lecture 3.2 example 4

preliminaries

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

simulation params & structures

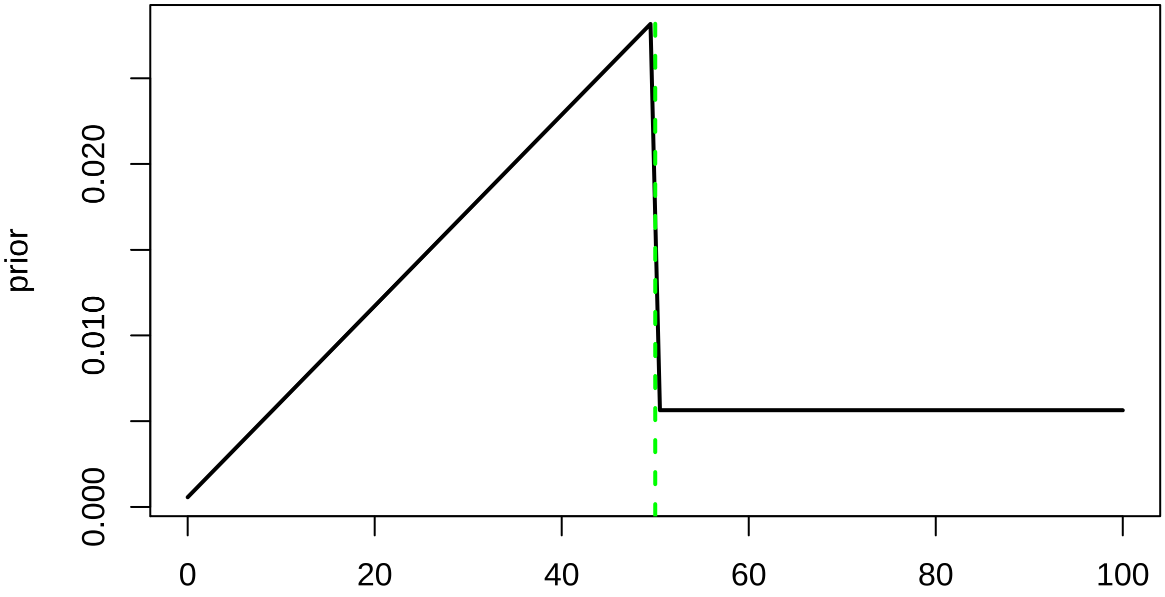
```
# simulation parameters
nObs = 10
muTrue = 50
sigTrue = 20

# build grid
nGridPoints = 100
muGridMin = 0
muGridMax = 100
muGrid = seq(muGridMin, muGridMax,length.out = nGridPoints)
muGridSize = (muGridMax - muGridMin) / nGridPoints
```

define key functions

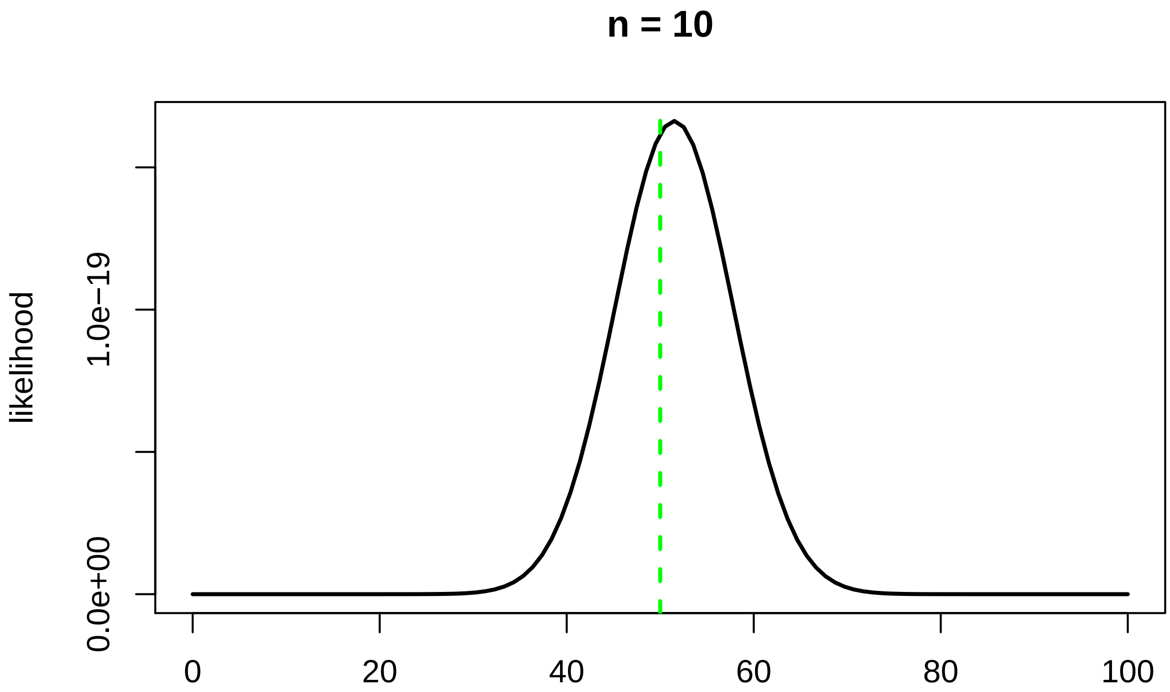
```
# compute posterior
computePost = function(data, sigTrue, prior){
  #initialize posterior matrix
  post = rep(-1, nGridPoints )
  #fill out the posterior
  for (t in 1:nGridPoints) {
      muVal = muGrid[t]
      #compute data likelihood
      loglike = sum(log(dnorm(data, muVal, sigTrue)))
      # 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)
}
```

simulate and visualize relationship between prior, likelihood and posterior

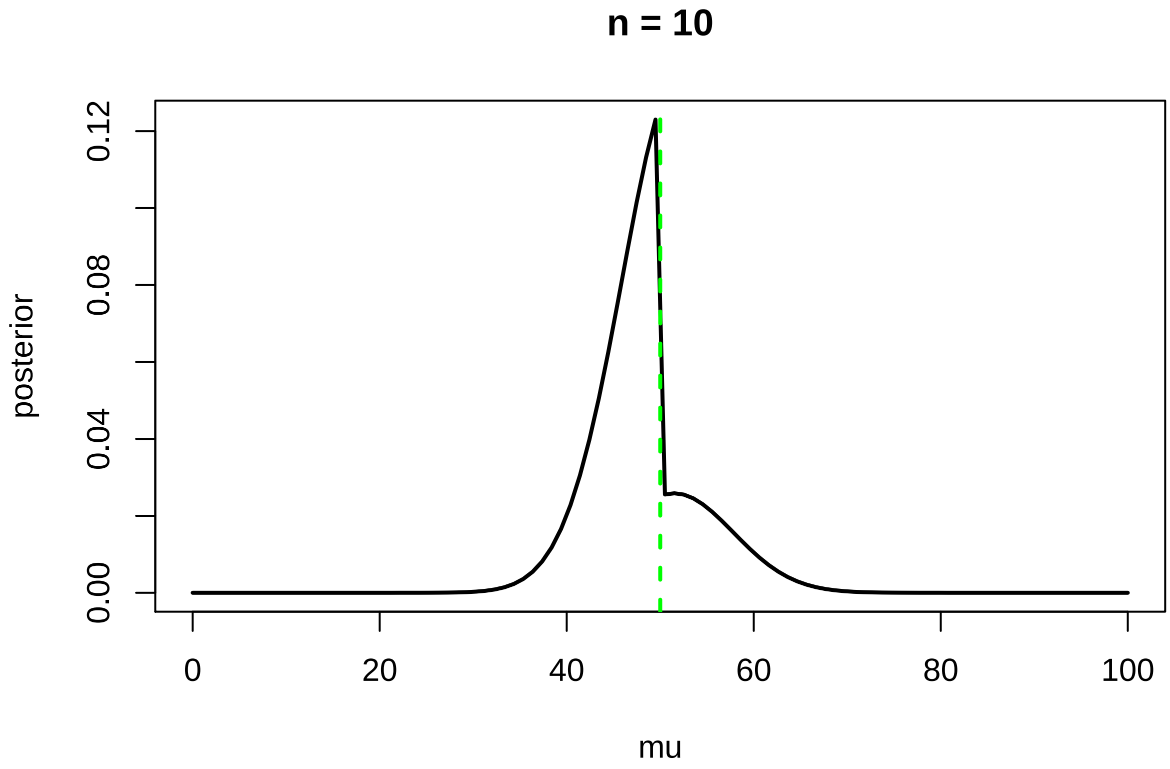
```
# define priors for mu
muPrior = 1:nGridPoints
muPrior[muPrior>50] = 10
muPrior = muPrior / ( sum(muPrior) * muGridSize )

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

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



```
plot(muGrid, likeh, type="l", lwd=2,
     xlab = "", ylab = "likelihood",
     main = "n = 10")
abline(v=muTrue, lwd = 2, lty = 2, col="green")
```



```r
plot(muGrid,post, type="l", lwd=2,
     xlab = "mu", ylab = "posterior",
     main = "n = 10")
abline(v=muTrue, lwd = 2, lty = 2, col="green")
```


n = 10
posterior
mu