Gibbs sampling for simple linear regression

Chapter 3.2.1: Gibbs sampling

For observation $i = 1, \ldots, n$, let Y_i be the response and X_i be the covariate. The model is

$$Y_i \sim \text{Normal}(\alpha + \beta X_i, \sigma^2).$$

We select priors

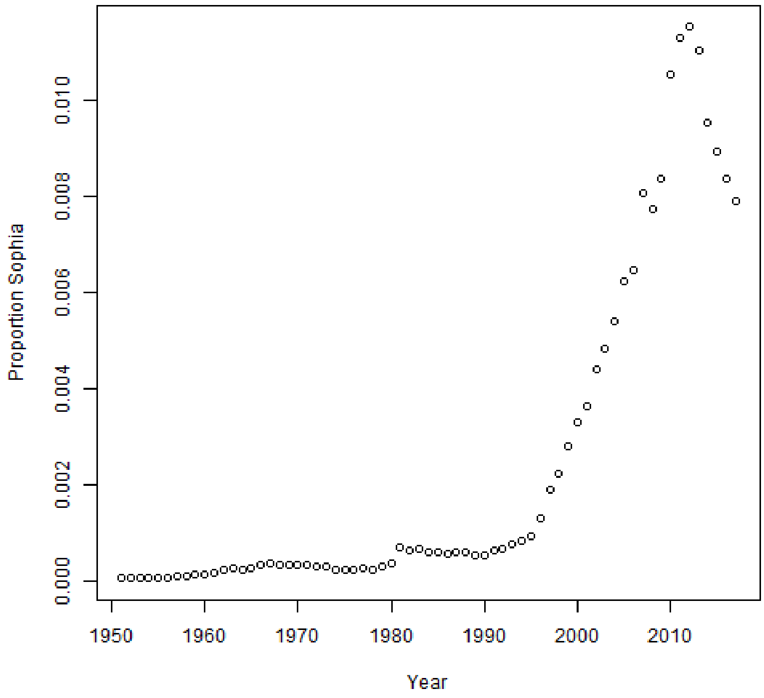
$$\alpha, \beta \sim \text{Normal}(\mu_0, \sigma_0^2) \quad \sigma^2 \sim \text{InvGamma}(a, b).$$

To illustrate the method we regress the log odds of a baby being named “Sophia" (Y) onto the year (X). To improve convergence we take X to be the year - 1984 (so that X is centered on zero).

```
### Load data and fit least squares
library(babynames)
dat <- babynames
dat <- dat[dat$name=="Sophia" & dat$sex=="F" & dat$year>1950,]
dat
```

```
## # A tibble: 67 x 5
##     year sex   name        n      prop
##    <dbl> <chr> <chr>   <int>     <dbl>
##  1  1951 F     Sophia    153 0.0000828
##  2  1952 F     Sophia    110 0.0000578
##  3  1953 F     Sophia    130 0.0000674
##  4  1954 F     Sophia    112 0.0000563
##  5  1955 F     Sophia    152 0.0000758
##  6  1956 F     Sophia    121 0.0000588
##  7  1957 F     Sophia    188 0.0000896
##  8  1958 F     Sophia    226 0.000109
##  9  1959 F     Sophia    277 0.000133
## 10  1960 F     Sophia    262 0.000126
## # ... with 57 more rows
```

```
yr  <- dat$year
p   <- dat$prop

X   <- dat$year - 1980
Y   <- log(p/(1-p))
n   <- length(X)

plot(yr,p,xlab="Year",ylab="Proportion Sophia")
```



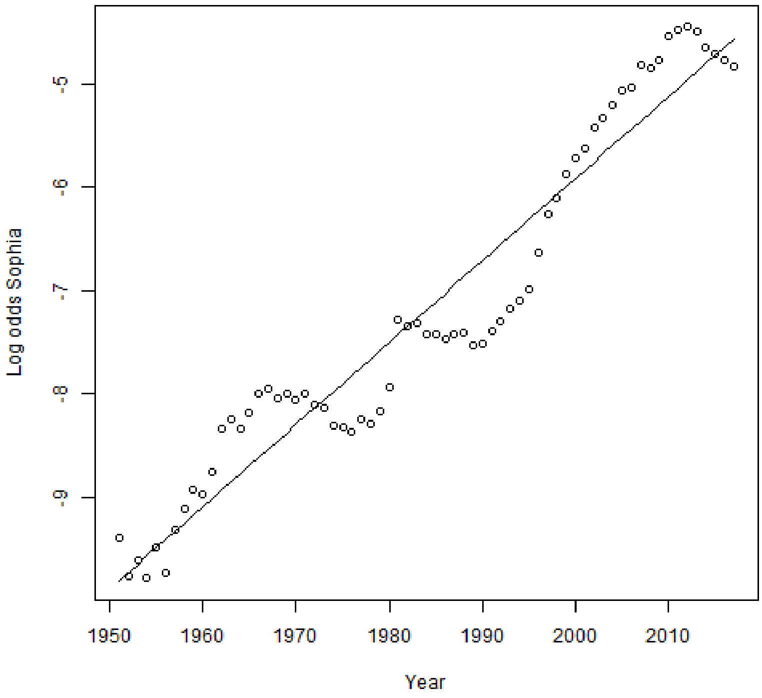

```
OLS <- lm(Y~X)
summary(OLS)
```

```
##
## Call:
## lm(formula = Y ~ X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.79800 -0.36517  0.03036  0.38820  0.61809
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.506061   0.053838 -139.42   <2e-16 ***
## X            0.079399   0.002726   29.12   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4315 on 65 degrees of freedom
## Multiple R-squared:  0.9288, Adjusted R-squared:  0.9277
## F-statistic: 848.2 on 1 and 65 DF,  p-value: < 2.2e-16
```

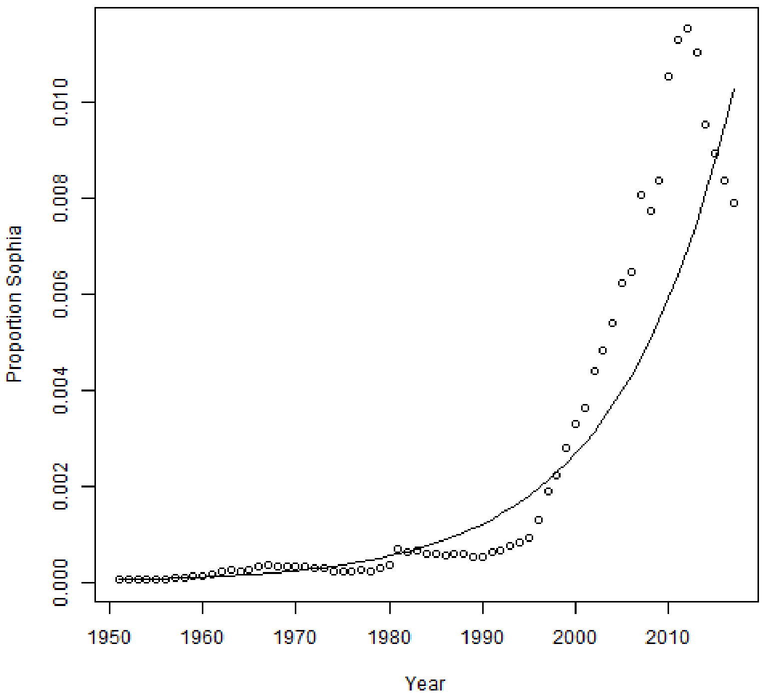
```
plot(yr,Y,xlab="Year",ylab="Log odds Sophia")
OLS$coef
```

```
## (Intercept)           X
## -7.50606055  0.07939896
```

```
y_hat <- OLS$coef[1]+OLS$coef[2]*X
lines(yr,y_hat)
```




```
# Plot fitted values on the proportion scale
plot(yr,p,xlab="Year",ylab="Proportion Sophia")
p_hat <- exp(y_hat)/(1+exp(y_hat))
lines(yr,p_hat)
```



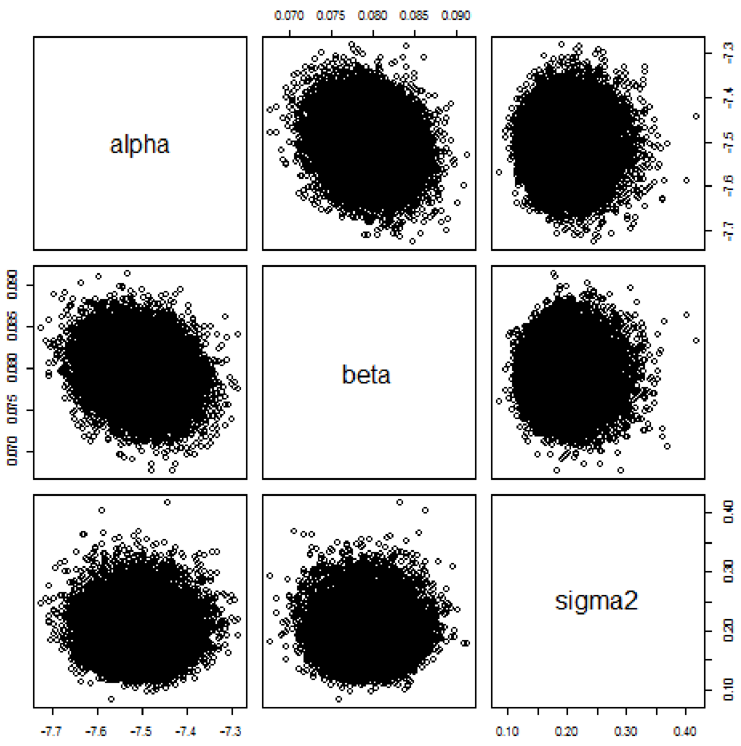

```
### Priors

 mu0 <- 0
 s20 <- 1000
 a   <- 0.01
 b   <- 0.01
```

MCMC!

```
n.iters <- 30000
keepers <- matrix(0,n.iters,3)
colnames(keepers)<-c("alpha","beta","sigma2")

# Initial values
alpha        <- OLS$coef[1]
beta         <- OLS$coef[2]
s2           <- var(OLS$residuals)
keepers[1,] <- c(alpha,beta,s2)

for(iter in 2:n.iters){

  # sample alpha

   V      <- n/s2+mu0/s20
   M      <- sum(Y-X*beta)/s2+1/s20
   alpha <- rnorm(1,M/V,1/sqrt(V))

  # sample beta

   V      <- sum(X^2)/s2+mu0/s20
   M      <- sum(X*(Y-alpha))/s2+1/s20
   beta   <- rnorm(1,M/V,1/sqrt(V))

  # sample s2|mu,Y,Z

   A  <- n/2 + a
   B  <- sum((Y-alpha-X*beta)^2)/2 + b
   s2 <- 1/rgamma(1,A,B)

  # keep track of the results
   keepers[iter,] <- c(alpha,beta,s2)

}
```

Plots of the joint posterior distribution.

```
pairs(keepers)
```



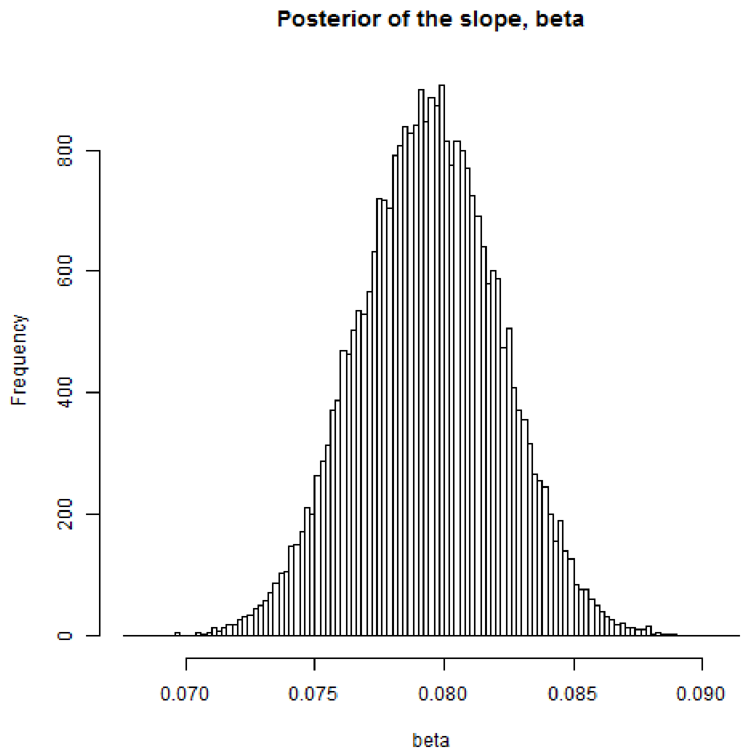
Summarize the marginal distributions in a table

```
output <- matrix(0,3,4)
rownames(output) <- c("Intercept","Slope","sigma2")
colnames(output) <- c("Mean","SD","Q025","Q975")

output[,1] <- apply(keepers,2,mean)
output[,2] <- apply(keepers,2,sd)
output[,3] <- apply(keepers,2,quantile,0.025)
output[,4] <- apply(keepers,2,quantile,0.975)

kable(output,digits=3)
```

	Mean	SD	Q025	Q975
Intercept	-7.506	0.055	-7.614	-7.399
Slope	0.079	0.003	0.074	0.085
sigma2	0.192	0.035	0.135	0.271

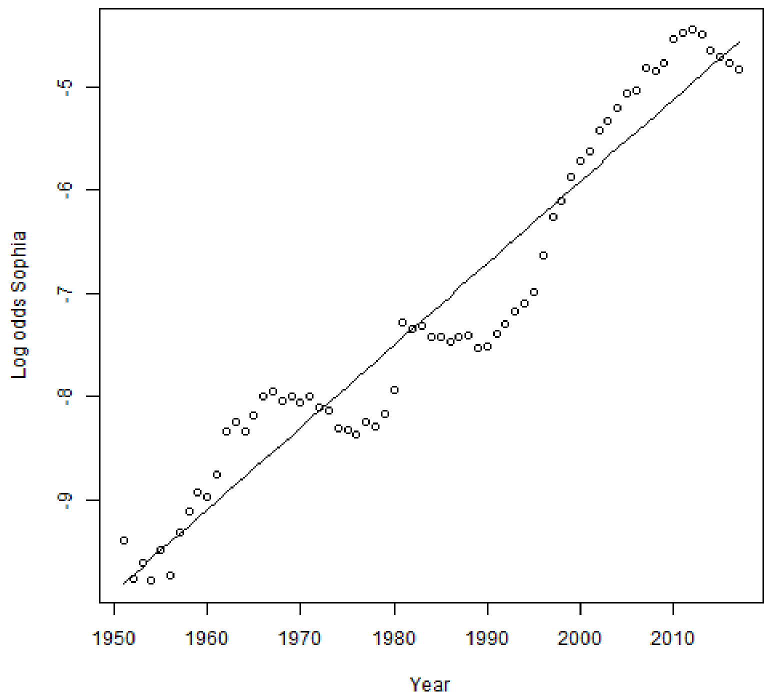
Plot the marginal posterior $f(\beta \mid Y)$.

```
beta <- keepers[,2]
hist(beta,main="Posterior of the slope, beta",breaks=100)
```



Plot the fitted regression line

```
fit_bayes <- output[1:2,1]
plot(yr,Y,xlab="Year",ylab="Log odds Sophia")
lines(yr,fit_bayes[1]+fit_bayes[2]*X)
```



Processing math: 100%