# lecture 4.5 example 1

## preliminaries

```r
#clear workspace
rm(list=ls())
#initalize random seed
set.seed(123)
```

## set simulation parameters

```r
nObs = 100
xGrid = seq(-1,1, by=0.1)

xPredict = c(-4,-2,0,2,4)
nPredictSamples = 10000

beta0 = 0
beta1 = 1
sigma = 1
```

## build priors

```r
boundBetaGrid = 2
sizeBetaGrid = 0.05
boundSigmaGrid = 2
sizeSigmaGrid = 0.05

beta0Grid = seq(-boundBetaGrid,boundBetaGrid, by = sizeBetaGrid)
beta1Grid = seq(-boundBetaGrid,boundBetaGrid, by = sizeBetaGrid)
sigmaGrid = seq(0.001,boundSigmaGrid, by = sizeSigmaGrid)
nBeta0Grid = length(beta0Grid)
nBeta1Grid = length(beta1Grid)
nSigmaGrid = length(sigmaGrid)

# prior
priorSigma = 1 / sigmaGrid^2
```

## define model functions

```r
#likelihood
likelihood = function(y,x, b0L, b1L, sL){
  loglike = sum(log(dnorm(y-b0L-b1L*x, mean = 0, sd = sL)))
  like = exp(loglike)
  return(like)
}
```

```r
#compute posterior function
compPost = function(y,x){
  #initialize local posterior
  post = array( rep(-1, nBeta0Grid * nBeta1Grid * nSigmaGrid ),
               dim = c(nBeta0Grid, nBeta1Grid, nSigmaGrid ))
  # compute posterior
  for (nBeta0 in 1:nBeta0Grid) {
    b0 = beta0Grid[nBeta0]
    for (nBeta1 in 1:nBeta1Grid) {
      b1 = beta1Grid[nBeta1]
      for (nSigma in 1:nSigmaGrid) {
        s = sigmaGrid[nSigma]
        post[nBeta0,nBeta1,nSigma] = likelihood(y,x, b0, b1, s) * priorSigma[nSigma]
      }
    }
  }
  # normalize posterior
  post = post / (sum(post) *sizeBetaGrid^2 * sizeSigmaGrid)
  # return
  return(post)
}
```

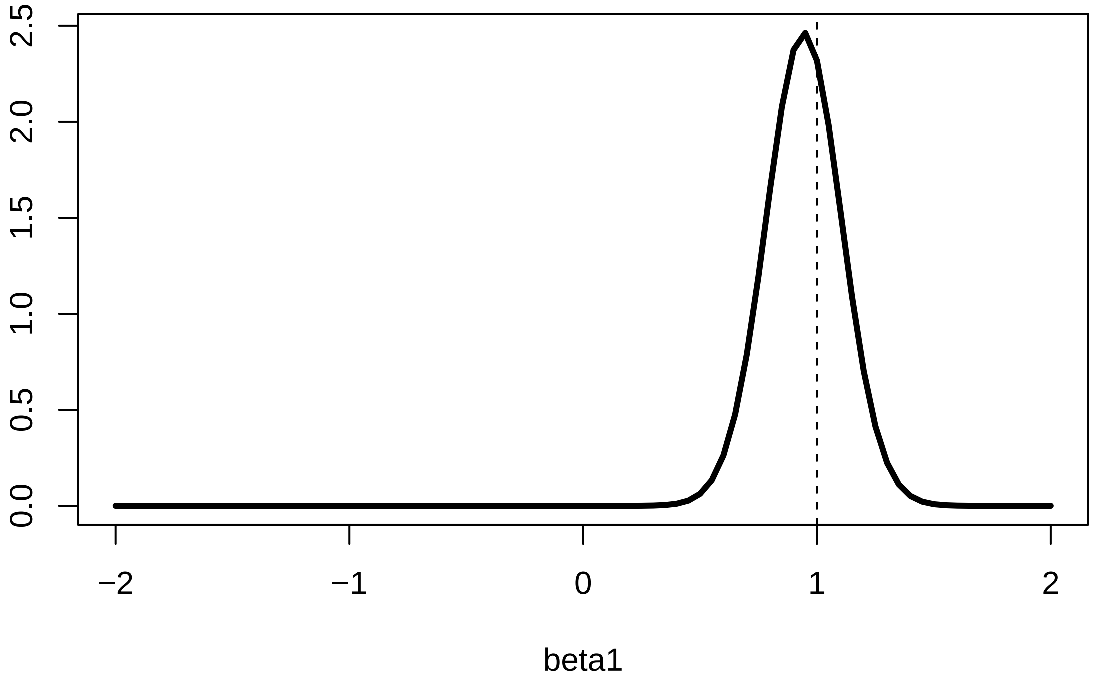
## fit model

```r
#initialize arrays
postFinal = array( rep(-1, nBeta0Grid * nBeta1Grid * nSigmaGrid ),
                   dim = c(nBeta0Grid, nBeta1Grid, nSigmaGrid ))


#generate data and fit posterior
x = sample(xGrid, nObs, replace = TRUE )
y = rnorm(nObs, mean = beta0 + beta1 * x, sd = sigma)
postFinal = compPost(y,x)
```

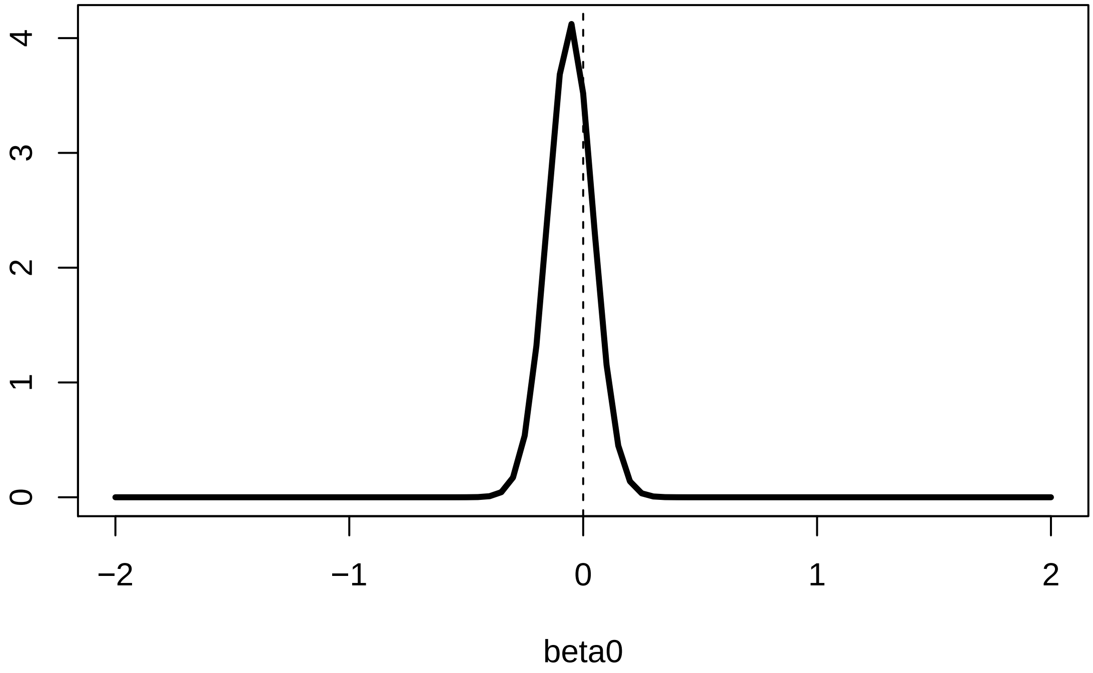
## check model fit

```r
#compute marginal posteriors
margPostBeta0 = apply(postFinal,c(1),sum)
margPostBeta0 = margPostBeta0 / (sum(margPostBeta0) * sizeBetaGrid)

margPostBeta1 = apply(postFinal,c(2),sum)
margPostBeta1 = margPostBeta1 / (sum(margPostBeta1) * sizeBetaGrid)

margPostSigma = apply(postFinal,c(3),sum)
margPostSigma = margPostSigma / (sum(margPostSigma) * sizeSigmaGrid)
```

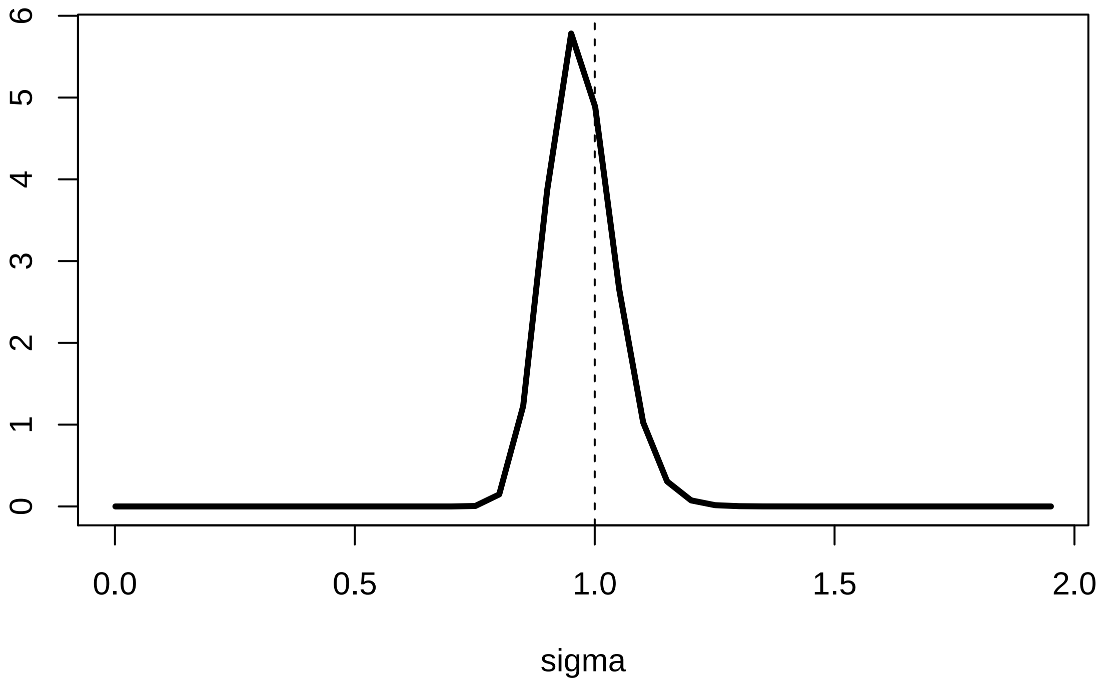
```r
plot(beta1Grid, margPostBeta1,
     xlab = "beta1", ylab="",
     type = "l", lwd = 3)
abline(v=beta1, lty=2)
```



```
plot(beta0Grid, margPostBeta0,
     xlab = "beta0", ylab="",
     type = "l", lwd = 3)
abline(v=beta0, lty=2)
```



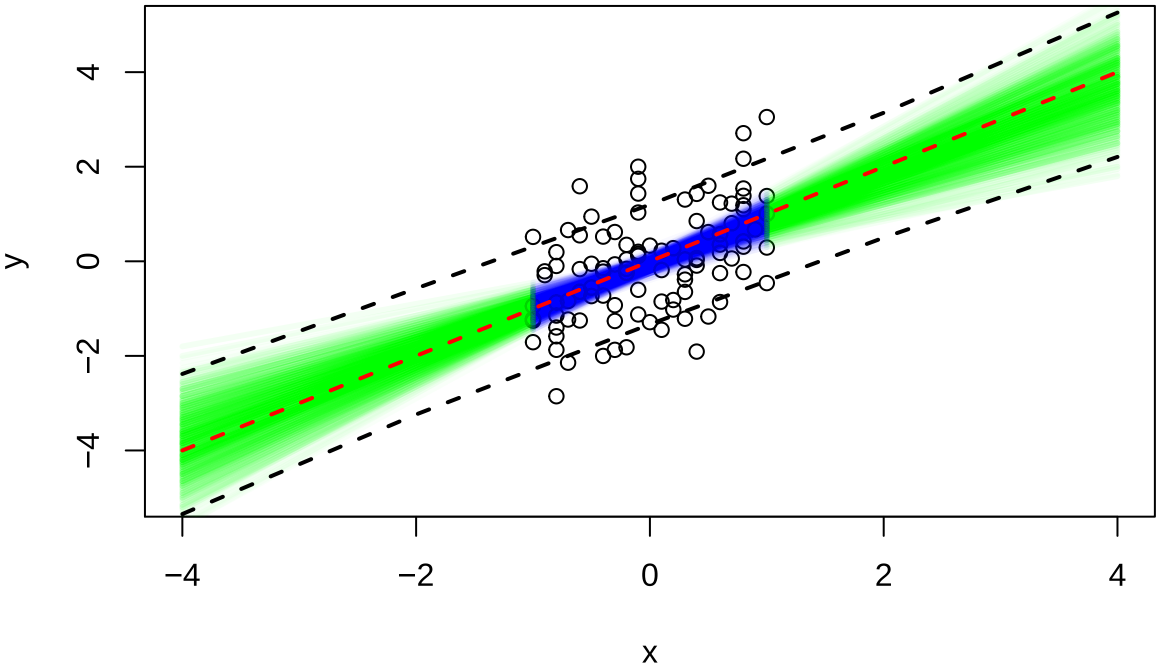
```
plot(sigmaGrid, margPostSigma,
     xlab = "sigma", ylab="",
     type = "l", lwd = 3)
abline(v=sigma, lty=2)
```



### make predictions and visualize results

```
#initialize storage arrays
predicted = array(rep(-1, length(xPredict) * nPredictSamples),
                  dim = c(length(xPredict), nPredictSamples))

#build conditional posteriors for sampling purposes
margBeta0 = apply(postFinal,c(1),sum)

margBeta1GivenBeta0 = array(rep(-1,nBeta0Grid * nBeta1Grid),
                            dim= c(nBeta0Grid, nBeta1Grid))
for (nBeta0 in 1:nBeta0Grid) {
  margBeta1GivenBeta0[nBeta0,] = apply(postFinal[nBeta0,,],c(1),sum)
}

#predictions
for (t in 1:length(xPredict)) {
  xNew = xPredict[t]
  for (sim in 1:nPredictSamples) {
    b0Index = sample(1:nBeta0Grid, 1, prob=margBeta0)
    b1Index = sample(1:nBeta1Grid, 1, prob=margBeta1GivenBeta0[b0Index,])
    sigIndex = sample(1:nSigmaGrid, 1, prob=postFinal[b0Index,b1Index,])
    b0Sample = beta0Grid[b0Index]
    b1Sample = beta1Grid[b1Index]
    sigSample = sigmaGrid[sigIndex]
    predicted[t,sim] = rnorm(1,b0Sample + xNew * b1Sample, sigSample)
  }
}

predict10 = rep(0,length(xPredict))
```

```r
for (t in 1:length(xPredict)) {
  predict10[t] = quantile(predicted[t,], probs = 0.1)
}

predict90 = rep(0,length(xPredict))
for (t in 1:length(xPredict)) {
  predict90[t] = quantile(predicted[t,], probs = 0.9)
}

# initialize plot
plot(x,y,xlim=c(-4,4),ylim=c(-5,5))
#plot posterior reg lines
xLeft = seq(-4,-1,by=0.1)
xRight = seq(1,4,by=0.1)
for (sim in 1:1000) {
  b0Index = sample(1:nBeta0Grid, 1, prob=margBeta0)
  b1Index = sample(1:nBeta1Grid, 1, prob=margBeta1GivenBeta0[b0Index,])
  b0Sample = beta0Grid[b0Index]
  b1Sample = beta1Grid[b1Index]
  points(xGrid, b0Sample + b1Sample*xGrid, type="l",lwd=3,
         col=rgb(red=0.0, green=0.0, blue=1.0, alpha=0.025))
  points(xLeft, b0Sample + b1Sample*xLeft, type="l",lwd=3,
         col=rgb(red=0.0, green=1.0, blue=0.0, alpha=0.025))
  points(xRight, b0Sample + b1Sample*xRight, type="l",lwd=3,
         col=rgb(red=0.0, green=1.0, blue=0.0, alpha=0.025))
}
points(seq(-4,4,by=0.1), beta0 + beta1*seq(-4,4,by=0.1), lwd=2, col=2, lty=2, type="l")
points(xPredict,predict10, type="l", lwd=2, lty=2)
points(xPredict,predict90, type="l", lwd=2, lty=2)
```


4
2
0
−2
−4
y
−4
−2
0
2
4
x