Homework 3

Stewart Renehan

5/8/2019

a)

We take the derivative of a with respect to the value we are attempting to minimize:

$$\frac{d}{da}[|y - Xa|^2 + \lambda a^T \Omega a] = -2X^T(y - Xa) + 2\lambda\Omega a$$

$$2X^T Xa + 2\lambda\Omega a = 2X^T y$$

$$a(X^T X + \lambda\Omega) = X^T y$$

$$\hat{a} = (X^T X + \lambda\Omega)^{-1} X^T y$$

Plugging this into

$$\hat{y} = X\hat{a}$$

gives:

$$\hat{y} = X(X^T X + \lambda\Omega)^{-1} X^T y$$

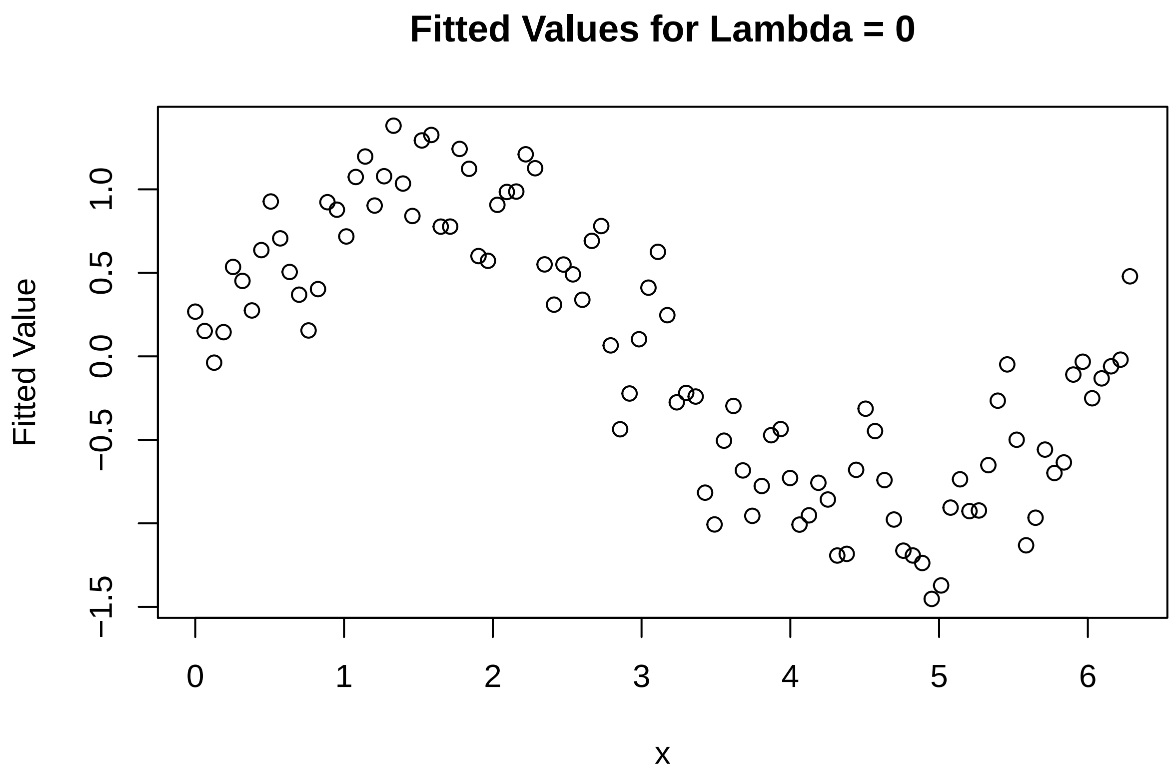
b)

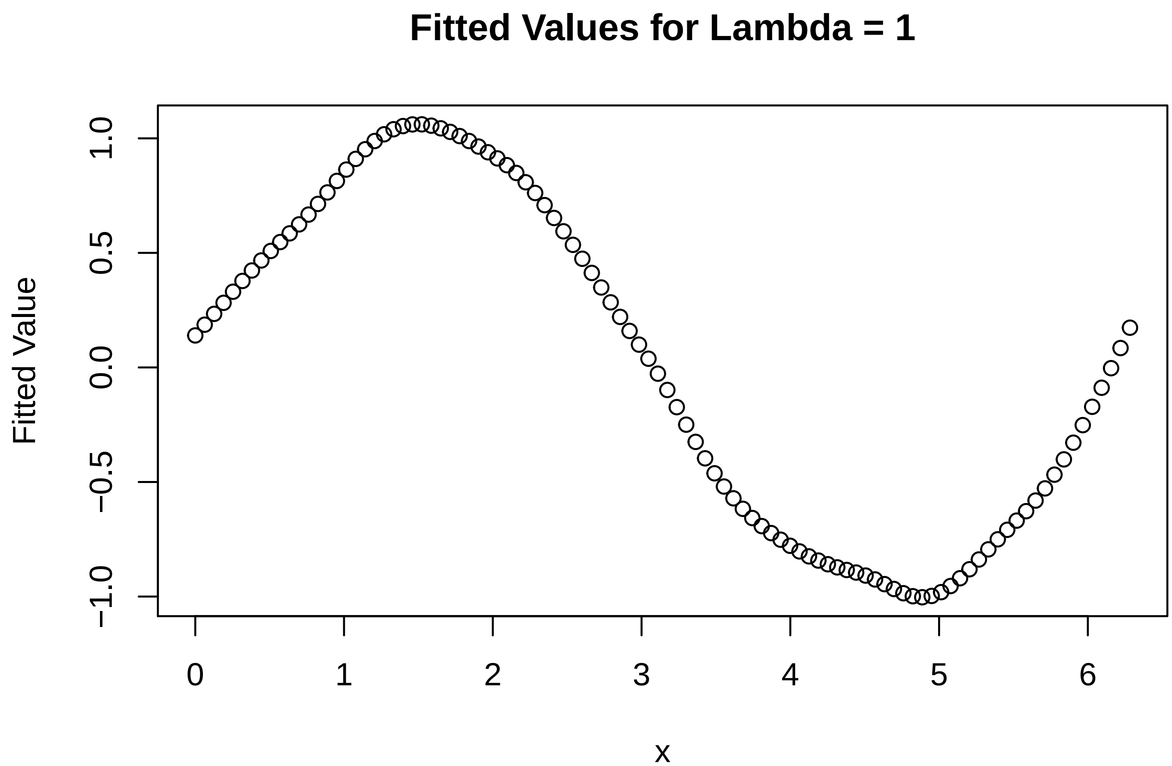
```
source('/Users/stewart/projects/stats/527/homework_3_test_data.R')
require(MASS)
```

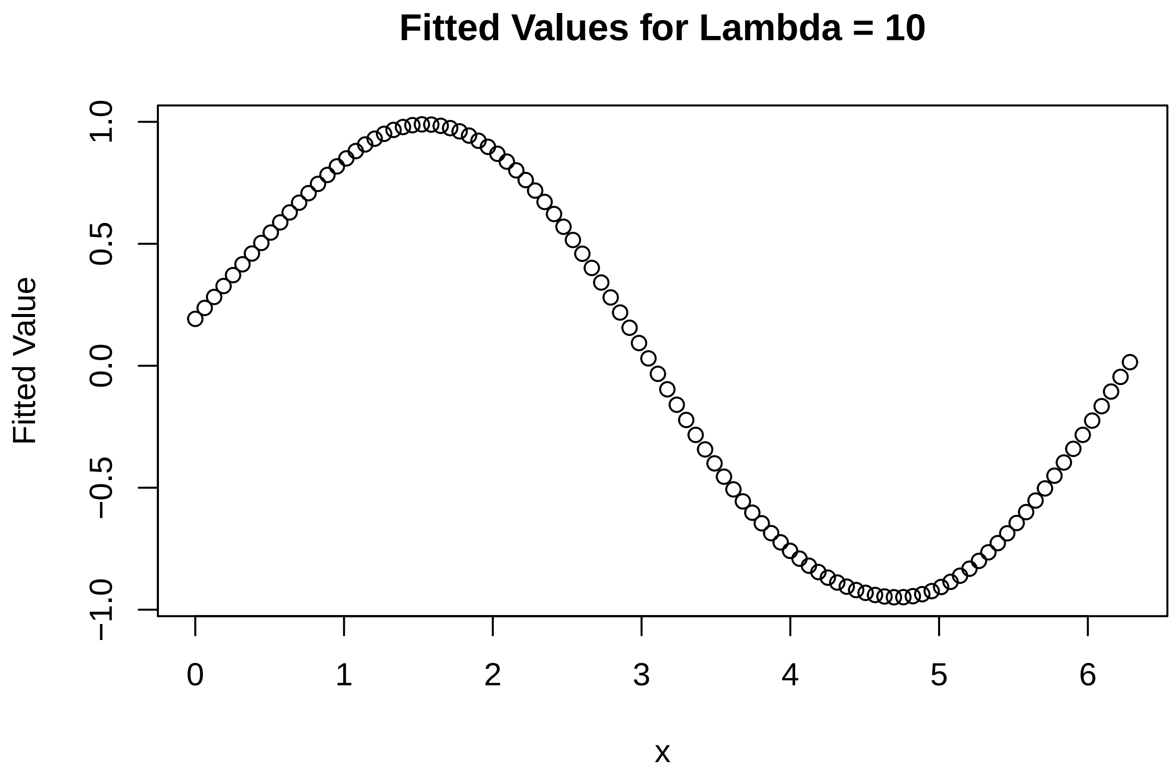
```
## Loading required package: MASS
```

```
truncated.power.design.matrix <- function() {
  design.matrix = matrix(0, ncol=length(x), nrow=length(x));
  design.matrix[, length(x)] = 1;
  for (i in 1:length(x) - 1) {
    design.matrix[i:length(x), i] = (x - x[i])[i:length(x)]
  }
  return(design.matrix);
}
X = truncated.power.design.matrix();
Omega = diag(length(x));
Omega[1, 1] = 0;
Omega[length(x), length(x)] = 0;
get.fitted.values <- function(lambda) {
  return(X %*% ginv((t(X) %*% X + (lambda * Omega))) %*% t(X) %*% y)
}
```

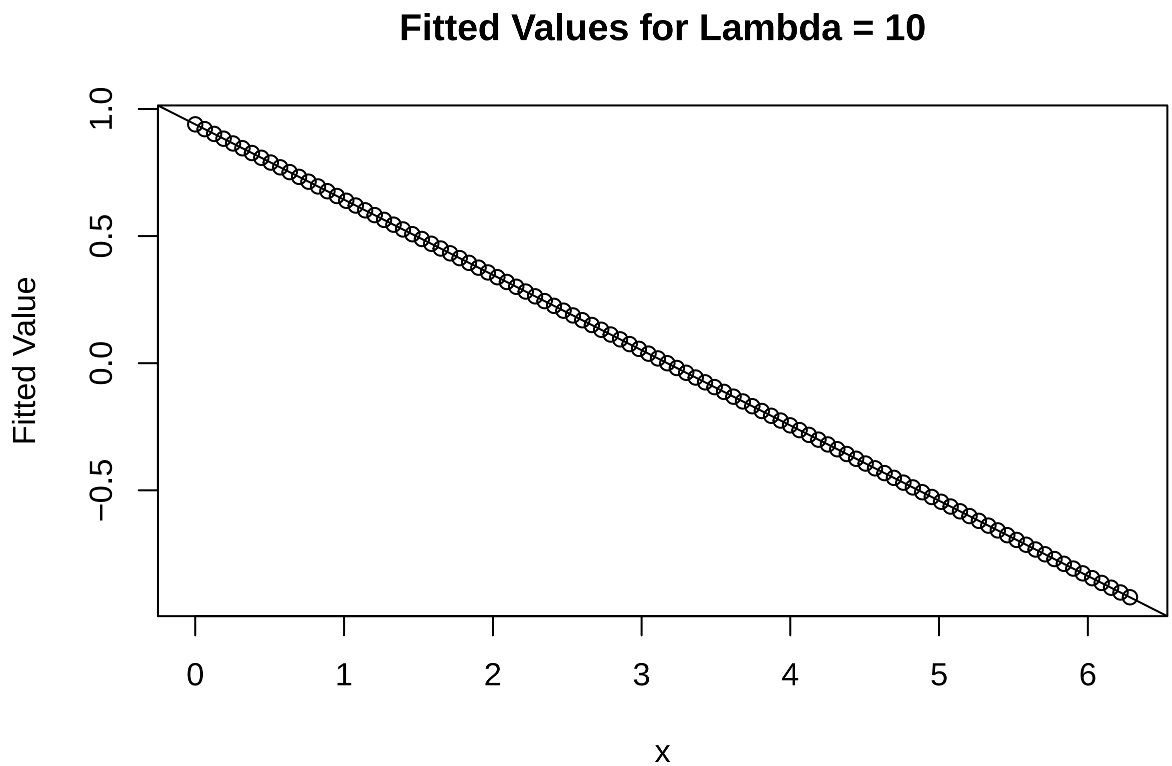
```
for (lambda in c(0, 1, 10)) {
  plot(x, get.fitted.values(lambda), ylab='Fitted Value', xlab='x', main=paste0('Fitted Values for Lambd
}
```




Fitted Values for Lambda = 1
Fitted Value
1.0
0.5
0.0
−0.5
−1.0
0
1
2
3
4
5
6
x




```
plot(x, get.fitted.values(10^6), ylab='Fitted Value', xlab='x', main=paste0('Fitted Values for Lambda =
abline(lm(y~x))
```



The line for $\lambda = 10^6$ is equal to the regression line, as shown above.

c)

Plugging in the value of N to the Reinsch form and multiplying by $I = XX^{-1}$ on the left and $I = X^{-T}X^T$ gives:

$$(I + \lambda X^{-T}\Omega X^{-1})^{-1} = XX^{-1}(I + \lambda X^{-T}\Omega X^{-1})^{-1}X^{-T}X^T$$

$$= X(X^T(I + \lambda X^{-T}\Omega X^{-1})X)^{-1}X^T$$

$$= X(X^TX + \lambda X^TX^{-T}\Omega X^{-1}X)^{-1}X^T$$

On the right side of the inverse, the X^TX^{-T} and $X^{-1}X$ both equal the identity, reducing the equation to:

$$X(X^TX + \lambda\Omega)^{-1}X^T = S_\lambda$$

Which is the same as above.

d)

$$
\begin{aligned}
cov(y, \hat{y}) &= cov(y, S_\lambda y) \\
&= S_\lambda cov(y, y) \\
&= S_\lambda \sigma^2 \\
=> \sum cov(y, \hat{y}) &= \sigma^2 \sum S_\lambda = \sigma^2 tr(S_\lambda) = p\sigma^2
\end{aligned}
$$

e)

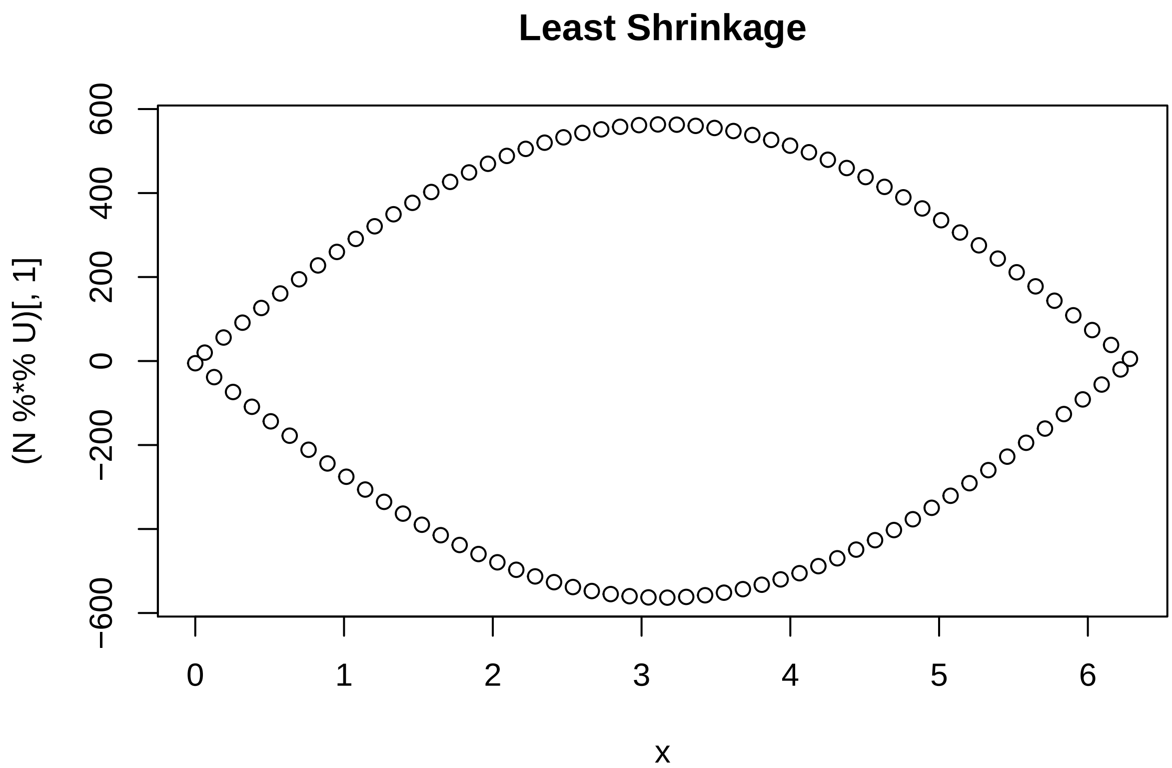
$$
S_\lambda = (I + \lambda N)^{-1} = (I + \lambda UDU^T)^{-1}
$$

$$
\begin{aligned}
&= UU^{-1}(I + \lambda UDU^T)^{-1}U^{-T}U^T \\
&= U(U^T(I + \lambda UDU^T)U)^{-1}U^T
\end{aligned}
$$

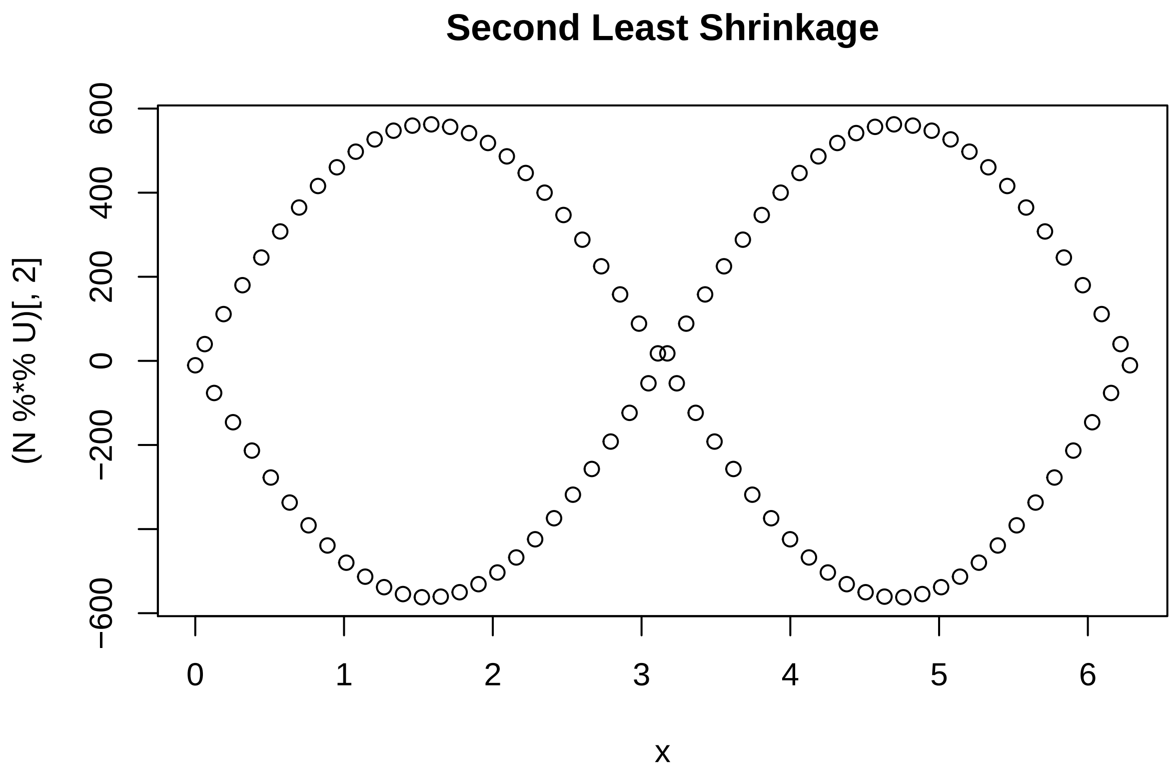
$$
\begin{aligned}
&= U(U^T IU + \lambda U^T UDU^T U)^{-1}U^T \\
&= U(I + \lambda D)^{-1}U^T
\end{aligned}
$$

f)

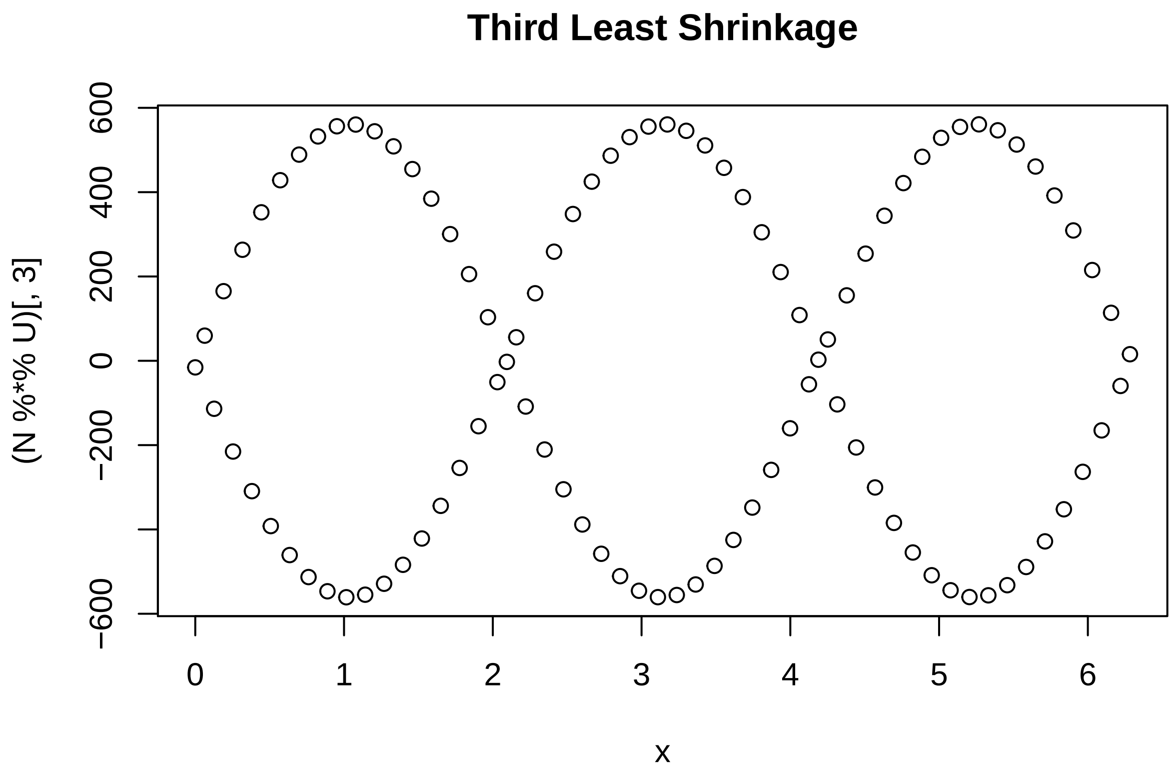
```
X = truncated.power.design.matrix();
Omega = diag(length(x));
Omega[1, 1] = 0;
Omega[length(x), length(x)] = 0;
N = ginv(t(X)) %*% Omega %*% ginv(X);
U = eigen(N)$vectors;
plot(x, (N %*% U)[,1], main='Least Shrinkage')
```



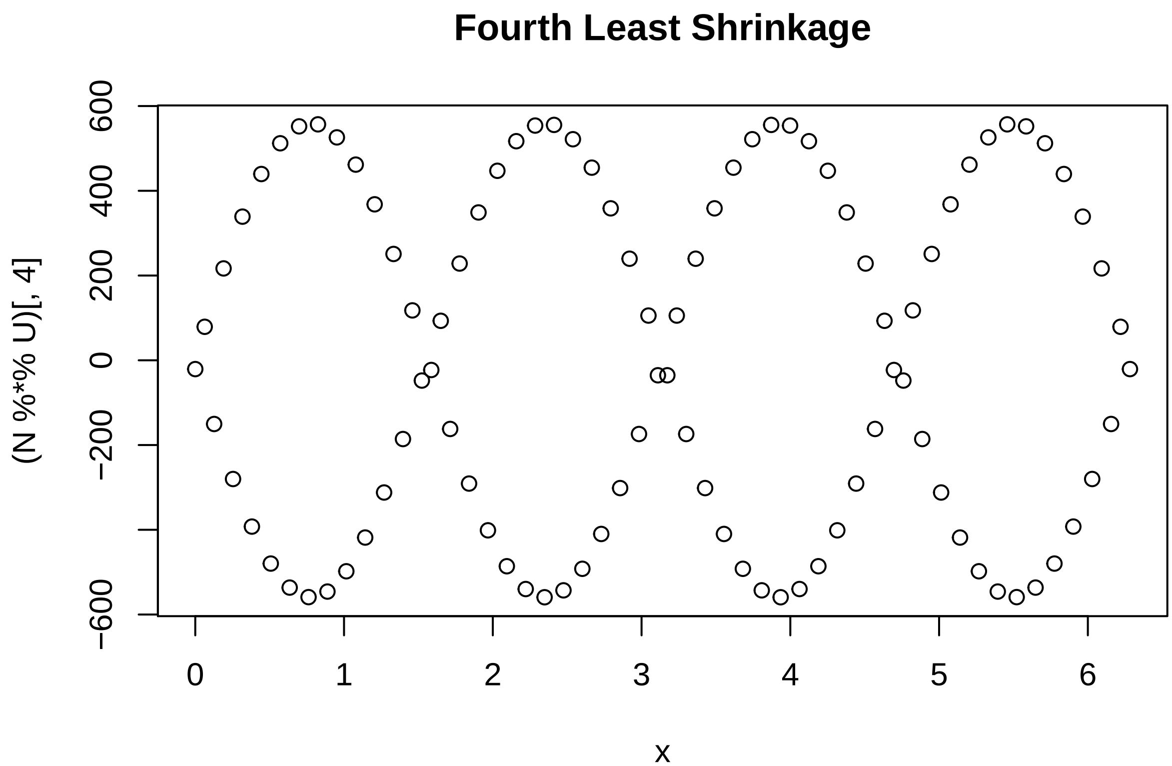

```
plot(x, (N %*% U)[,2], main='Second Least Shrinkage')
```



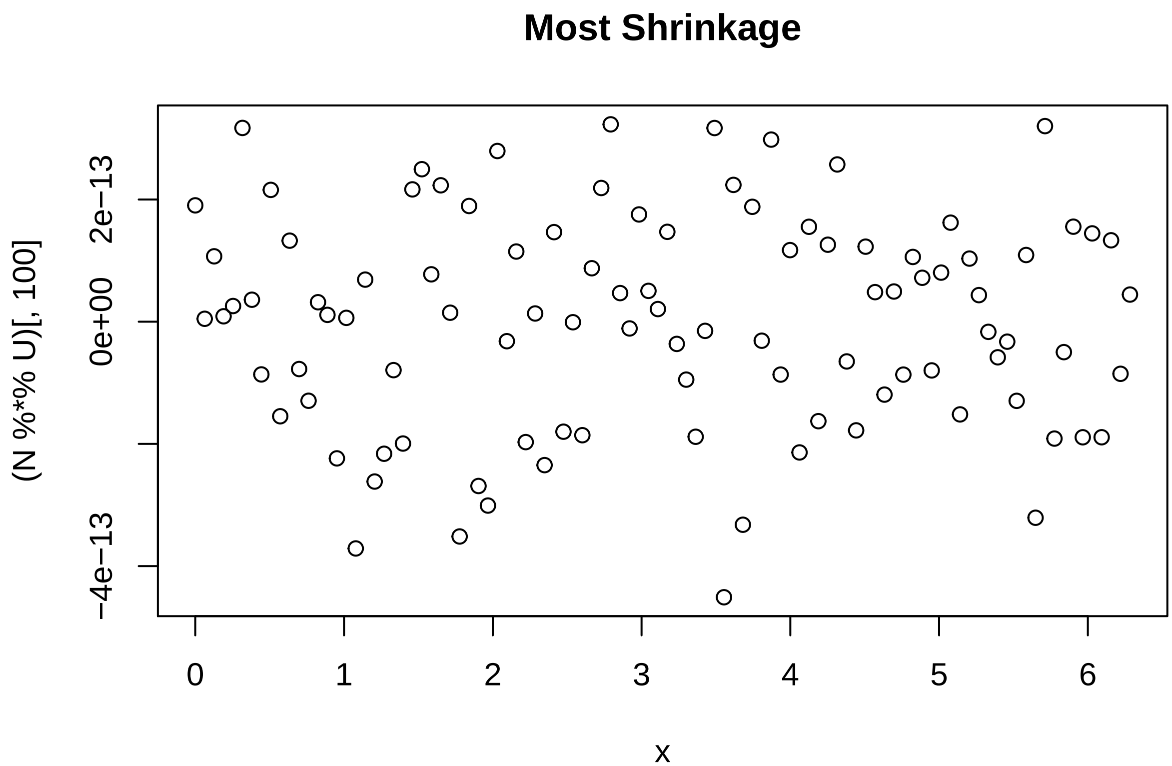

```
plot(x, (N %*% U)[,3], main='Third Least Shrinkage')
```



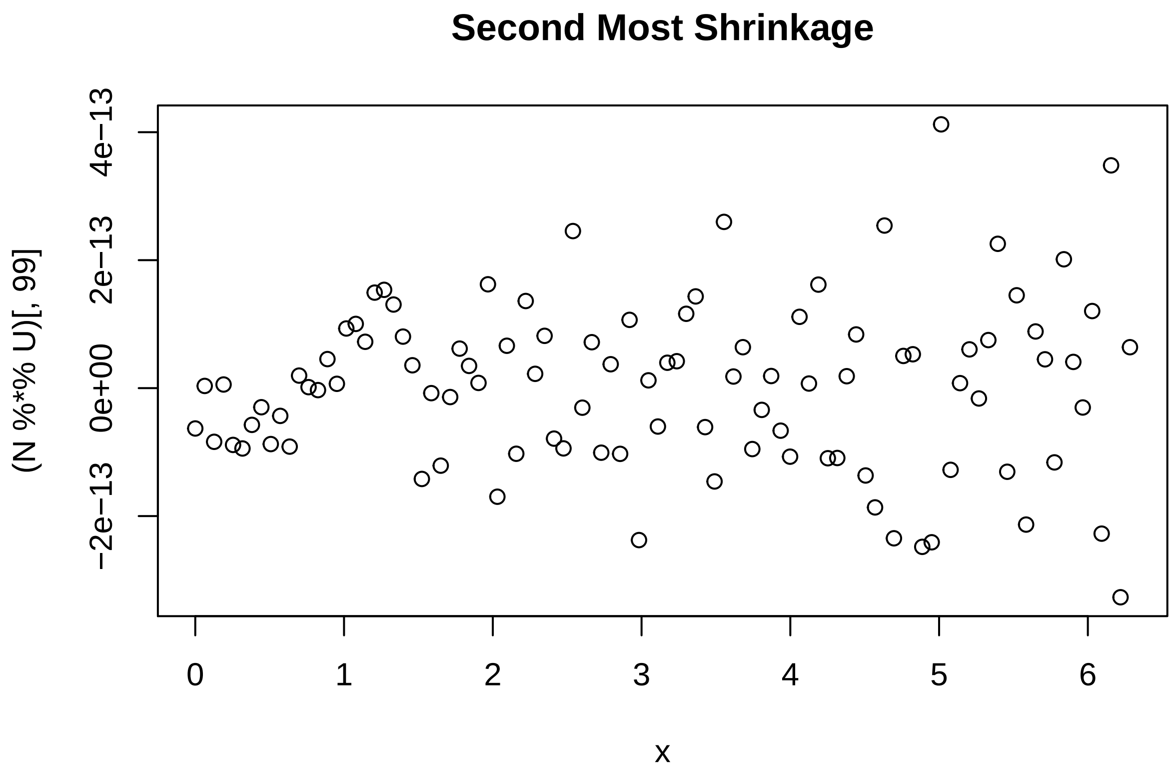

```
plot(x, (N %*% U)[,4], main='Fourth Least Shrinkage')
```



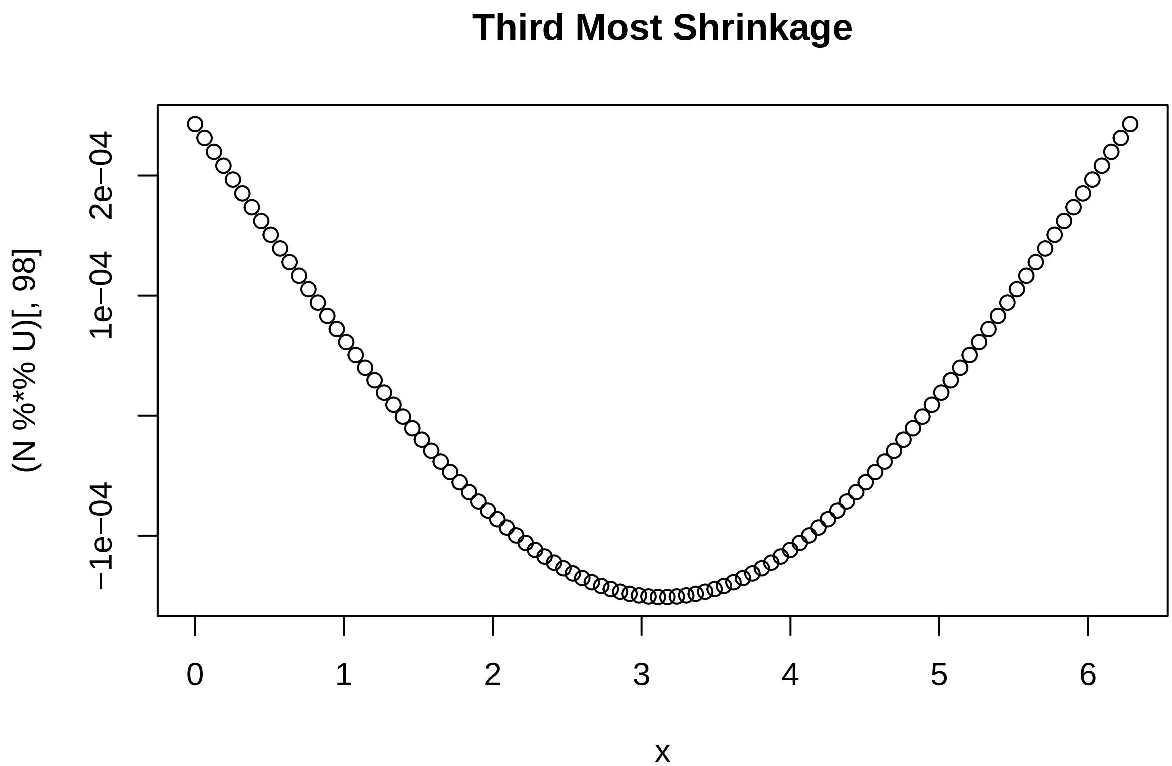

```r
plot(x, (N %*% U)[,100], main='Most Shrinkage')
```



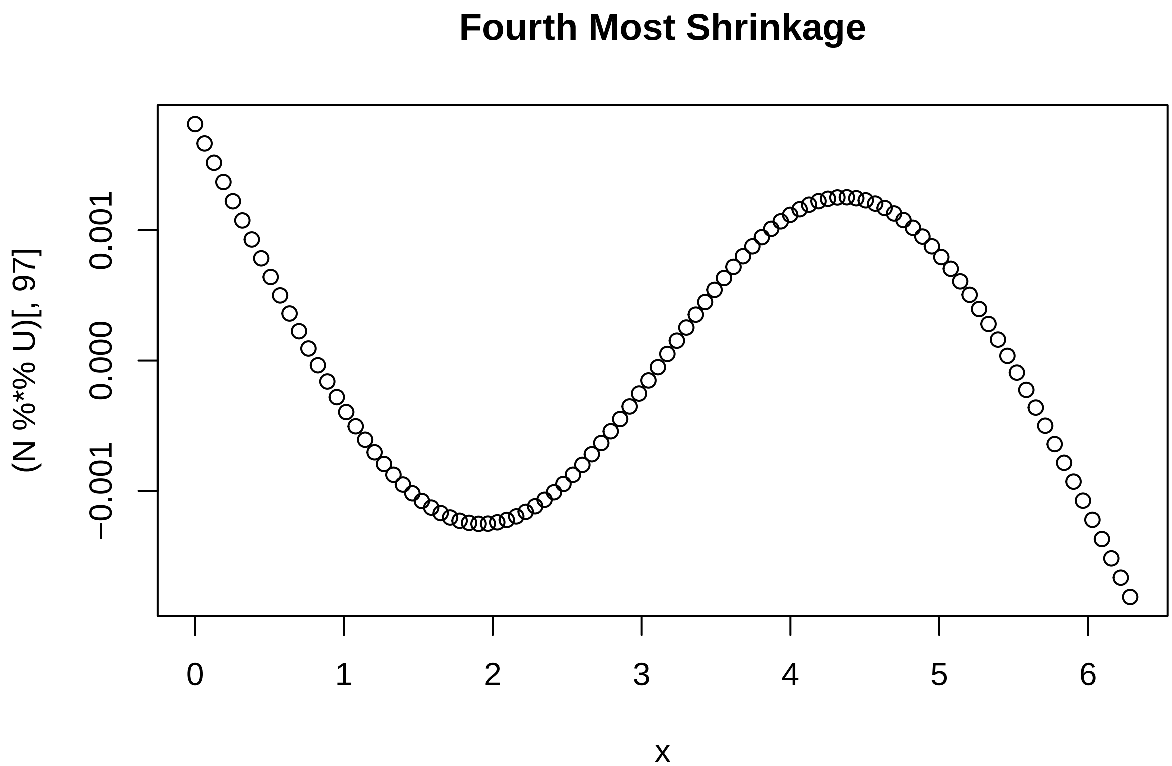

```r
plot(x, (N %*% U)[,99], main='Second Most Shrinkage')
```




```r
plot(x, (N %*% U)[,98], main='Third Most Shrinkage')
```




```
plot(x, (N %*% U)[,97], main='Fourth Most Shrinkage')
```



We see that the eigenvectors correspond to various frequencies of standing waves. So, we are essentially performing a fourier transform of the data onto various sine wave frequencies.

g)

The trace is invariant under permutations, so

$$tr(U^T(I+\lambda D)^{-1}U) = tr(U^TU(I+\lambda D)^{-1}) = tr(I(I+\lambda D)^{-1}))$$

$$= tr(I+\lambda D)^{-1}) = \sum_i (I+\lambda D_{ii})^{-1}$$