AdvStDaAn, Worksheet, Week 10

Michael Lappert

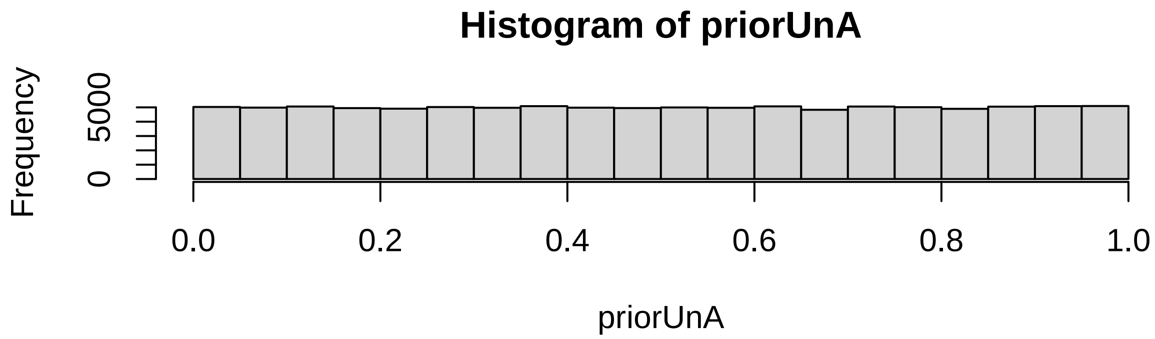
12 Mai, 2022

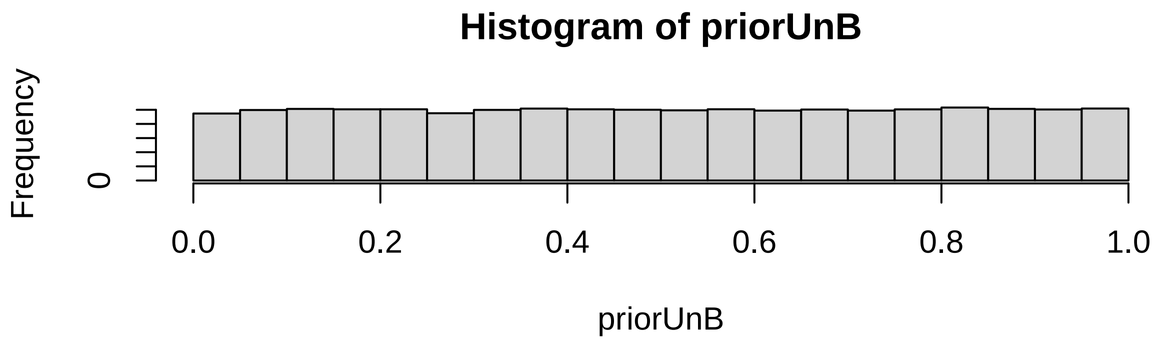
Contents

Task 1

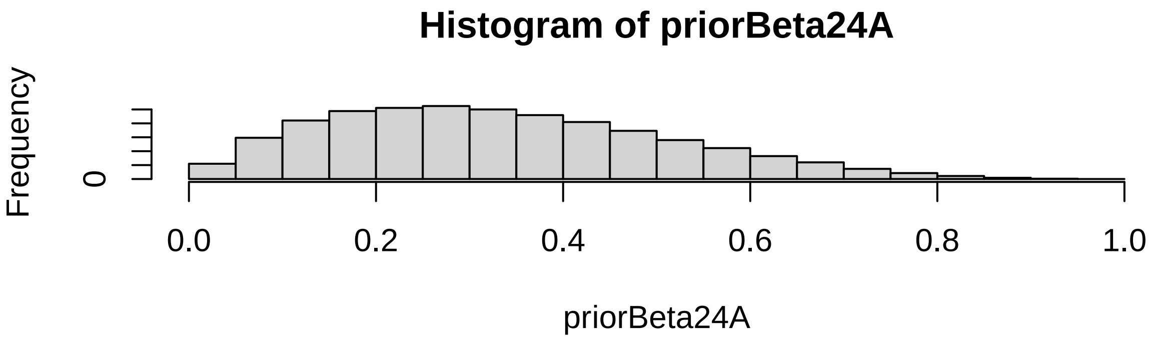
Study the influence of prior assumptions on the results of a Bayesian analysis. Remember Swedish Fish Inc.‘s two advertising alternatives: - method A: 6 out of 16 signed up and - method B: 10 out of 16 signed up. Use an uninformative prior, a Beta(2, 4) prior and the more informative Beta(3, 25) prior for the signup rates θ_A and θ_B and compare the resulting marginal posterior distributions.

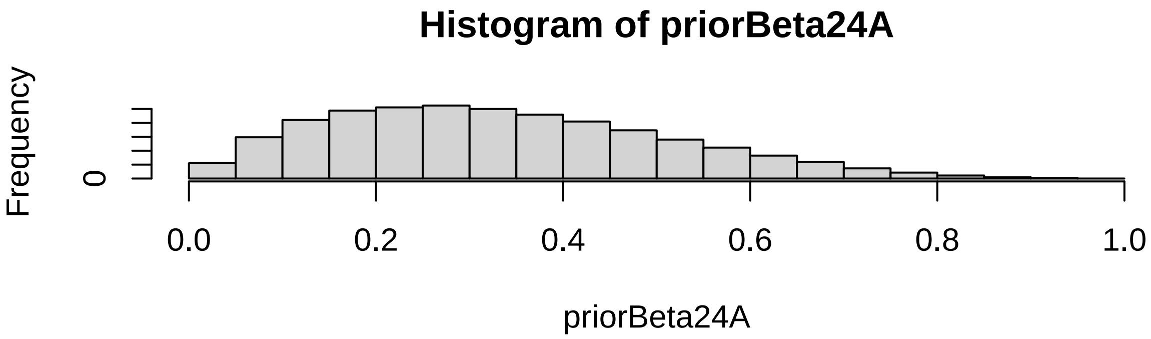
```r
# Simulate n random draws from the different priors
n = 100000

par(mfrow=c(2,1))
# uninformative prior:
priorUnA <- runif(n)
hist(priorUnA) # Eyball the prior
priorUnB <- runif(n)
hist(priorUnB) # Eyball the prior
```



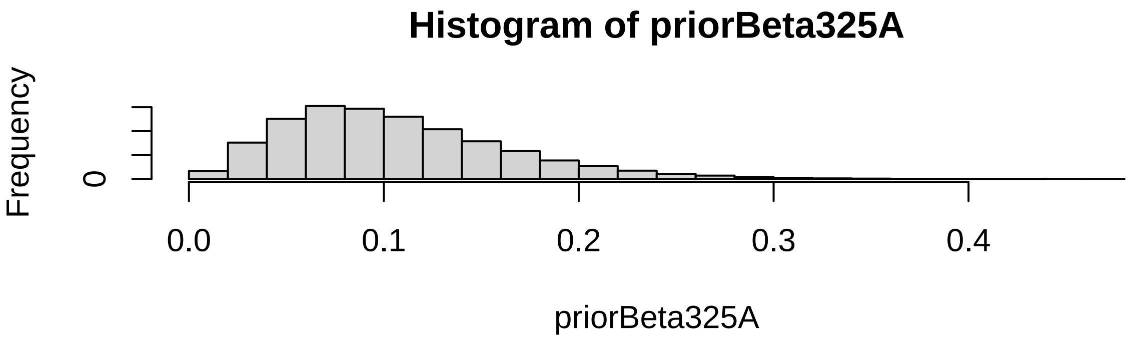


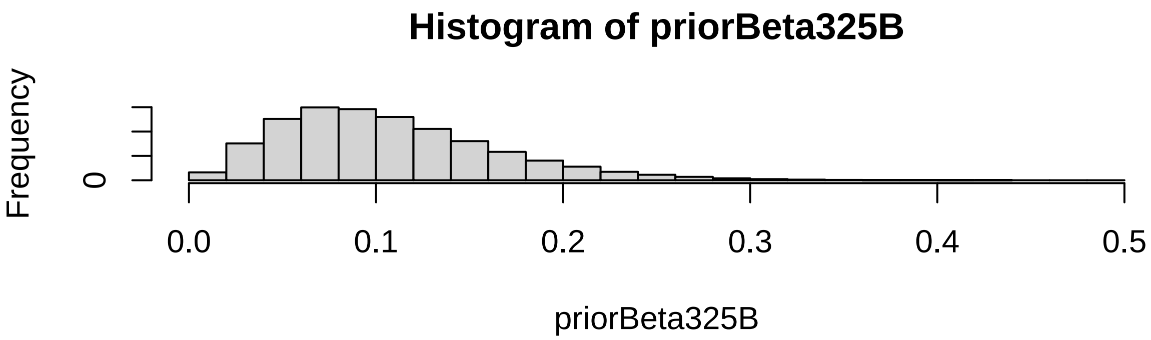
```r
# beta(2, 4) prior
priorBeta24A <- rbeta(n, 2, 4)
hist(priorBeta24A)
priorBeta24B <- rbeta(n, 2, 4)
hist(priorBeta24A)
```





```
# beta(3, 25) prior
priorBeta325A <- rbeta(n, 3, 25)
hist(priorBeta325A)
priorBeta325B <- rbeta(n, 3, 25)
hist(priorBeta325B)
```

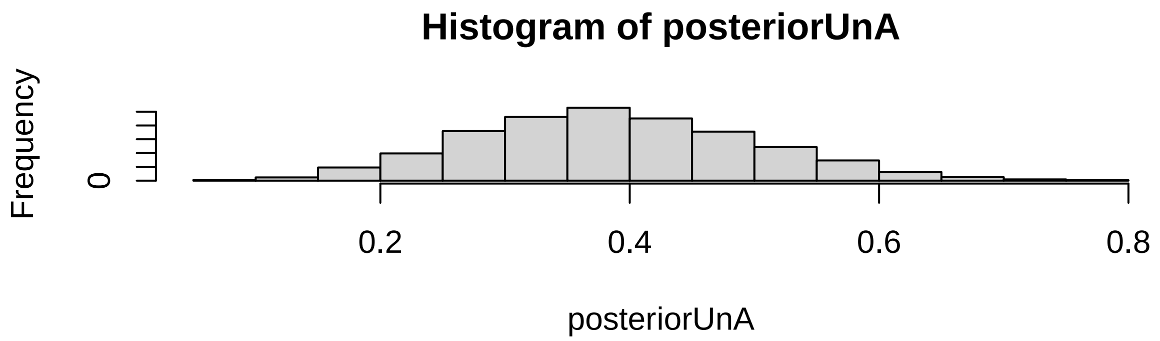


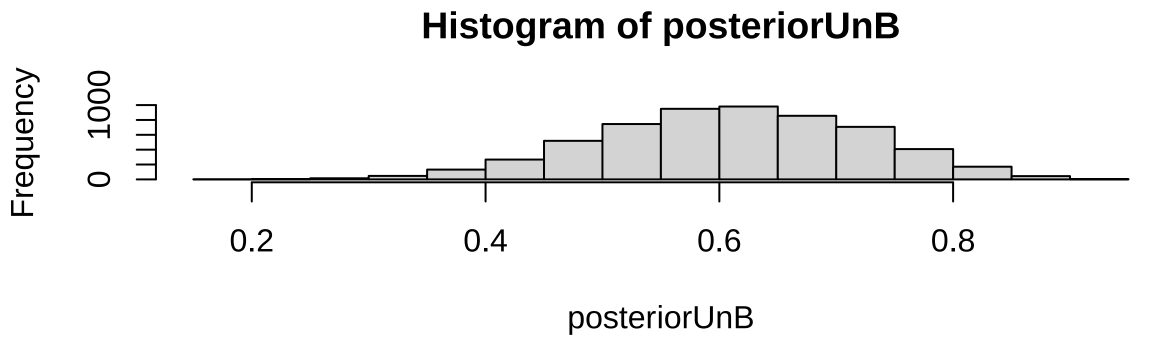


```r
# Define the generative model of uninformative prior:
generativemodelUnA <- function(theta) {
  rbinom(1, 16, theta)
}
generativemodelUnB <- function(theta) {
  rbinom(1, 16, theta)
}


# Simulate and store data from uninformative prior:
simdataUnA <- rep(NA, n)
for(i in 1:n) {
  simdataUnA[i] <- generativemodelUnA(priorUnA[i])
}
simdataUnB <- rep(NA, n)
for(i in 1:n) {
  simdataUnB[i] <- generativemodelUnA(priorUnB[i])
}

# Filter out all draws that do not match the data of the uniform prior:
posteriorUnA <- priorUnA[simdataUnA == 6]
hist(posteriorUnA)
length(posteriorUnA)
```

```
## [1] 5932
```

```
posteriorUnB <- priorUnB[simdataUnB == 10]
hist(posteriorUnB)
```





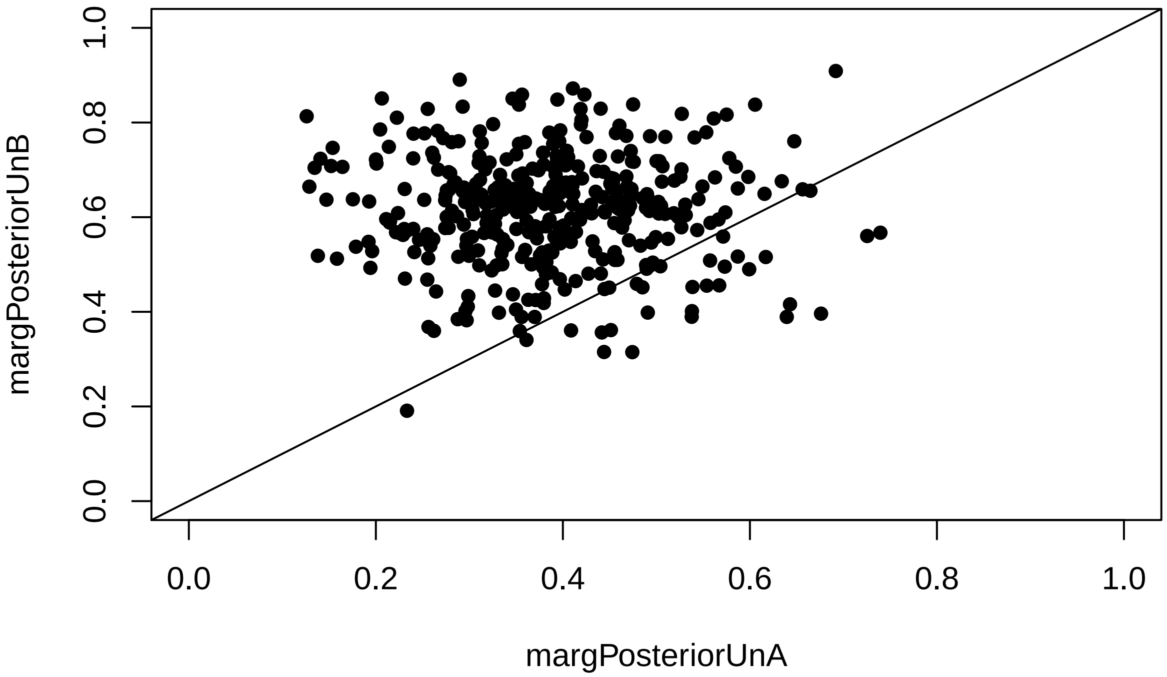
```
length(posteriorUnB)
```

```
## [1] 5851
```

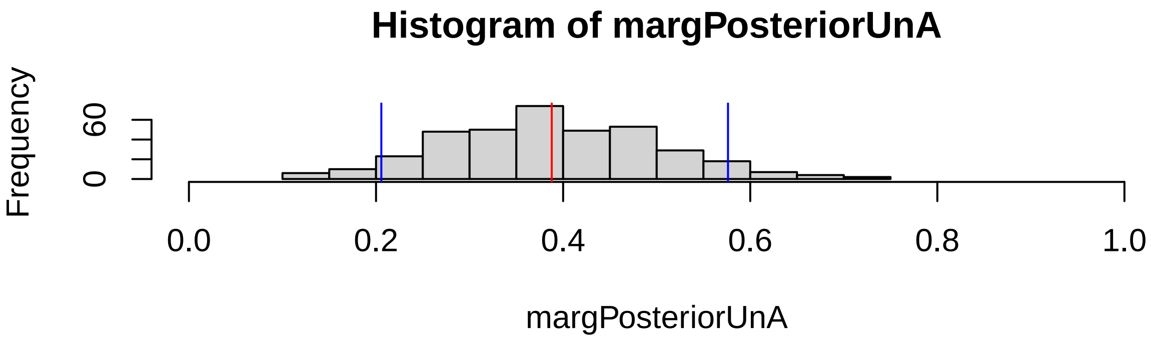
```
# Condition on observed data
ind = ( (simdataUnA==6) & (simdataUnB==10) )
ind[1:20]
```

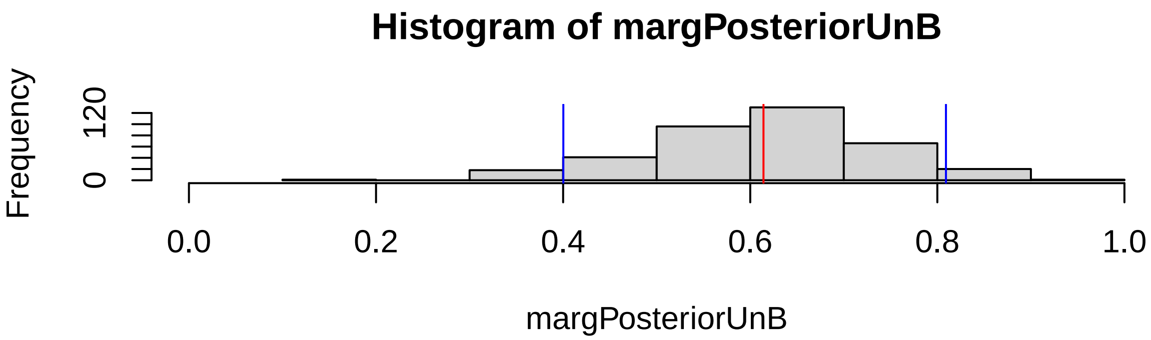
```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
margPosteriorUnA <- priorUnA[ind]
margPosteriorUnB <- priorUnB[ind]

# evaluate results
par(mfrow=c(1,1))
plot(margPosteriorUnA, margPosteriorUnB, cex=1, pch=16, xlim=c(0,1), ylim=c(0,1))
abline(0,1)
```



```
par(mfcol=c(2,1))
hist(margPosteriorUnA,xlim=0:1)
abline(v=mean(margPosteriorUnA),col="red")
abline(v=quantile(margPosteriorUnA,c(0.05,0.95)),col="blue")
hist(margPosteriorUnB,xlim=0:1)
abline(v=mean(margPosteriorUnB),col="red")
abline(v=quantile(margPosteriorUnB,c(0.05,0.95)),col="blue")
```





```
par(mfcol=c(1,1))
```

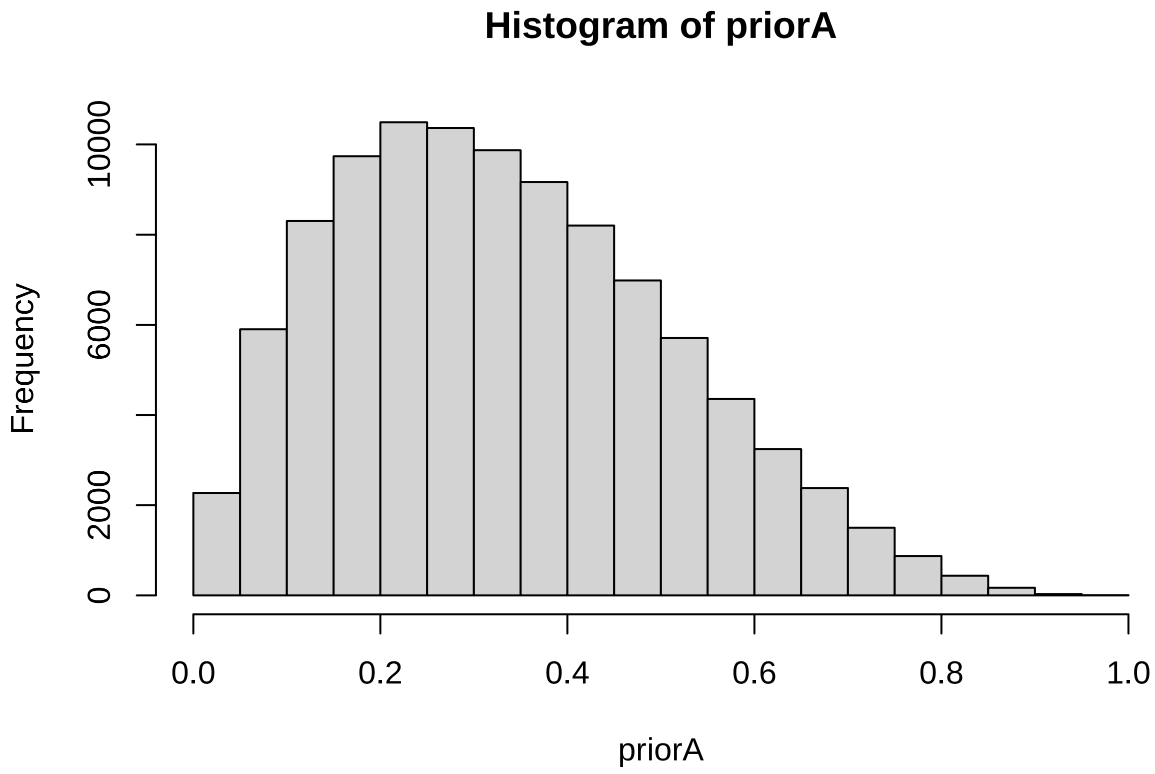
This would be the approach per prior. But in the solution was a way easier approach to do so. This one is used beneath.

Question 1, Task 1

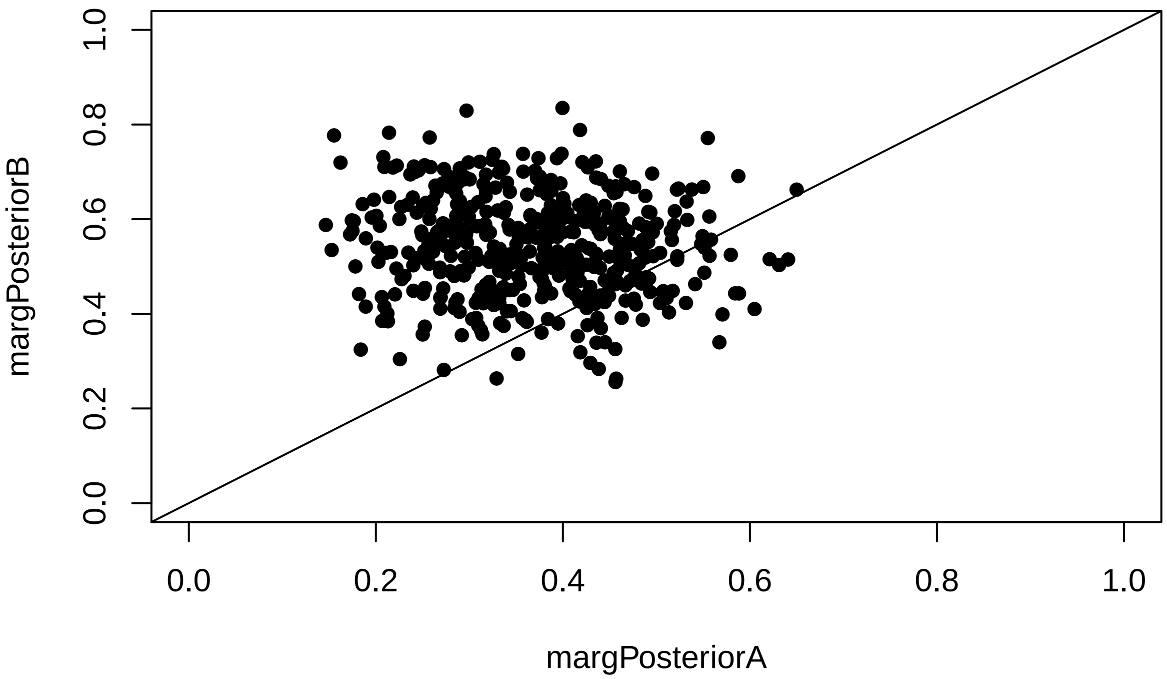
> Why are there these warning in the model data simulation process in this kind of approach? I adjusted n for the binomial sampling in rbinom to the singnups. Which is apparently wrong.

```
# number of samples
nSamples = 100000

# Data
asked    = 16 # number of asked/invited people
SingnupA =  6 # number of signups with method A
SingnupB = 10 # number of signups with method B

# Simulate prior
# Case 1
```

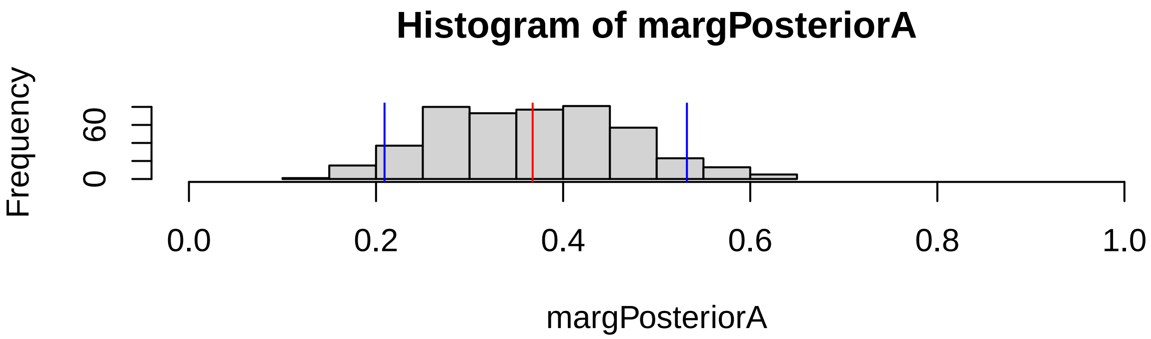
```
priorA = runif(nSamples,0,1)
priorB = runif(nSamples,0,1)
# Case 2
priorA = rbeta(nSamples,2,4)
priorB = rbeta(nSamples,2,4)
# Case 3
# priorA = rbeta(nSamples,3,25)
# priorB = rbeta(nSamples,3,25)
hist(priorA)
```

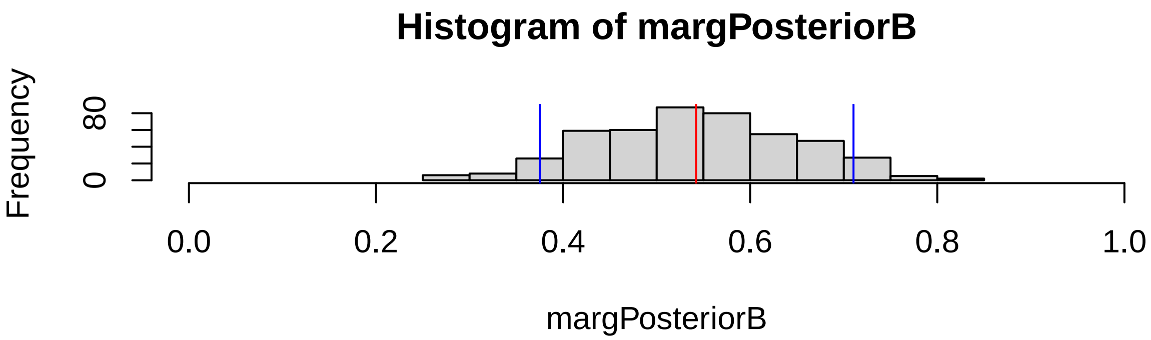


```
# Simulate generative model (likelihood)
simSingnupA = rbinom(nSamples,asked,priorA)
simSingnupB = rbinom(nSamples,asked,priorB)

# Condition on observed data
ind = ( (simSingnupA==SingnupA) & (simSingnupB==SingnupB) )

margPosteriorA = priorA[ind]
margPosteriorB = priorB[ind]

# evaluate results
par(mfrow=c(1,1))
plot(margPosteriorA,margPosteriorB,cex=1,pch=16,xlim=c(0,1),ylim=c(0,1))
abline(0,1)
```



```
par(mfcol=c(2,1))
hist(margPosteriorA, xlim=0:1)
abline(v=mean(margPosteriorA),col="red")
abline(v=quantile(margPosteriorA,c(0.05,0.95)),col="blue")
hist(margPosteriorB, xlim=0:1)
abline(v=mean(margPosteriorB),col="red")
abline(v=quantile(margPosteriorB,c(0.05,0.95)),col="blue")
```





```
par(mfcol=c(1,1))
print(mean(margPosteriorA)); mean(margPosteriorB)
```

```
## [1] 0.3674281
```

```
## [1] 0.5422162
```

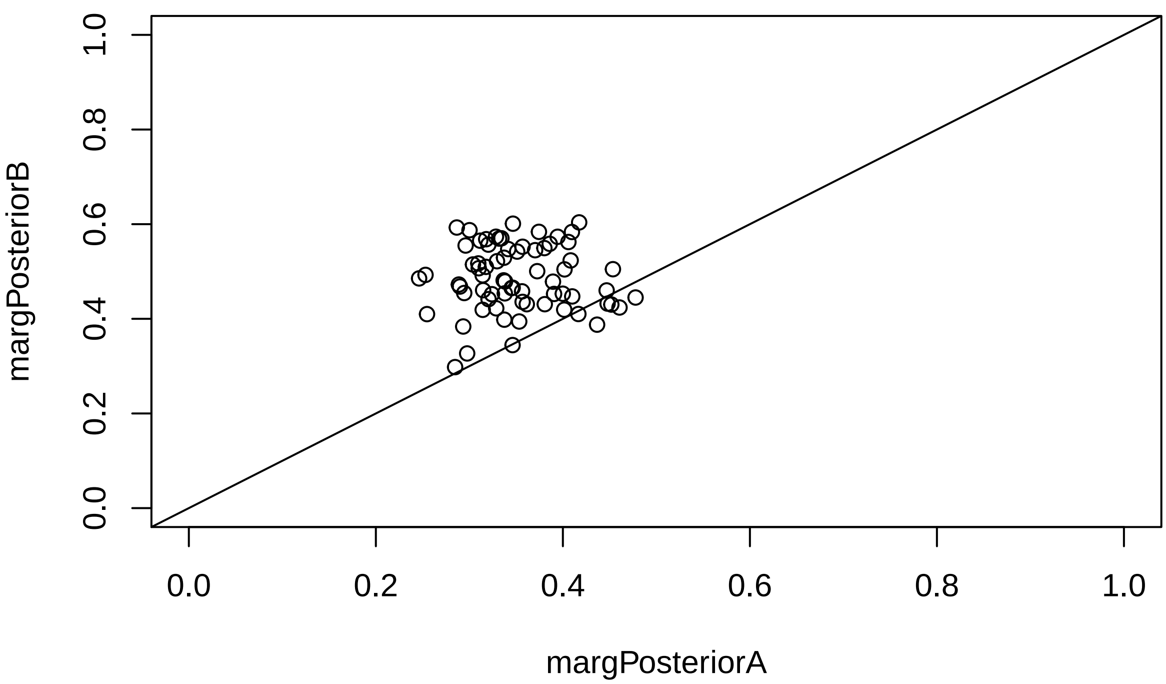
Question 2, Task 1

If i use a prior of beta(3, 25) i do not get any marginal posteriors. Why is that?

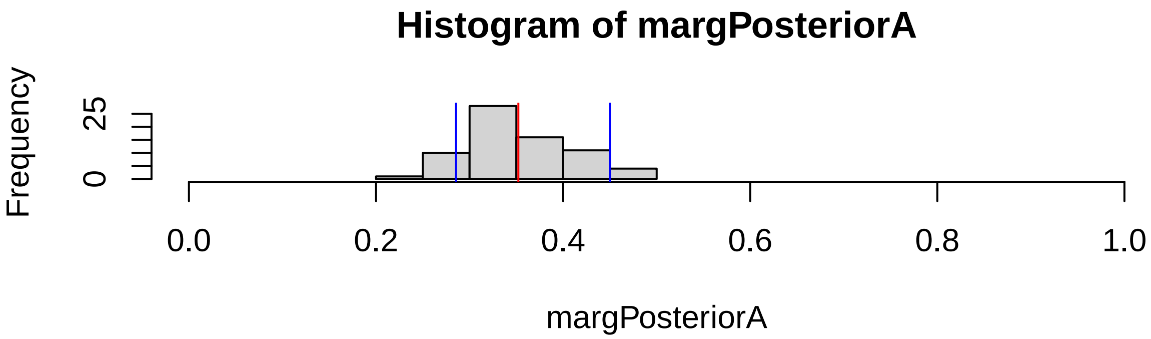
Task 2

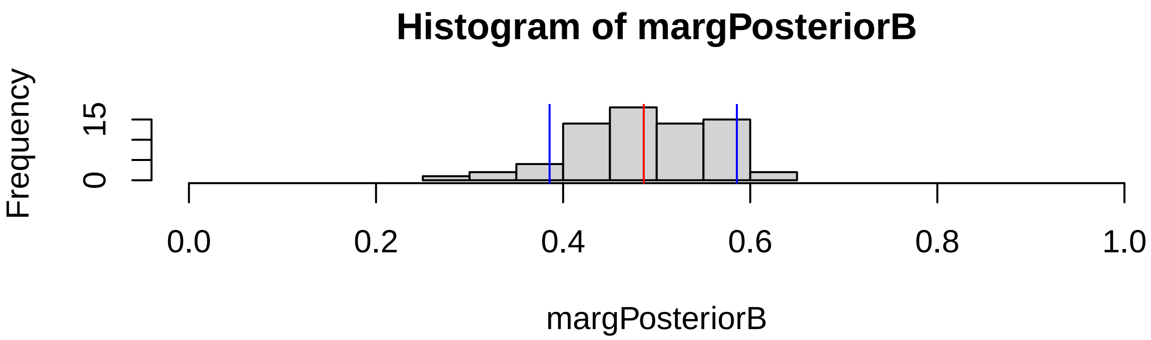
Evaluate the influence of prior assumptions and increasing data on the results of a Bayesian analysis. Consider Swedish Fish Inc. would have asked 32 (instead of 16) people, with the following results: - Method A: 12 out of 32 signed up and - method B: 20 out of 32 signed up. Again, use a uniform prior, a Beta(2, 4) prior and a Beta(3, 25) prior and compare the resulting marginal posterior distributions. Compare your results with the results from task 1.

```r
# number of samples
n <- 100000

# Signups
asked <- 32
signupA <- 12
signupB <- 20

# Simulate just the prior you wish to investigate:
## uniform prior
priorA <- runif(n, 0, 1)
priorB <- runif(n, 0, 1)

## beta(2,4) prior
priorA <- rbeta(n, 2, 4)
priorB <- rbeta(n, 2, 4)

## beta(3, 25) prior
priorA <- rbeta(n, 10, 20)
priorB <- rbeta(n, 10, 20)

# simulate generative model
simSignupA <- rbinom(n, asked, priorA)
simSignupB <- rbinom(n, asked, priorB)

# condition on observed data
ind <- ( (simSignupA == signupA) & (simSignupB == signupB) )

margPosteriorA <- priorA[ind]
margPosteriorB <- priorB[ind]

# inspect results
par(mfrow=c(1,1))
plot(margPosteriorA, margPosteriorB, xlim = c(0, 1), ylim = c(0, 1))
abline(0, 1)
```



```
par(mfrow=c(2, 1))
hist(margPosteriorA, xlim=0:1)
abline(v = mean(margPosteriorA), col = 'red')
abline(v = quantile(margPosteriorA, probs = c(0.05, 0.95)), col = 'blue')

hist(margPosteriorB, xlim = 0:1)
abline(v = mean(margPosteriorB), col = 'red')
abline(v = quantile(margPosteriorB, probs = c(0.05, 0.95)), col = 'blue')
```





```
print(paste('Mean marginal PosteriorA:', round(mean(margPosteriorA), 4)))
```

```
## [1] "Mean marginal PosteriorA: 0.3521"
```

```
print(paste('Mean marginal PosteriorB:', round(mean(margPosteriorB), 4)))
```

```
## [1] "Mean marginal PosteriorB: 0.4861"
```

The differences compared to the first task are about 1%, so it is not that much of a difference.

Task 3

John has two children. Given that at least one is a boy, what's the probability he has two boys?

Hint: Draw a tree.

$\Omega = \{BB, BG, GB, GG\}$ -> All possibilities
$A = \{BB, BG, GB\}$ -> P(A) that at least one is a boy $= 3/4 = 0.75$
$B = \{BB\}$ -> P(B) of two boys $= 1/4 = 0.25$
$B|A = \{BB\}|\{BB, BG, GB\}$ -> P(B|A) that he has two boys given at least one is a boy $= 1/3 = 0.33$

$P(B, A) = P(A) * P(B|A) = 0.75 * 0.33 = 0.25$

Question Task 3

- Why is the above solution not correct? Why is it just 1/3 and not 1/4?

Task 4

Automated email classification

Suppose an online shop wants to forward all unspecified emails to the technical support (helpdesk) or the sales department (sales).

In general **75%** of all unspecified emails have contents that belong to the sales department, the remaining emails have contents belonging to the helpdesk. You noticed, that most emails containing the string cost in the body of the email, should be forwarded to the sales department.To quanfify this, you carry out an experiment:

- 100 unspecified emails have been evaluated, that had contents belonging to the helpdesk.

10 of them contained the string cost: P(cost | heldpesk) = 0.1.

- 100 unspecified emails heve been evaluated, that had contents belonging to the sales department. 80 of them contained the string cost: P(cost | sales) = 0.8.

Calculate the probability of the naive Bayes classifier: P(sales | cost).

P(sales | costs)

$= \frac{P(costs|sales)*P(sales)}{P(costs)}$

$= \frac{P(costs|sales)*P(sales)}{P(costs|sales)*P(sales)+P(costs|helpdesk)*P(helpdesk)}$

$= \frac{0.8*0.75}{0.8*0.75+0.1*0.25}$

= 0.96

Task 5

Bluetooth earphones

You want to buy new earphones. You can decide between two almost identical products:

- Earphones A has 500 likes and 50 dislikes.
- Earphones B has 21 likes and 2 dislikes.

Based on the data and assuming similar preferences among consumers: What's the probability that you are satisfied with earphones A and earphones B, when you are commonly satisfied with about 80% of your online shopping activities (try to find an appropriate beta prior with mode=0.8 and standard deviation about 12%)?

There are two parameters to fit for the beta prior (mean and variance):

mode(X) = $\frac{a-1}{a+b-2}$
Var(X) = $\frac{ab}{(a+b+1)(a+b)^2}$

$\frac{a-1}{a+b-2} = 0.8$
a = 4b-3

Substitute in mode: $\frac{4b-3-1}{4b-3+b-2} \sim 0.12^2$

Now try out different values for b:

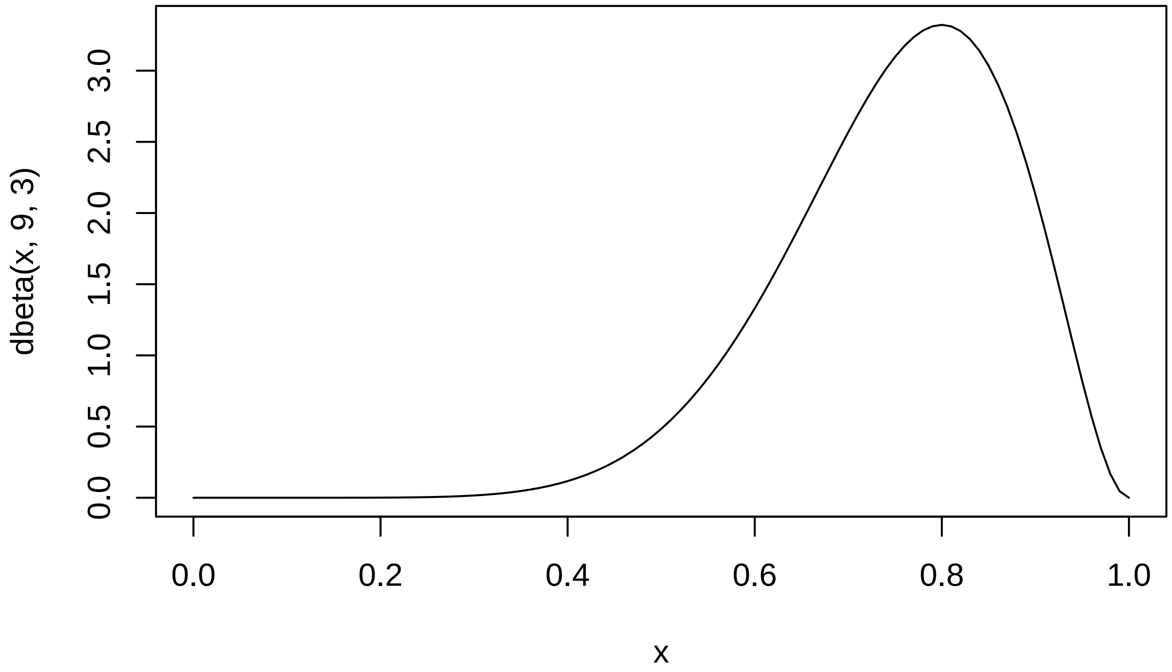
```
# Find prior by searching
b = 1:5
(4*b-3)*b/(5*b-2)/(5*b-3)^2
```

```
## [1] 0.083333333 0.025510204 0.014423077 0.009996155 0.007635645
```

```
for(i in b){
  print(paste('When b =', i))
  tmpb = (4*i-3)*i/(5*i-2)/(5*i-3)^2
  print(paste('mode =', tmpb))
  i = i+1
}
```

```
## [1] "When b = 1"
## [1] "mode = 0.0833333333333333"
## [1] "When b = 2"
## [1] "mode = 0.0255102040816327"
## [1] "When b = 3"
## [1] "mode = 0.0144230769230769"
## [1] "When b = 4"
## [1] "mode = 0.00999615532487505"
## [1] "When b = 5"
## [1] "mode = 0.00763564498742364"
```

```
# => b = 3 => a = 9
curve(dbeta(x,9,3))
```



Now solve this task via simulation:

```
likes_ABTest = function(nSamples = 100)
{
    likesAB = matrix(NA,nrow=nSamples,ncol=2)
    j = 1
    for(i in 1:10000000){
        xA = rbeta(1,9,3)   # priorA
        xB = rbeta(1,9,3)   # priorB
        nLikesA = rbinom(1,550,xA)    # simulate generative model
        nLikesB = rbinom(1,23,xB)     # simlate generative model
        if(nLikesA == 500 && nLikesB == 21) # condition on observed data
        {
           likesAB[j,] = c(xA,xB)
           if(j>=nSamples)
           {
               break;
           }
           j = j+1
        }
    }
    print(i)
    invisible(likesAB)
}
likes_ABTest()
# get 1000 samples
```

```
likes_ABTest(1000) -> result
plot(result[,1],result[,2],xlim=c(0,1),ylim=c(0,1))
abline(0,1)
sum(result[,1]>result[,2])

mean(result[,1])
mean(result[,2])

# Probability that you like earphones A
a = rbinom(1000,1,result[,1])
sum(a)/1000
#Probability that you like earphones B
b = rbinom(1000,1,result[,2])
sum(b)/1000
```

Question Task 5

> Where comes the common satisfaction with the online shopping activities in? Does this not play a role here?

Task 6

Series of coin tossing

You conducted a coin tossing experiment with 10 i.i.d. consecutive coin tosses. You counted a series of 7 (or more) equal consecutive states. Compute a 90% credible region of the form $[0, c] \cap [1 - c, 1]$ for some c in the interval (0, 0.5) for the probability of tossing a head by simulation.

Use the R function rle to compute the lengths of runs (series of equal consecutive values) in a vector.

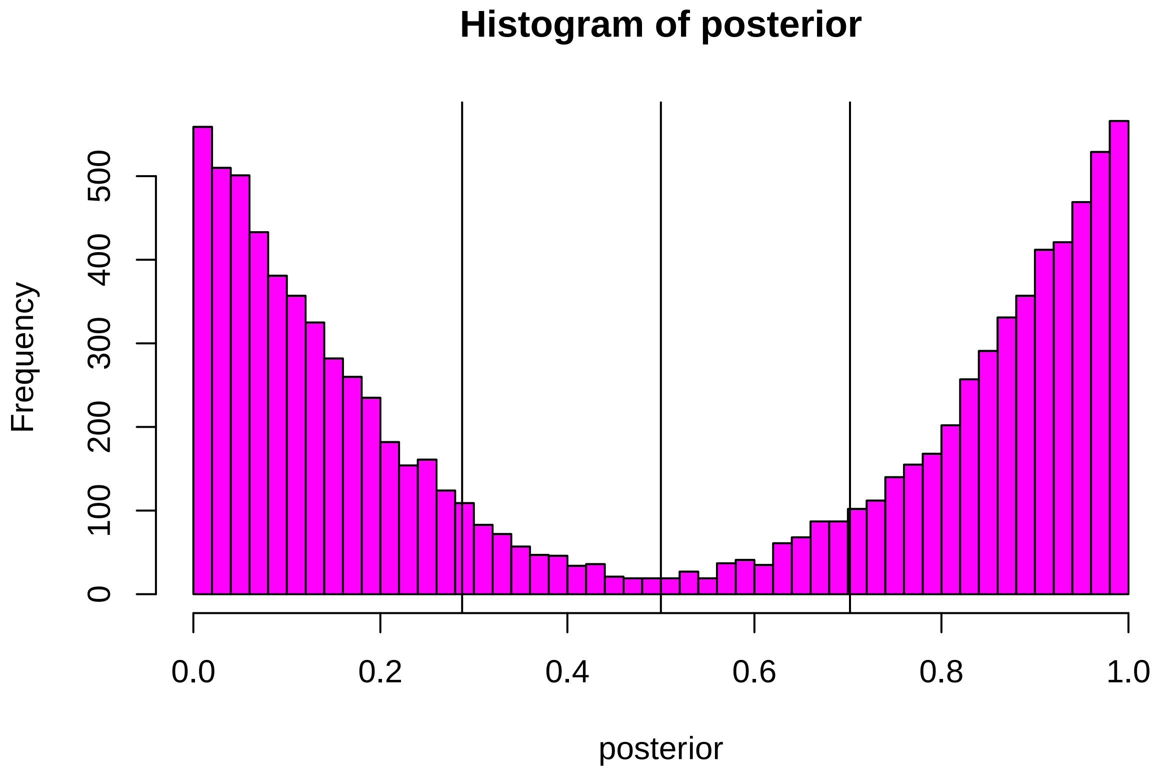
Because there is no prior information one should use a flat prior (beta(1,1))

```
seriesTossing = function(nSamples = 100)
{
    posteriorHead = rep(NA,nSamples)    # init posterior
    j = 1
    for(i in 1:10000000)
    {
        priorHead     = rbeta(1,1,1)
        # print(priorHead)
        observations  = rbinom(10,1,priorHead)
        # print(observations)
        lenghtrun = max(rle(observations)$lengths)
        # print(lenghtrun)
        if(lenghtrun >= 7)
```

```
            {
                posteriorHead[j] = priorHead
                if(j>=nSamples)
                {
                    break;
                }
                j = j+1
            }
        }
        print(i)
        invisible(posteriorHead)
}
# get 1000 samples
seriesTossing(10000) -> posterior
```

```
## [1] 30287
```

```
hist(posterior,breaks=50,col="magenta")
abline(v=.5)

abline(v=quantile(posterior,c(.45,.55)))
```



```
quantile(posterior,c(.45,.55))
```

```
##       45%       55%
## 0.2874472 0.7022240
```

```
# Answer:
# c = .29
```

Question Task 6

> Q: Why cant we just use a uniform prior here?
 > A: A beta distribution with a = b = 1 is equal to a unifrom distribution.