Lecture 2.2 example 5

preliminaries

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

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

define params

```
#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

```
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)

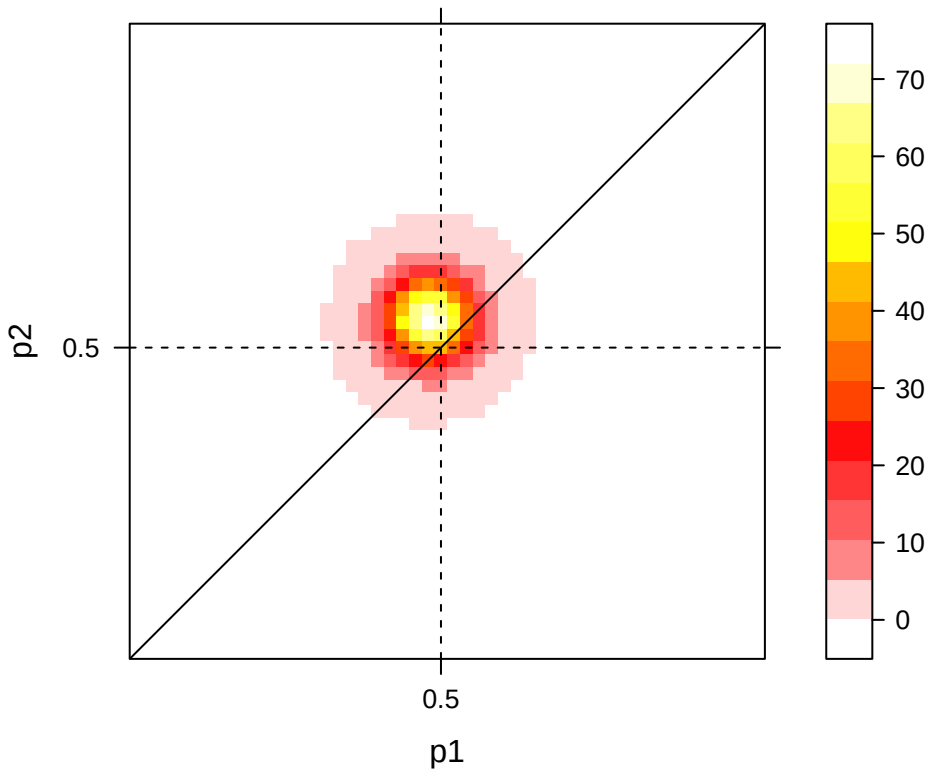
```
data = read.csv("~/Desktop/data_globe_tossing.csv")
data1 = data$ball_1
data2 = data$ball_2
nWater_1 = sum(data1)
nWater_2 = sum(data2)
n1 = length(data1)
n2 = length(data2)
```

compute posterior

```
# define prior matrix
# >> note: assume independent non-uniform priors
prior = dbeta(x = pGrid, shape1 = aPrior, shape2 = bPrior)

priorM = matrix(rep(1, nGridPoints ^ 2 ),
                nrow = nGridPoints,
                ncol = nGridPoints,
                byrow = TRUE)
for (row in 1:nGridPoints) {
  for (col in 1:nGridPoints) {
    priorM[row,col] = prior[row] * prior[col]
  }
}

# compute posterior
postM = matrix(rep(-1, nGridPoints ^ 2 ),
               nrow = nGridPoints,
               ncol = nGridPoints,
               byrow = TRUE)

for (row in 1:nGridPoints) {
  for (col in 1:nGridPoints) {
    p1 = pGrid[row]
    p2 = pGrid[col]
    postM[row,col] = dbinom(nWater_1,n1,p1) * dbinom(nWater_2,n2,p2) * priorM[row,col]
  }
}
postM = postM / ( sum(postM) * gridSize ^ 2 )
```

plot results

```
library(lattice)
new.palette=colorRampPalette(c("white","red","yellow","white"),space="rgb")
levelplot(postM, col.regions=new.palette(20),
          xlab = "p1", ylab = "p2",
          scales=list(x=list(at=c(25), labels=c(0.5)),
                      y=list(at=c(25), labels=c(0.5))),
          panel = function(...){
            panel.levelplot(...)
            panel.abline(0,1, col = "black")
            panel.abline(v=25, col = "black", lty=2)
            panel.abline(h=25, col = "black", lty=2)})
```



compute p2 > p1

```
sum = 0
for (row in 1:nGridPoints) {
  for (col in row:nGridPoints) {
    sum = sum + (postM[row,col] * gridSize ^ 2)
  }
}
paste("posterior prob p2 > p1 = ", sum)
```

```
## [1] "posterior prob p2 > p1 =  0.865185060765815"
```