){
        cursigsq <- sigsqvalues[i]
        postmean  <- (n*mean(y)/cursigsq + mu0/tausq0)/(n/cursigsq + 1/tausq0)
        for (j in 1:n){
            logvals[i] <- logvals[i] + dnorm(y[j],mean=postmean,sd=sqrt(cursigsq),log=T)
        }
        logvals[i] <- logvals[i] - 0.5*log(n/cursigsq + 1/tausq0)
        logvals[i] <- logvals[i] - (0.5*nu0+1)*log(cursigsq) - nu0*sigsq0/(2*cursigsq)
        print (i)
    }
    out <- exp(logvals-max(logvals))
    out
}

## setting hyperparameter values
tausq0 <- 0.1
sigsq0 <- 0.1
nu0 <- 0.001
mu0 <- 0.2

sigsqgrid <- ppoints(100)
sigsqprobs <- evaluatepostsigsq(sigsqgrid,ba,mu0,nu0,tausq0,sigsq0)
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## [1] 11
## [1] 12
## [1] 13
## [1] 14
## [1] 15
## [1] 16
## [1] 17
## [1] 18
## [1] 19
## [1] 20
## [1] 21
## [1] 22
## [1] 23
## [1] 24
## [1] 25
## [1] 26
## [1] 27
## [1] 28
## [1] 29
## [1] 30
## [1] 31
## [1] 32
## [1] 33
## [1] 34
## [1] 35
## [1] 36
## [1] 37
## [1] 38
## [1] 39
## [1] 40
## [1] 41
## [1] 42
## [1] 43
## [1] 44
## [1] 45
## [1] 46
## [1] 47
## [1] 48
## [1] 49
## [1] 50
## [1] 51
## [1] 52
## [1] 53
## [1] 54
## [1] 55
## [1] 56
## [1] 57
## [1] 58
## [1] 59
## [1] 60
## [1] 61
## [1] 62
## [1] 63
## [1] 64
## [1] 65
## [1] 66
## [1] 67
## [1] 68
## [1] 69
## [1] 70
## [1] 71
## [1] 72
## [1] 73
## [1] 74
## [1] 75
## [1] 76
## [1] 77
## [1] 78
## [1] 79
## [1] 80
## [1] 81
## [1] 82
## [1] 83
## [1] 84
## [1] 85
## [1] 86
## [1] 87
## [1] 88
## [1] 89
## [1] 90
## [1] 91
## [1] 92
## [1] 93
## [1] 94
## [1] 95
## [1] 96
## [1] 97
## [1] 98
## [1] 99
## [1] 100
```

```
par(mfrow=c(1,1))
```



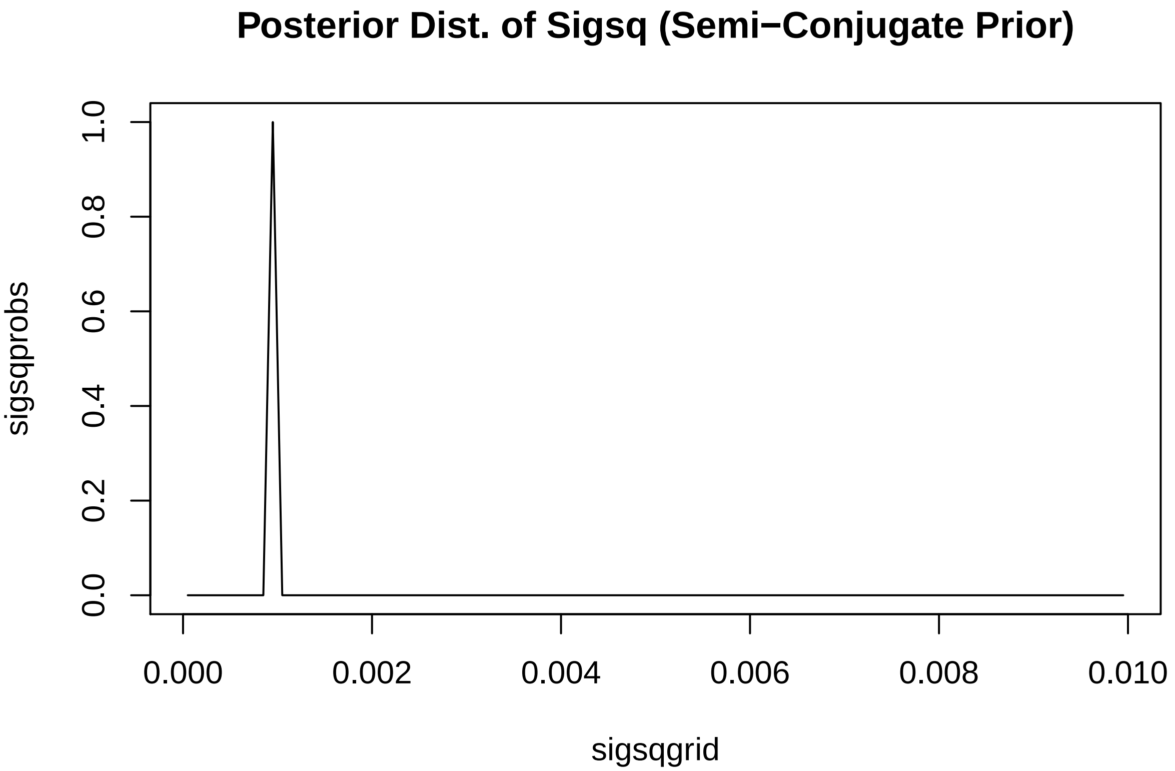
```
plot(sigsqgrid,sigsqprobs,type="l",main="Posterior Dist. of Sigsq (Semi-Conjugate Prior)")
```



```
sigsqgrid <- ppoints(100)*0.01
sigsqprobs <- evaluatepostsigsq(sigsqgrid,ba,mu0,nu0,tausq0,sigsq0)
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## [1] 11
## [1] 12
## [1] 13
## [1] 14
## [1] 15
## [1] 16
## [1] 17
## [1] 18
## [1] 19
## [1] 20
## [1] 21
## [1] 22
## [1] 23
## [1] 24
## [1] 25
## [1] 26
## [1] 27
## [1] 28
## [1] 29
## [1] 30
## [1] 31
## [1] 32
## [1] 33
## [1] 34
## [1] 35
## [1] 36
## [1] 37
## [1] 38
## [1] 39
## [1] 40
## [1] 41
## [1] 42
## [1] 43
## [1] 44
## [1] 45
## [1] 46
## [1] 47
## [1] 48
## [1] 49
## [1] 50
## [1] 51
## [1] 52
## [1] 53
## [1] 54
## [1] 55
## [1] 56
## [1] 57
## [1] 58
## [1] 59
## [1] 60
## [1] 61
## [1] 62
## [1] 63
## [1] 64
## [1] 65
## [1] 66
## [1] 67
## [1] 68
## [1] 69
## [1] 70
## [1] 71
## [1] 72
## [1] 73
## [1] 74
## [1] 75
## [1] 76
## [1] 77
## [1] 78
## [1] 79
## [1] 80
## [1] 81
## [1] 82
## [1] 83
## [1] 84
## [1] 85
## [1] 86
## [1] 87
## [1] 88
## [1] 89
## [1] 90
## [1] 91
## [1] 92
## [1] 93
## [1] 94
## [1] 95
## [1] 96
## [1] 97
## [1] 98
## [1] 99
## [1] 100
```

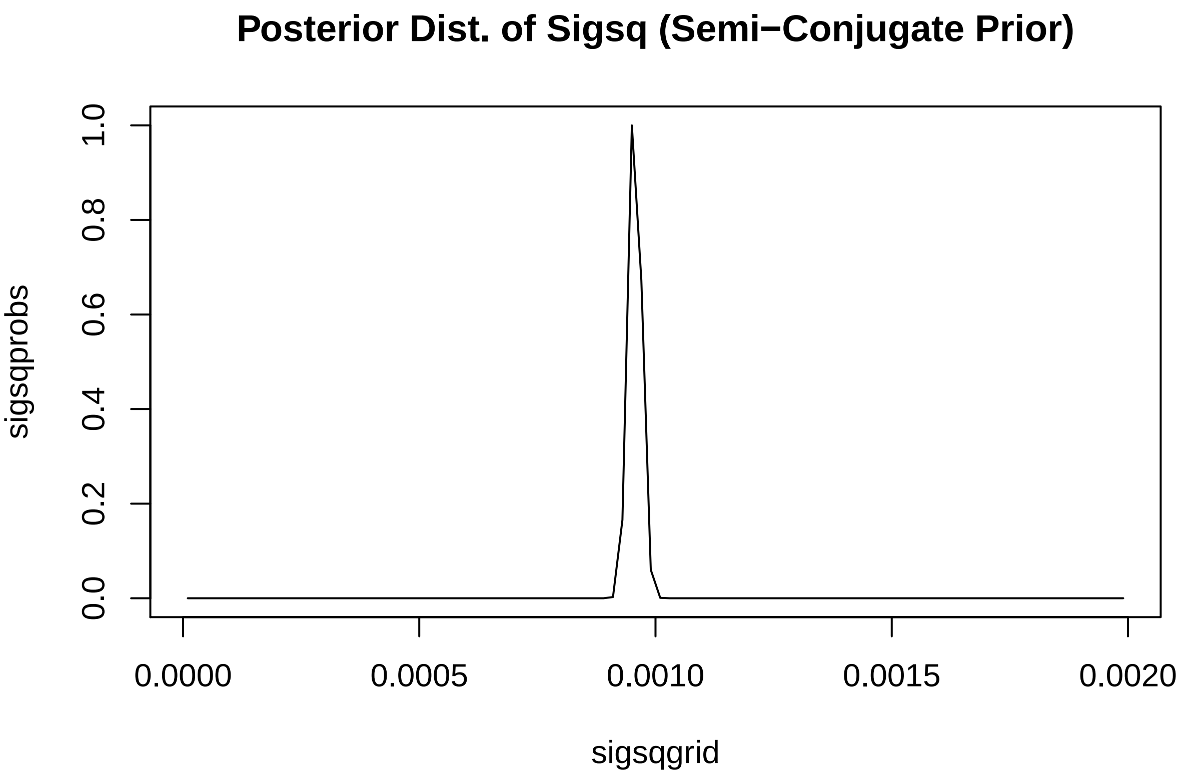
```
par(mfrow=c(1,1))
plot(sigsqgrid,sigsqprobs,type="l",main="Posterior Dist. of Sigsq (Semi-Conjugate Prior)")
```



```
sigsqgrid <- ppoints(100)*0.002
sigsqprobs <- evaluatepostsigsq(sigsqgrid,ba,mu0,nu0,tausq0,sigsq0)
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## [1] 11
## [1] 12
## [1] 13
## [1] 14
## [1] 15
## [1] 16
## [1] 17
## [1] 18
## [1] 19
## [1] 20
## [1] 21
## [1] 22
## [1] 23
## [1] 24
## [1] 25
## [1] 26
## [1] 27
## [1] 28
## [1] 29
## [1] 30
## [1] 31
## [1] 32
## [1] 33
## [1] 34
## [1] 35
## [1] 36
## [1] 37
## [1] 38
## [1] 39
## [1] 40
## [1] 41
## [1] 42
## [1] 43
## [1] 44
## [1] 45
## [1] 46
## [1] 47
## [1] 48
## [1] 49
## [1] 50
## [1] 51
## [1] 52
## [1] 53
## [1] 54
## [1] 55
## [1] 56
## [1] 57
## [1] 58
## [1] 59
## [1] 60
## [1] 61
## [1] 62
## [1] 63
## [1] 64
## [1] 65
## [1] 66
## [1] 67
## [1] 68
## [1] 69
## [1] 70
## [1] 71
## [1] 72
## [1] 73
## [1] 74
## [1] 75
## [1] 76
## [1] 77
## [1] 78
## [1] 79
## [1] 80
## [1] 81
## [1] 82
## [1] 83
## [1] 84
## [1] 85
## [1] 86
## [1] 87
## [1] 88
## [1] 89
## [1] 90
## [1] 91
## [1] 92
## [1] 93
## [1] 94
## [1] 95
## [1] 96
## [1] 97
## [1] 98
## [1] 99
## [1] 100
```

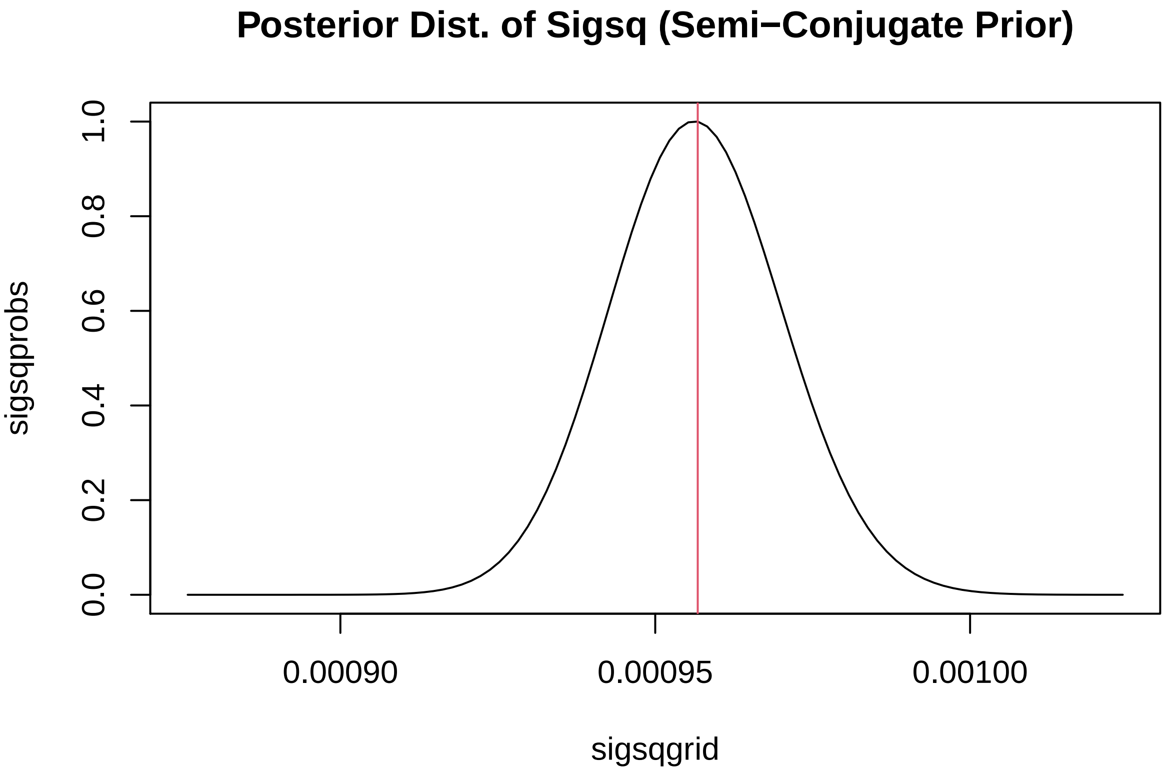
```r
par(mfrow=c(1,1))
plot(sigsqgrid,sigsqprobs,type="l",main="Posterior Dist. of Sigsq (Semi-Conjugate Prior)")
```



```r
sigsqgrid <- ppoints(100)*0.00015+0.000875
sigsqprobs <- evaluatepostsigsq(sigsqgrid,ba,mu0,nu0,tausq0,sigsq0)
```

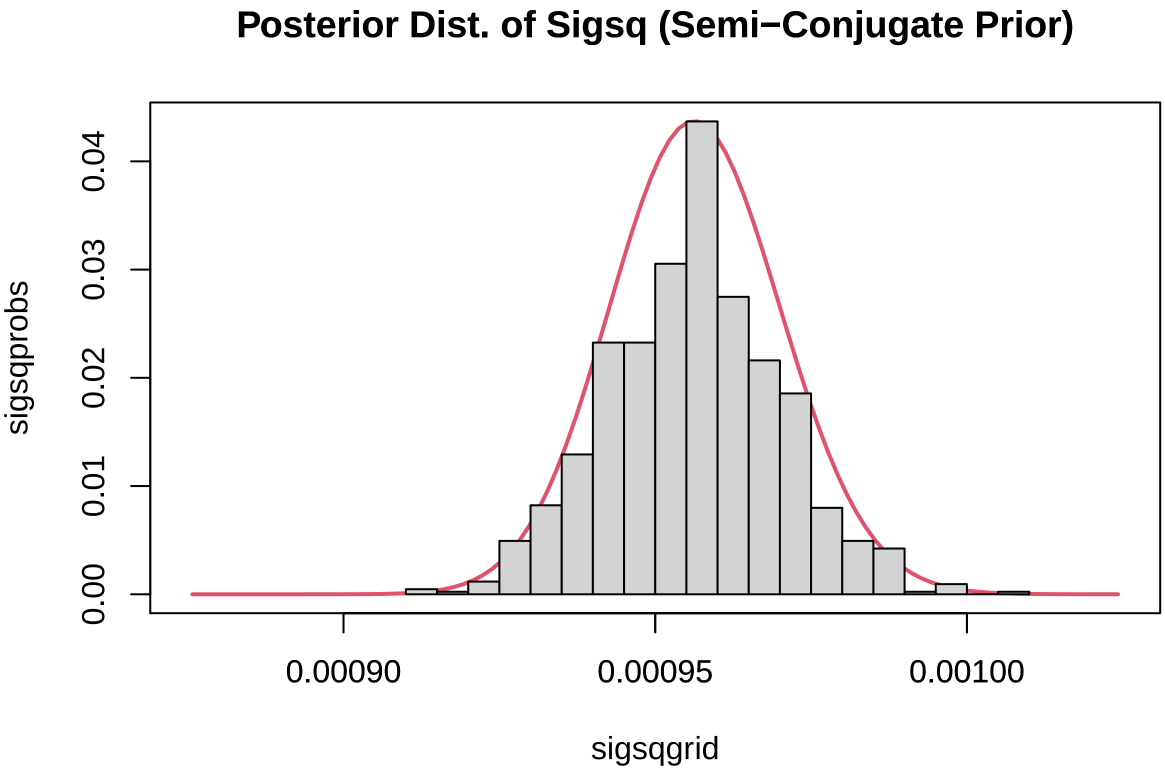
```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## [1] 11
## [1] 12
## [1] 13
## [1] 14
## [1] 15
## [1] 16
## [1] 17
## [1] 18
## [1] 19
## [1] 20
## [1] 21
## [1] 22
## [1] 23
## [1] 24
## [1] 25
## [1] 26
## [1] 27
## [1] 28
## [1] 29
## [1] 30
## [1] 31
## [1] 32
## [1] 33
## [1] 34
## [1] 35
## [1] 36
## [1] 37
## [1] 38
## [1] 39
## [1] 40
## [1] 41
## [1] 42
## [1] 43
## [1] 44
## [1] 45
## [1] 46
## [1] 47
## [1] 48
## [1] 49
## [1] 50
## [1] 51
## [1] 52
## [1] 53
## [1] 54
## [1] 55
## [1] 56
## [1] 57
## [1] 58
## [1] 59
## [1] 60
## [1] 61
## [1] 62
## [1] 63
## [1] 64
## [1] 65
## [1] 66
## [1] 67
## [1] 68
## [1] 69
## [1] 70
## [1] 71
## [1] 72
## [1] 73
## [1] 74
## [1] 75
## [1] 76
## [1] 77
## [1] 78
## [1] 79
## [1] 80
## [1] 81
## [1] 82
## [1] 83
## [1] 84
## [1] 85
## [1] 86
## [1] 87
## [1] 88
## [1] 89
## [1] 90
## [1] 91
## [1] 92
## [1] 93
## [1] 94
## [1] 95
## [1] 96
## [1] 97
## [1] 98
## [1] 99
## [1] 100
```

```
par(mfrow=c(1,1))
plot(sigsqgrid,sigsqprobs,type="l",main="Posterior Dist. of Sigsq (Semi-Conjugate Prior)")

## optimal point estimate approximated by grid point with highest value
sigsqhat <- sigsqgrid[which(sigsqprobs==max(sigsqprobs))]
abline(v=sigsqhat,col=2)
```

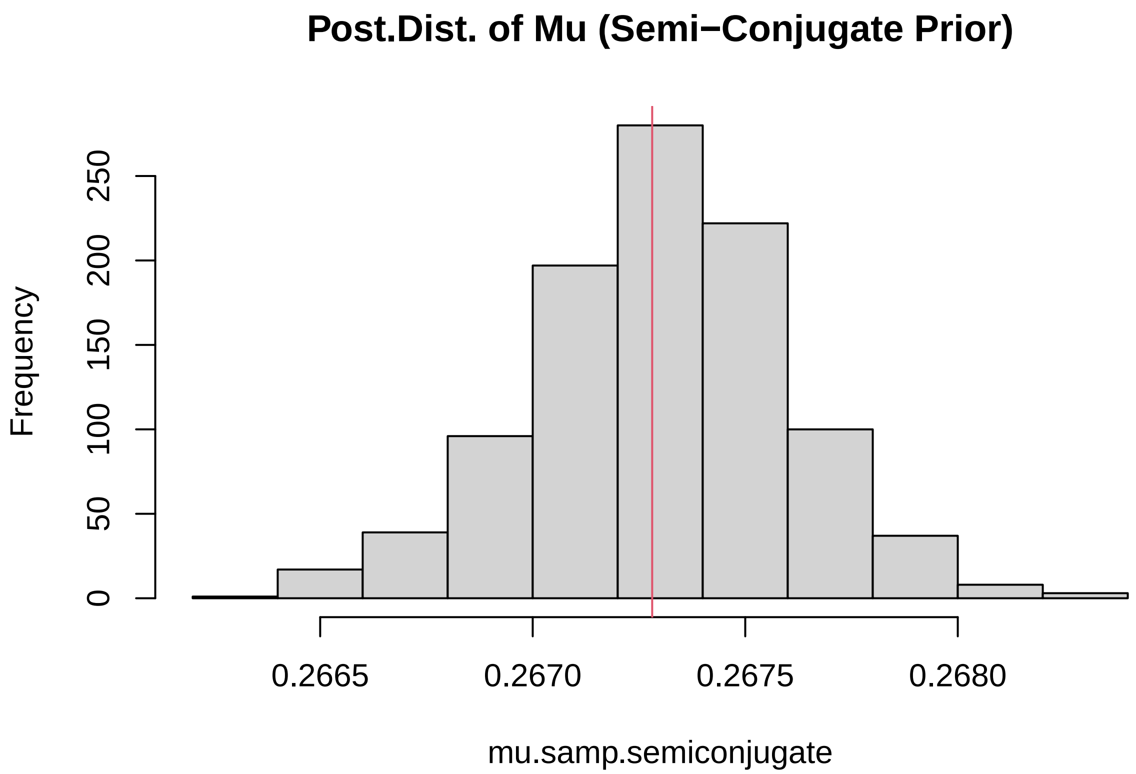


```
## grid sampling: sample 1000 values of sigsq proportional to sigsqprobs

sigsqprobs <- sigsqprobs/sum(sigsqprobs)
```

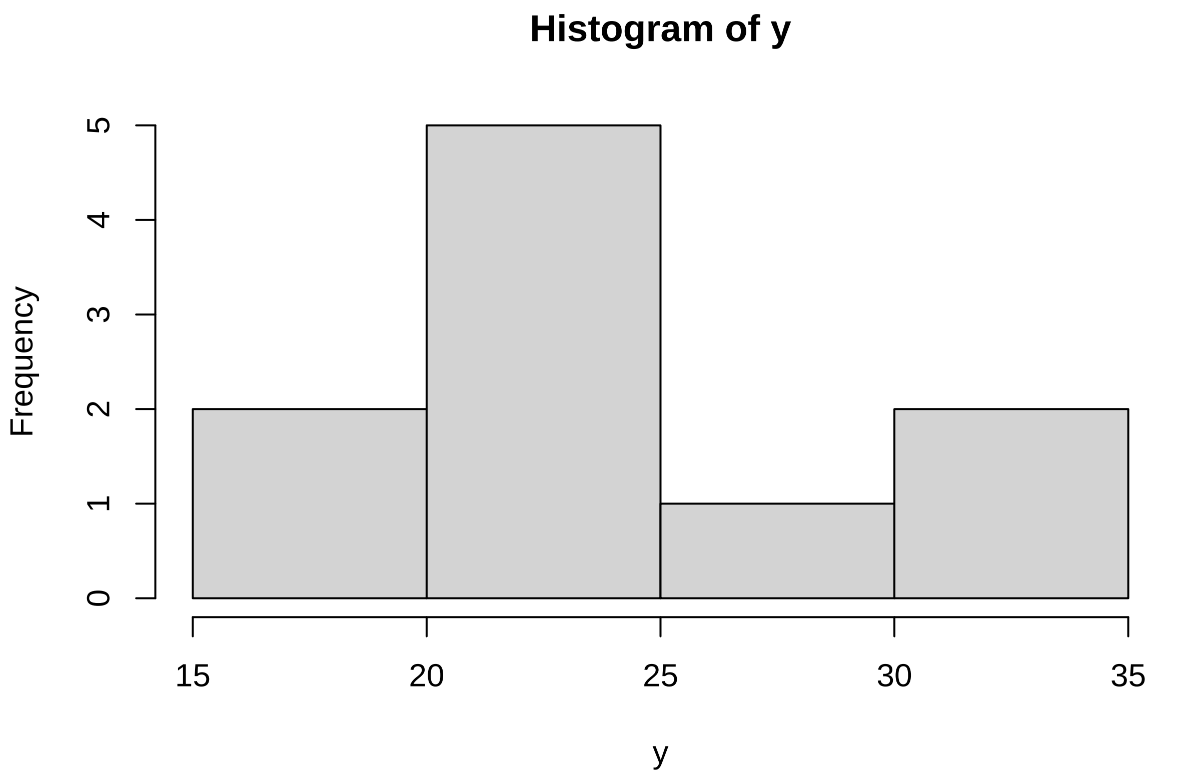
```
sigsq.samp <- sample(sigsqgrid,size=1000,replace=T,prob=sigsqprobs)

plot(sigsqgrid,sigsqprobs,type="l",main="Posterior Dist. of Sigsq (Semi-Conjugate Prior)",xlim=c(0.0008
par(new=T)
hist(sigsq.samp,prob=T,xlim=c(0.000875,0.001025),main="Posterior Dist. of Sigsq (Semi-Conjugate Prior)"
```



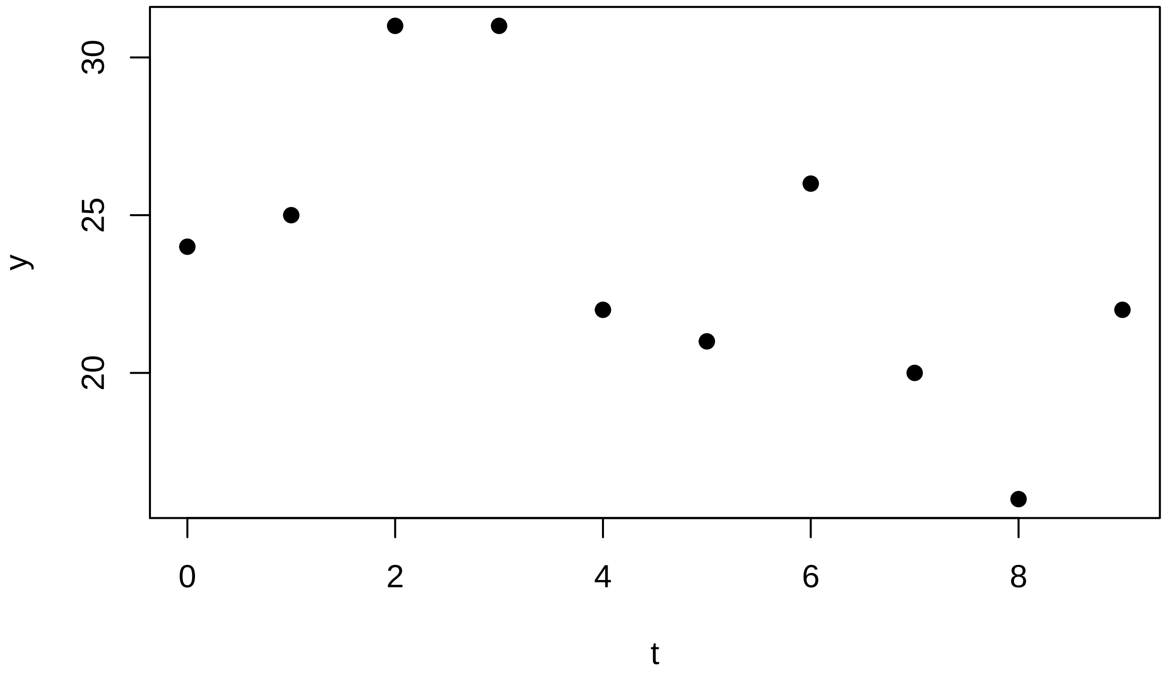
```
## sampling mu given sampled sigmasq

mu.samp.semiconjugate <- rep(NA,1000)
for (i in 1:1000){
    postvar <- 1/(n/sigsq.samp[i] + 1/tausq0)
    postmean <- (n*mean(ba)/sigsq.samp[i] + mu0/tausq0)*postvar
    mu.samp.semiconjugate[i] <- rnorm(1,mean=postmean,sd=sqrt(postvar))
}
hist(mu.samp.semiconjugate,main="Post.Dist. of Mu (Semi-Conjugate Prior)")
abline(v=mean(ba),col=2)
```



```
############################################################
### Grid Search and Grid Sample: Poisson Planes Dataset ####
############################################################

## input data:
data <- read.table("data/planes.txt",skip=1)
y <- data[,2]
t <- data[,1]-1976
n <- length(y)
hist(y)
```

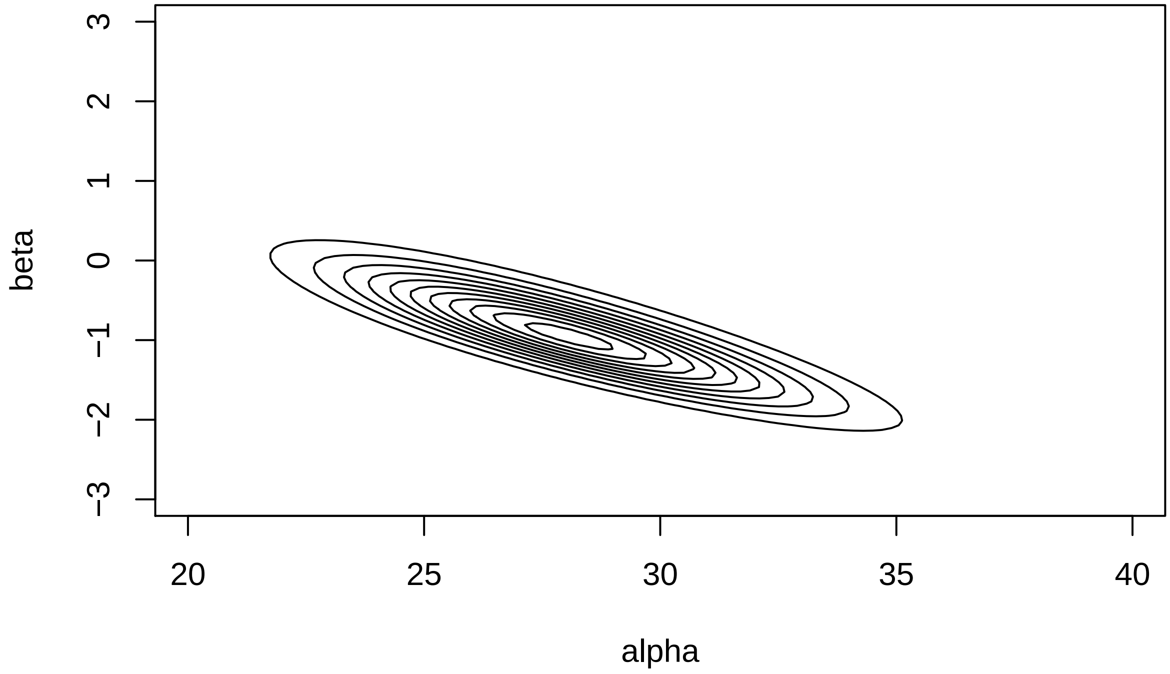


```
plot(t,y,pch=19)
```

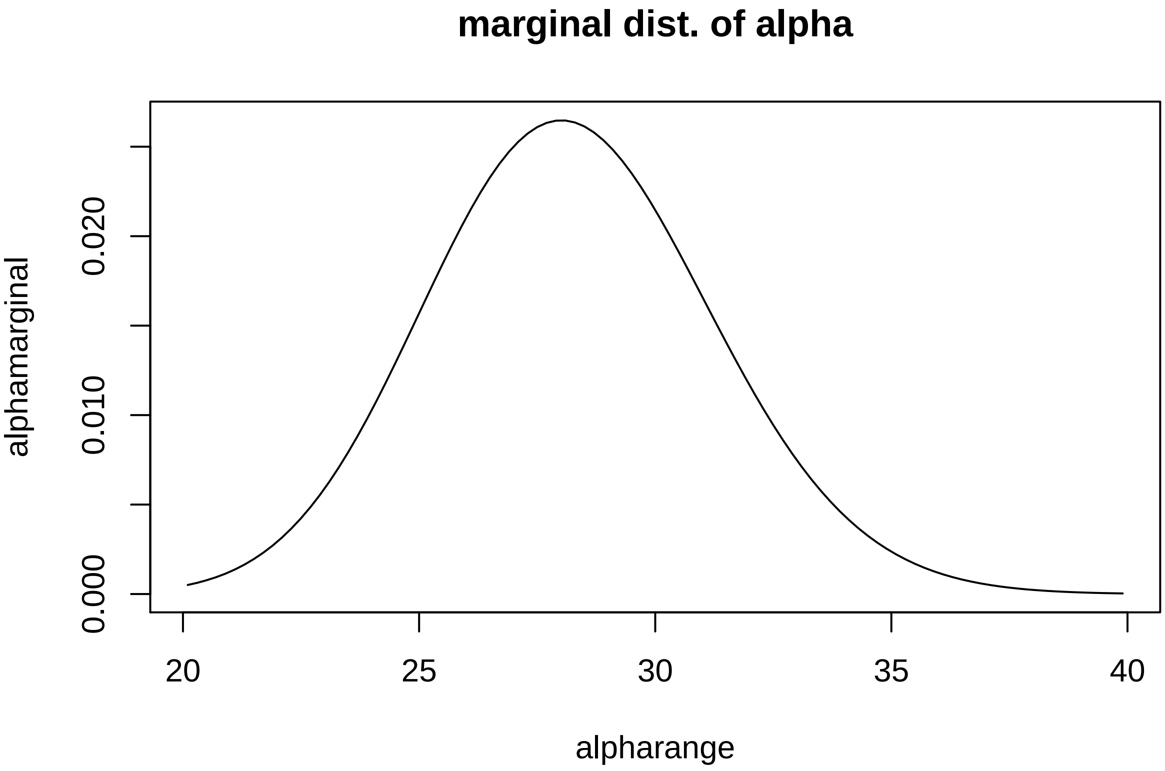


```
## graphing posterior over range of alpha and beta:
posteriorplanes <- function(alpha,beta){
  logpost <- -Inf
  if (alpha + beta*max(t) > 0){
    logpost <- 0
    for (i in 1:n){
```

```r
      logpost <- logpost + y[i]*log(alpha+beta*t[i])
      logpost <- logpost - (alpha+beta*t[i])
    }
  }
  logpost
}

numgrid <- 100
alpharange <- ppoints(numgrid)*20   # alpha between 0 and 20
betarange <- ppoints(numgrid)*6  # beta between 0 and 6

numgrid <- 100
alpharange <- ppoints(numgrid)*20+20   # alpha between 20 and 40
betarange <- ppoints(numgrid)*6-3  # beta between -3 and 3
full <- matrix(NA,nrow=numgrid,ncol=numgrid)
for (i in 1:numgrid){
  for (j in 1:numgrid){
    full[i,j] <- posteriorplanes(alpharange[i],betarange[j])
  }
}
full <- exp(full - max(full))
full <- full/sum(full)
contour(alpharange,betarange,full,xlab="alpha",ylab="beta",drawlabels=F)
```



```r
## calculating probabilities for grid sampler:

alphamarginal <- rep(NA,numgrid)
for (i in 1:numgrid){
  alphamarginal[i] <- sum(full[i,])
}
betaconditional <- matrix(NA,nrow=numgrid,ncol=numgrid)
for (i in 1:numgrid){
  for (j in 1:numgrid){
    betaconditional[i,j] <- full[i,j]/sum(full[i,])
  }
}

## plotting marginal distribution of alpha
par(mfrow=c(1,1))
plot(alpharange,alphamarginal,type="l",main="marginal dist. of alpha")
```



```
## plotting conditional distribution of beta given alpha
alpharange[25]
```

```
## [1] 24.9
```

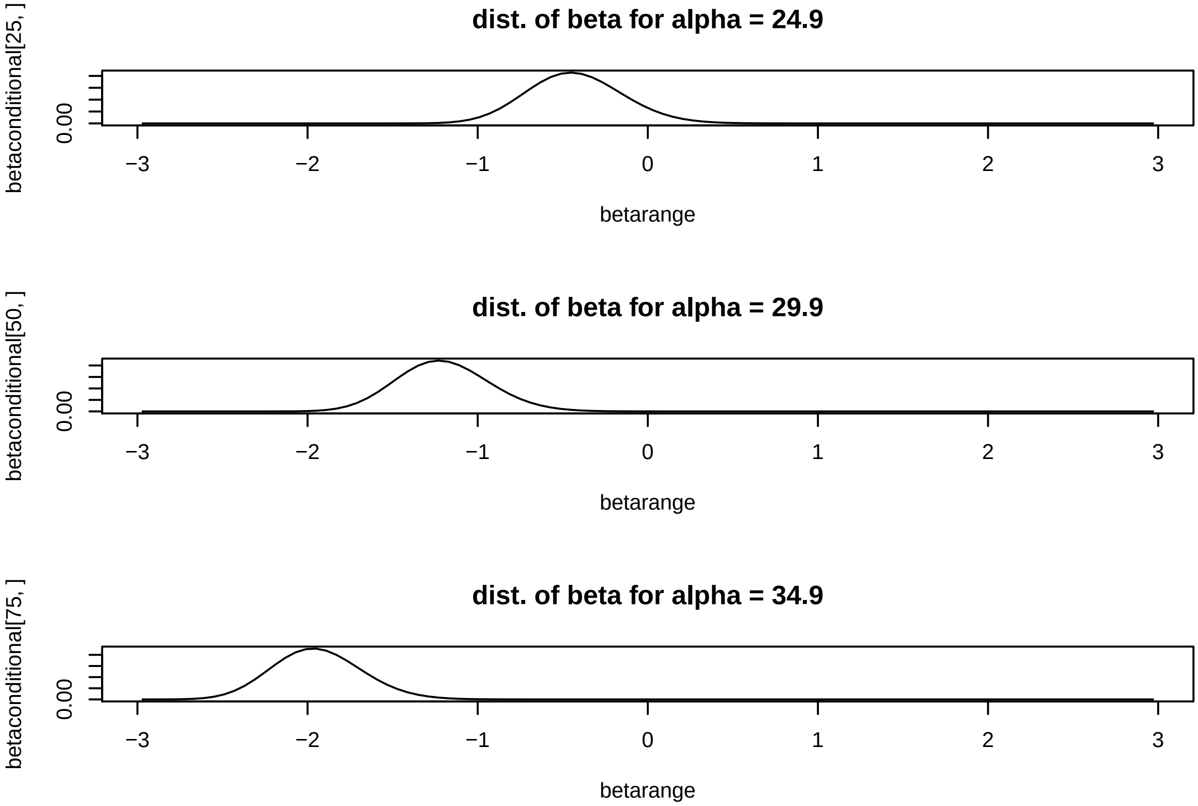
```
alpharange[50]
```

```
## [1] 29.9
```

```
alpharange[75]
```

```
## [1] 34.9
```

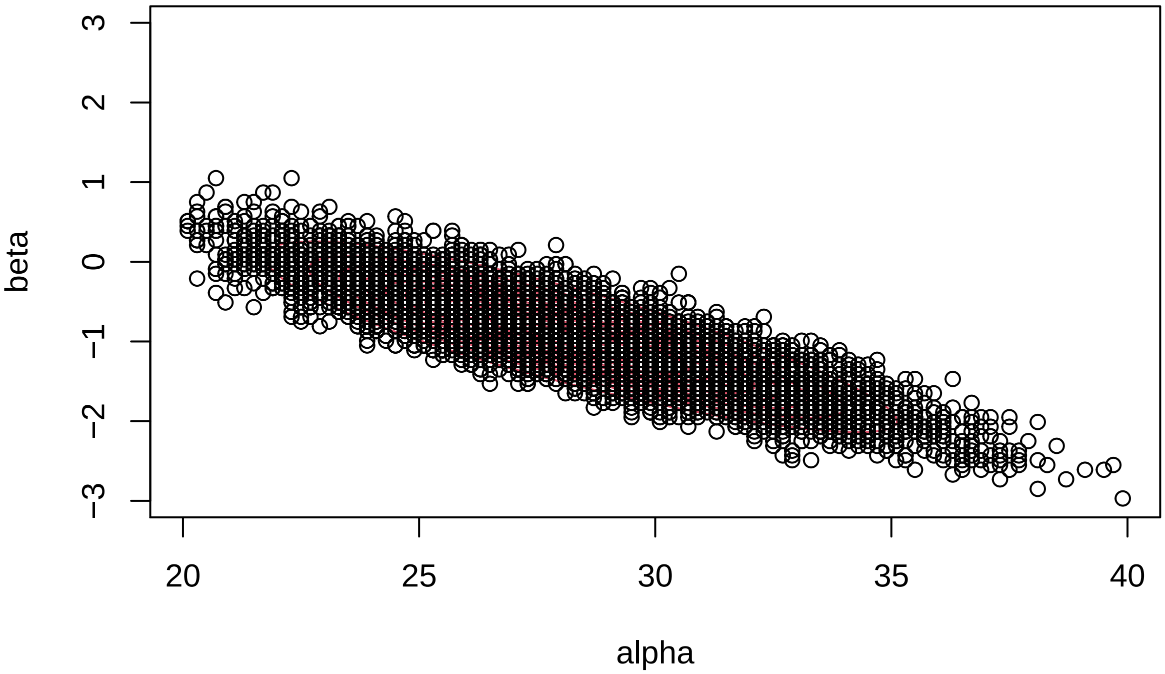
```
par(mfrow=c(3,1))
plot(betarange,betaconditional[25,],type="l",main="dist. of beta for alpha = 24.9")
plot(betarange,betaconditional[50,],type="l",main="dist. of beta for alpha = 29.9")
plot(betarange,betaconditional[75,],type="l",main="dist. of beta for alpha = 34.9")
```


dist. of beta for alpha = 24.9

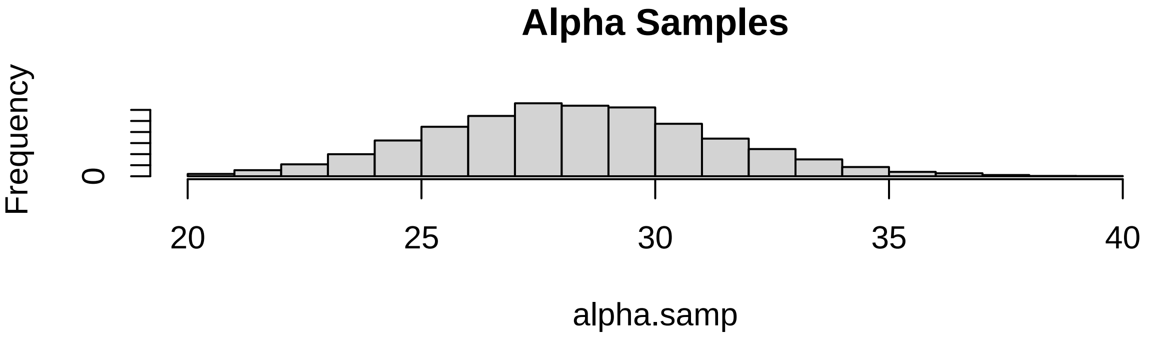
dist. of beta for alpha = 29.9

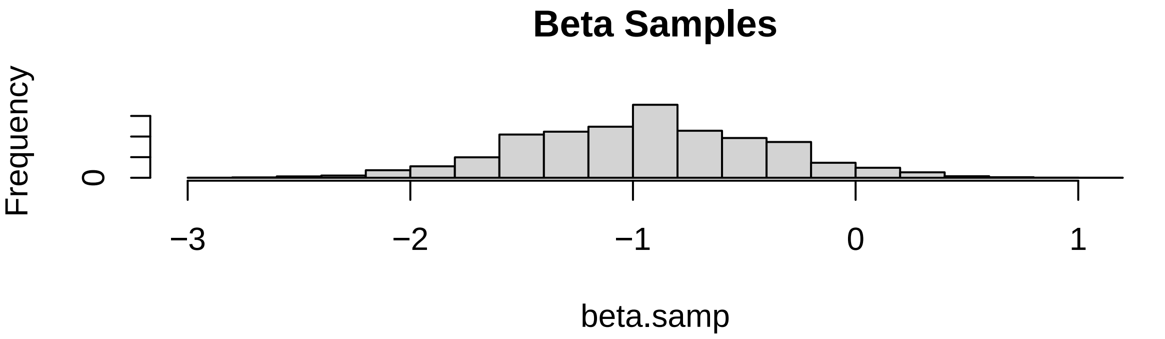
dist. of beta for alpha = 34.9

```r
## sampling grid values:

alpha.samp <- rep(NA,10000)
beta.samp <- rep(NA,10000)
for (m in 1:10000){
  a <- sample(1:100,size=1,replace=T,prob=alphamarginal)
  b <- sample(1:100,size=1,replace=T,prob=betaconditional[a,])
  alpha.samp[m] <- alpharange[a]
  beta.samp[m] <- betarange[b]
}

par(mfrow=c(1,1))
contour(alpharange,betarange,full,xlab="alpha",ylab="beta",drawlabels=F,col=2)
points(alpha.samp,beta.samp)
```



```
## calculating posterior means/intervals for alpha and beta

par(mfrow=c(2,1))
hist(alpha.samp,main="Alpha Samples")
hist(beta.samp,main="Beta Samples")
```





```
mean(alpha.samp)
```

```
## [1] 28.28166
```

```
mean(beta.samp)
```

```
## [1] -0.945912
```

```
alpha.sampsort <- sort(alpha.samp)
beta.sampsort <- sort(beta.samp)

alpha.sampsort[250]
```

```
## [1] 22.5
```

```
alpha.sampsort[9750]
```

```
## [1] 34.5
```

```
beta.sampsort[250]
```

```
## [1] -2.01
```

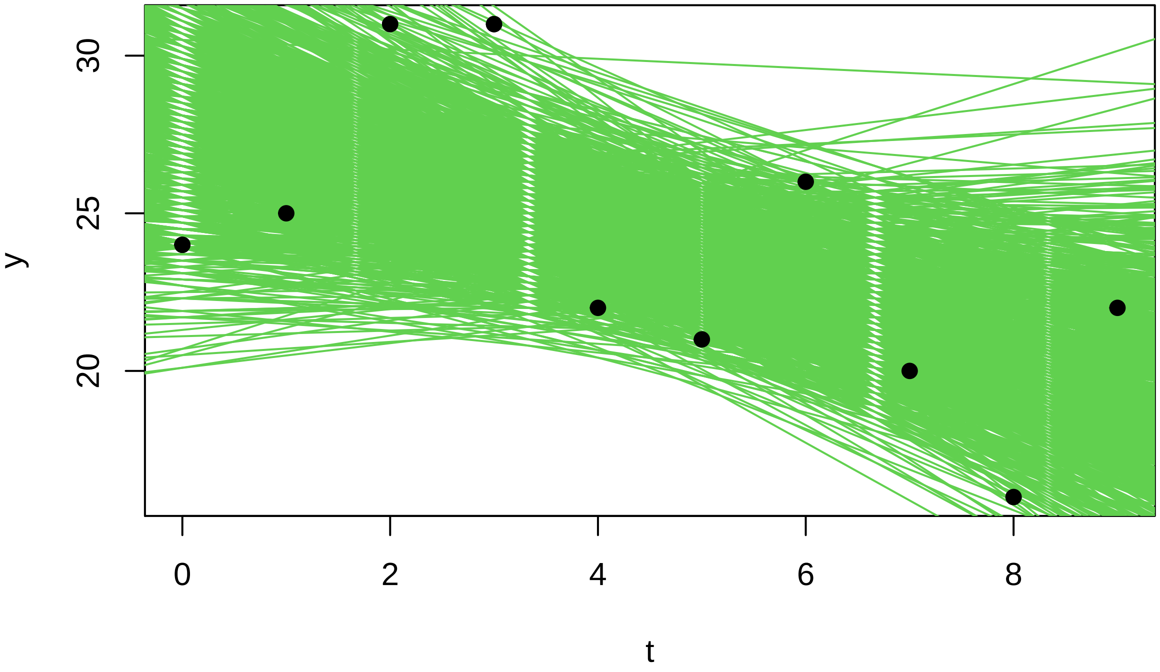
```
beta.sampsort[9750]
```

```
## [1] 0.09
```

```
sum(beta.samp >= 0)/10000
```

```
## [1] 0.043
```

```
par(mfrow=c(1,1))
plot(t,y,pch=19)
for (i in 1:1000){
  abline(alpha.samp[i],beta.samp[i],col=3)
}
points(t,y,pch=19)
```



```
## predicted new observation for 1986 (t = 10):

pred.rate <- alpha.samp + beta.samp*10

pred.accidents <- rep(NA,10000)
for (i in 1:10000){
  pred.accidents[i] <- rpois(1,pred.rate[i])
}

mean(pred.accidents)
```

```
## [1] 18.8306
```

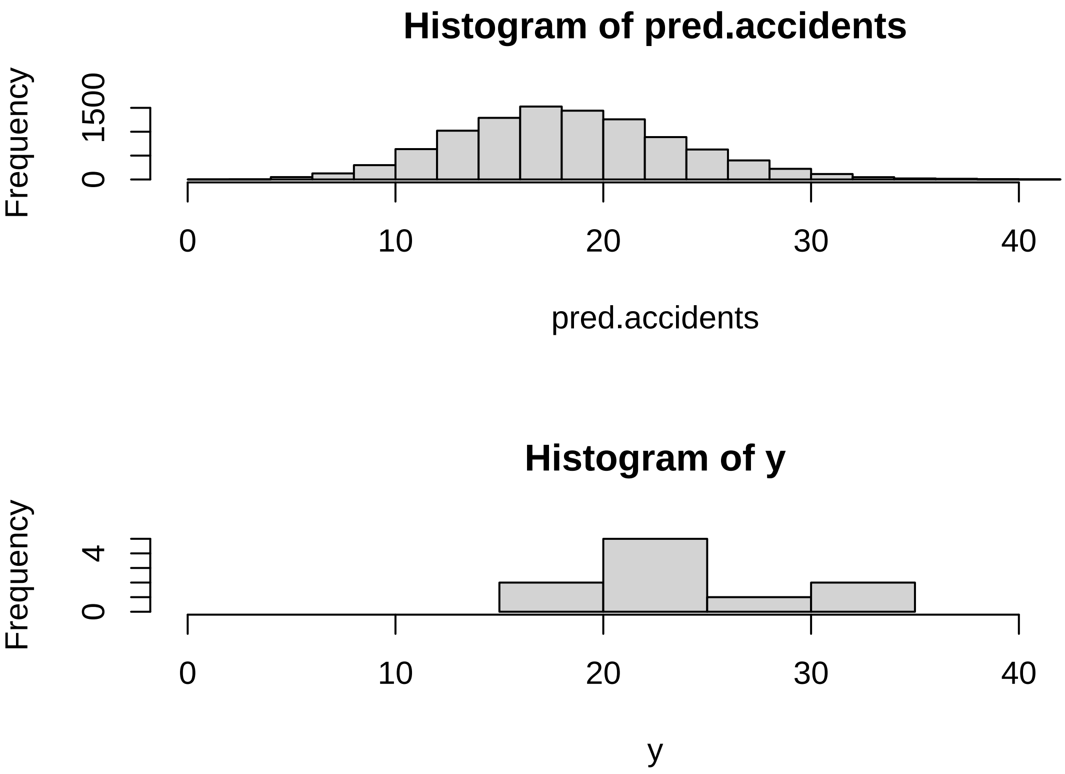
```
sort(pred.accidents)[250]
```

```
## [1] 9
```

```
sort(pred.accidents)[9750]
```

```
## [1] 30
```

```
par(mfrow=c(2,1))
hist(pred.accidents,xlim=c(0,45))
hist(y,xlim=c(0,45))
```


Histogram of pred.accidents
Frequency
pred.accidents
Histogram of y
Frequency
y