Solutions to Assignment #1

1. (a) $\widehat{Z} = H_m Z H_n$. Thus

$$H_m \widehat{Z} H_n = H_m^2 Z H_n^2 = mnZ$$

since $H_n^2 = nI$. Thus $Z = H_m \widehat{Z} H_n/(mn)$.

(b) Functions for the hard and soft thresholding can either use the original image as an input (with its W-H transform computed within the function) or the W-H transform as an input. The examples given below show the latter approach:

```
hard.thres <- function(x,lambda) {
                 a <- x[1,1]
                 x <- ifelse(abs(x)<=lambda,0,x)
                 x[1,1] <- a
                 x <- fwht2d(x)/(nrow(x)*ncol(x))
                 x
                 }
soft.thres <- function(x,lambda) {
                 a <- x[1,1]
                 x <- sign(x)*pmax(abs(x)-lambda,0)
                 x[1,1] <- a
                 x <- fwht2d(x)/(nrow(x)*ncol(x))
                 x
                 }
```

Note that these two functions do not shrink the $(1,1)$ element of the matrix – this is usually a good practice to follow although in this case, shrinking the $(1,1)$ element (which only relevant for soft thresholding) does not change the denoised images significantly unless λ is very large.

(c) Figures 1 and 2 show the hard and soft thresholded images using $\lambda = 60, 80, 100, 120$ for each method.

The value of λ influences (a) the contrast between the dark and light regions of the image, and (b) the sharpness of the boundary between the dark and light regions. As λ increases. we do see an increase in the contrast between the dark and light regions coupled with a lessening of sharpness of the boundary between the two regions.

This denoising method as applied here is very crude. First, we use a single threshold value for all of elements (with the exception of the $(1,1)$ element) of the matrix W-H transform, Second (and perhaps more importantly), we are applying the transform to the entire image rather than applying the transform to smaller sub-images and thresholding each of these transforms.

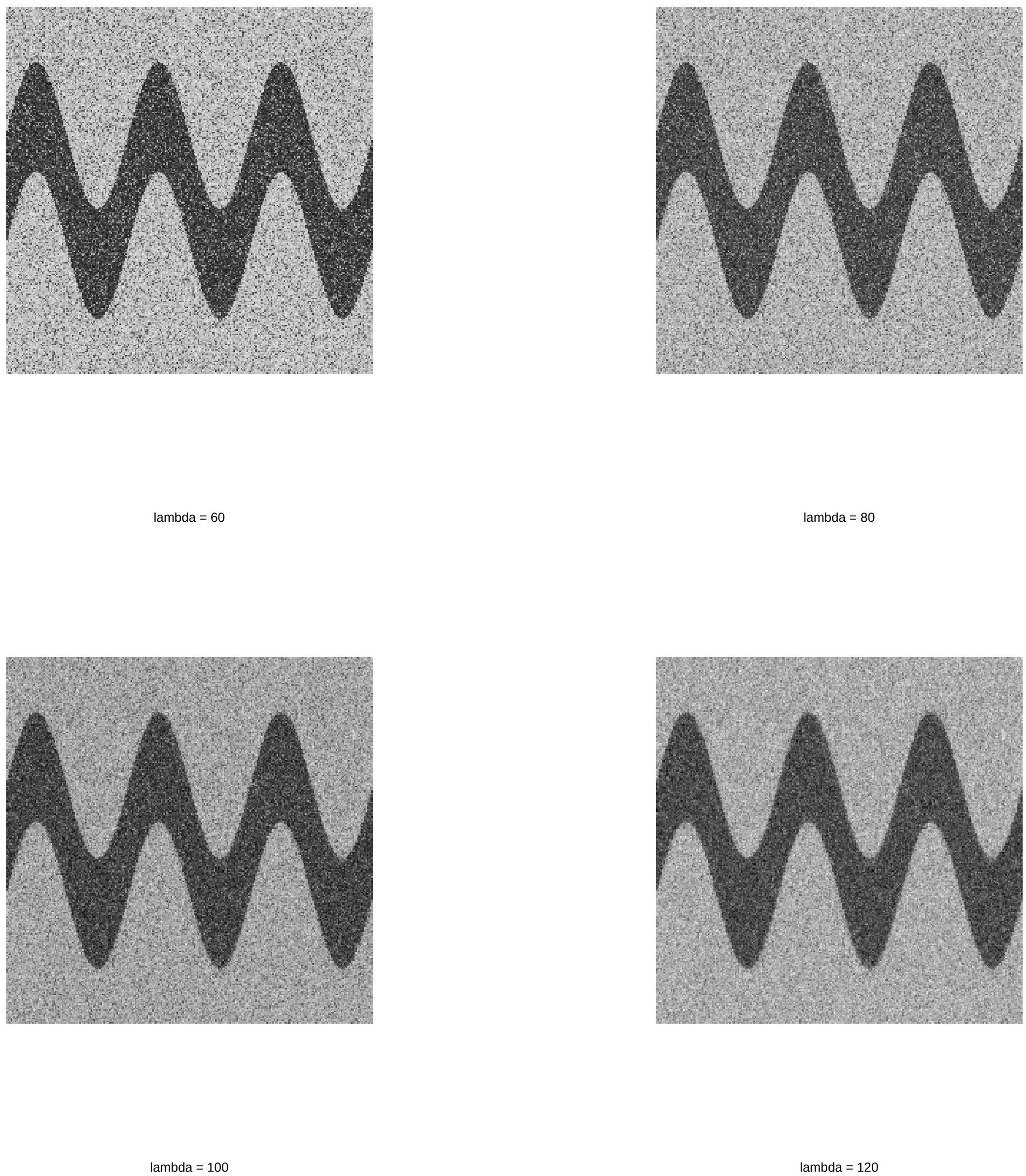

Figure 1: Hard thresholded images

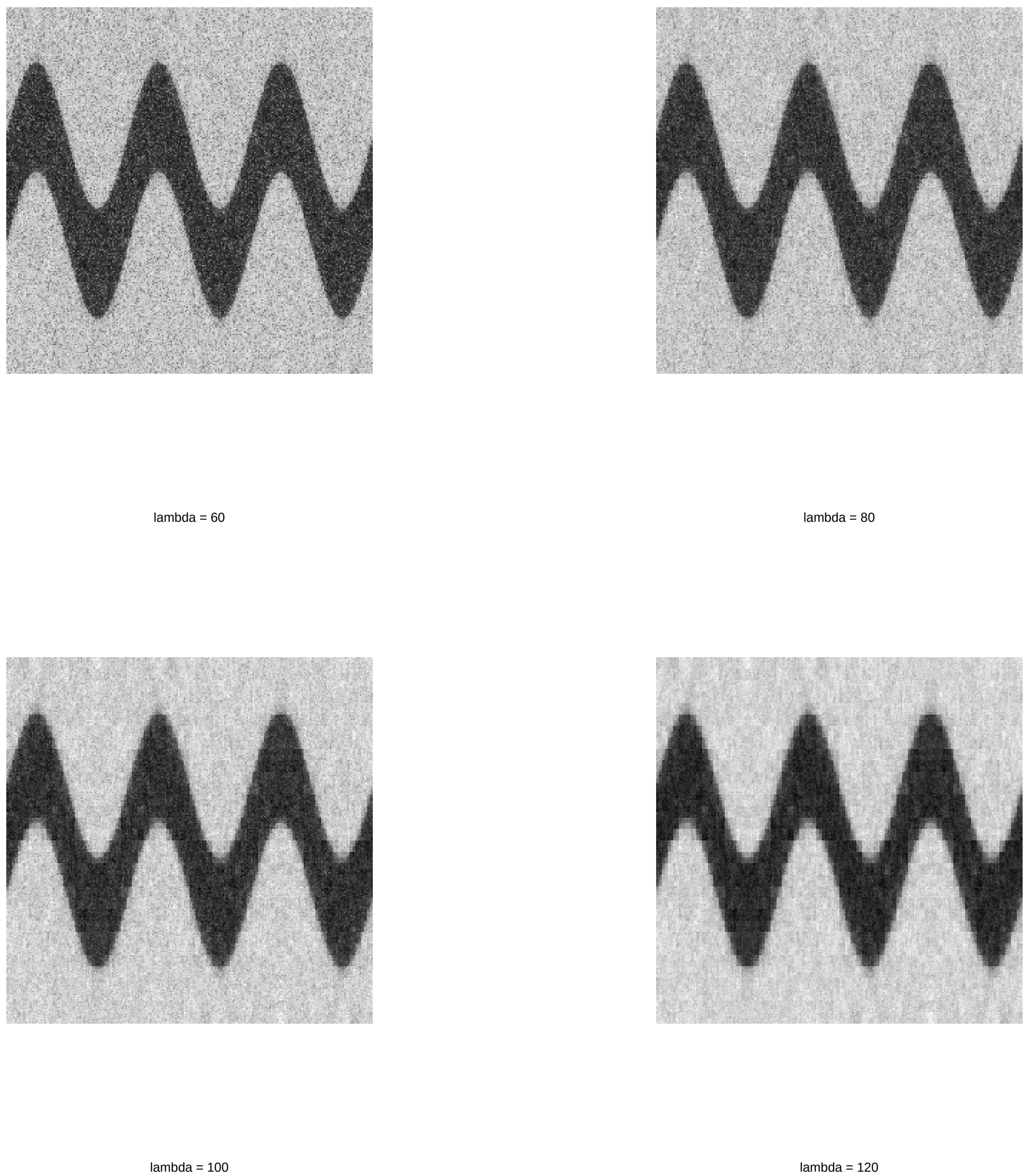

Figure 2: Soft thresholded images

2. (a) We use the facts that (a) $E(t^S) = \sum_{n=0}^{\infty} E(t^S|N = n)P(N = n)$ and (b) $E(t^S|N = n) = \phi(t)^n$. Thus

$$E(t^S) = \sum_{n=0}^{\infty} \phi(t)^n P(N = n) = g(\phi(t))$$

for t such that $g(\phi(t))$ is finite, that is, for t such that $\phi(t) < 1/\theta$.

(b) Note that $N < m$ implies that $S < m\ell$ and so

$$1 - P(N \geq m) = P(N < m) \leq P(S < m\ell) = 1 - P(S \geq m\ell)$$

and so

$$P(S \geq m\ell) \leq P(N \geq m).$$

Thus if $P(N \geq m) \leq \epsilon$ then $P(S \geq m\ell) \leq \epsilon$.

(c) For a given value of $t > 1$ with $\phi(t) < 1/\theta$, $P(S \geq M) < \epsilon$ if $M \geq M(t)$ where

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

by setting the upper bound equal for $P(S \geq M)$ to ϵ and solving for $M(t)$. Since (for each t) $P(S \geq M(t)) < \epsilon$, it follows that if $M = \inf_t M(t)$ then $P(S \geq M) < \epsilon$.

(d) Since $\phi(t) = (1+t)^{10}/2^{10}$, we have $\phi(t) < 1/\theta$ if $t < 2\theta^{-1/10} - 1$. Thus for $\theta = 0.9$, we need to evaluate $M(t)$ for $1 < t < 1.021184$. This is done using the following R code:

```
> eps <- 1e-5
> theta <- 0.9
> tt <- c(1000001:1021183)/1000000
> phi <- (1 + tt)^10/2^10
> M <- (log(1-theta) - log(1 - theta*phi) - log(eps))/log(tt)
> min(M)
[1] 724.1235
```

Thus we can define M to be any number greater than 724 (although we could use 724 without problem). In what follows, I will use 725 although (for example) 729 has only 2 and 3 as its prime factors (As a point of comparison, $P(N \geq m) < \epsilon = 10^{-5}$ for $m \geq \lceil \ln(\epsilon)/\ln(\theta) \rceil = 110$ and so the upper bound from part (b) is 1100.)

Since the $\{X_i\}$ are Binomial with parameters 10 and 1/2, we can use the R function `dbinom` to define $p(x)$ for $x = 0, 1, \cdots, 10$ with $p(11) = \cdots = p(724) = 0$.

```
> px <- c(dbinom(c(0:10),10,1/2),rep(0,714))
> pxhat <- fft(px)
> pshat <- (1 - theta)/(1 - theta*pxhat)
> ps <- Re(fft(pshat,inv=T))/725
> plot(c(0:724),ps,type="h",lwd=2,xlim=c(0,150),xlab="s",ylab="P(S = s)")
```

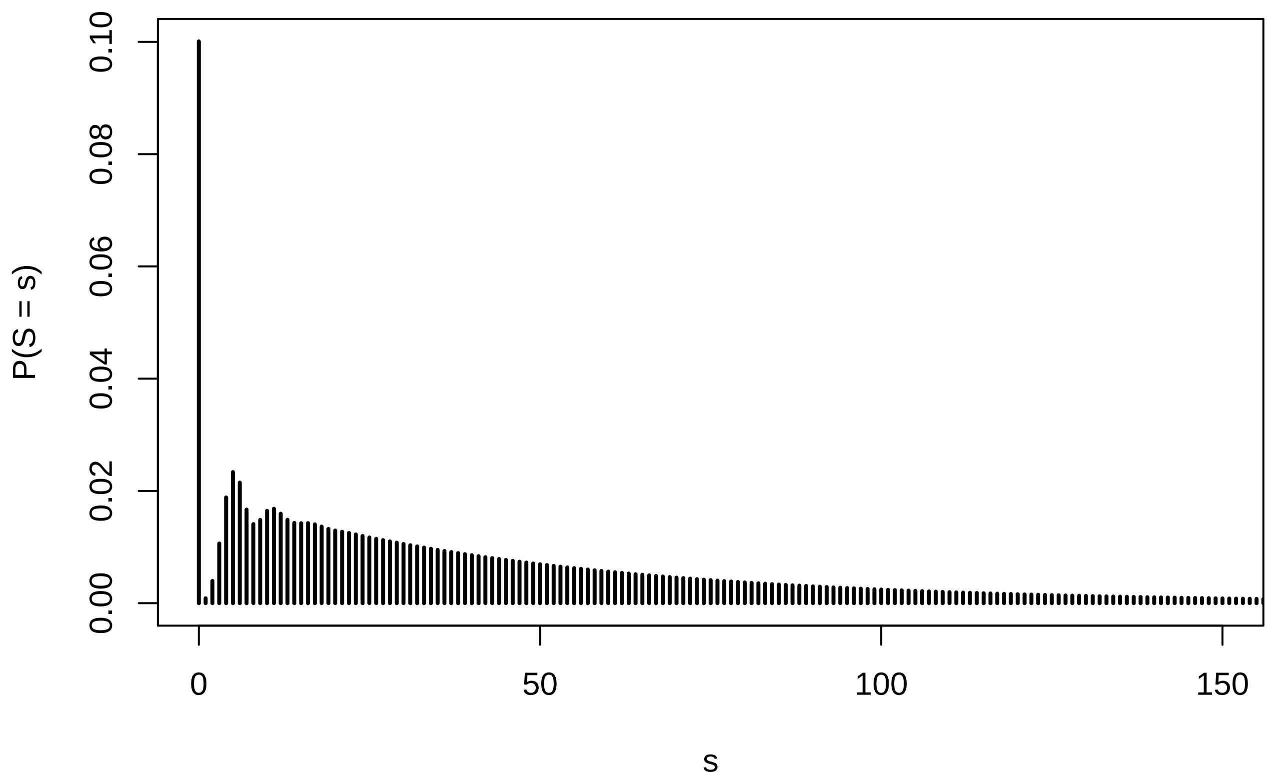

Figure 3: Probability distribution of S

Figure 3 shows $P(S=s)$ for $s=0,\cdots,150$. Note that $P(S=0)$ is relatively large and that the distribution of S has a relatively long tail, that is, $P(S=s)$ tends to 0 quite slowly as s increases.

Supplemental problems

3. (a) Note that

$$\begin{aligned}\sum_{i=1}^{k+1}(x_i-\bar{x}_{k+1})^2 &= \sum_{i=1}^{k}(x_i-\bar{x}_k+\bar{x}_k-\bar{x}_{k+1})^2+(x_{k+1}-\bar{x}_{k+1})^2 \\ &= \sum_{i=1}^{k}(x_i-\bar{x}_k)^2+k(\bar{x}_k-\bar{x}_{k+1})^2+(x_{k+1}-\bar{x}_{k+1})^2\end{aligned}$$

Then we have

$$\begin{aligned}(\bar{x}_k-\bar{x}_{k+1})^2 &= \frac{1}{(k+1)^2}(x_{k+1}-\bar{x}_k)^2 \\ (x_{k+1}-\bar{x}_{k+1})^2 &= \frac{k^2}{(k+1)^2}(x_{k+1}-\bar{x}_k)^2.\end{aligned}$$

Substituting, the desired identity follows.

(b) For any x_0, we have

$$\begin{aligned}\sum_{i=1}^{n}(x_i-\bar{x})^2 &= \sum_{i=1}^{n}(x_i-x_0+x_0-\bar{x})^2\\ &= \sum_{i=1}^{n}(x_i-x_0)^2-2\sum_{i=1}^{n}(x_i-x_0)(\bar{x}-x_0)+n(\bar{x}-x_0)^2\\ &= \sum_{i=1}^{n}(x_i-x_0)^2-n(\bar{x}-x_0)^2\end{aligned}$$

since

$$2\sum_{i=1}^{n}(x_i-x_0)(\bar{x}-x_0)=2n(\bar{x}-x_0)^2.$$

(c) There is no absolutely optimal approach to choosing the value of x_0 to use in the formula in part (b). The idea is to try to choose x_0 (without completely scanning the data) to make $(\bar{x}-x_0)^2$ small — but of course, without going through the data, we cannot compute $\bar{x}$. One feasible approach if n is extremely large is to take a small random sample of size m (where m is very much smaller than n) from the data and set x_0 equal to the mean of these numbers. Other implementations of this approach merely set $x_0 = x_1$, the first datum. Generally speaking, as long as we take x_0 within the range of the data (i.e. between the minimum and maximum), we will avoid catastrophic cancellation in most cases.

4. (a) Note that $P(N \le x) = P(N \le \lfloor x \rfloor)$ since N takes the values $0, 1, 2, \cdots$. Thus

$$P(N \le x) = (1-\theta)\sum_{y=0}^{\lfloor x \rfloor}\theta^y = \frac{(1-\theta)(1-\theta^{\lfloor x \rfloor+1})}{1-\theta} = 1-\theta^{\lfloor x \rfloor+1}$$

and so $F(x) = P(N \le x) = 1-\theta^{n+1}$ for $n \le x < n+1$.

(b) For $x \ge 0$, $F(x)$ takes the values $1-\theta$, $1-\theta^2$, $1-\theta^3, \cdots$ with jumps at $x = 0, 1, 2, \cdots$ Thus

$$F^{-1}(t) = n \ \text{ if } 1-\theta^n < t \le 1-\theta^{n+1}$$

for $0 < t < 1$.

(c) $N = F^{-1}(U)$ where $U \sim \text{Unif}(0,1)$ has a Geometric distribution; from part (b), $N = n$ if $1-\theta^n < U \le 1-\theta^{n+1}$ or, equivalently, $\theta^{n+1} \le 1-U < \theta^n$.

(d) If X has an Exponential distribution with mean $-1/\ln(\theta) > 0$ and $N = \lfloor X \rfloor$ then

$$\begin{aligned}P(N=n) &= P(n \le X < n+1)\\ &= \exp(n\ln(\theta)) - \exp((n+1)\ln(\theta))\\ &= \theta^n-\theta^{n+1}\\ &= (1-\theta)\theta^n.\end{aligned}$$