AdvStDaAn, Worksheet, Week 9

Michael Lappert

28 April, 2022

Contents

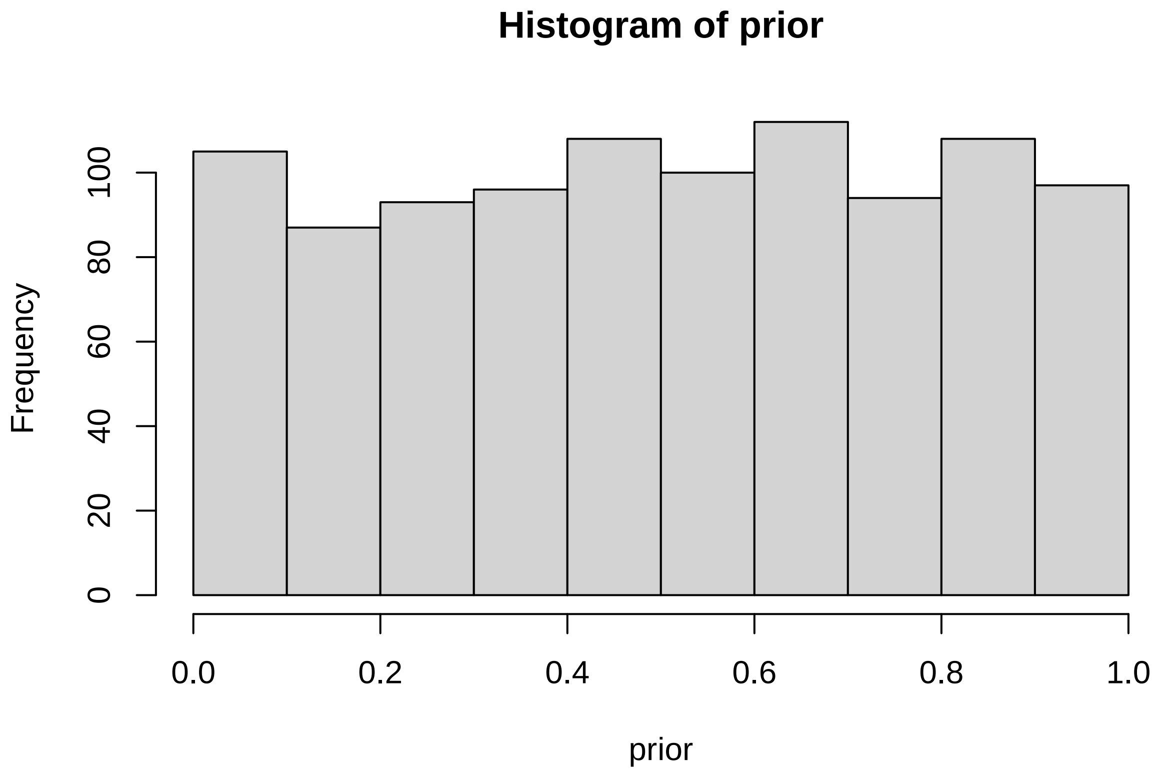
Task 1

Problem statement: Swedish Fish Incorporated is delivering fish by mail order. They are now trying to enter the Swiss market. The marketing department has done a pilot study with method A: Sending a mail with a colorful brochure that invites people to sign up for a one year salmon subscription. The marketing department sent out 16 mails. Six out of 16 recipients signed up.

Build a Bayesian model that answers the question: What would the rate of sign-up be if method A was used on a larger number of people? Consider you send a mail to 20 people and 8 recipients signed up. Assume that you know nothing about the sign-up rate apriori, i.e. choose a unifom prior.

Simulate n random draws from the prior:

```r
n = 1000
prior = runif(n) # Here we sample n draws from the prior
par(mfrow=c(1,1))
hist(prior) # It’s always good to eyeball the prior to make sure it looks ok
```



Define the generative model:

```
generativemodel = function(theta) {
  rbinom(1, 20, theta)
}
```

Question generative model

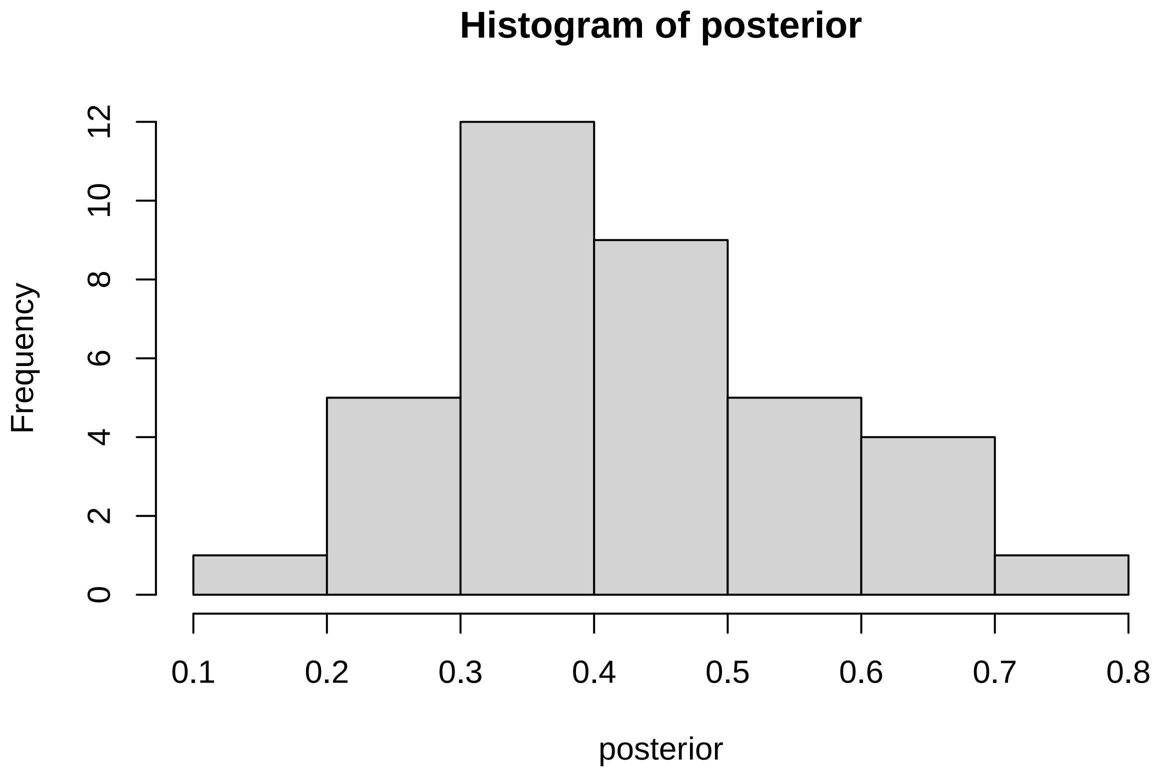
Why did you choose n = 1 and size = 20 in rbinom()? - Answer: n = 1 because it is used in the for loop afterwards (so 1 simulation per n) and 20 because the email was sent to 20 people. ***

Simulate and store data using parameters from the prior and the generative model:

```
simdata = rep(NA, n)
for(i in 1:n) {
  simdata[i] = generativemodel( prior[i] )
}
```

Filter out all draws that do not match the data:

```
posterior = prior[simdata == 8]
hist(posterior)
```



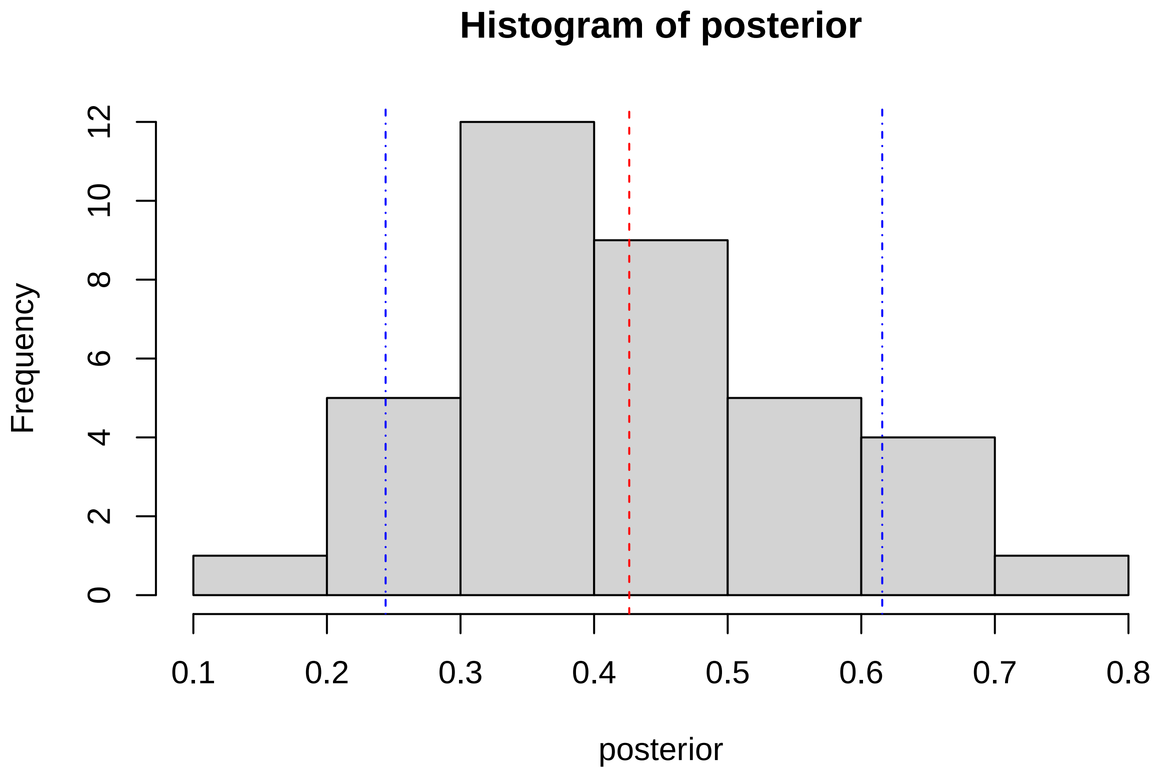
```
# Are there enought draws left after the filtering?
length(posterior)
```

```
## [1] 37
```

There are no rules about the minimum draws left after filtering, but you probably want to aim for >1000 draws.

Now, summarize the posterior (posterior mean, 90% credible interval):

```
hist(posterior)
abline(v = mean(posterior), col = "red", lty = 2)
abline(v = quantile(posterior,c(.05,.95)), col = 'blue', lty = 4)
```



Task 2

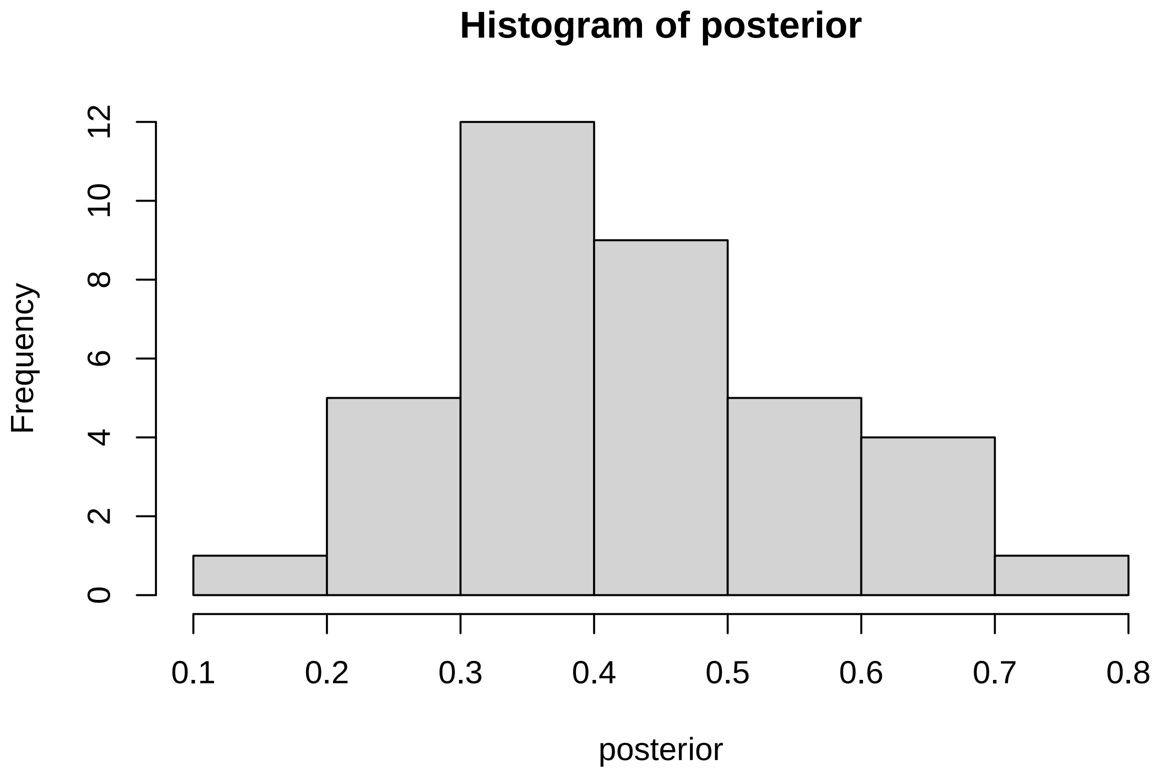
Problem statement: What's the probability that method A is better than telemarketing?

Marketing just told us that the rate of sign-up would be 20% if salmon subscribers were snared by a telemarketing campaign instead (to us it's very unclear where marketing got this very precise number from). So given the model and the data that we developed in the last question, what's the probability that method A has a higher rate of sign-up than telemarketing?

Hint 1: If you have a vector of samples representing a probability distribution, which you should have from the last question, calculating the amount of probability above a certain value is done by simply counting the number of samples above that value and dividing by the total number of samples.

Hint 2: The answer to this question is a one-liner.

```
hist(posterior)
```



```
length(posterior)
```

```
## [1] 37
```

```
length(posterior[posterior > 0.2])
```

```
## [1] 36
```

```
sum(posterior[posterior > 0.2]) / length(posterior)
```

```
## [1] 0.4211291
```

So the probability that method A is better than telemarketing is 42.1%.

Task 3

If method A was used on 100 people what would be the number of sign-ups?

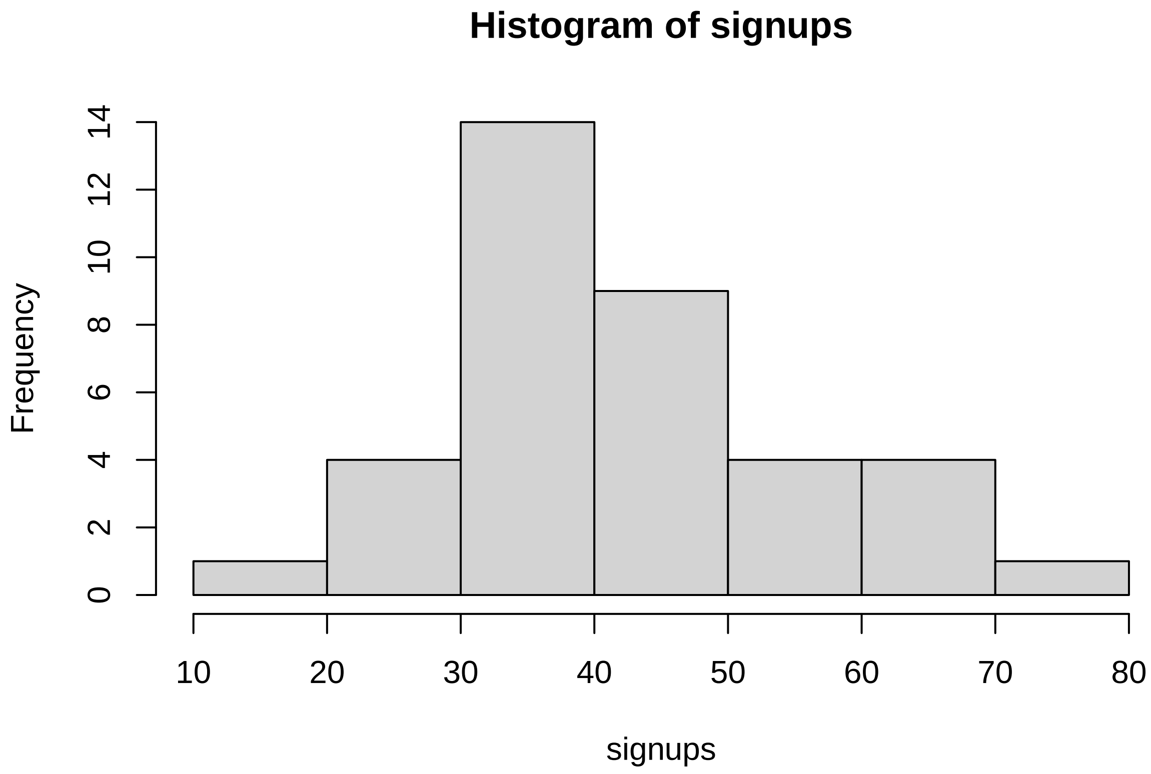
Hint 1: The answer is again not a single number but a distribution over probable number of sign-ups.

Hint 2: As before, the binomial distribution is a good candidate for how many people that sign up out of the 100 possible.

Hint 3: Make sure you don't throw away uncertainty, for example by using a summary of the posterior distribution calculated in the first question. Use the full original posterior sample!

Hint 4: The general pattern when using the posterior distribution is to go through the sampled values one-by-one, and perform a transformation (say, plugging in the value in a binomial distribution), and collect the new values in a vector.

```
signups = rbinom(length(posterior), 100, posterior)
hist(signups)
```



Task 4

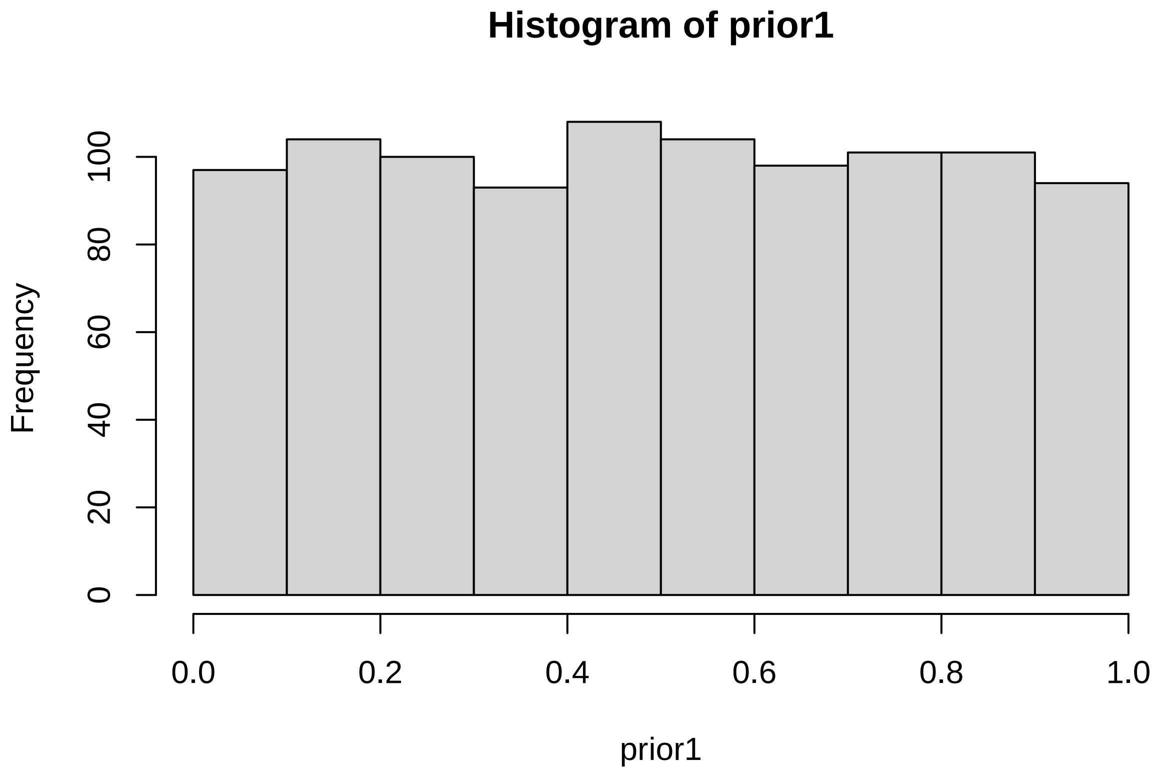
Consider the case, you sent out flyers to 160 randomly selected people and 60 out of 160 recipients signed up.

Calculate the posterior distribution of the signup rate θ and compare it with our original posterior distribution (6 out of 16 signups). Plot the two resulting posterior distributions, based on a uniform prior. Calculate and compare the posteriori means and the equal-tailed credible intervals.

What't the maximum likelihood estimate for the signup rate θ in both cases?

First calculating the original data (6 signups out of 16)

```
# Simulate n1 random draws from the prior:
n1 = 1000
prior1 = runif(n1)
hist(prior1) # Eyball the prior
```

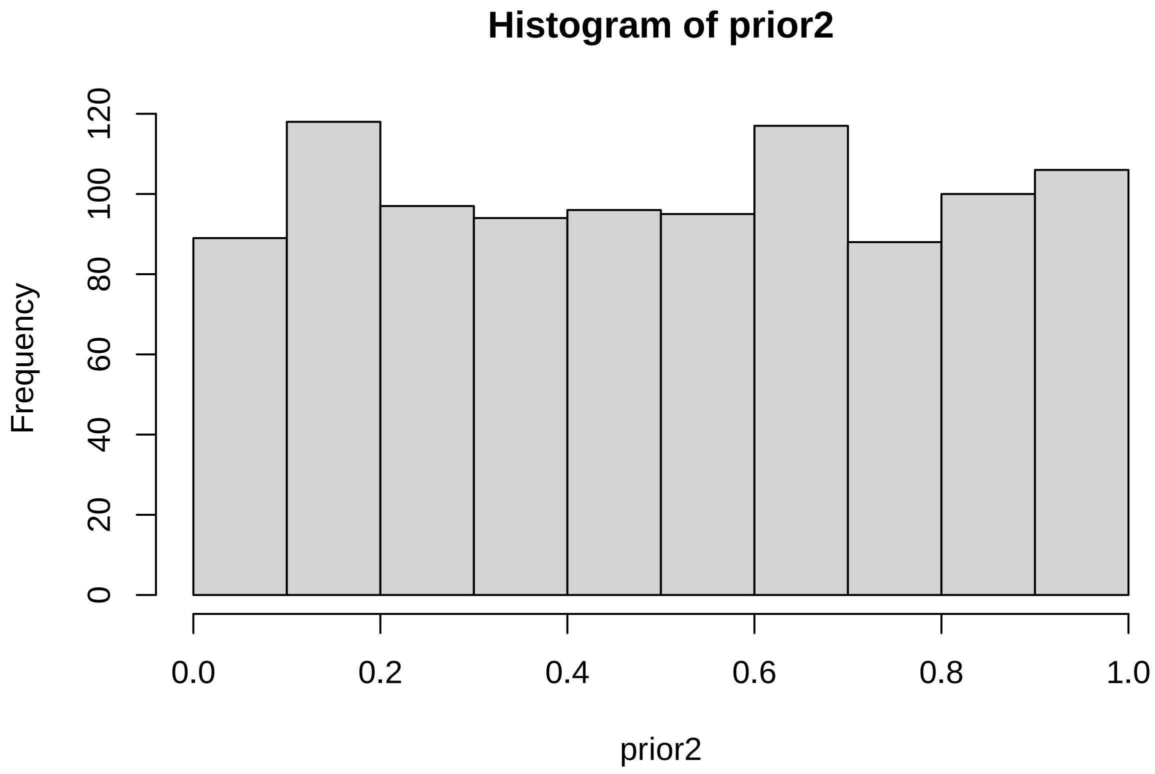


```
# Define the generative model:
generativemodel1 <- function(theta) {
  rbinom(1, 16, theta)
}

# Simulate and store data using parameters from the prior and the generative model:
simdata1 <- rep(NA, n1)
for(i in 1:n1) {
  simdata1[i] <- generativemodel1(prior1[i])
}

# Filter out all draws that do not match the data:
posterior1 <- prior1[simdata1 == 6]
length(posterior1)
```

```
## [1] 54
```

Second calculate the second case with 60 signups out of 160:

```
# Simulate n2 random draws from the prior:
n2 = 1000
prior2 = runif(n2)
hist(prior2) # Eyball the prior
```

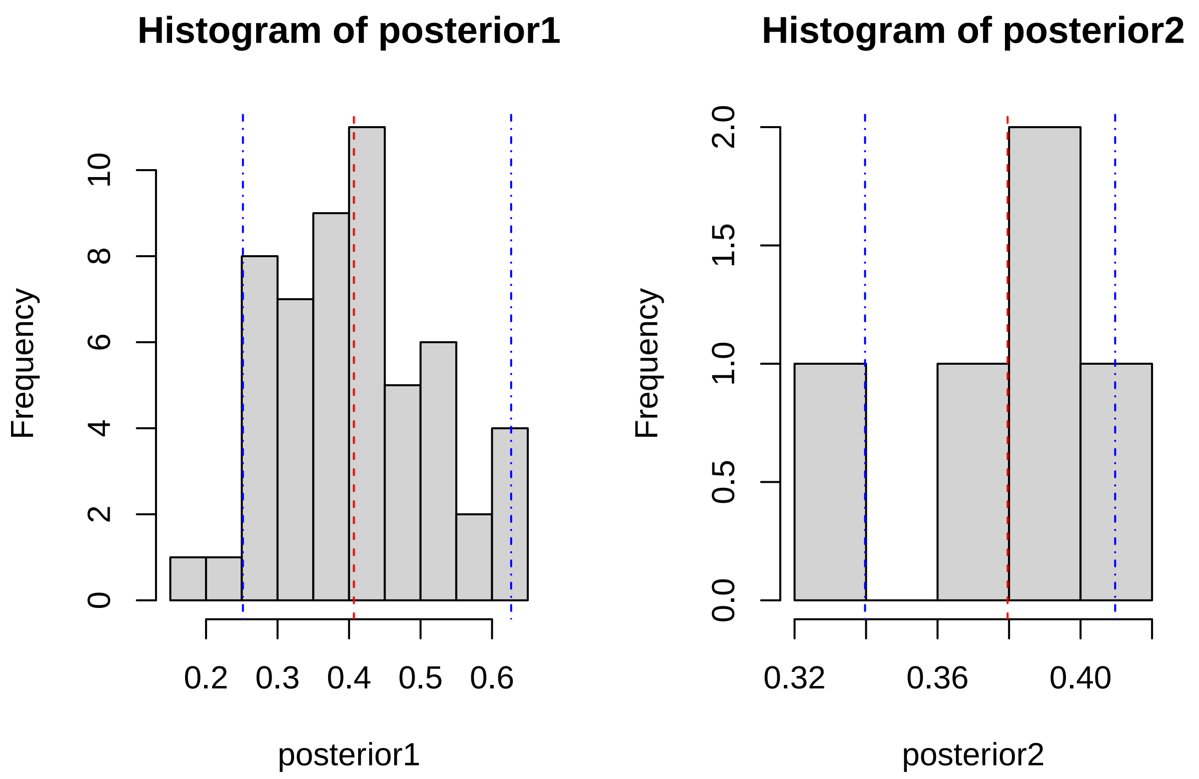


```
# Define the generative model:
generativemodel2 <- function(theta) {
  rbinom(1, 160, theta)
}

# Simulate and store data using parameters from the prior and the generative model:
simdata2 <- rep(NA, n2)
for(i in 1:n2) {
  simdata2[i] <- generativemodel2(prior2[i])
}

# Filter out all draws that do not match the data:
posterior2 <- prior2[simdata2 == 60]
length(posterior2)
```

```
## [1] 5
```

Then comparing the resulting posteriors:

```
par(mfrow=c(1,2))
# Hisogram posterior1
hist(posterior1)
abline(v = mean(posterior1), col = "red", lty = 2)
abline(v = quantile(posterior1,c(.05,.95)), col = 'blue', lty = 4)

# Hisogram posterior2
hist(posterior2)
abline(v = mean(posterior2), col = "red", lty = 2)
abline(v = quantile(posterior2,c(.05,.95)), col = 'blue', lty = 4)
```



```
# Means of posteriors
mean(posterior1)
```

```
## [1] 0.4067053
```

```
mean(posterior2)
```

```
## [1] 0.3795913
```

```
# Equal-tailed credible intervals
quantile(posterior1,c(.05,.95))
```

```
##         5%       95% 
## 0.2515754 0.6267057
```

```
quantile(posterior2,c(.05,.95))
```

```
##         5%       95% 
## 0.3397172 0.4096803
```

The higher we would set n1/n2 the closer the means and quantiles would get. Because they are acually from the same distribution they have also the same maximum likelihood estmate -> 6/16 = 60/160 = 0.375

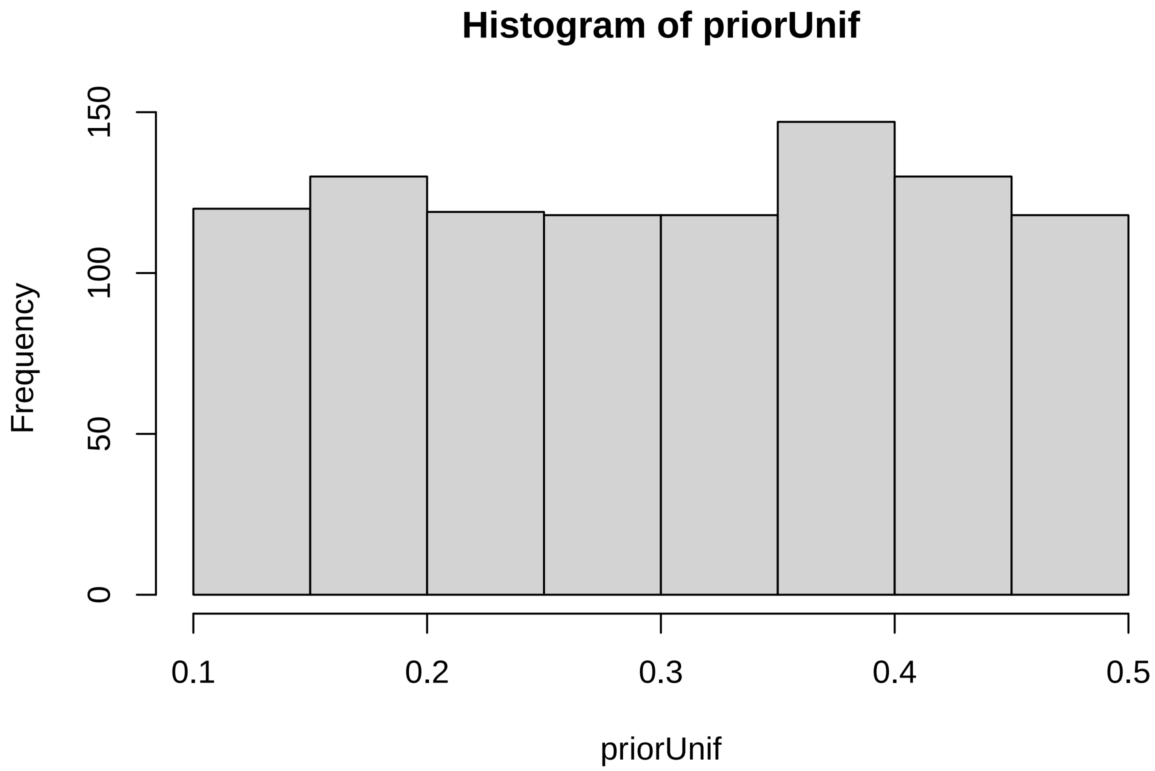
Task 5

Design a prior for the signup rate, that is uniformly distributed between [0.1, 0.5].

Design a prior for the signup rate from the beta family of distributions with expectation 0.2 and standard deviation 0.1.

Calculate a posterior sample for the signup rate with each of the above priors and observed 6 out of 16 signups. Compare the resulting posterior distribution with a Beta(2, 20) prior on the signup rate (see lecture).

```
n = 1000

# Prior uniformly distributed between 0.1 and 0.5
priorUnif = runif(n, min = 0.1, 0.5)
hist(priorUnif)
```



Prior from the beta family with expectation 0.2 and sd = 0.1 (standard deviation is the square root of the variance)

For this we have to solve the equation system of two equations for (expected value and variance of the beta distribution)

$\frac{a}{a+b} = 0.1$ and $\frac{a*b}{(a+b+1)(a+b)^2} = 0.1^2$

so

$b = 4a$

and plotting this into the variance function results in

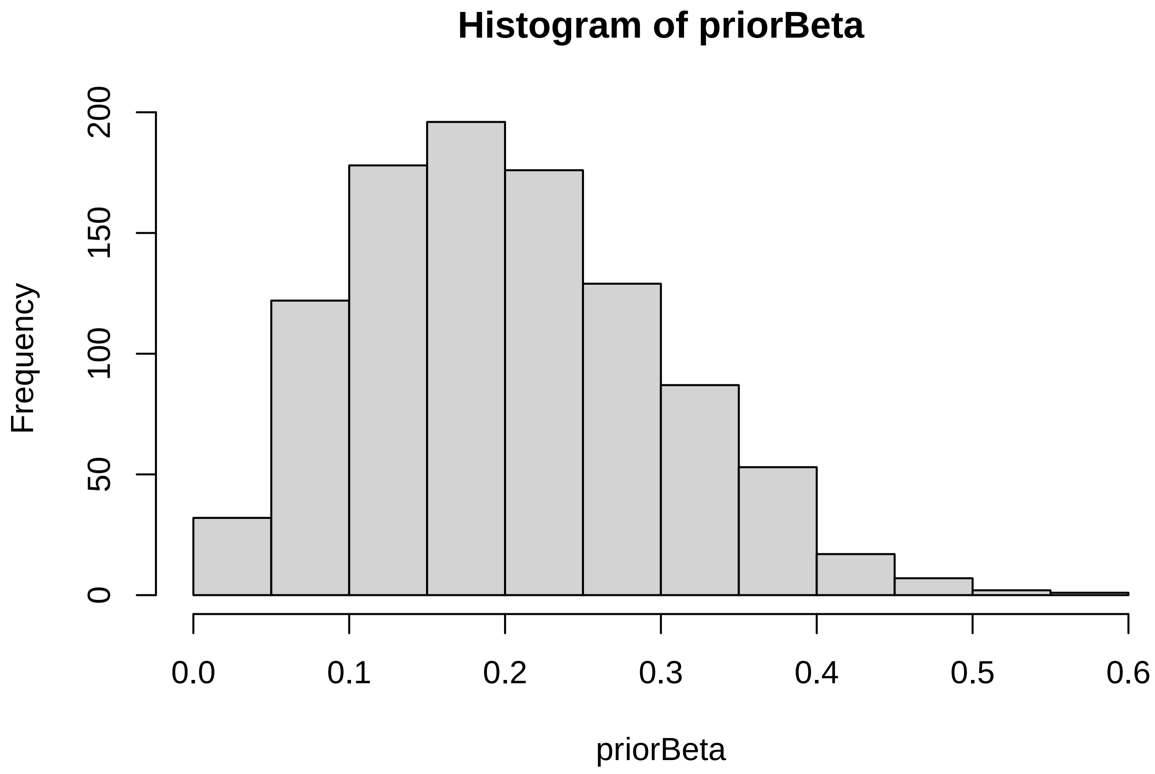
$\frac{4a^2}{(5a+2)(5a)^2} = 0.1^2$ $4a^2 = 0.1^2 * (5a+1)(5a)^2$ $400a^2 = (5a+1)(5a)^2$ $a = 3$

plugging this in the expected value function then gives

$b = 12$

We can this now use to compute the beta prior:

```r
a = 3
b = 12
priorBeta = rbeta(n, a, b)
hist(priorBeta)
```



from here on we proceed as seen before for both priors **Uniform prior**

```
# Define the generative model:
generativemodelUnif <- function(theta) {
  rbinom(1, 16, theta)
}

# Simulate and store data using parameters from the prior and the generative model:
simdataUnif <- rep(NA, n)
for(i in 1:n) {
  simdataUnif[i] <- generativemodelUnif(priorUnif[i])
}

# Filter out all draws that do not match the data:
posteriorUnif <- priorUnif[simdataUnif == 6]
length(posteriorUnif)
```

```
## [1] 127
```

Beta prior

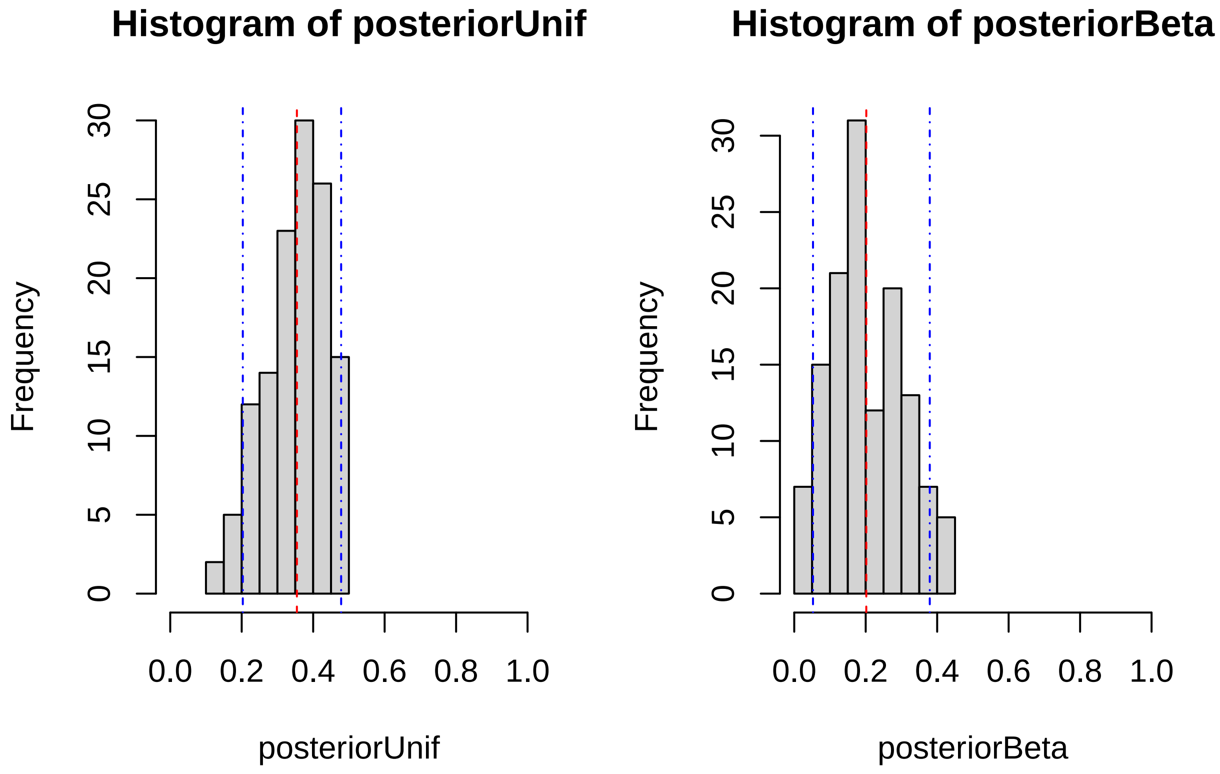
```
# Define the generative model:
generativemodelBeta <- function(theta) {
  rbinom(1, 16, theta)
}
```

```
# Simulate and store data using parameters from the prior and the generative model:
simdataBeta <- rep(NA, n)
for(i in 1:n) {
  simdataBeta[i] <- generativemodelBeta(priorUnif[i])
}

# Filter out all draws that do not match the data:
posteriorBeta <- priorBeta[simdataBeta == 6]
length(posteriorBeta)
```

```
## [1] 131
```

Looking at the results:

```
par(mfrow=c(1,2))
# Hisogram posteriorUnif
hist(posteriorUnif, xlim = c(0,1))
abline(v = mean(posteriorUnif), col = "red", lty = 2)
abline(v = quantile(posteriorUnif,c(.05,.95)), col = 'blue', lty = 4)

# Hisogram posteriorBeta
hist(posteriorBeta, xlim = c(0,1))
abline(v = mean(posteriorBeta), col = "red", lty = 2)
abline(v = quantile(posteriorBeta,c(.05,.95)), col = 'blue', lty = 4)
```



```
# Means of posteriors
mean(posteriorUnif)
```

```
## [1] 0.3544529
```

```
mean(posteriorBeta)
```

```
## [1] 0.201618
```

```
# Equal-tailed credible intervals
quantile(posteriorUnif,c(.05,.95))
```

```
##        5%       95%
## 0.2031527 0.4782720
```

```
quantile(posteriorBeta,c(.05,.95))
```

```
##         5%        95%
## 0.05254497 0.37930694
```