4. Show the equivalence of the two forms of the optimization used to produce the fitted ridge regression parameter. (That is, show that there is a $t(\lambda)$ such that $\beta_\lambda = \beta_0(\lambda)$ and a $\lambda(t)$ such that $\hat{\beta}_t = \beta_\lambda(t)$.)

Hint sent out by e-mail:

You may take as given that there is an explicit formula for $\beta_\lambda$ that is continuous in $\lambda$ and thus has error sum of squares varying continuously from $\left( Y - \hat{Y}_{OLS} \right)' \left( Y - \hat{Y}_{OLS} \right)$ to $Y'Y$ as $\lambda$ runs from 0 to $\infty$. Suppose $\hat{\beta}_\lambda$ solves the constrained optimization problem. There is a $\hat{\beta}_\lambda$ with same error sum of squares. Consider its squared norm and that of $\beta_t$. How do they compare?

Additional related observation:

Notice that it is also clear from the representation of $\hat{Y}_\lambda$ in terms of the orthonormal vectors $u_j$ appearing in the singular value decomposition of $X$ that $\left( Y - \hat{Y}_\lambda \right)' \left( Y - \hat{Y}_\lambda \right)$ is strictly increasing in $\lambda$.

Solution:

As in the hint, first suppose that $\hat{\beta}_t$ solves the constrained optimization problem. There is a $\hat{\beta}_\lambda$ with say, $\lambda = \lambda^*$, having the same error sum of squares. Consider its squared norm and that of $\beta_t$. The possibility that $\left\| \hat{\beta}_t \right\|^2 < \left\| \beta_t \right\|^2$ would violate the definition of $\hat{\beta}_\lambda$ as an (unconstrained) optimizer of the penalized error sum of squares. On the other hand, if $\left\| \hat{\beta}_\lambda \right\|^2 < \left\| \beta_t \right\|^2$ continuity of $\hat{\beta}_\lambda$ and strict monotonicity of $\left( Y - \hat{Y}_\lambda \right)' \left( Y - \hat{Y}_\lambda \right)$ in $\lambda$ implies that there is a $\lambda < \lambda^*$ with $\left\| \hat{\beta}_\lambda \right\|^2 < \left\| \hat{\beta}_t \right\|^2$ but $\left( Y - \hat{Y}_\lambda \right)' \left( Y - \hat{Y}_\lambda \right) < \left( Y - \hat{Y}_\lambda \right)' \left( Y - \hat{Y}_\lambda \right)$. But this would violate the definition of $\hat{\beta}_t$ as a constrained optimizer of the error sum of squares. So $\left\| \hat{\beta}_t \right\|^2 = \left\| \beta_t \right\|^2$ and we may take $\lambda(\lambda) = \lambda^*$.

On the other hand, suppose that $\hat{\beta}_\lambda$ solves the unconstrained optimization problem for the penalized error sum of squares. I claim that it also solves the constrained optimization problem...
for $t = \|\hat{\beta}_{\lambda}^{\text{ridge}}\|^2$. If not, there is another $\hat{\beta}$ with $\|\hat{\beta}\|^2 \leq t$ with

$$(Y - X\hat{\beta})'(Y - X\hat{\beta}) < (Y - \hat{Y}_\lambda)'(Y - \hat{Y}_\lambda).$$

But that would mean that $\hat{\beta}_{\lambda}^{\text{ridge}}$ cannot be an unconstrained optimizer.
4. Show the equivalence of the two forms of the optimization used to produce the fitted ridge regression parameter. (That is, show that there is a \( t(\lambda) \) such that \( \hat{\beta}_{\lambda}^\text{ridge} = \hat{\beta}_{t(\lambda)}^\text{ridge} \) and a \( \lambda(t) \) such that \( \hat{\beta}_t^\text{ridge} = \hat{\beta}_{\lambda(t)}^\text{ridge} \).)

(5 pts) Need to show: There is a \( t(\lambda) \) such that \( \hat{\beta}_{\lambda}^\text{ridge} = \hat{\beta}_{t(\lambda)}^\text{ridge} \).

From the definition of the Ridge regression, \( \hat{\beta}_{\lambda}^\text{ridge} = (X'X + \lambda I)^{-1}X'Y \) is a solution of

\[
\arg\min_{\beta \in \mathbb{R}^p} (Y - X\beta)'(Y - X\beta) + \lambda \beta'\beta. \tag{1}
\]

Let \( t(\lambda) = \|\hat{\beta}_{\lambda}^\text{ridge}\|^2 \) and \( \hat{\beta}_{t(\lambda)} \) is one solution of the constrained problem

\[
\arg\min_{\beta \text{ with } \|\beta\|^2 \leq t(\lambda)} (Y - X\beta)'(Y - X\beta) \tag{2}
\]

which implies

\[
\|\hat{\beta}_{t(\lambda)}\|^2 \leq \|\hat{\beta}_{\lambda}^\text{ridge}\|^2 \tag{3}
\]

and

\[
(Y - X\hat{\beta}_{t(\lambda)})(Y - X\hat{\beta}_{t(\lambda)}) < (Y - X\hat{\beta}_{\lambda}^\text{ridge})(Y - X\hat{\beta}_{\lambda}^\text{ridge}). \tag{4}
\]

Note that when the equality holds in the Eq. (4) then \( \hat{\beta}_{\lambda}^\text{ridge} \) is one solution of the Eq. (2), and we can set \( \hat{\beta}_{t(\lambda)} \equiv \hat{\beta}_{\lambda}^\text{ridge} \) and finish the proof.

Otherwise, suppose \( \hat{\beta}_{t(\lambda)} \neq \hat{\beta}_{\lambda}^\text{ridge} \) then we have

\[
(Y - X\hat{\beta}_{t(\lambda)})(Y - X\hat{\beta}_{t(\lambda)}) + \lambda \hat{\beta}_{t(\lambda)}'\hat{\beta}_{t(\lambda)} < (Y - X\hat{\beta}_{\lambda}^\text{ridge})(Y - X\hat{\beta}_{\lambda}^\text{ridge}) + \lambda \hat{\beta}_{t(\lambda)}'\hat{\beta}_{t(\lambda)} \quad \text{by (4)}
\]

\[
\leq (Y - X\hat{\beta}_{\lambda}^\text{ridge})(Y - X\hat{\beta}_{\lambda}^\text{ridge}) + \lambda \hat{\beta}_{\lambda}^\text{ridge}'\hat{\beta}_{\lambda}^\text{ridge} \quad \text{by (3)}
\]

which implies \( \hat{\beta}_{t(\lambda)} \) is a solution of the Eq. (1) rather than \( (X'X + \lambda I)^{-1}X'Y \) which is a contradiction. Therefore, we can set \( \hat{\beta}_{t(\lambda)} \equiv \hat{\beta}_{\lambda}^\text{ridge} \) and finish the proof.

(5 pts) Need to who: There is a \( \lambda(t) \) such that \( \hat{\beta}_t^\text{ridge} = \hat{\beta}_{\lambda(t)}^\text{ridge} \).
\[
\hat{\beta}^{\text{ridge}}_t = \arg\min_{\beta} (Y - X\beta)'(Y - X\beta) \quad \text{with} \quad \|\beta\|^2 \leq t
\]  

(5)

i.e. \( \min_{\beta \in \mathbb{R}^p} (Y - X\beta)'(Y - X\beta) \) subject to \( \|\beta\|^2 \leq t \)

First, for any fixed \( t \), the Lagrange multiplier \( \lambda \) is dependent on \( t \), we have an unconstrained optimization problem providing the same solution as \( \hat{\beta}^{\text{ridge}}_t \). That is

\[
\min_{\beta \in \mathbb{R}^p} (Y - X\beta)'(Y - X\beta) + \lambda(t)(t - \beta'\beta)
\]

let \( g(\beta, t) \equiv (Y - X\beta)'(Y - X\beta) + \lambda(t)(t - \beta'\beta) \)

\[\Rightarrow\] solve \( \begin{cases} 
\frac{\partial g}{\partial \beta} = -2X'Y + 2X'X - 2\lambda(t)\beta = 0 \\ 
\frac{\partial g}{\partial \lambda(t)} = t - \beta'\beta = 0
\end{cases} \)

yield \( \begin{cases} 
\beta = (X'X + \lambda(t)I)^{-1}X'Y \\ 
t = \beta'\beta
\end{cases} \)

\[\Rightarrow\] \( \hat{\beta}_{\lambda(t)} = (X'X + \lambda(t)I)^{-1}X'Y \)

\[ t = \|\hat{\beta}_{\lambda(t)}\|^2 \]

Let \( \hat{\beta}^{\text{ridge}}_t \equiv \hat{\beta}_{\lambda(t)} \) and \( \lambda(t) \) exists. Also, \( \hat{\beta}_{\lambda(t)} \) is in the form of solution of

\[
\arg\min_{\beta \in \mathbb{R}^p} (Y - X\beta)'(Y - X\beta) + \lambda(t)\beta'\beta
\]  

(6)

By the definition of the Ridge regression, we can set \( \hat{\beta}^{\text{ridge}}_{\lambda(t)} = \hat{\beta}_{\lambda(t)} \).

Second, however, the remained thing needs to verify is that \( \hat{\beta}_{\lambda(t)} \) is the optimal solution of the Eq.(6) for all other \( \beta \)'s where \( \beta'\beta < t \). Suppose there exists \( \beta_0 \) where \( \beta_0'\beta_0 < t \) and

\[
(Y - X\beta_0)'(Y - X\beta_0) < (Y - X\beta_{\lambda(t)})'(Y - X\beta_{\lambda(t)})
\]  

(7)

Again, when the equality holds in the Eq.(7), then \( \beta_0 \) is one solution of the Eq.(6), and we can set \( \hat{\beta}_{\lambda(t)} = \beta_0 \) and finish the proof. Otherwise,

\[
(Y - X\beta_0)'(Y - X\beta_0) + \lambda(t)\beta_0'\beta_0 < (Y - X\beta_{\lambda(t)})'(Y - X\beta_{\lambda(t)}) + \lambda(t)\hat{\beta}_{\lambda(t)}'\hat{\beta}_{\lambda(t)} \quad \text{since} \quad \beta_0'\beta_0 < t = \hat{\beta}_{\lambda(t)}'\hat{\beta}_{\lambda(t)} \\
< (Y - X\beta_{\lambda(t)})'(Y - X\beta_{\lambda(t)}) + \lambda(t)\hat{\beta}_{\lambda(t)}'\hat{\beta}_{\lambda(t)} \quad \text{by} \quad (7)
\]
which implies $\beta_0$ is a solution of the Eq. (6) rather than $\beta_{\lambda(t)}$ which is a contradiction. Hence, there exists a $\lambda(t)$ such that $\beta_t^{\text{ridge}} = \beta_{\lambda(t)}$ and finish the proof.

5. Consider the linear space of functions on $[0, 1]$ of the form

$$f(t) = a + bt + ct^2 + dt^3$$

Equip this space with the inner-product $\langle f, g \rangle \equiv \int_0^1 f(t)g(t)dt$ and norm $\|f\| = \langle f, f \rangle^{1/2}$ (to create a small Hilbert space). Use the Gram-Schmidt process to orthogonalize the set of functions $\{1, t, t^2, t^3\}$ and produce an orthonormal basis for the space.

(10 pts) For $t \in [0, 1]$, I rewrite the vector $x_i(t) \equiv t^{(i-1)}$ as a function of $t$ for $i = 1, \ldots, 4$, and $z_i(t)$ are the orthogonal vectors to be computed by Gram-Schmidt process in the followings,

\[
\begin{align*}
z_1(t) &= 1 \\
z_2(t) &= x_2(t) - \frac{\langle x_2(t), z_1(t) \rangle}{\langle z_1(t), z_1(t) \rangle} z_1(t) \\
&= t - \frac{1}{2} \\
z_3(t) &= x_3(t) - \frac{\langle x_3(t), z_1(t) \rangle}{\langle z_1(t), z_1(t) \rangle} z_1(t) - \frac{\langle x_3(t), z_2(t) \rangle}{\langle z_2(t), z_2(t) \rangle} z_2(t) \\
&= t^2 - t + \frac{1}{6} \\
z_4(t) &= x_4(t) - \frac{\langle x_4(t), z_1(t) \rangle}{\langle z_1(t), z_1(t) \rangle} z_1(t) - \frac{\langle x_4(t), z_2(t) \rangle}{\langle z_2(t), z_2(t) \rangle} z_2(t) - \frac{\langle x_4(t), z_3(t) \rangle}{\langle z_3(t), z_3(t) \rangle} z_3(t) \\
&= t^3 - \frac{3}{2} t^2 + \frac{3}{5} t - \frac{1}{20}.
\end{align*}
\]

(5 pts) The normalizing constants are $\|z_1(t)\| = 1$, $\|z_2(t)\| = \sqrt{\frac{1}{12}}$, $\|z_3(t)\| = \sqrt{\frac{1}{180}}$, and $\|z_3(t)\| = \sqrt{\frac{1}{2800}}$. So, the orthonormal basis for the space of $f(t)$ on $[0, 1]$ is

\[
\left\{ 1, \sqrt{12} \left( t - \frac{1}{2} \right), \sqrt{180} \left( t^2 - t + \frac{1}{6} \right), \sqrt{2800} \left( t^3 - \frac{3}{2} t^2 + \frac{3}{5} t - \frac{1}{20} \right) \right\}.
\]
6. Consider the matrix
\[ X = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 1 \\ 1 & 2 & 1 \\ 2 & 2 & 1 \end{bmatrix} \]

a) Use R and find the QR and singular value decompositions of \( X \). What are the two corresponding bases for \( C(X) \)?

(4 pts) From QR, the basis of \( C(X) \) is
\[
\begin{bmatrix}
-0.3162278 \\
-0.6324555 \\
-0.3162278 \\
-0.6324555
\end{bmatrix}, \begin{bmatrix}
0.07254763 \\
-0.58038100 \\
0.79802388 \\
0.14509525
\end{bmatrix}, \begin{bmatrix}
0.8029551 \\
0.1147079 \\
0.1147079 \\
-0.5735393
\end{bmatrix}
\]

From SVD, the basis of \( C(X) \) is
\[
\begin{bmatrix}
-0.3558795 \\
-0.4908122 \\
-0.4908122 \\
-0.6257449
\end{bmatrix}, \begin{bmatrix}
1.532113 \times 10^{-16} \\
-7.071068 \times 10^{-1} \\
7.071068 \times 10^{-1} \\
1.206853 \times 10^{-17}
\end{bmatrix}, \begin{bmatrix}
0.7895256 \\
0.09541184 \\
0.09541184 \\
-0.59870138
\end{bmatrix}
\]

R Output
\[
R> X <- \text{matrix(c(1, 2, 1, 2, 1, 1, 2, 1, 1, 1, 1), nrow = 4, ncol = 3)}
\]
\[
R> \text{qr(X)$qr}
\]
\[
\begin{bmatrix}
[,1] & [,2] & [,3]
[1,] -3.1622777 -2.8460499 -1.8973666 \\
[2,] 0.6324555 1.3784049 0.4352858 \\
[3,] 0.3162278 -0.7805941 0.4588315 \\
[4,] 0.6324555 -0.071068 -0.59870138
\end{bmatrix}
\]

R Output
\[
R> \text{X.SVD <- svd(X)}
\]
\[
\$d
[1] 4.7775257 1.0000000 0.4186267
\]

\[
\$u
\begin{bmatrix}
[,1] & [,2] & [,3]
[1,] -0.3558795 1.532113e-16 0.78952506 \\
[2,] -0.4908122 -7.071068e-01 0.09541184 \\
[3,] -0.4908122 7.071068e-01 0.09541184 \\
[4,] -0.6257449 1.206853e-17 -0.59870138
\end{bmatrix}
\]

\[
\$v
\begin{bmatrix}
[,1] & [,2] & [,3]
[1,] -0.6446445 -7.071068e-01 -0.2905744 \\
[2,] -0.6446445 7.071068e-01 -0.2905744 \\
[3,] -0.4109342 2.020953e-16 0.9116650
\end{bmatrix}
\]
b) Use the singular value decomposition of $X$ to find the eigen (spectral) decomposition of $X'X$ (what are the eigenvalues and eigenvectors?)

(2 pts) The three eigen values of $X'X$ are $22.8247517$, $1.0000000$, $0.1752483$ which are square of the singular values, and the corresponding eigen vectors are

$$
\begin{bmatrix}
-0.6446445 \\
-0.6446445 \\
-0.4109342
\end{bmatrix}, \quad \begin{bmatrix}
-7.071068 \times 10^{-1} \\
7.071068 \times 10^{-1} \\
2.020953 \times 10^{-16}
\end{bmatrix}, \quad \begin{bmatrix}
-0.2905744 \\
-0.2905744 \\
0.9116650
\end{bmatrix}.
$$

R Output

```
> X.SVD$d^2
[1] 22.8247517 1.0000000 0.1752483
```

c) Find the best $\text{rank} = 1$ and $\text{rank} = 2$ approximations to $X$.

(4 pts) By taking the first one and two largest singular values and corresponding vectors to reconstruct the best approximations to $X$, the followings show the results.

R Output

```
> (rank.1 <- matrix(X.SVD$u[, 1], ncol = 1) %*% diag(X.SVD$d[1], 1, 1) %*% t(X.SVD$v[, 1]))
      [,1]      [,2]      [,3]
[1,] 1.096040 1.096040 0.6986799
[2,] 1.511606 1.511606 0.9635863
[3,] 1.511606 1.511606 0.9635863
[4,] 1.927173 1.927173 1.2284928
> (rank.2 <- X.SVD$u[, 1:2] %*% diag(X.SVD$d[1:2]) %*% t(X.SVD$v[, 1:2]))
      [,1]      [,2]      [,3]
[1,] 1.096040 1.096040 0.6986799
[2,] 2.011606 2.011606 0.9635863
[3,] 1.011606 2.011606 0.9635863
[4,] 1.927173 1.927173 1.2284928
```
7. Consider "data augmentation" methods of penalized least squares fitting.

a) Augment a centered $\mathbf{X}$ matrix with $p$ new rows $\sqrt{\lambda} \mathbf{I}$ and $\mathbf{Y}$ by adding $p$ new entries 0. Argue that OLS fitting with the augmented data set returns $\hat{\beta}_\text{ridge}$ as a fitted coefficient vector.

(5 pts) Let

\[
\mathbf{A} = \begin{bmatrix} \mathbf{X} & \sqrt{\lambda} \mathbf{I} \end{bmatrix}^{(N+p) \times p} \quad \text{and} \quad \mathbf{W} = \begin{bmatrix} \mathbf{Y} \\ \mathbf{0} \end{bmatrix}^{p \times 1}
\]

Then,

\[
\hat{\beta}_\text{ols} = (\mathbf{A}'\mathbf{A})^{-1}\mathbf{A}'\mathbf{W} = \left( \begin{bmatrix} \mathbf{X}' & \sqrt{\lambda} \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \sqrt{\lambda} \mathbf{I} \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{X}' & \sqrt{\lambda} \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{Y}' \\ \mathbf{0} \end{bmatrix} = (\mathbf{X}'\mathbf{X} + \lambda \mathbf{I})^{-1}\mathbf{X}'\mathbf{Y} = \hat{\beta}_\text{ridge}.
\]

b) Show how the "elastic net" fitted coefficient vector $\hat{\beta}_{\lambda_1,\lambda_2}^{\text{elastic net}}$ could be found using lasso software and an appropriate augmented data set.

(5 pts) Following the notation in the part a) and from the lecture outline, the elastic net coefficient vector can be obtained by augmenting data $\sqrt{\lambda_2} \mathbf{I}$ and $\mathbf{0}$ to original $\mathbf{X}$ and $\mathbf{Y}$, respectively, then inputing the augmented data into lasso software for any given $\lambda_1 > 0$ and $\lambda_2 > 0$. Then,

\[
\hat{\beta}_{\lambda_1,\lambda_2}^{\text{elastic net}} = \arg\min_{\beta \in \mathbb{R}^p} (\mathbf{Y} - \mathbf{X}\beta)'(\mathbf{Y} - \mathbf{X}\beta) + \lambda_1 \sum_{j=1}^p |\beta_j| + \lambda_2 \sum_{j=1}^p \beta_j^2
\]

\[
= \arg\min_{\beta \in \mathbb{R}^p} \left[ (\mathbf{Y} - \mathbf{X}\beta)'(\mathbf{Y} - \mathbf{X}\beta) + \lambda_2 \sum_{j=1}^p \beta_j^2 \right] + \lambda_1 \sum_{j=1}^p |\beta_j|
\]

\[
= \arg\min_{\beta \in \mathbb{R}^p} \left[ (\mathbf{W} - \mathbf{A}\beta)'(\mathbf{W} - \mathbf{A}\beta) + \lambda_1 \sum_{j=1}^p |\beta_j| \right]
\]

\[
= \hat{\beta}_{\lambda_1 \mathbf{W}, \mathbf{A}}^{\text{lasso}}.
\]
8. Beginning in Section 5.6 Izenman’s book uses an example where PET yarn density is to be predicted from its NIR spectrum. This is a problem where $N = 28$ data vectors $x_j$ of length $p = 268$ are used to predict the corresponding outputs $y_i$. Izenman points out that the yarn data are to be found in the pls package in R. Get those data and make sure that all inputs are standardized and the output is centered.

a) Using the pls package, find the 1, 2, 3, and 4 component PCR and PLS $\hat{\beta}$ vectors. (Output is too long to display and skipped.)

Note that the function var() in R uses $N - 1$ by default to yield unbiased estimators. If you manually adjust by $\sqrt{(N - 1)/N}$ as state in the lecture outline, then the following answers will be slightly different. Unfortunately, I use $N - 1$ to obtain the best subsets in terms of $R^2$.

(2 pts) The following code center $Y$ and standard $X$, then provides the coefficient vectors of given components of PCR and PLS.

```
library(pls)
data(yarn)
y <- yarn$density - mean(yarn$density)
X <- yarn$NIR
X <- t((t(X) - colMeans(X)) / sqrt(apply(X, 2, var)))
yarn.pcr <- pcr(y ~ X, ncomp = 4)
yarn.pcr$coefficients[,1]
yarn.pcr$coefficients[,2]
yarn.pcr$coefficients[,3]
yarn.pcr$coefficients[,4]
yarn.plsr <- plsr(y ~ X, ncomp = 4)
yarn.plsr$coefficients[,1]
yarn.plsr$coefficients[,2]
yarn.plsr$coefficients[,3]
yarn.plsr$coefficients[,4]
```

b) Find the singular values for the matrix $X$ and use them to plot the function $df(\lambda)$ for ridge regression. Identify values of $\lambda$ corresponding to effective degrees of freedom 1, 2, 3, and 4. Find corresponding ridge $\hat{\beta}$ vector.

(5 pts) The following code gives the singular values (first five are 60.81398, 47.27684, 34.52588, 8.452855, and 4.027232), the plot for the effective degrees of freedom $df(\lambda)$ ($df(\lambda) \to 0$ as $\lambda \to \infty$), the corresponding $\lambda$ given in the table, and $\hat{\beta}^{\text{ridge}}$.

```
### Singular values
(d.j <- svd(X)$d)
d.j.2 <- d.j^2
function.df.lambda <- function(lambda, d.j.2, df.lambda = 0){
```

---

- 7 -
sum(d.j.2 / (d.j.2 + lambda)) - df.lambda

### plot df(lambda)
lambda <- seq(0, 10, by = 0.01)
df.lambda <- do.call("c", lapply(lambda, function.df.lambda, d.j.2))
plot(lambda, df.lambda, type = "l",
     main = "Effective degrees of freedom",
     xlab = expression(lambda), ylab = expression(df(lambda)))

### solve lambda
(lambda.1 <- uniroot(function.df.lambda, c(0, 10000), d.j.2, df.lambda = 1)$root)
(lambda.2 <- uniroot(function.df.lambda, c(0, 10000), d.j.2, df.lambda = 2)$root)
(lambda.3 <- uniroot(function.df.lambda, c(0, 10000), d.j.2, df.lambda = 3)$root)
(lambda.4 <- uniroot(function.df.lambda, c(0, 10000), d.j.2, df.lambda = 4)$root)

### obtain ridge coefficient
p <- ncol(X)
yarn.ridge.1 <- solve(t(X) %*% X + diag(lambda.1, p)) %*% t(X) %*% y
yarn.ridge.2 <- solve(t(X) %*% X + diag(lambda.2, p)) %*% t(X) %*% y
yarn.ridge.3 <- solve(t(X) %*% X + diag(lambda.3, p)) %*% t(X) %*% y
yarn.ridge.4 <- solve(t(X) %*% X + diag(lambda.4, p)) %*% t(X) %*% y
yarn.ridge <- cbind(yarn.ridge.1, yarn.ridge.2, yarn.ridge.3, yarn.ridge.4)
c) Plot on the same set of axes $\hat{\beta}_j$ versus $j$ for the PCR, PLS and ridge vectors for number of components/degrees of freedom 1. (Plot them as “functions,” connecting consecutive plotted $(j, \hat{\beta}_j)$ points with line segments.) Then do the same for 2, 3, and 4 numbers of components/degrees of freedom.

(3 pts) From the following plot, as $\lambda$ is small, the three methods can provide very different estimations some $\beta$. We can expect that as df($\lambda$) increases the three methods provide similar estimations for all $\beta$. 

R Code

```r
ylim <- range(yarn.pcr$coefficients, yarn.plsr$coefficients, yarn.ridge)
xlim <- c(1, p)
par(mfrow = c(2, 2), mar = c(5, 5, 2, 1))
for(m in 1:4){
  plot(NULL, NULL, xlim = xlim, ylim = ylim, main = substitute(df(lambda) == m, list(m = m)),
       xlab = expression(j), ylab = expression(hat(beta)[j]))
  abline(h = 0, lty = 1)
}````
(d) It is (barely) possible to find that the best (in terms of $R^2$) subsets of $M = 1, 2, 3, \text{ and } 4$ predictors are respectively, $\{x_{40}\}, \{x_{212}, x_{246}\}, \{x_{25}, x_{160}, x_{215}\}$ and $\{x_{160}, x_{169}, x_{231}, x_{243}\}$. Find their corresponding coefficient vectors. Use the lars package in R and find the lasso coefficient vectors $\hat{\beta}$ with exactly $M = 1, 2, 3, \text{ and } 4$ non-zero entries with the largest possible $\sum_{j=1}^{268} |\hat{\beta}_j|$ (for the counts of non-zero entries).

(2 pts) The coefficients for the best (in terms of $R^2$) subsets of $M = 1, 2, 3$ and 4 are as shown in the following output.

R Output
---
```
R> library(pls)
R> library(lars)
R> data(yarn)
R> y <- yarn$density - mean(yarn$density)
R> X <- yarn$NIR
R> X <- t((t(X) - colMeans(X)) / sqrt(apply(X, 2, var)))
R> (beta.best.1 <- as.vector(lm(y ~ X[, 40] - 1)$coef))
[1]  -26.32484
R> (beta.best.2 <- as.vector(lm(y ~ X[, c(212, 246)] - 1)$coef))
R> (beta.best.3 <- as.vector(lm(y ~ X[, c(25, 160, 215)] - 1)$coef))
R> (beta.best.4 <- as.vector(lm(y ~ X[, c(160, 169, 231, 243)] - 1)$coef))
```

(3 pts) Form the lars(), the $x$’s are obtained from steps 2, 13, 18 and 19, and their coefficients are given in the followings with exactly $M = 1, 2, 3, \text{ and } 4$ non-zero entries with the largest possible $\sum_{j=1}^{268} |\hat{\beta}_j|$.

R Output
---
```
R> yarn.lasso <- lars(X, y, type = "lasso", intercept = FALSE,
+ normalize = FALSE, max.steps = 268)
R> nonzero <- lapply(1:nrow(yarn.lasso$beta),
+ function(i) which(yarn.lasso$beta[i,] != 0))
R> nonzero.count <- lapply(1:nrow(yarn.lasso$beta),
+ function(i) sum(yarn.lasso$beta[i,] != 0))
R> abs.beta <- rowSums(abs(yarn.lasso$beta))
R> for(M in 1:4){
+ id <- which(abs.beta == max(abs.beta[nonzero.count == M]) &
+ nonzero.count == M)
+ id.nonzero <- as.vector(nonzero[[id]])
+ cat("M = ", M, " (", id, ")\n", sep = "")
+ cat(" x: ", id, non\n"
```

---

---
e) If necessary, re-order/sort the cases by their values of $y_i$ (from smallest to largest) to get a new indexing. Then plot on the same set of axes $y_i$ versus $i$ and $\hat{y}_i$ versus $i$ for ridge, PCR, PLS, best subset, and lasso regressions for number of components/degrees of freedom/number of nonzero coefficients equal to 1. (Plot them as ”functions,” connecting consecutive plotted, $(i, y_i)$ or $(i, \hat{y}_i)$ points with line segments.) Then do the same for 2, 3, and 4 numbers of components/degrees of freedom/counts of non-zero coefficients.

(5 pts) The following code will draw the summarized plot. The measures used for different methods may not be appropriate for comparison in terms of degrees of freedom. Basically, we can expect the $\hat{y}_i$ should approach to $y$ as the measures increase. For lasso regressions, counting zero coefficients in the active set or picking by smaller $\sum_{j=1}^{268} \beta_j^{lasso}$ will yield very unlikely conclusions.

```R
yarn.pcr.y.hat <- cbind(yarn.pcr$fitted.values[,1], yarn.pcr$fitted.values[,2], yarn.pcr$fitted.values[,3], yarn.pcr$fitted.values[,4])
yarn.plsr.y.hat <- cbind(yarn.plsr$fitted.values[,1], yarn.plsr$fitted.values[,2], yarn.plsr$fitted.values[,3], yarn.plsr$fitted.values[,4])
yarn.ridge.y.hat <- X %*% cbind(yarn.ridge.1, yarn.ridge.2, yarn.ridge.3, yarn.ridge.4)
yarn.best.y.hat <- cbind(X[, 40] * beta.best.1, X[, c(212, 246)] * beta.best.2, X[, c(25, 160, 215)] * beta.best.3, X[, c(160, 169, 231, 243)] * beta.best.4)
yarn.lasso.y.hat <- X %*% t(yarn.lasso$beta[c(2, 13, 18, 19), ])
x.id <- 1:length(y)
y.id <- order(y)
ylim <- range(y, yarn.pcr.y.hat, yarn.plsr.y.hat, yarn.ridge.y.hat, yarn.best.y.hat, yarn.lasso.y.hat)
xlim <- range(x.id)
par(mfrow = c(2, 2), mar = c(5, 5, 2, 1))
for(m in 1:4){
    plot(NULL, NULL, xlim = xlim, ylim = ylim, main = substitute(M == m, list(m = m)),
         xlab = expression(paste("Index of sorted ", y[j])), ylab = expression(hat(y)[j]))
    for(i in 1:length(x.id)){
        # Plot the data points
        points(x.id[i], y.id[i], pch = 16, cex = 0.5)
        # Plot the fitted values
        lines(x.id[i], yarn.pcr.y.hat[i, ], col = 2)
        lines(x.id[i], yarn.plsr.y.hat[i, ], col = 3)
        lines(x.id[i], yarn.ridge.y.hat[i, ], col = 4)
        lines(x.id[i], yarn.best.y.hat[i, ], col = 5)
        lines(x.id[i], yarn.lasso.y.hat[i, ], col = 6)
    }
    # Plot the residuals
    for(i in 1:length(x.id)){
        points(x.id[i], resid(yarn.pcr.y.hat)[i], pch = 16, cex = 0.5)
        points(x.id[i], resid(yarn.plsr.y.hat)[i], pch = 16, cex = 0.5)
        points(x.id[i], resid(yarn.ridge.y.hat)[i], pch = 16, cex = 0.5)
        points(x.id[i], resid(yarn.best.y.hat)[i], pch = 16, cex = 0.5)
        points(x.id[i], resid(yarn.lasso.y.hat)[i], pch = 16, cex = 0.5)
    }
    # Add a grid
    grid(lty = 1, lwd = 0.5)
}
```

---

- 11 -
abline(h = 0, lty = 1)
lines(x.id, y[y.id], lty = 1, col = 1, lwd = 2)
lines(x.id, yarn.pcr.y.hat[y.id, m], lty = 2, col = 2, lwd = 2)
lines(x.id, yarn.plsr.y.hat[y.id, m], lty = 3, col = 3, lwd = 2)
lines(x.id, yarn.ridge.y.hat[y.id, m], lty = 4, col = 4, lwd = 2)
lines(x.id, yarn.best.y.hat[y.id, m], lty = 5, col = 5, lwd = 2)
lines(x.id, yarn.lasso.y.hat[y.id, m], lty = 6, col = 6, lwd = 2)
legend(1, 75, c("y", "PCR", "PLS", "Ridge", "Best", "Lasso"),
       lty = 1:6, col = 1:6, cex = 0.8, lwd = 2)

Total: 65 pts (Q4: 10, Q5: 15, Q6: 10, Q7:10, Q8:20). Any resonable solutions are acceptable.