This is obtained from the work of Rasmus Bååth: http://sumsar.net/files/posts/2017-bayesian-tutorial-exercises/modeling_exercise2.html

Exercise 2: Bayesian A/B testing for Swedish Fish Incorporated with Stan

Swedish Fish Incorporated is the largest Swedish company delivering fish by mail order, but you probably already knew that. The marketing department have done a pilot study and tried two different marketing methods:

A: Sending a mail with a colorful brochure that invites people to sign up for a one year salmon subscription.

B: Sending a colorful brochure that invites people to sign up for a one year salmon subscription *and that includes a free salmon.*

The marketing department sent out 16 mails of type A and 16 mails of type B. Six Danes that received a mail of type A signed up for one year of salmon, and ten Danes that received a mail of type B signed up!

The marketing department now wants to know, which method should we use, A or B?

At the bottom of this document you'll find a solution. But try yourself first!

Question I: Build a Bayesian model in Stan that answers the question: What is the probability that method B is better than method A?

Hint 1: As part of you generative model you'll want to use the binomial distribution, which you can use in Stan like this:

```
s ~ binomial(size, rate);
```

This should be read as: "The number of successes `s` is distributed as a binomial distribution with `size` trials, where the rate of success is `rate`.

Hint 2: A commonly used prior for the unknown probability of success in a binomial distribution is a uniform distribution from 0 to 1. You can use this distribution in Stan like this:

```
rate ~  uniform(0, 1);
```

Hint 3: Here is a code scaffold that you can build upon which estimates the rate for one group. For an A/B test you would have to extend this model to include two groups. Comparing the rates of the two groups can be done either in the `generated quantities` block in Stan or by post-processing the samples from the stan model.

```
library(rstan)

# The Stan model as a string.
model_string <- "
# Here we define the data we are going to pass into the model
data {
  int n; # Number of trials
  int s;  # Number of successes
}
```

```
# Here we define what 'unknowns' aka parameters we have.
parameters {
  real<lower=0, upper=1> rate;
}

# The generative model
model {
  rate ~ uniform(0, 1);
  s ~ binomial(n, rate);
}

# In the generated quantiles block you can calculate 'derivatives' of
# the parameters. Here is a silly example calculating the square of the
# rate. Variables have to be defined before they are assigned to.
generated quantities {
  real rate_squared;
  rate_squared = rate^2;
}
"

data_list <- list(n = 16, s = 6)

# Compiling and producing posterior samples from the model.
stan_samples <- stan(model_code = model_string, data = data_list)

# Plotting and summarizing the posterior distribution
stan_samples
traceplot(stan_samples)
plot(stan_samples)

# Export the samples to a data.frame for easier handling.
posterior <- as.data.frame(stan_samples)

# Now we could, for example, calculate the probability that the rate is higher
# than, say, 20%
sum(posterior$rate > 0.2) / length(posterior$rate )
```

Question II: Change the model so that it uses a more informative prior. What is now the probability that method B is better than method A?

The marketing department are starting to believe that it was a fluke that such a large proportion of the Danes signed up. In all other European markets the proportion that signs up for a year of salmon is around 5% to 15%, even when given a free salmon. Use this information and make the priors in your model more informative.

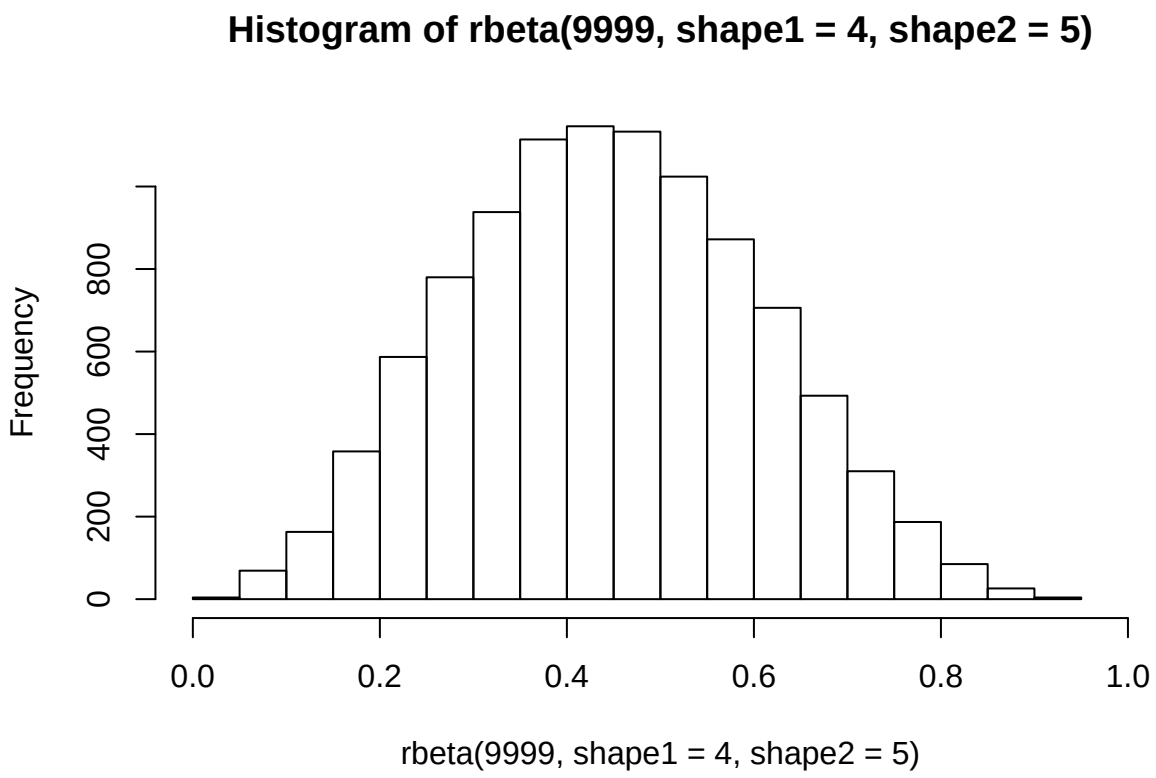
Hint 1: This can be done in a million ways and there isn't any "right" answer to this question. Just do something quick 'n dirty.

Hint 2: It would however be cool if you used a prior that wasn't uniform. A good distribution, when crafting priors with support over [0, 1], is the beta distribution. You can sample from a beta distribution using the `rbeta(1, shape1, shape2)` function in R, where `shape1` and `shape2` surprisingly defines the shape of the distribution.

Hint 3: An easy way to plot a beta distribution (and to explore what the `shape` parameters really do) is to

run the following and to play around with `shape1` and `shape2`:

```
hist(rbeta(9999, shape1 = 4, shape2 = 5), xlim=c(0, 1))
```



Question III: So what should we do? Make a simple decision analysis.

The economy department gives you the following information:

- A mail of type A costs 30 kr to send out.
- A mail of type B costs 300 kr to send out (due to the cost of the free salmon).
- A salmon subscription brings in 1000 kr in revenue.

Which method, A or B, is most likely to make Swedish Fish Incorporated the most money?

Hint 1: This should require no changes to your model. It should suffice to post process the samples.

Hint 2: If `rateA` is the probability that someone will sign up when receiving a type A mail then the expected profit is `1000 * prob_a - 30`

Hint 3: The cool thing with working with *samples* from posterior distributions is that if we calculate a 'derivative' per each row of the sample then the resulting 'derivative distribution' will be correct! That is the derived quantity `rate_squared` in the model scaffold given in question I could equally well be calulated after having run the model, like this:

```
posterior <- as.data.frame(stan_samples)
rate_squared <- posterior$rate^2
```

Solutions (but this can be done in many ways)

###Question I

```
library(rstan)
```

```
## Warning: package 'rstan' was built under R version 3.6.1
```

```
## Warning: package 'StanHeaders' was built under R version 3.6.1
```

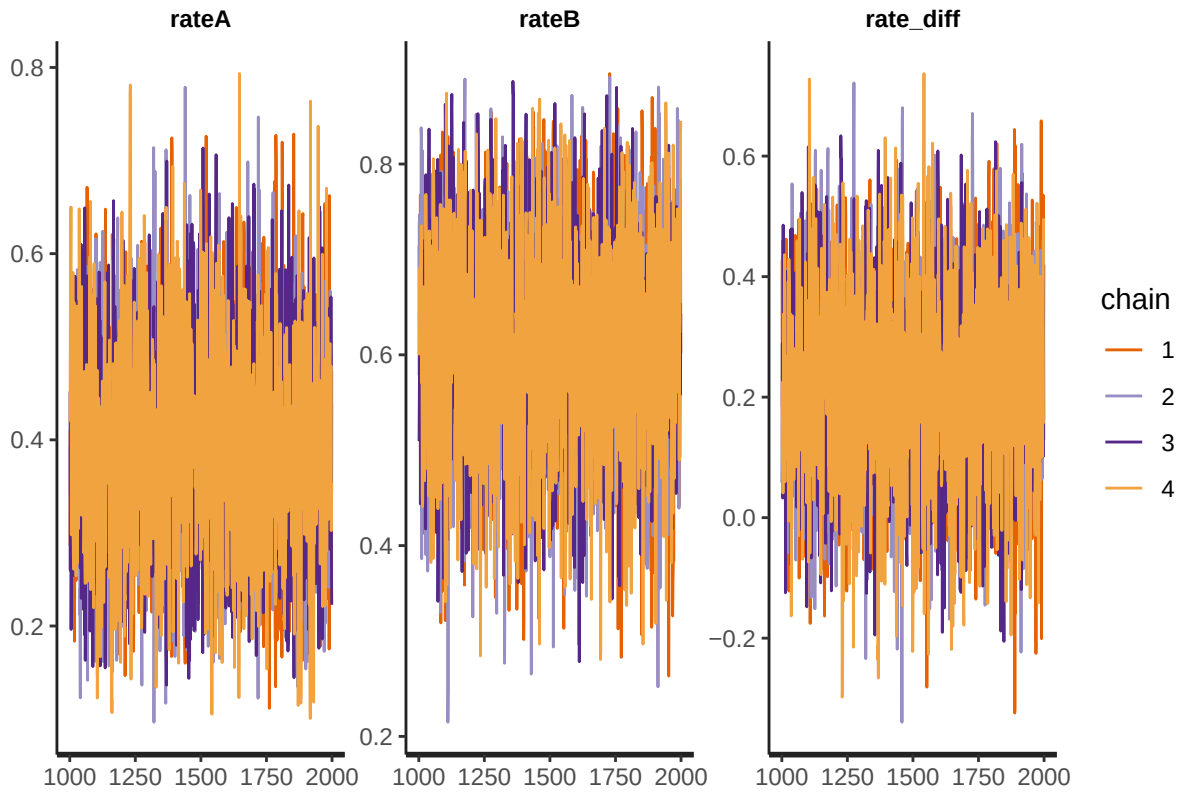
```
## Warning: package 'ggplot2' was built under R version 3.6.1
```

```
# The Stan model as a string.
model_string <- "
data {
  # Number of trials
  int nA;
  int nB;
  # Number of successes
  int sA;
  int sB;
}

parameters {
  real<lower=0, upper=1> rateA;
  real<lower=0, upper=1> rateB;
}

model {
  rateA ~ uniform(0, 1);
  rateB ~ uniform(0, 1);
  sA ~ binomial(nA, rateA);
  sB ~ binomial(nB, rateB);
}

generated quantities {
  real rate_diff;
  rate_diff = rateB - rateA;
}
"

data_list <- list(nA = 16, nB = 16, sA = 6, sB = 10)

# Compiling and producing posterior samples from the model.
stan_samples <- stan(model_code = model_string, data = data_list)
```

```
# Plotting and summarizing the posterior distribution
stan_samples
```

```
## Inference for Stan model: 1bddad524ccb220ab6a6a99f60bff794.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##             mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
## rateA       0.39    0.00 0.11   0.18   0.31   0.38   0.46   0.61  3407    1
## rateB       0.61    0.00 0.11   0.39   0.54   0.62   0.69   0.82  2898    1
## rate_diff   0.22    0.00 0.16  -0.09   0.12   0.23   0.33   0.52  3213    1
## lp__      -25.07    0.03 1.04 -27.90 -25.46 -24.74 -24.34 -24.08  1686    1
##
## Samples were drawn using NUTS(diag_e) at Fri Dec 20 19:48:00 2019.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

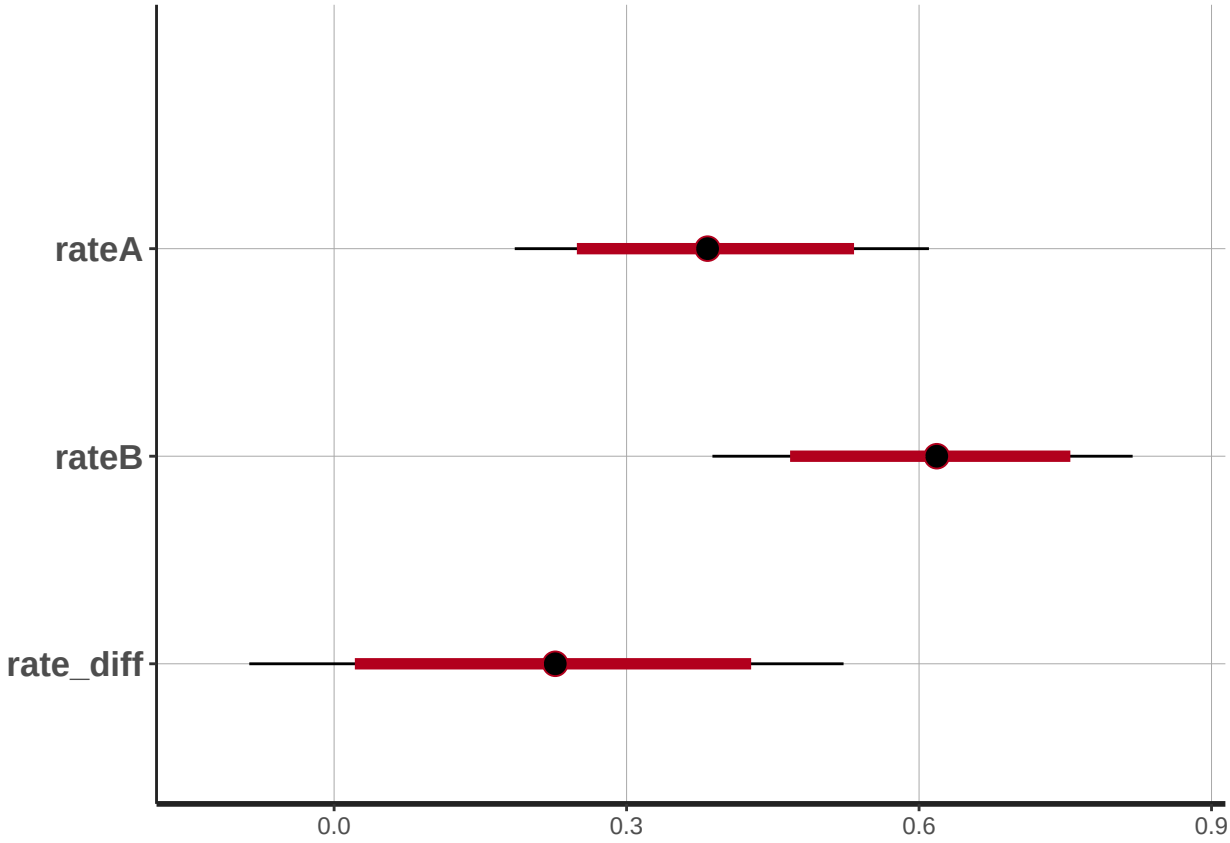
```
traceplot(stan_samples)
```




```
plot(stan_samples)
```

```
## ci_level: 0.8 (80% intervals)
```

```
## outer_level: 0.95 (95% intervals)
```




```
# So, which rate is likely higher? A or B?

# Export the samples to a data.frame for easier handling.
posterior <- as.data.frame(stan_samples)
sum(posterior$rate_diff > 0) / length(posterior$rate_diff)
```

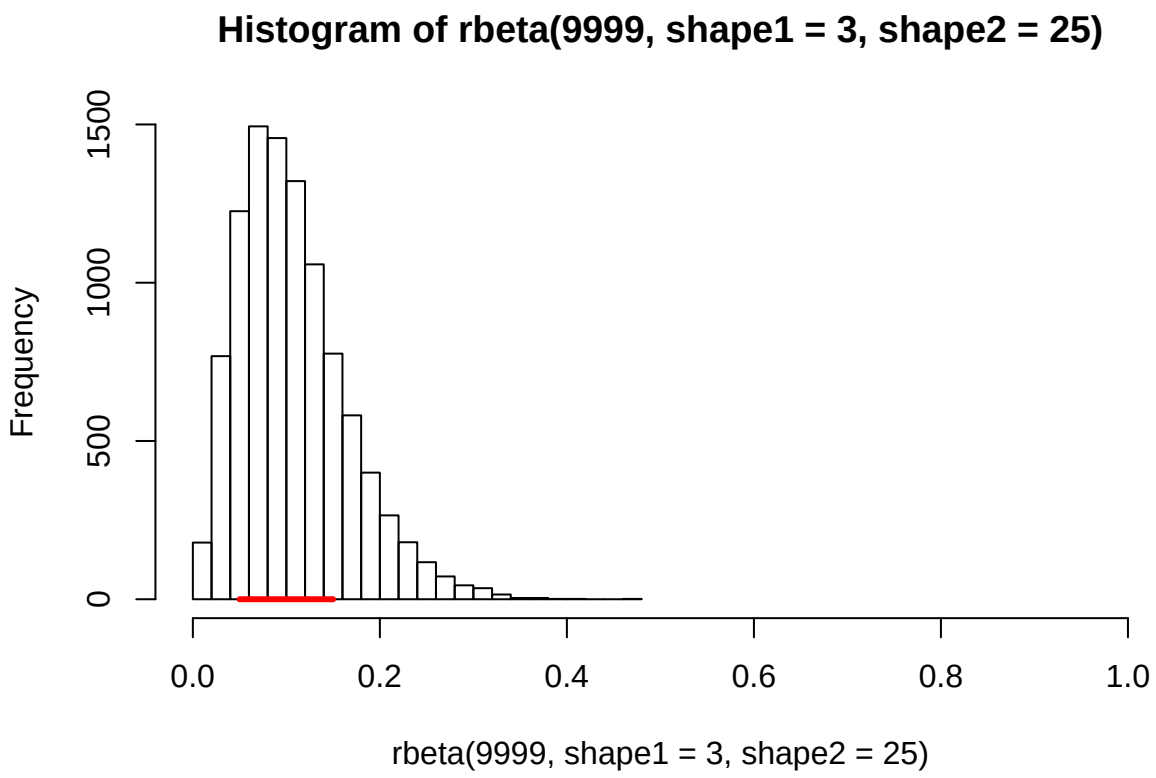
```
## [1] 0.9185
```

```
# So with around 90% probability rate B is higher than rate A.
```

Question II

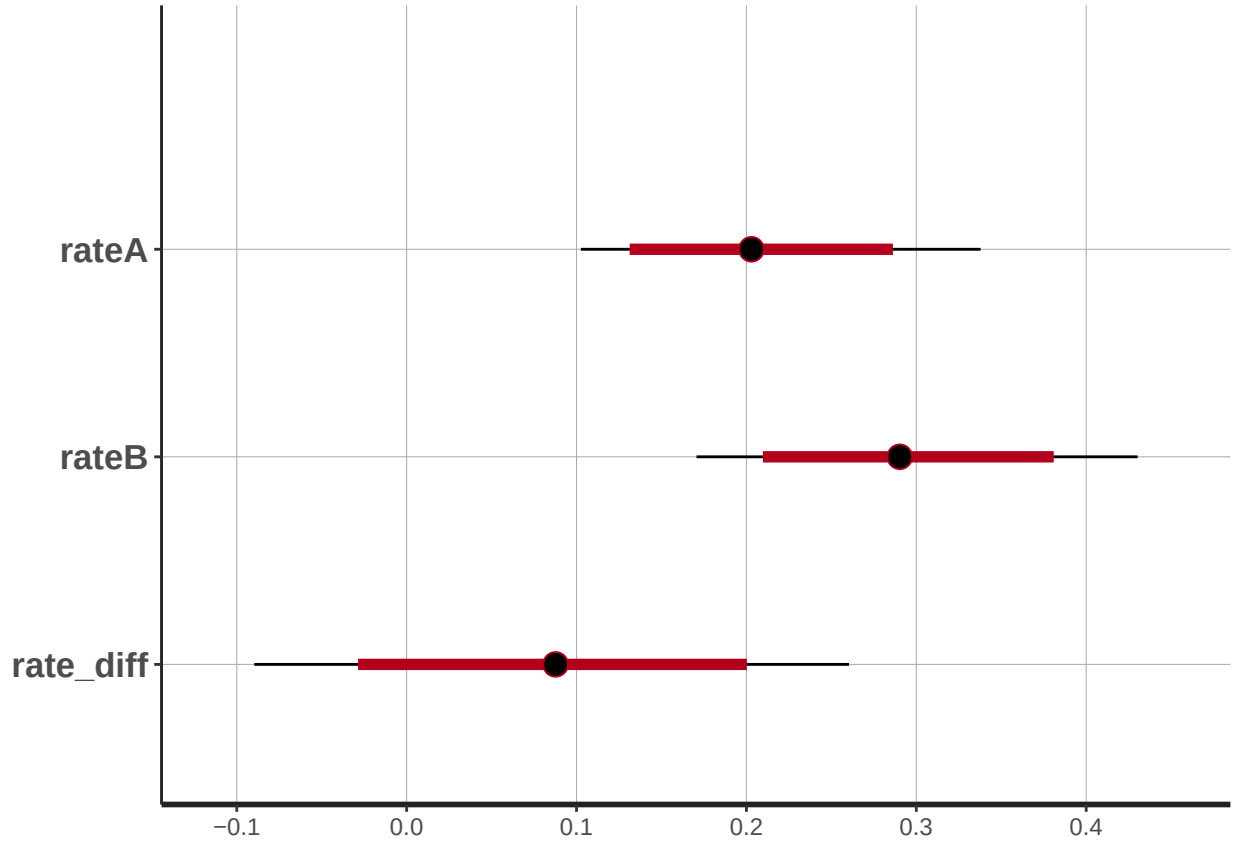
There are an unlimited ways of doing this, and here I'm just going to go with something that I think is decent but surely not perfect.

```
# We will represent the background knowledge using the following beta distribution which is mostly focu

hist(rbeta(9999, shape1 = 3, shape2 = 25), xlim=c(0, 1), 30)
lines(c(0.05, 0.15), c(0,0), col="red", lwd = 3)
```




```
#Except for the prior, the model below is exactly the same as in question I.

library(rstan)

# The Stan model as a string.
model_string <- "
data {
  # Number of trials
  int nA;
  int nB;
  # Number of successes
  int sA;
  int sB;
}

parameters {
  real<lower=0, upper=1> rateA;
  real<lower=0, upper=1> rateB;
}

model {
  rateA ~ beta(3, 25);
  rateB ~ beta(3, 25);
  sA ~ binomial(nA, rateA);
  sB ~ binomial(nB, rateB);
}
```

```
generated quantities {
  real rate_diff;
  rate_diff = rateB - rateA;
}
"

data_list <- list(nA = 16, nB = 16, sA = 6, sB = 10)

# Compiling and producing posterior samples from the model.
stan_samples <- stan(model_code = model_string, data = data_list)

# Plotting and summarizing the posterior distribution
plot(stan_samples)
```




```
posterior <- as.data.frame(stan_samples)
sum(posterior$rate_diff > 0) / length(posterior$rate_diff)
```

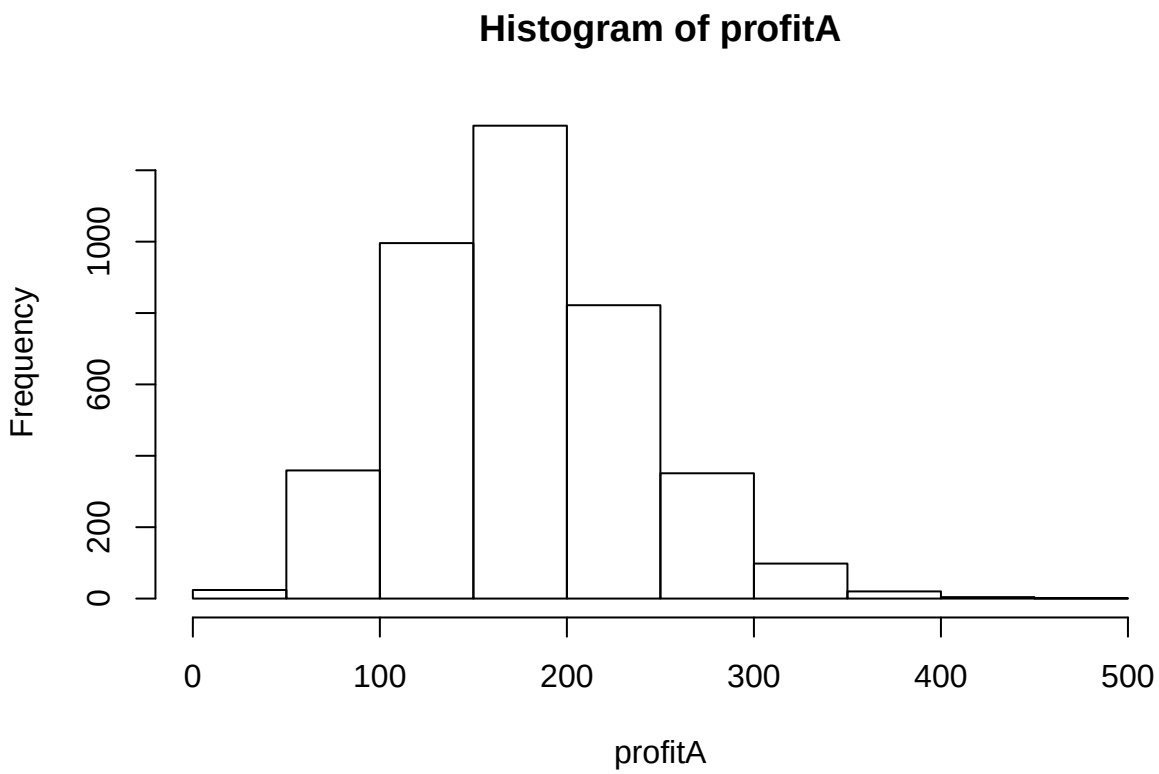
```
## [1] 0.831
```

```
# So rate B is still estimated to be higher than A with around
# 80% probability, but both rates are estimated to be much lower.
```

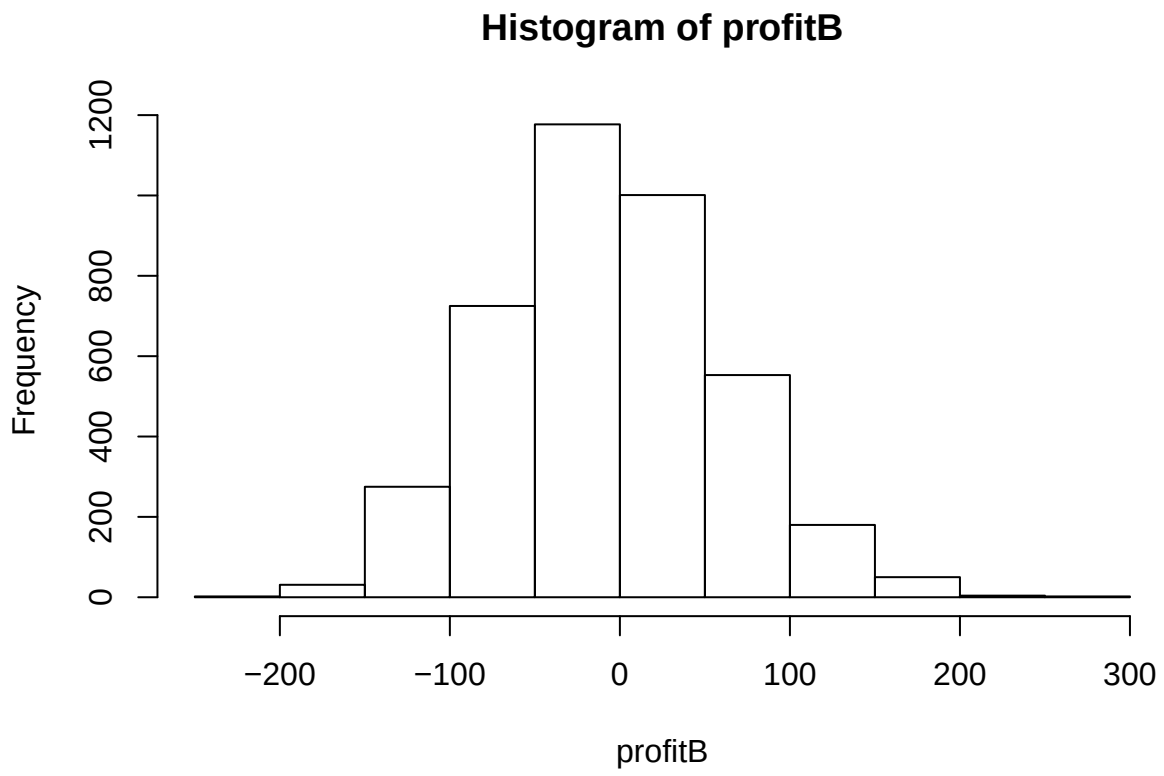
Question III

Here we don't have to make any changes to the model, it is enough to "post-process" the posterior distribution in `posterior`.

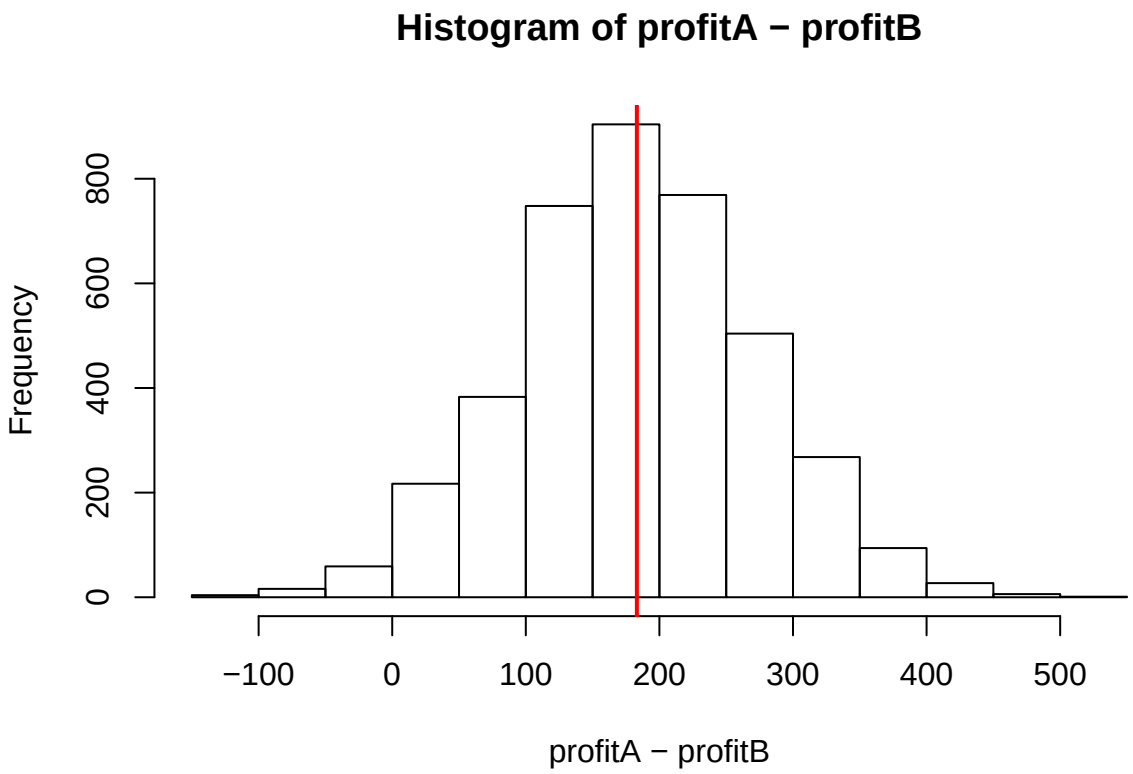
```
posterior <- as.data.frame(stan_samples)
# calculating the estimated posterior profit using method A (or B)
# a cost of 30 kr + the average profit per sent out add
profitA <- -30 + posterior$rateA * 1000
profitB <- -300 + posterior$rateB * 1000
hist(profitA)
```




```
hist(profitB)
```




```
hist(profitA - profitB)
expected_profit_diff <- mean(profitA - profitB)
abline(v = expected_profit_diff, col = "red", lwd =2)
```



The expected profit when using method A is around 190 kr higher than for method B (which actually has a negative expected profit). So I guess sending free salmon to people isn't the best idea. But note that we got this result after having made the decision analysis based on the model with the *informative* priors. If we use the non-informative priors we get a different result, and it's up to you, the analyst, to decide which version of the model you decide to use.