

# Lecture 3.2 example 2

## preliminaries

```
#clear work space  
rm(list=ls())  
#set random seed  
set.seed(123)
```

## simulation params & structures

```
# simulation parameters  
nObs = 100  
muTrue = 75  
sigTrue = 20  
  
# prior parameters  
muMean = 50  
muSd = 50  
  
# build grid  
nGridPoints = 100  
muGridMin = 0  
muGridMax = 100  
muGrid = seq(muGridMin, muGridMax, length.out = nGridPoints)  
muGridSize = (muGridMax - muGridMin) / nGridPoints
```

## define key functions

```
# compute posterior  
computePost = function(data, sigTrue, prior){  
  #initialize posterior matrix  
  post = rep(-1, nGridPoints )  
  #fill out the posterior  
  for (t in 1:nGridPoints) {  
    muVal = muGrid[t]  
    #compute data likelihood  
    loglike = sum(log(dnorm(data, muVal, sigTrue)))  
    # update posterior matrix cell  
    post[t] = exp(loglike) * prior[t]  
  }  
  # normalize the posterior & return  
  post = post / ( sum(post) * muGridSize)  
  return(post)  
}  
  
# compute likelihood  
computeLike = function(data, sigTrue, prior){  
  #initialize likelihood matrix
```

```

like = rep(-1, nGridPoints )
#fill out the likelihood
for (t in 1:nGridPoints) {
  muVal = muGrid[t]
  #compute data likelihood
  like[t] = prod(dnorm(data, muVal, sigTrue))
}
return(like)
}

```

visualize relationship between prior, likelihood and posterior

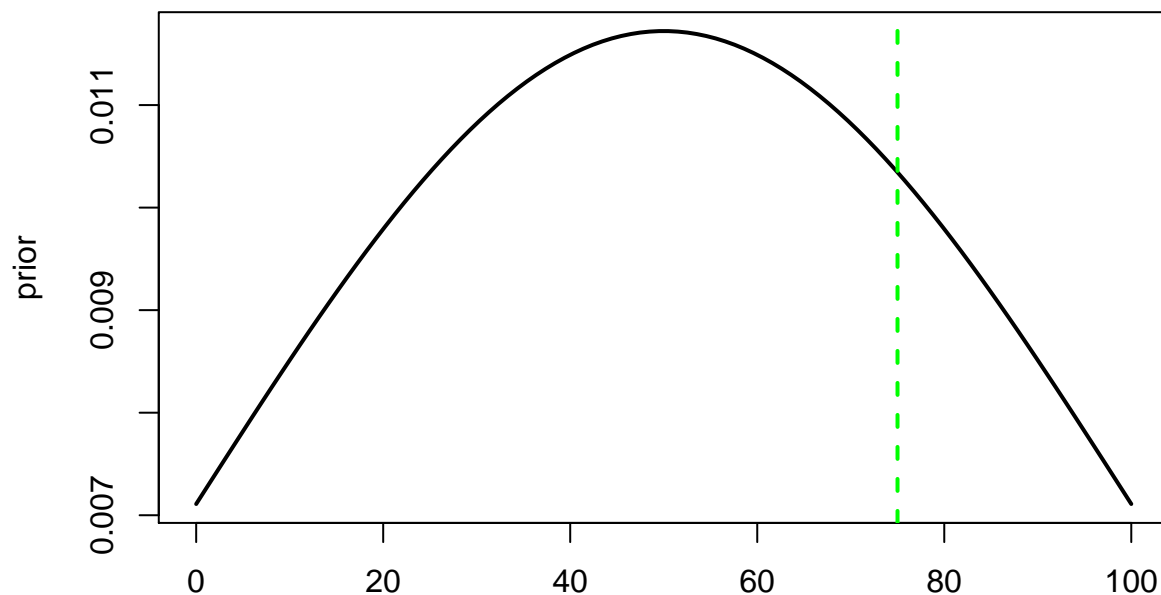
```

#build priors for mu
muPrior = dnorm(muGrid,muMean,muSd)
muPrior = muPrior / ( sum(muPrior) * muGridSize )

# simulate dataset
data = rnorm(nObs, muTrue, sigTrue)
# compute posterior
post = computePost(data[1:10], sigTrue, muPrior)
# compute likelihood
likeh = computeLike(data[1:10], sigTrue, muPrior)

plot(muGrid, muPrior, type="l", lwd=2,
     xlab = "", ylab = "prior")
abline(v=muTrue, lwd = 2, lty = 2, col="green")

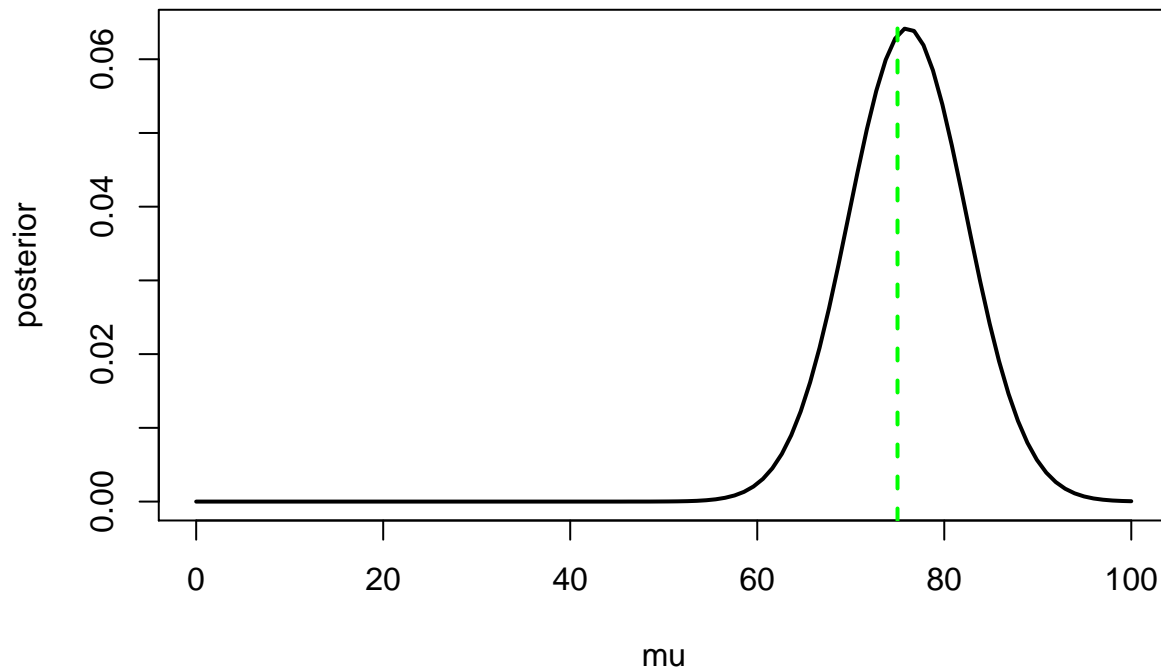
```



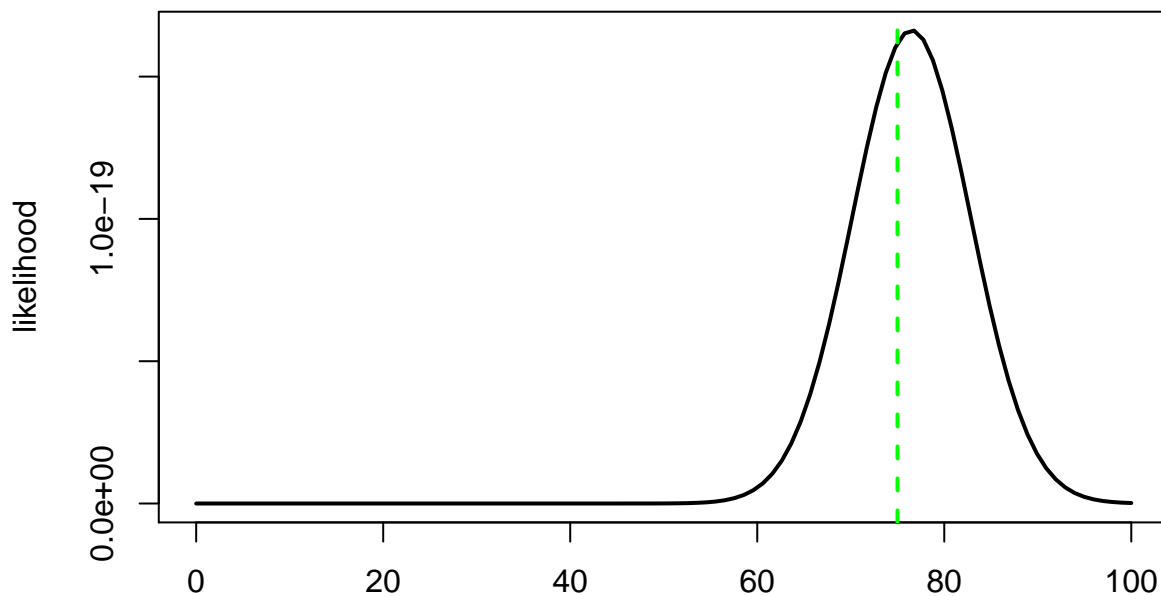
```

plot(muGrid,post, type="l", lwd=2,
     xlab = "mu", ylab = "posterior")
abline(v=muTrue, lwd = 2, lty = 2, col="green")

```



```
plot(muGrid, likeh, type="l", lwd=2,
      xlab = "", ylab = "likelihood")
abline(v=muTrue, lwd = 2, lty = 2, col="green")
```

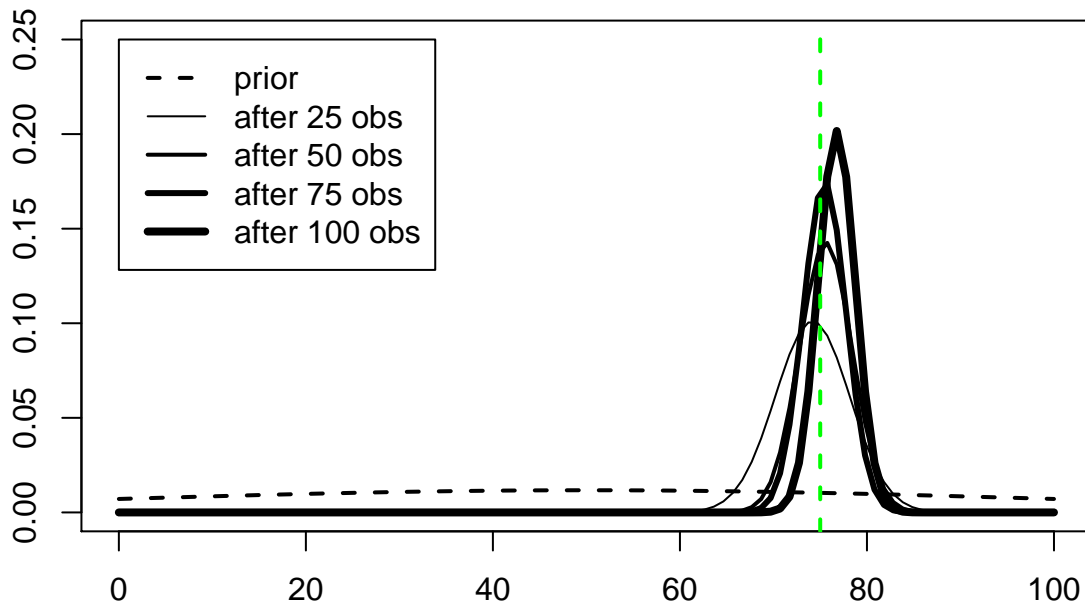


plot posterior evolution

```
plot(muGrid, computePost(data[1:25], sigTrue, muPrior),
      type="l", lwd=1, xlab = "", ylab = "", ylim = c(0,0.25),
      main = "posterior evolution")
points(muGrid, computePost(data[1:50], sigTrue, muPrior), type="l", lwd=2)
points(muGrid, computePost(data[1:75], sigTrue, muPrior), type="l", lwd=3)
points(muGrid, computePost(data[1:100], sigTrue, muPrior), type="l", lwd=4)
```

```
points(muGrid, muPrior, , type="l", lwd=2, lty = 2)
abline(v=muTrue, lwd = 2, lty = 2, col="green")
legend(0,0.25, legend = c("prior", "after 25 obs", "after 50 obs", "after 75 obs", "after 100 obs"),
      lty = c(2,1,1,1,1), lwd = c(2,1,2,3,4))
```

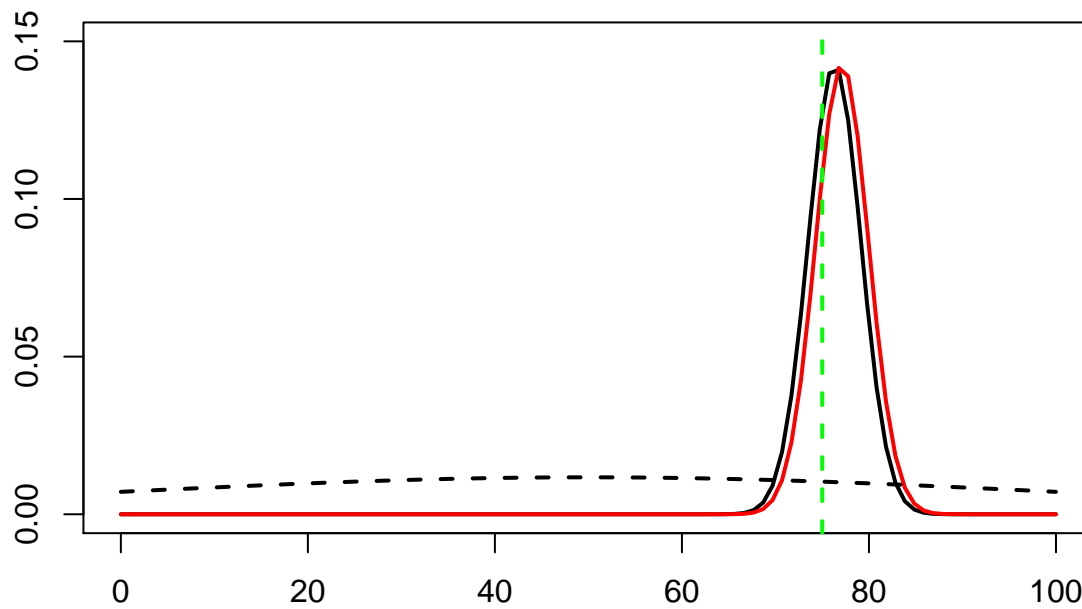
### posterior evolution



### compare posterior in even and odd trials

```
dataEven = data[(1:nObs) %% 2 == 0]
dataOdd = data[(1:nObs) %% 2 == 1]

plot(muGrid, computePost(dataEven, sigTrue, muPrior),
     type="l", lwd=2, xlab = "", ylab = "", ylim = c(0,0.15))
points(muGrid, computePost(dataOdd, sigTrue, muPrior),
       type="l", lwd=2, col = "red")
points(muGrid, muPrior, , type="l", lwd=2, lty = 2)
abline(v=muTrue, lwd = 2, lty = 2, col="green")
```



## role of priors

```
muPriorA = dnorm(muGrid, 50, 50)
muPriorA = muPriorA / ( sum(muPriorA) * muGridSize )
muPriorB = dnorm(muGrid, -50, 50)
muPriorB = muPriorB / ( sum(muPriorB) * muGridSize )
muPriorC = dnorm(muGrid, 50, 5)
muPriorC = muPriorC / ( sum(muPriorC) * muGridSize )

plot(muGrid, computePost(data, sigTrue, muPriorA), col = "black",
     type="l", lwd=2, xlab = "", ylab = "", ylim = c(0,0.25))
points(muGrid, computePost(data, sigTrue, muPriorB), col = "red",
       type="l", lwd=2)
points(muGrid, computePost(data, sigTrue, muPriorC), col = "blue",
       type="l", lwd=2)
points(muGrid, muPriorA, col = "black", type="l", lwd=2, lty=2)
points(muGrid, muPriorB, col = "red", type="l", lwd=2, lty=2)
points(muGrid, muPriorC, col = "blue", type="l", lwd=2, lty=2)
abline(v=muTrue, lwd = 2, lty = 2, col="green")
legend(0,0.25,
      legend = c("prior = Normal(50,2500)", "prior = Normal(-50,2500)", "prior = Normal(50,25)"),
      lty = c(1,1,1), col = c("black","red","blue"))
```

