

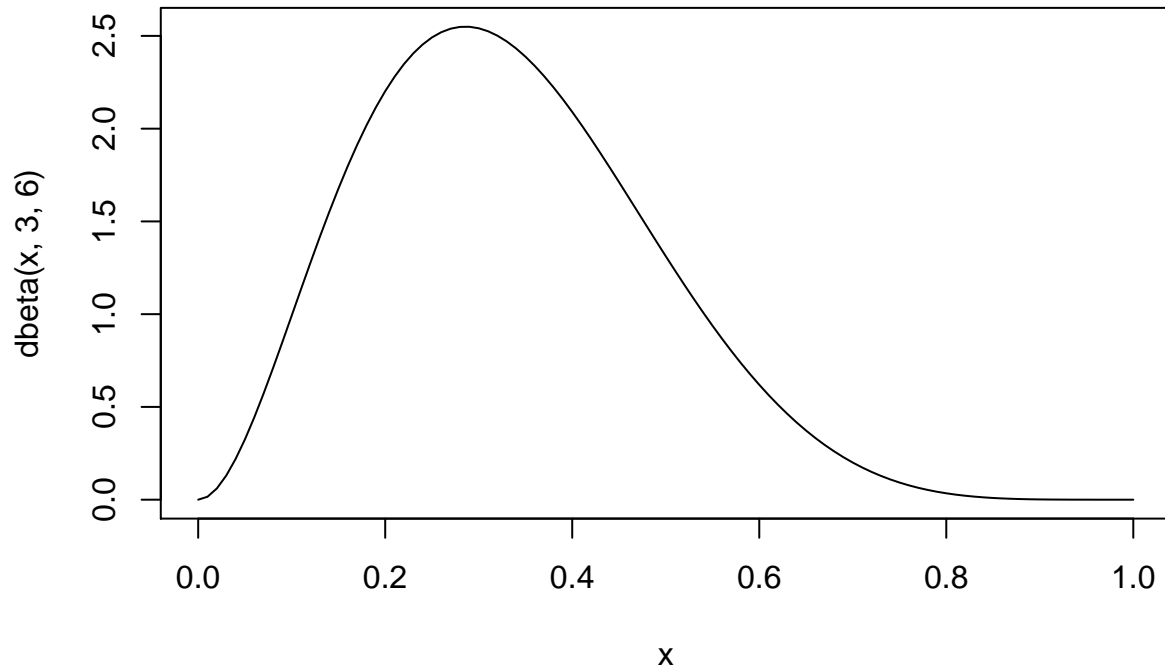
Rejection sampling

Motivation

This code snippet is courtesy of Florian Hartig: <https://theoreticalecology.wordpress.com/2015/04/22/a-simple-explanation-of-rejection-sampling-in-r/>

The idea is to sample from a beta distribution with parameters: $\alpha = 3$ and $\beta = 6$

```
curve(dbeta(x, 3,6),0,1)
```

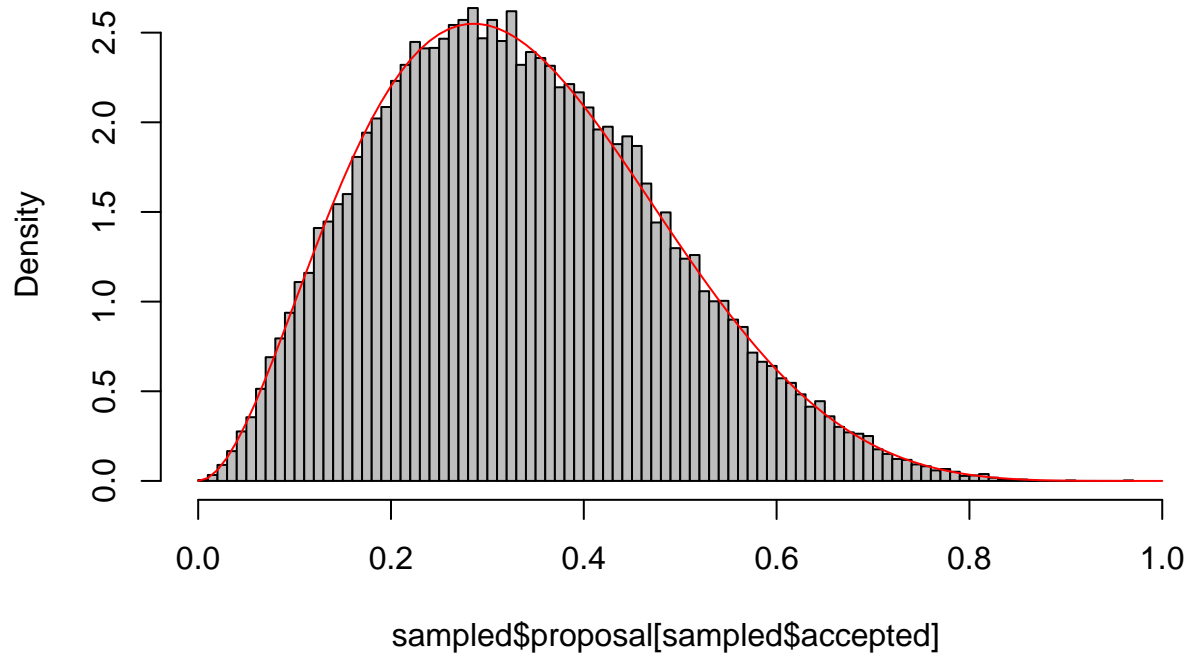


For this we can draw randomly from a uniform distribution and evaluate those values with the beta density function, providing a sort of acceptance weights if we set them proportional to the initially used uniform distribution. Here, we are dividing the densities by the maximal density to normalize the values to the $[0,1]$ interval just like the observations from $U(0,1)$

```
sampld <- data.frame(proposal = runif(100000,0,1))
sampld$targetDensity <- dbeta(sampld$proposal, 3,6)
maxDens = max(sampld$targetDensity, na.rm = T)
sampld$accepted = ifelse(runif(100000,0,1) < sampld$targetDensity / maxDens, TRUE, FALSE)

hist(sampld$proposal[sampld$accepted], freq = FALSE, col = "grey", breaks = 100)
curve(dbeta(x, 3,6),0,1, add = T, col = "red")
```

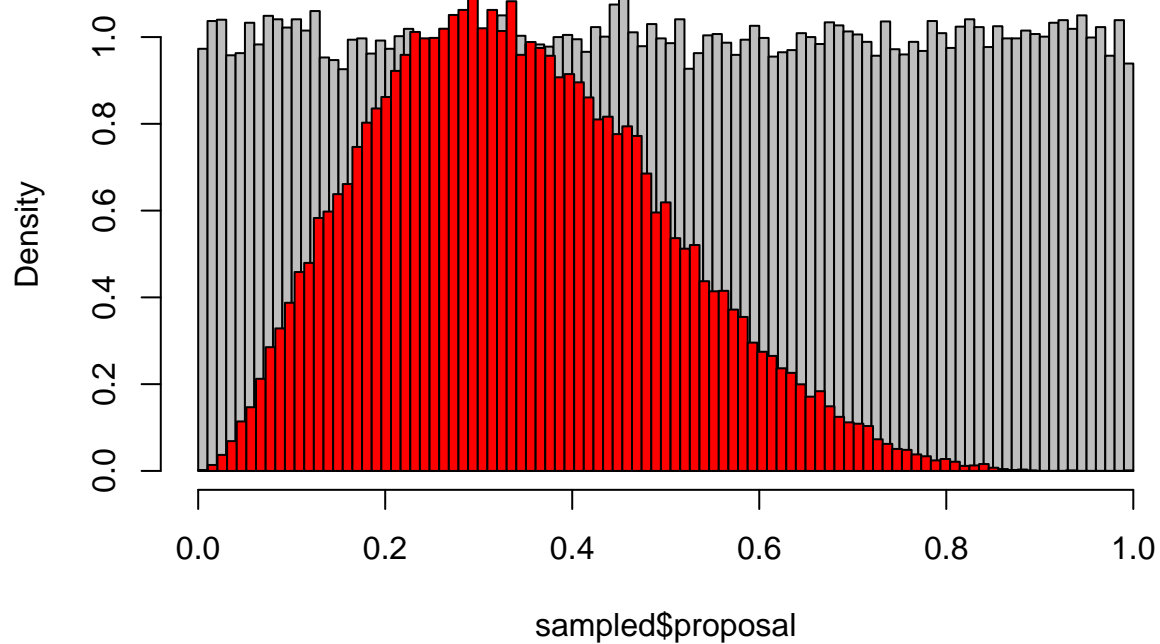
Histogram of sampled\$proposal[sampled\$accepted]



A reason why the random observations from the prior distribution have to be proportional to the posterior density is that the density of the random observation has to encapsulate the posterior density in order to ensure that all possible values of the posterior can be sampled.

```
hist(sampled$proposal, freq = F, col = "grey", breaks = 100)
par(new = TRUE)
hist(sampled$proposal[sampled$accepted], freq = FALSE, col = "red", breaks = 100, xlab = "", ylab = "",
      xaxt = "n", yaxt = "n", main = "")
```

Histogram of sampled\$proposal

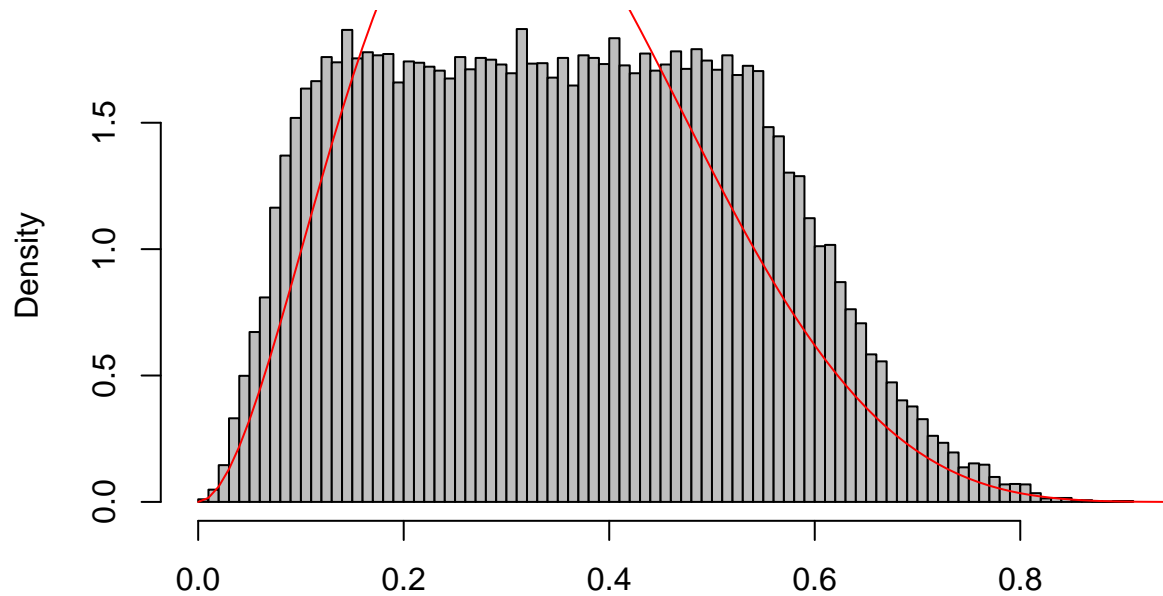


In this plot the grey area is the density of randomly drawn observations from the uniform distribution. While the red area is the normalized density of the beta distribution. It is clear to see that the density works like a weight for the values drawn between 0 and 1. If we wouldn't have normalized the density, the prior distribution would not have been able to capture the entirety of the beta density.

```
sampled <- data.frame(proposal = runif(100000,0,1))
sampled$targetDensity <- dbeta(sampled$proposal, 3,6)
maxDens = max(sampled$targetDensity, na.rm = T)
sampled$accepted = ifelse(runif(100000,0,1) < sampled$targetDensity, TRUE, FALSE)

hist(sampled$proposal[sampled$accepted], freq = FALSE, col = "grey", breaks = 100)
curve(dbeta(x, 3,6),0,1, add = T, col = "red")
```

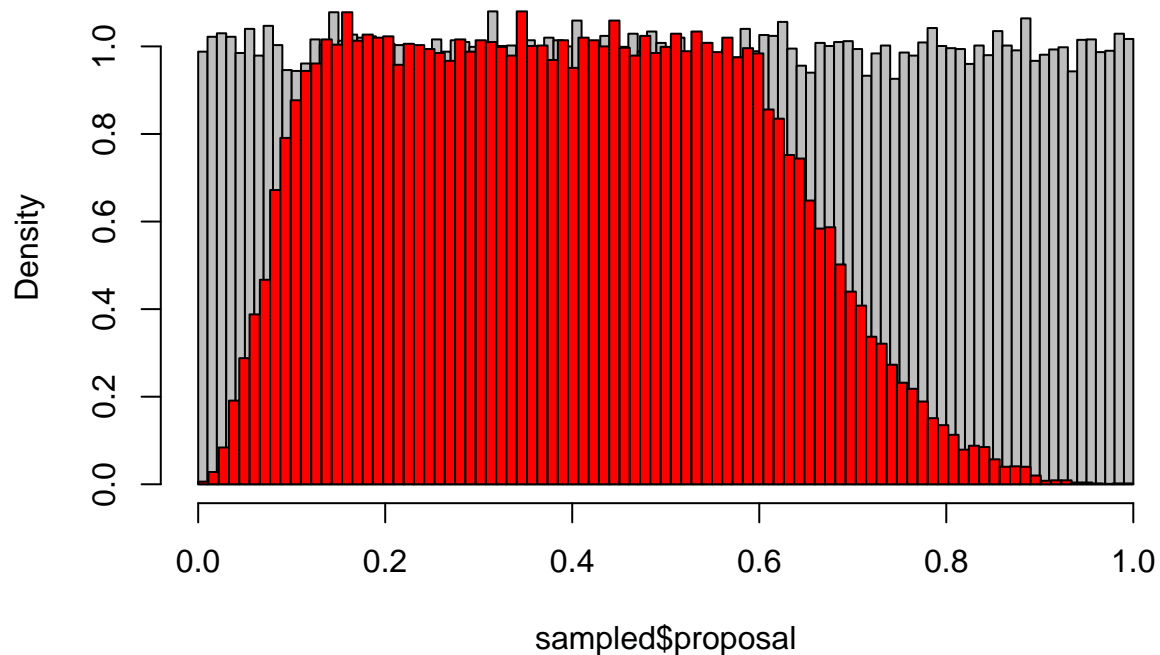
Histogram of sampled\$proposal[sampled\$accepted]



sampled\$proposal[sampled\$accepted]

```
hist(sampled$proposal, freq = FALSE, col = "grey", breaks = 100)
par(new = TRUE)
hist(sampled$proposal[sampled$accepted], freq = FALSE, col = "red",
      breaks = 100, xlab = "", ylab = "",
      xaxt = "n", yaxt = "n", main = "")
```

Histogram of sampled\$proposal



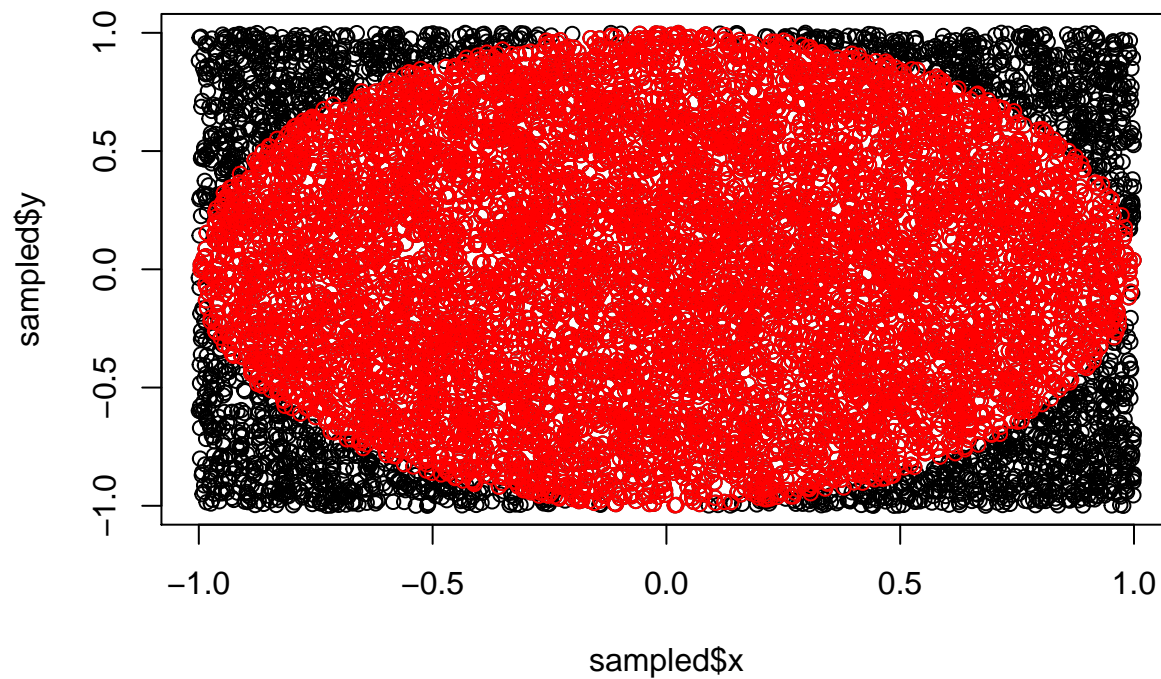
Own attempt

For better understanding this I try to sample from a circle:

1. Given two values x and y , a unity circle can be defined by the function $x^2 + y^2 = 1$
2. Thus we sample x and y from $U(-1,1)$ and reject if $x^2 + y^2 > 1$

```
library(data.table)
k <- 10000
sampled <- data.table(x = runif(k, -1, 1), y = runif(k, -1, 1))
sampled[,eval := x^2 + y^2]
sampled$accepted <- ifelse(sampled$eval <= 1, TRUE, FALSE)

plot(sampled$x, sampled$y)
points(sampled$x[sampled$accepted], sampled$y[sampled$accepted], col = "red")
```



More formal