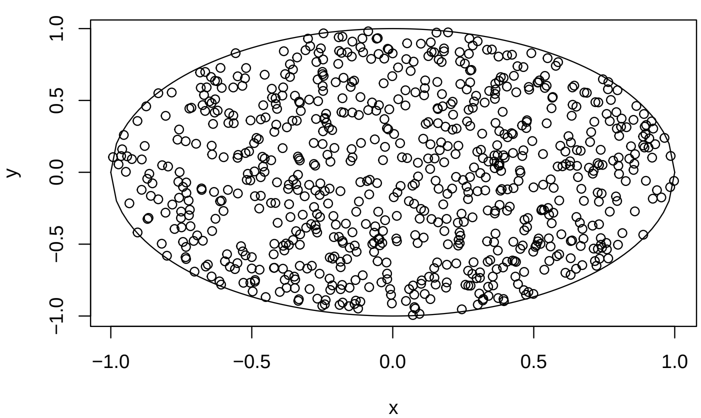
STA 602 Lab 5

Yicheng Shen

17 October, 2022

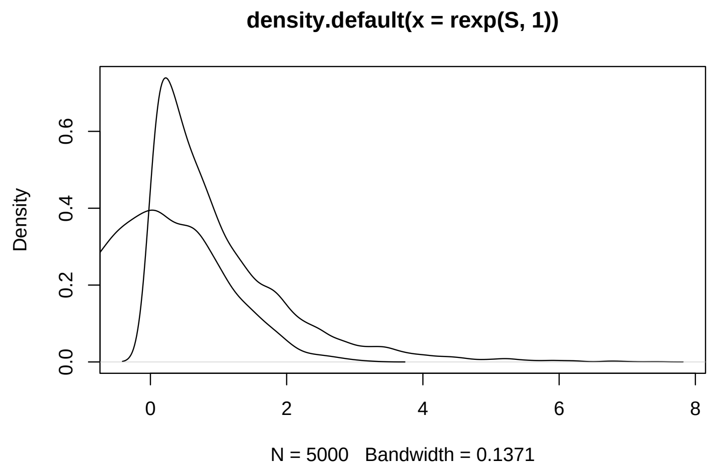
Exercise 1

```
x <- runif(1000,-1,1)
y <- runif(1000,-1,1)
idx <- (x^2 + y^2 < 1)

plot(x[idx], y[idx], xlab="x", ylab="y")
curve((  1 * (1 - x^2)^0.5 ), add=TRUE, from=-1 , to =1)
curve(( -1 * (1 - x^2)^0.5 ), add=TRUE, from=-1 , to =1)
```



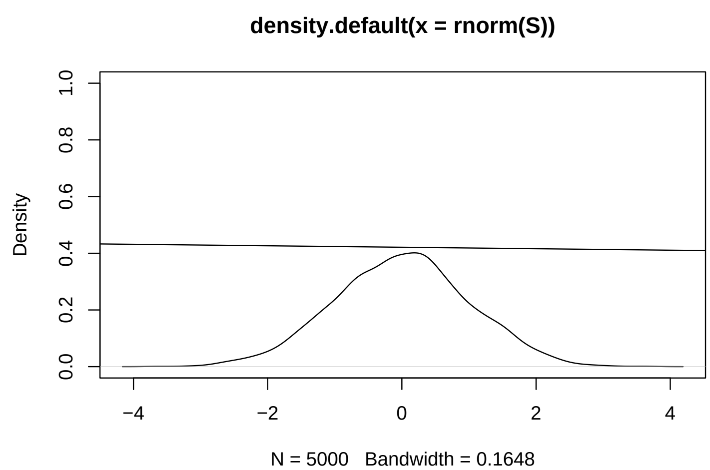
Can we use Exponential(1) as the candidate density to draw samples for N(0,1) on the interval (-3,3)? No.

```
S <- 5000
plot(density(rexp(S,1)))
lines(density(rnorm(S)))
```



Can we use N(0,1) as candidate density to draw samples from the standard Cauchy distribution on the interval (-2,2)? Yes.

```
plot(density(rnorm(S)), ylim = c(0,1))
lines(density(rcauchy(S, -2, 2)))
```

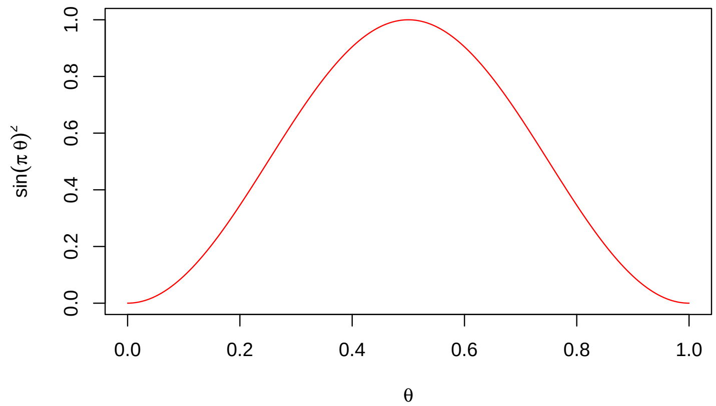


If we only know the kernel of a uni-variate posterior distribution we are interested in, can we use RS? Yes.

What if it's a 100-variate posterior distribution? No.

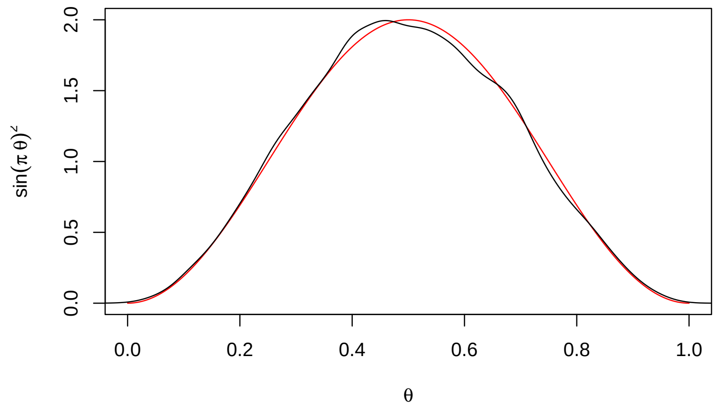
Exercise 2

```
theta <- seq(0,1,length.out = 200)
g_theta <- sin(pi * theta)^2

plot(theta, g_theta, type = "l", col="red",
     xlab = expression(theta), ylab = expression(sin(pi~theta)^2))
```




```r
# target density function
f <- function(x){
  sin(pi*x)^2
}

nsim <- 10000

# acceptance count, total count
ac <- tc <- 0

# to store results
res <- numeric(nsim)

# candidate density function
g <- function(x) { dunif(x) }

M <- 0.0001

while (ac < nsim){
  tc <- tc + 1
  x <- runif(1)
  u <- runif(1)

  if ( u < f(x) )
  {
    res[ac] <- x
    ac <- ac + 1
  }

}
```

```r
plot(theta, g_theta*2, type = "l", col="red",
     xlab = expression(theta), ylab = expression(sin(pi~theta)^2), ylim = c(0,2))
lines(density(res))
```



Exercise 3

one is the density of distribution one is kernel.

Exercise 4

Exercise 5

Exercise 6

the ratio of normalizing constant

$1/N \sum \frac{\tilde{f}}{\tilde{g}} h(x')$ converge to $E_f[h]$

Exercise 7

Exercise 8

Plug in posterior distributions.

Exercise 9

```
tumors <- read.csv(file = url("http://www.stat.columbia.edu/~gelman/book/data/rats.asc"),
                   skip = 2, header = T, sep = " ")[,c(1,2)]
y <- tumors$y
N <- tumors$N
n <- length(y)

stan_dat <- list(n = n, N = N, y =y, a = 1, b = 1)
fit_pool <- stan('lab-02-pool.stan', data = stan_dat, chains = 2, refresh = 0)
## Trying to compile a simple C file
pool_output <- rstan::extract(fit_pool)
```

```r
as <- c(1,10,25,100)
bs <- c(1,10,25,100)

# posterior parameters under the current Beta(1,1) prior
post_a0 <- 1 + sum(y)
post_b0 <- 1 + sum(N) - sum(y)

IS_sensi_mean <- matrix(0, length(as), length(bs))  # IS estimated posterior mean
exp_sensi_mean <- matrix(0, length(as), length(bs))  # theoretical posterior mean

for (i in (1:length(as))){
  for (j in (1:length(bs))){

    # fill in your codes here

  }
}

# plot the theoretical posterior means against the IS estimated posterior means
plot(c(IS_sensi_mean), c(exp_sensi_mean), xlab = "IS", ylab = "Expected")
abline(a = 0, b = 1)  # 45 degree line
```