Lecture 3.3 example 1

preliminaries

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

set simulation parameters

Note: Change precH and n to generate different examples in lecture

```
nSim = 10000
n     = 1
prec = 1 / 2^2
mu0 = 0
muH = 0
prec0 = 1
precH = prec0 / 10

# useful objects
sigma = sqrt(1/prec)
sigma0 = sqrt(1/prec0)
sigmaH = sqrt(1/precH)
```

define functions

```
priorSample = function(nS, muP, sigmaP) {
  return(rnorm(nS, muP, sigmaP))
}

sampleData = function(nS, mu) {
  return(rnorm(nS, mu, sigma))
}
```

run simulations

```
#initalize results arrays
error0 = rep(-1, nSim)
errorH = rep(-1, nSim)

#simulation loop
for (t in 1:nSim) {
  # draw muT
  muT = priorSample(1, mu0, sigma0)
  # simulate data
  data = sampleData(n, muT)
```

```
  # define useful objects
  sampleMean = mean(data)
  # compute posterior based on true priors (analytic solution)
  post0Mean = (mu0 * prec0 + sampleMean * n * prec) / (prec0 + n * prec)
  post0Var = 1 / (prec0 + n * prec)
  # compute posterior based on H alternative priors (analytic solution)
  postHMean = (mu0 * precH + sampleMean * n * prec) / (precH + n * prec)
  postHVar = 1 / (precH + n * prec)
  # compute errors
  error0[t] = abs(post0Mean - muT)
  errorH[t] = abs(postHMean - muT)
}
```

visualize results

```
plot(density(errorH), ylim = c(0,2), xlim = c(0,4),
     col = "red", lwd=3, lty=2,
     xlab = "|postMean - muTrue|", ylab="",
     main = "")
lines(density(error0), lwd=3)
legend("topright", legend = c("using true prior", "using alternative prior"),
       col=c("black","red"), lty=c(1,2), lwd=c(3,3))
```