STA2102 Homework 1

Kevin Zhang 1002225264

October 8, 2020

Question 1

a)

$$\frac{H_m \hat{Z} H_n}{mn} = \frac{H_m H_m Z H_n H_n}{mn} = \frac{mIZnI}{mn} = Z$$

Notice that H_m and H_n are symmetric and $H_m H_m^\intercal = mI$ and $H_n H_n^\intercal = nI$.

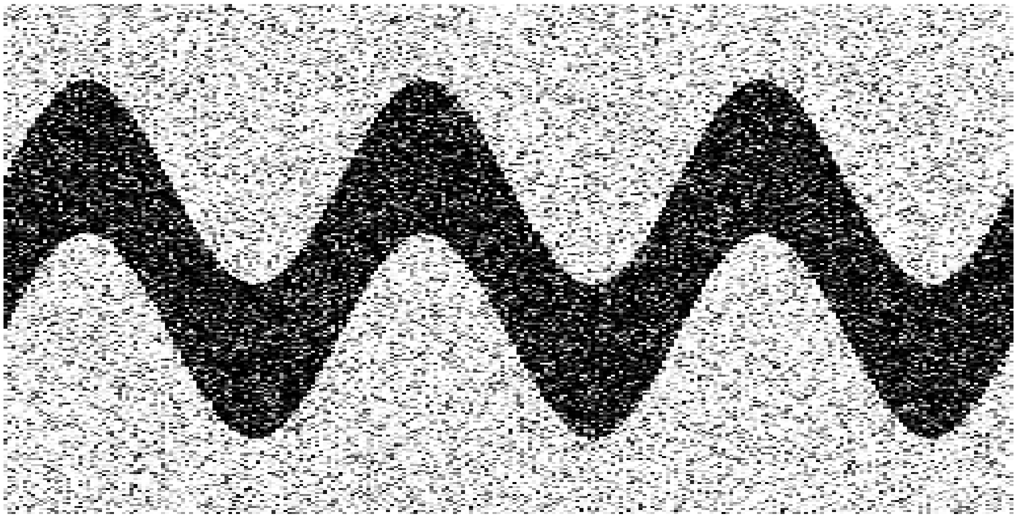
b)

```
hard_th <- function(x, lambda){
  I = nrow(x)
  J = ncol(x)
  for(i in 1:I){
    for(j in 1:J){
      x[i,j] = ifelse(abs(x[i,j])>lambda, x[i,j], 0)
    }
  }
  return(x)
}
soft_th <- function(x, lambda){
  I = nrow(x)
  J = ncol(x)
  for(i in 1:I){
    for(j in 1:J){
      x[i,j] = sign(x[i,j]) * max(abs(x[i,j])-lambda, 0)
    }
  }
  return(x)
}
fwht2d <- function(x){
  h <- 1
  len <- ncol(x)
  while(h < len){
    for(i in seq(1,len,by=h*2)){
      for (j in seq(i,i+h-1)){
        a <- x[,j]
        b <- x[,j+h]
        x[,j] <- a + b
        x[,j+h] <- a - b
      }
```

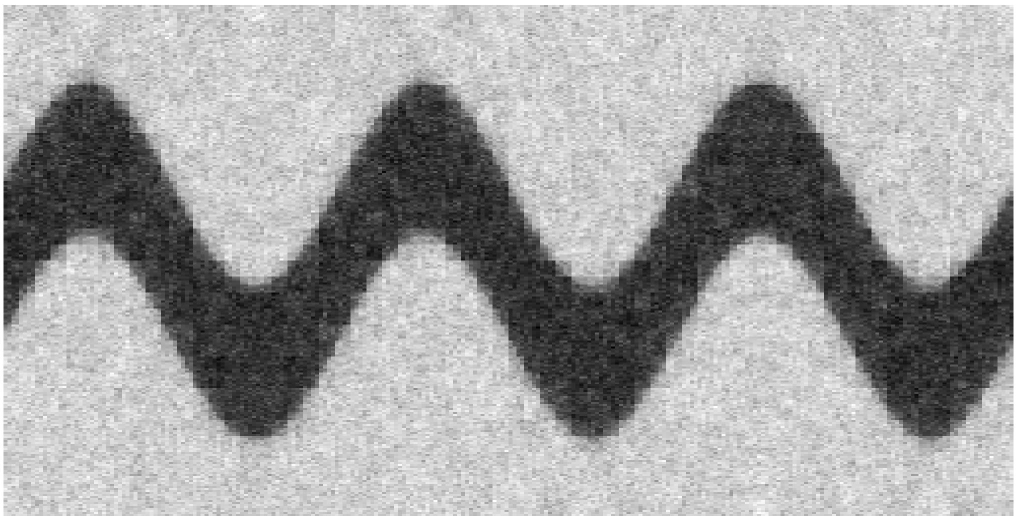
```
    }
    h <- 2*h
  }
  h <- 1
  len <- nrow(x)
  while(h < len){
    for(i in seq(1,len,by=h*2)){
      for(j in seq(i,i+h-1)){
        a <- x[j,]
        b <- x[j+h,]
        x[j,] <- a + b
        x[j+h,] <- a - b
      }
    }
    h <- 2*h
  }
  x
}
```

c)

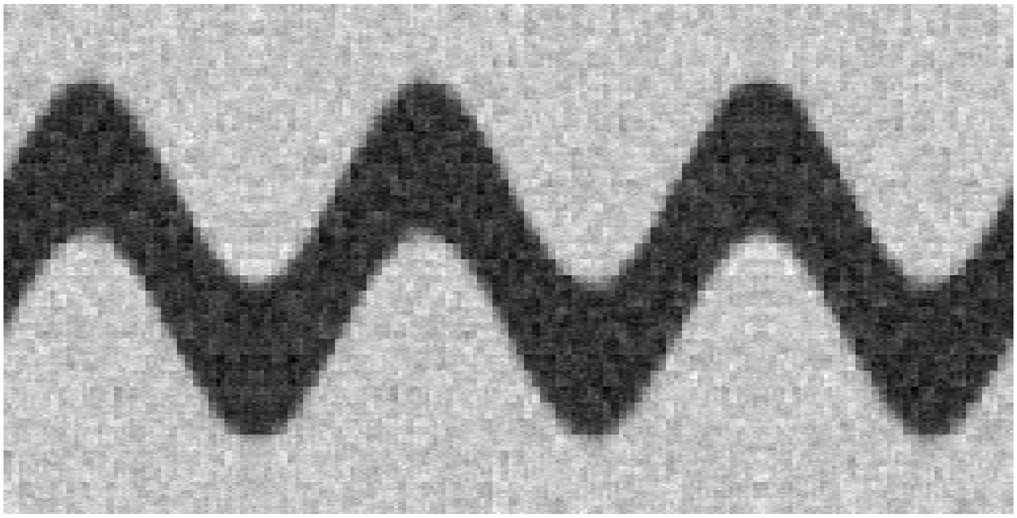
```
design <- matrix(scan("design.txt"),ncol=256,byrow=T)
colours <- grey(seq(0,1,length=256))
image(design, axes=F, col=colours)
```



```
xhat <- fwht2d(design)
xhat_soft <- soft_th(xhat, 100)
xhat_hard <- hard_th(xhat, 200)
x_soft <- fwht2d(xhat_soft)/ncol(xhat_soft)^2
x_hard <- fwht2d(xhat_hard)/ncol(xhat_hard)^2
image(x_soft, axes=F, col=colours)
```



```
image(x_hard, axes=F, col=colours)
```



Question 2

a)

$$E(t^S) = \sum_{n=0}^{\infty} E(t^X|N=n)P(N=n) = \sum_{n=0}^{N} (\phi(t))^n (1-\theta)\theta^n = \frac{1-\theta}{1-\theta\phi(t)} = g(\phi(t))$$

Note that $\sum_{k=0}^{\infty} ar^k = \frac{a}{1-r}$.

b)

Suppose $N < m$, then $S = \sum_{i=1}^{N} X_i \leq \sum_{i=1}^{m-1} X_i \leq (m-1)l < ml$. Therefore, by contrapositive, $S \geq ml$ implies $N \geq m$. Hence, $P(S \geq ml) \leq P(N \geq m) \leq \epsilon$.

c)

$$P(S \geq M) \leq \frac{1}{t^M}\left(\frac{1-\theta}{1-\theta\phi(t)}\right) \implies P(S \geq M) \leq \inf_{1<\phi(t)<\theta^{-1}} \frac{1}{t^M}\left(\frac{1-\theta}{1-\theta\phi(t)}\right)$$

$$\text{suppose } \inf_{1<\phi(t)<\theta^{-1}} \frac{1}{t^M}\left(\frac{1-\theta}{1-\theta\phi(t)}\right) = \epsilon \text{ i.e. } P(S \geq M) \leq \epsilon$$

$$\text{since } \ln(x) \text{ is continuous, then } \inf_{1<\phi(t)<\theta^{-1}} \ln(1-\theta) - \ln(1-\theta\phi(t)) - M\ln(t) = \ln(\epsilon)$$

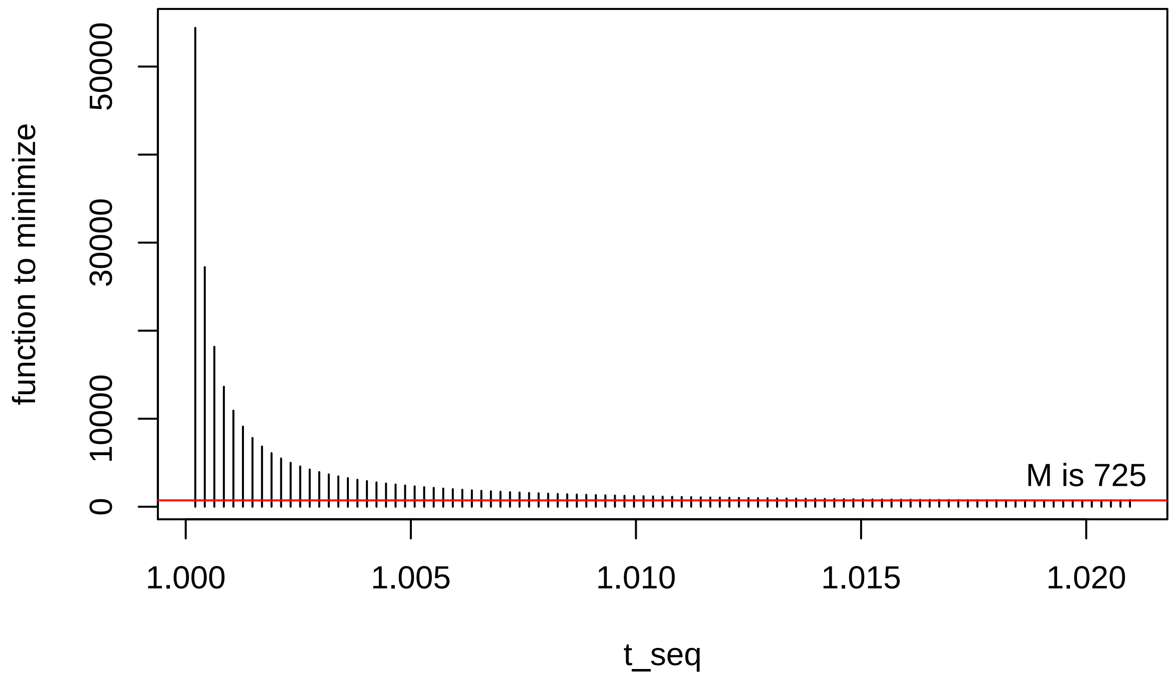
$$\implies \inf_{1<\phi(t)<\theta^{-1}} \frac{\ln(1-\theta) - \ln(1-\theta\phi(t)) - \ln(\epsilon)}{\ln(t)} = M$$

d)

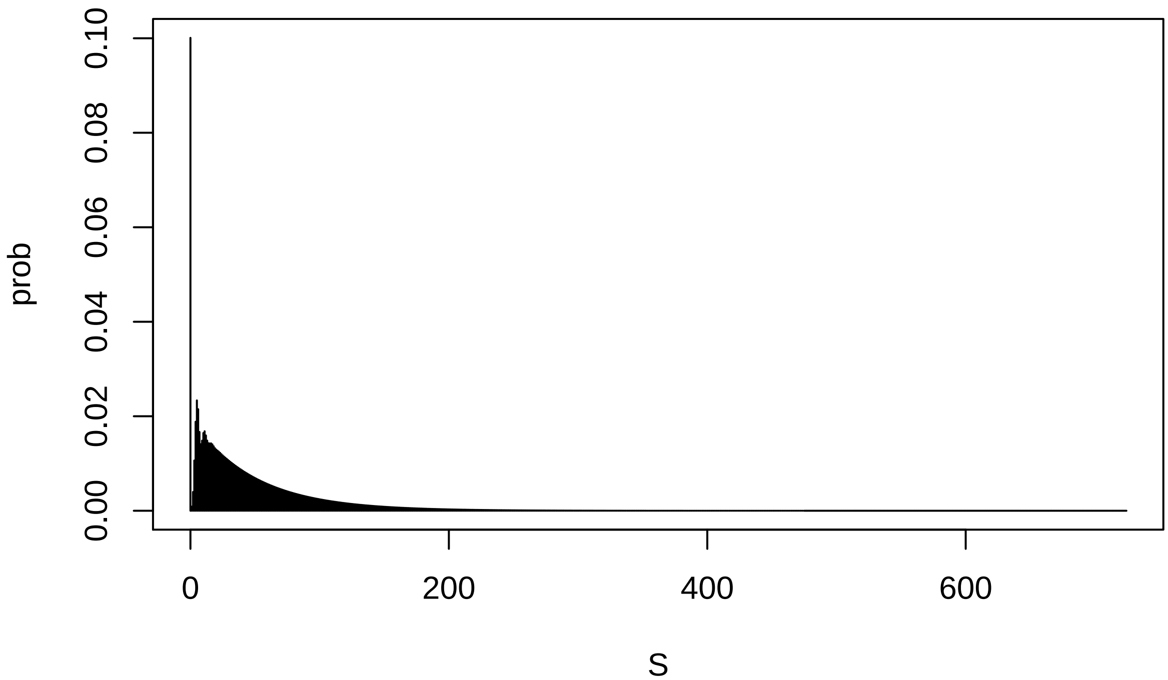
```r
p <- numeric(0)
for(i in 0:10){
  p <- c(p, choose(10, i)*0.5^10)
}
phi <- function(t){
  return((1+t)^10/2^10)
}
get_dist_s <- function(x_max, epsilon, phi, theta){ # x_max = l
  t_max <- (1/0.9*2^10)^0.1-1 # the max value for t such that phi(t)<1/theta
  t_seq <- seq(1, t_max, by=(t_max-1)/100)
  t_seq <- t_seq[-1] # remove t=1
  t_seq <- t_seq[-length(t_seq)] # remove t_max
  M <- Inf
  for(t in t_seq){
    M_temp <- (log(1-theta)-log(1-theta*phi(t))-log(epsilon))/log(t)
    if(M_temp < M){
      M <- ceiling(M_temp)
    } # M = min of the function
  }
  p_dft <- numeric(0) # dft of all the p
  for(j in 0:(M-1)){
    p_temp <- 0
    for(x in 0:x_max){
        p_temp <- p_temp + exp(-2*pi*complex(real=0, imaginary=1)*j/M*x)*p[x+1]
    }
    p_dft <- c(p_dft, p_temp)
  }
  g <- numeric(0) # g of p
  for(p_j in p_dft){
    g <- c(g, (1-theta)/(1-theta*p_j))
  }
  ps <- numeric(0) # vector of p(S=s)
  for(s in 0:(M-1)){
    ps_temp <- 0
    for(j in 0:(M-1)){
      ps_temp <- ps_temp + exp(2*pi*complex(real=0,imaginary=1)*s/M*j)*g[j+1]
    }
    ps <- c(ps, round(Re(ps_temp/M), 8))
  }
```

```
  return(list(ps, M)) #jth element of ps is p(S=j-1)
}
```

```
t_max <- (1/0.9*2^10)^0.1-1 # the max value for t such that phi(t)<1/theta
t_seq <- seq(1, t_max, by=(t_max-1)/100)
t_seq <- t_seq[-1] # remove t=1
t_seq <- t_seq[-length(t_seq)] # remove t_max
x_max <- 10
epsilon <- 10^-5
theta <- 0.9
plot(t_seq, (log(1-theta)-log(1-theta*phi(t_seq))-log(epsilon))/log(t_seq),
     ylab='function to minimize', type='h')
M <- get_dist_s(x_max, epsilon, phi, theta)[[2]]
ps <- get_dist_s(x_max, epsilon, phi, theta)[[1]]
abline(h=M, col='red')
text(x=1.02, y=5*M, label=sprintf('M is %d', M))
```




```
plot(0:(M-1), ps, type='h', xlab='S', ylab='prob')
```


0.00
0.02
0.04
0.06
0.08
0.10
prob
0
200
400
600
S