lecture 2.2 example 3

preliminaries

```r
# clear environment
rm(list = ls())

# set random seed
set.seed(123)
```

define params

```r
#define grid size
nGridPoints = 50
pGrid = seq(from = 0, to = 1,length.out = nGridPoints)
gridSize = 1 / nGridPoints

# prior params
aPrior = 5
bPrior = 5
```

define simulation functions

```r
computePost = function(data, prior) {
  nWater = sum(data)
  nData = length(data)
  likelihood = dbinom(x = nWater, size = nData, prob = pGrid)
  post = likelihood * prior
  post = post / ( sum(post) * gridSize )
  return(post)
}
```

load data

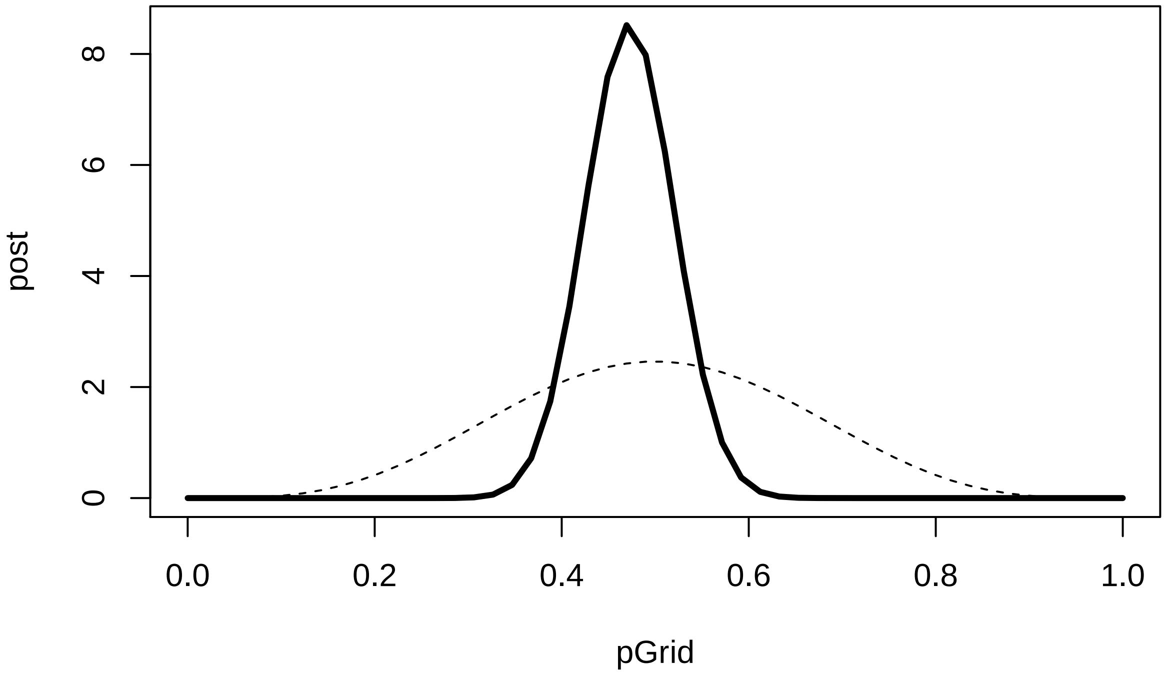
Assumed to be in the Desktop (in Mac system)

```r
data = read.csv("~/Desktop/data_globe_tossing.csv")
```

compute posterior with full dataset

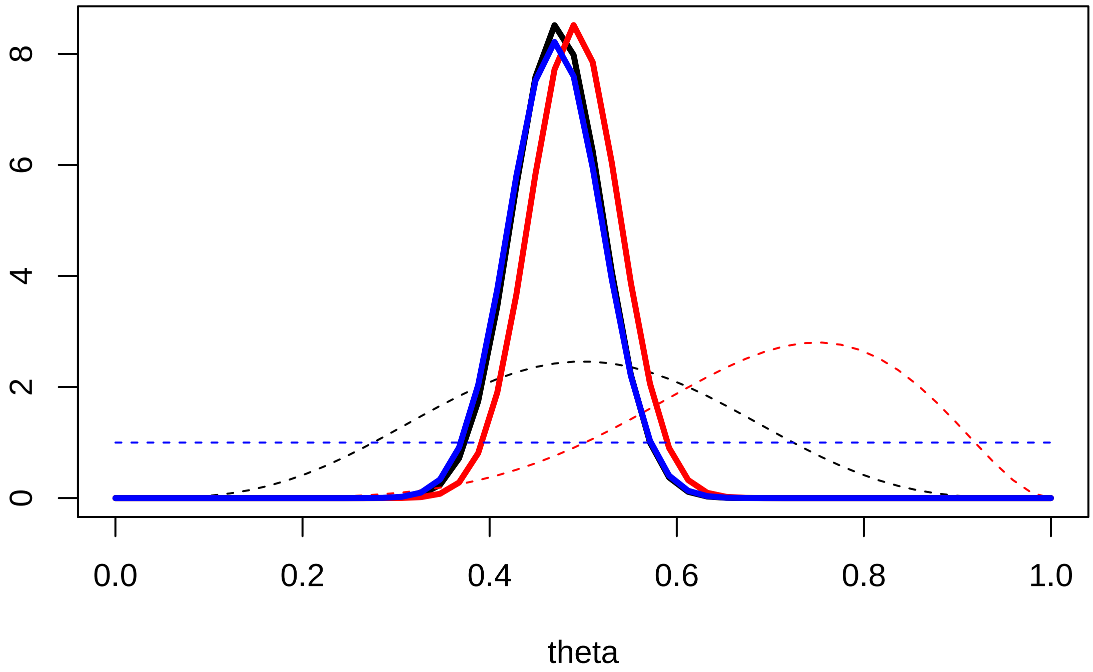
```r
prior = dbeta(x = pGrid, shape1 = aPrior, shape2 = bPrior)
post = computePost(data$ball_1, prior)

plot(pGrid, post, type="l", lwd = 3)
points(pGrid, prior, type="l", lty=2)
```



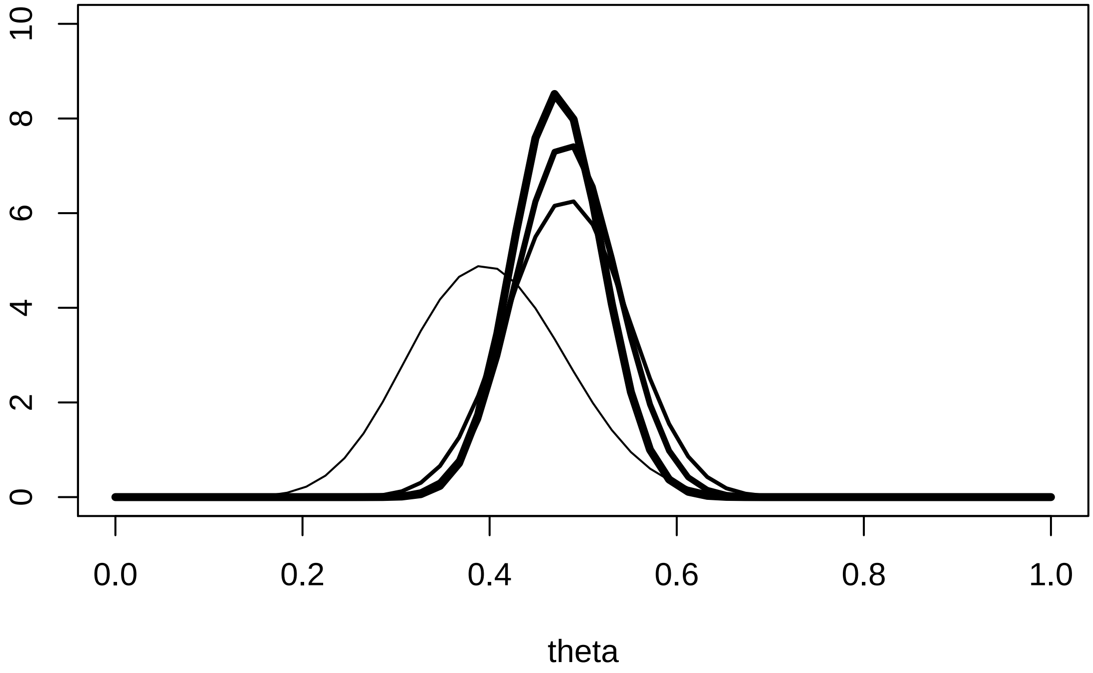
role of priors

```
prior_1 = dbeta(x = pGrid, shape1 = 5, shape2 = 5)
prior_2 = dbeta(x = pGrid, shape1 = 7, shape2 = 3)
prior_3 = dbeta(x = pGrid, shape1 = 1, shape2 = 1)
post_1 = computePost(data$ball_1, prior_1)
post_2 = computePost(data$ball_1, prior_2)
post_3 = computePost(data$ball_1, prior_3)
plot(pGrid, post_1, type="l", lwd = 3, xlab = "theta", ylab = "")
points(pGrid, prior_1, type="l", lty=2)
points(pGrid, post_2, type="l", lwd = 3, col = "red")
points(pGrid, prior_2, type="l", lty=2, col = "red")
points(pGrid, post_3, type="l", lwd = 3, col = "blue")
points(pGrid, prior_3, type="l", lty=2, col = "blue")
```



## posterior evolution

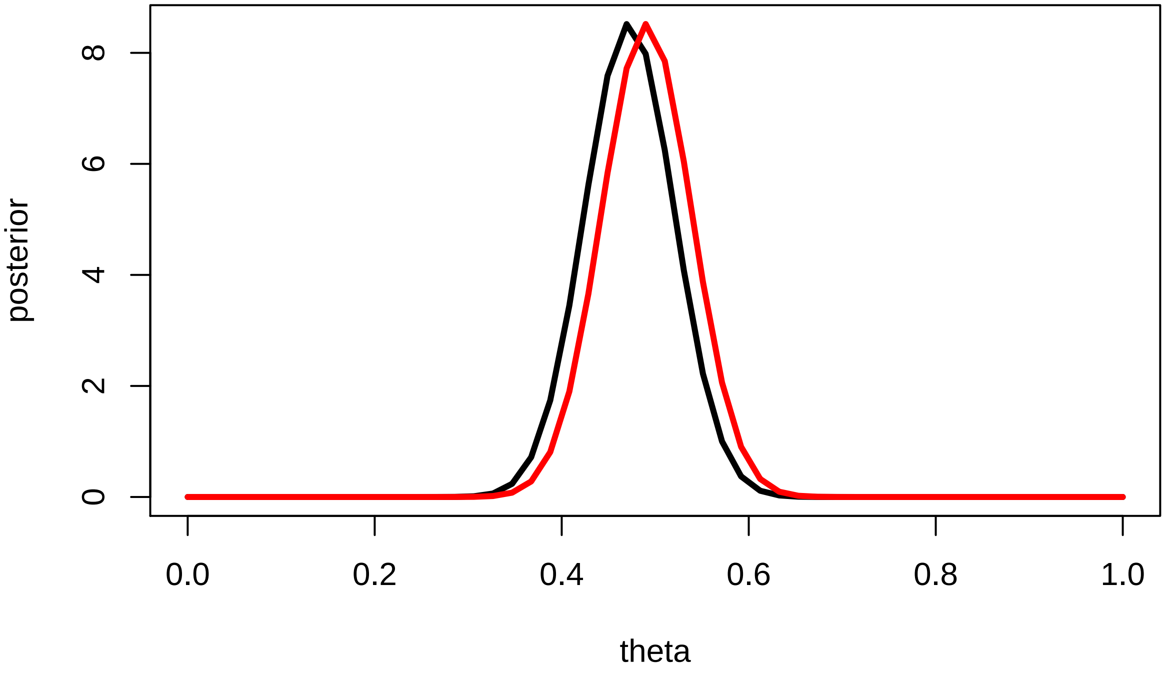
```
post = computePost(data$ball_1[1:25], prior)
plot(pGrid, post, type="l", ylim=c(0,10),
     xlab = "theta", ylab = "")
for (i in 1:3) {
  post = computePost(data$ball_1[1:(25 + i * 25)], prior)
  points(pGrid, post, type="l", lwd = 1+i)
}
```



## odd vs even trials

```r
# select even and odd trials
dataEven = data$ball_1[(1:length(data$ball_1))%%2 == 0]
dataOdd = data$ball_1[(1:length(data$ball_1))%%2 != 0]

# fit posterior separatedly and plot results
post_Even = computePost(dataEven, prior)
post_Odd = computePost(dataOdd, prior)
plot(pGrid, post_1, type="l", lwd = 3,
     xlab = "theta", ylab = "posterior")
points(pGrid, post_2, type="l", lwd = 3, col = "red")
```


posterior
theta