Homework #7

1. **(Page 125, chap. 3, #14)**. This problem focuses on **multicollinearity**.

   (a) Perform the following commands in R:

   ```
   > set.seed (1)
   > x1 = runif (100)
   > x2 = 0.5*x1 + rnorm(100)/10
   > y = 2 + 2*x1 + 0.3*x2 + rnorm(100)
   ```

   The last line corresponds to creating a linear model in which y is a function of x_1 and x_2. Write out the form of the linear model. What are the regression coefficients?

   Form of the model:

   $$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$$

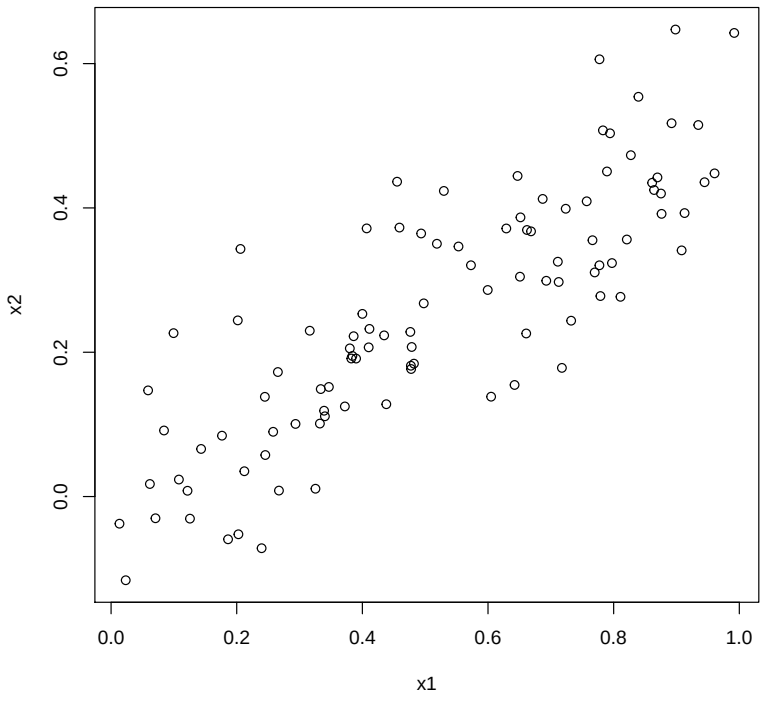
   $\beta_1 = 2$ and $\beta_2 = 0.3$ are the regression coefficients.

   (b) What is the correlation between x_1 and x_2? Create a scatterplot displaying the relationship between the variables.

```
cor(x1,x2)

## [1] 0.8351212

plot(x1,x2)
```



(c) Using this data, fit a least squares regression to predict y using x_1 and x_2. Describe the results obtained. What are $\hat{\beta}_0$, $\hat{\beta}_1$, and $\hat{\beta}_2$? What are the true β_0, β_1, and β_2? Can you reject the null hypothesis $H_0 : \beta_1 = 0$? How about the null hypothesis $H_0 : \beta_2 = 0$?

```
reg <- lm(y ~ x1 + x2)
summary(reg)

##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8311 -0.7273 -0.0537  0.6338  2.3359
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1305     0.2319   9.188 7.61e-15 ***
## x1            1.4396     0.7212   1.996   0.0487 *
## x2            1.0097     1.1337   0.891   0.3754
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.056 on 97 degrees of freedom
## Multiple R-squared:  0.2088,Adjusted R-squared:  0.1925
## F-statistic:  12.8 on 2 and 97 DF,  p-value: 1.164e-05
```

The estimation $\hat{\beta}_0 = 2.1305$, $\hat{\beta}_1 = 1.4396$, and $\hat{\beta}_2 = 1.0097$. And the true $\beta_0 = 2$, $\beta_1 = 2$, and $\beta_2 = 0.3$. Under the significance level of 0.05, I can reject the null hypothesis $H_0 : \beta_1 = 0$ but I cannot reject the null hypothesis $H_0 : \beta_2 = 0$.

(d) Now fit a least squares regression to predict y using only x_1. Comment on your results. Can you reject the null hypothesis $H_0 : \beta_1 = 0$?

```
reg.x1 <- lm(y ~ x1)
summary(reg.x1)

##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.89495 -0.66874 -0.07785  0.59221  2.45560
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1124     0.2307   9.155 8.27e-15 ***
## x1            1.9759     0.3963   4.986 2.66e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.055 on 98 degrees of freedom
## Multiple R-squared:  0.2024,Adjusted R-squared:  0.1942
## F-statistic: 24.86 on 1 and 98 DF,  p-value: 2.661e-06
```

I can reject the null hypothesis $H_0 : \beta_1 = 0$.

(e) Now fit a least squares regression to predict y using only x_2. Comment on your results. Can you reject the null hypothesis $H_0 : \beta_2 = 0$?

```
reg.x2 <- lm(y ~ x2)
summary(reg.x2)

##
## Call:
## lm(formula = y ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.62687 -0.75156 -0.03598  0.72383  2.44890
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.3899     0.1949   12.26  < 2e-16 ***
## x2            2.8996     0.6330    4.58 1.37e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.072 on 98 degrees of freedom
## Multiple R-squared:  0.1763,Adjusted R-squared:  0.1679
## F-statistic: 20.98 on 1 and 98 DF,  p-value: 1.366e-05
```

I can reject the null hypothesis $H_0 : \beta_2 = 0$.

(f) Do the results obtained in (c)–(e) contradict each other? Explain your answer.

No, given x_1 and x_2 collinear, including both of them in the model will reduce the power of t-test. Also they will diminish the explanatory effect of each other. So it makes sense to have the predictor significant in the model including only one of them, while at least one of them is not signficant in the model including both of them.

(g) Now suppose we obtain one additional observation, which was unfortunately mismeasured. Use the following R code.

```
> x1=c(x1, 0.1)
> x2=c(x2, 0.8)
> y=c(y,6)
```

Re-fit the linear models from (c) to (e) using this new data. What effect does this new observation have on the each of the models? In each model, is this observation an outlier? A high-leverage point? Both? Explain your answers. How do the slopes from all the considered models react on the newly added data point?

```
x1=c(x1, 0.1)
x2=c(x2, 0.8)
y=c(y,6)
reg <- lm(y ~ x1 + x2)
reg.x1 <- lm(y ~ x1)
reg.x2 <- lm(y ~ x2)
summary(reg)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.73348 -0.69318 -0.05263  0.66385  2.30619
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2267     0.2314   9.624 7.91e-16 ***
## x1            0.5394     0.5922   0.911  0.36458
## x2            2.5146     0.8977   2.801  0.00614 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.075 on 98 degrees of freedom
## Multiple R-squared:  0.2188,Adjusted R-squared:  0.2029
## F-statistic: 13.72 on 2 and 98 DF,  p-value: 5.564e-06
```

```
summary(reg.x1)
```

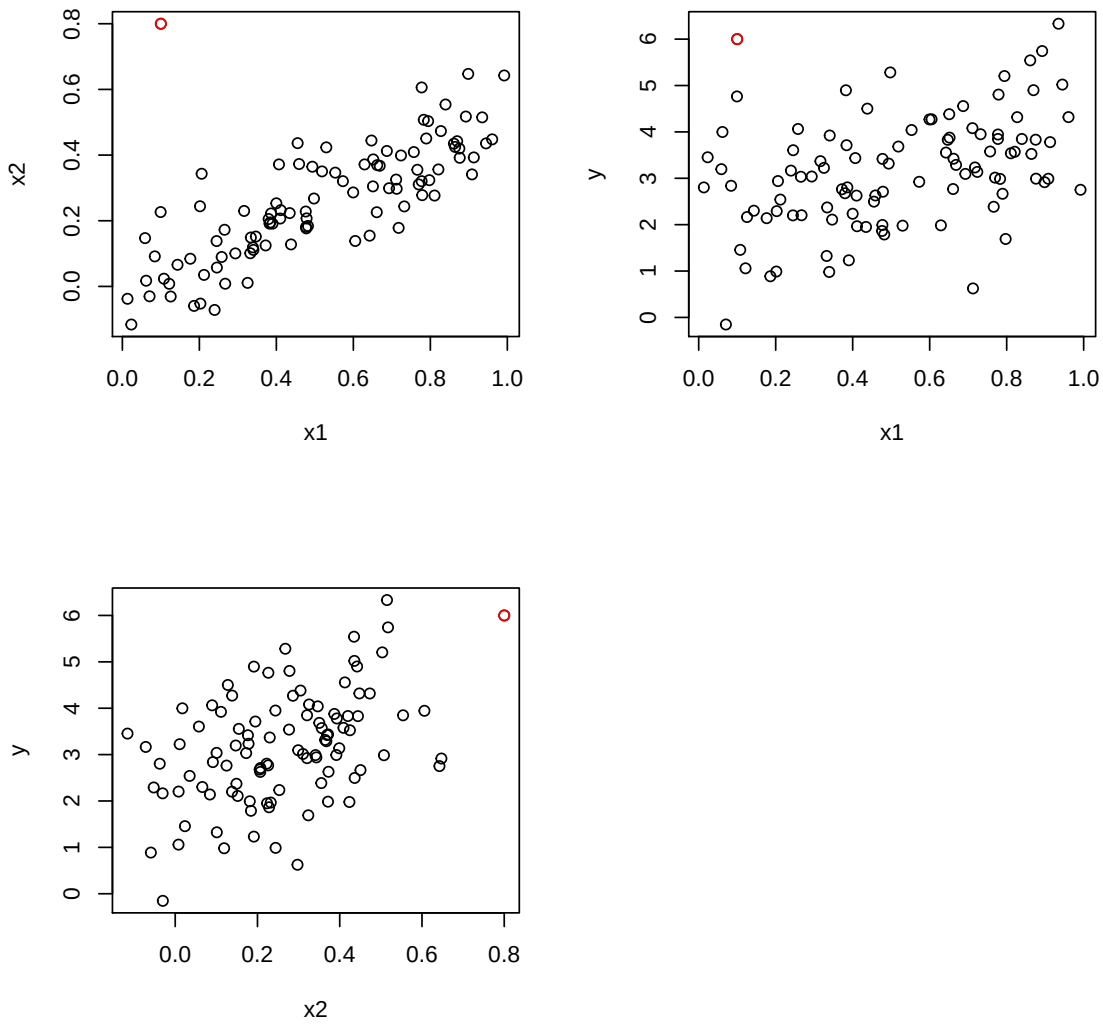
```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8897 -0.6556 -0.0909  0.5682  3.5665
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2569     0.2390   9.445 1.78e-15 ***
## x1            1.7657     0.4124   4.282 4.29e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.111 on 99 degrees of freedom
## Multiple R-squared:  0.1562,Adjusted R-squared:  0.1477
## F-statistic: 18.33 on 1 and 99 DF,  p-value: 4.295e-05
```

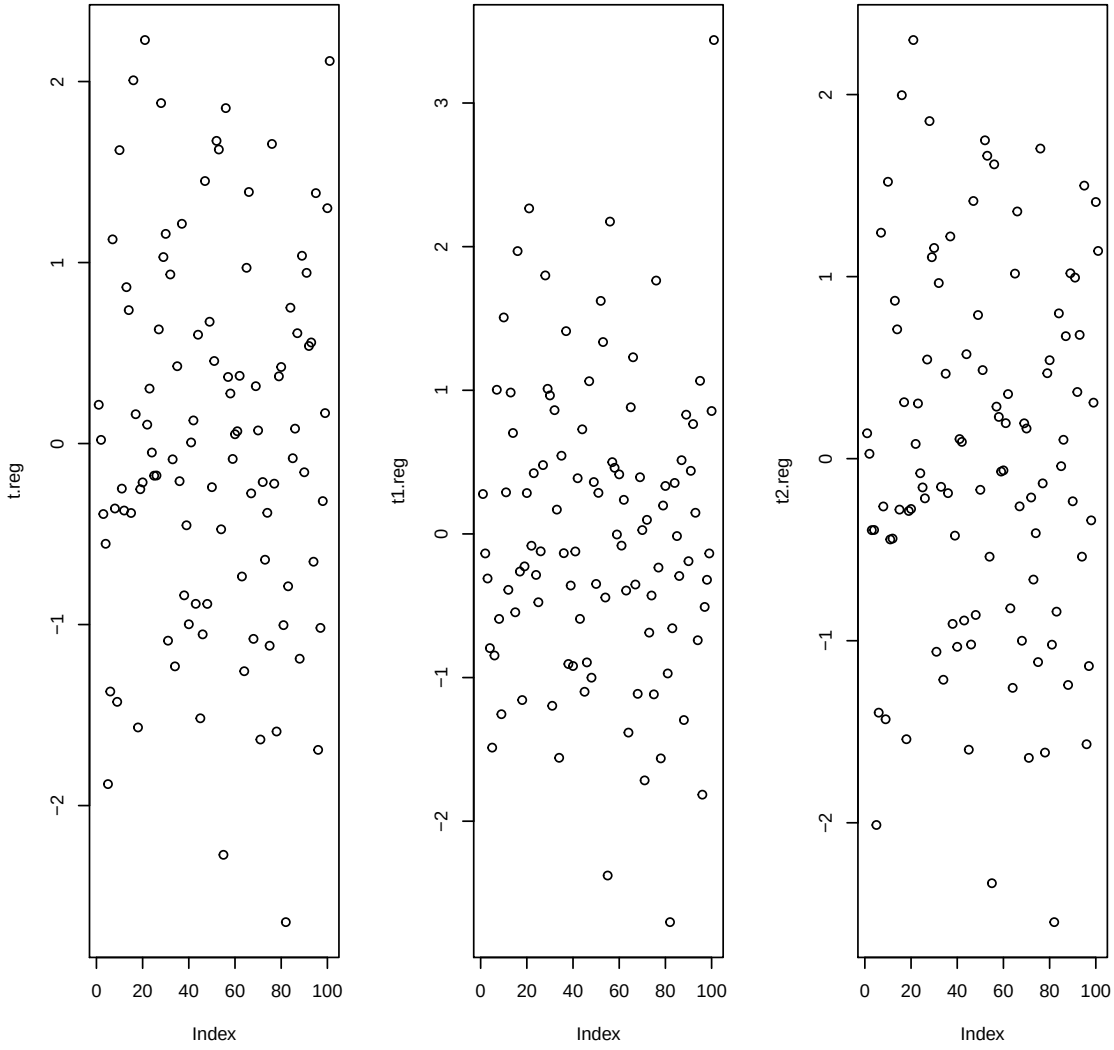
```
summary(reg.x2)
```

```
##
## Call:
## lm(formula = y ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.64729 -0.71021 -0.06899  0.72699  2.38074
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.3451     0.1912  12.264  < 2e-16 ***
## x2            3.1190     0.6040   5.164 1.25e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.074 on 99 degrees of freedom
## Multiple R-squared:  0.2122,Adjusted R-squared:  0.2042
## F-statistic: 26.66 on 1 and 99 DF,  p-value: 1.253e-06
```

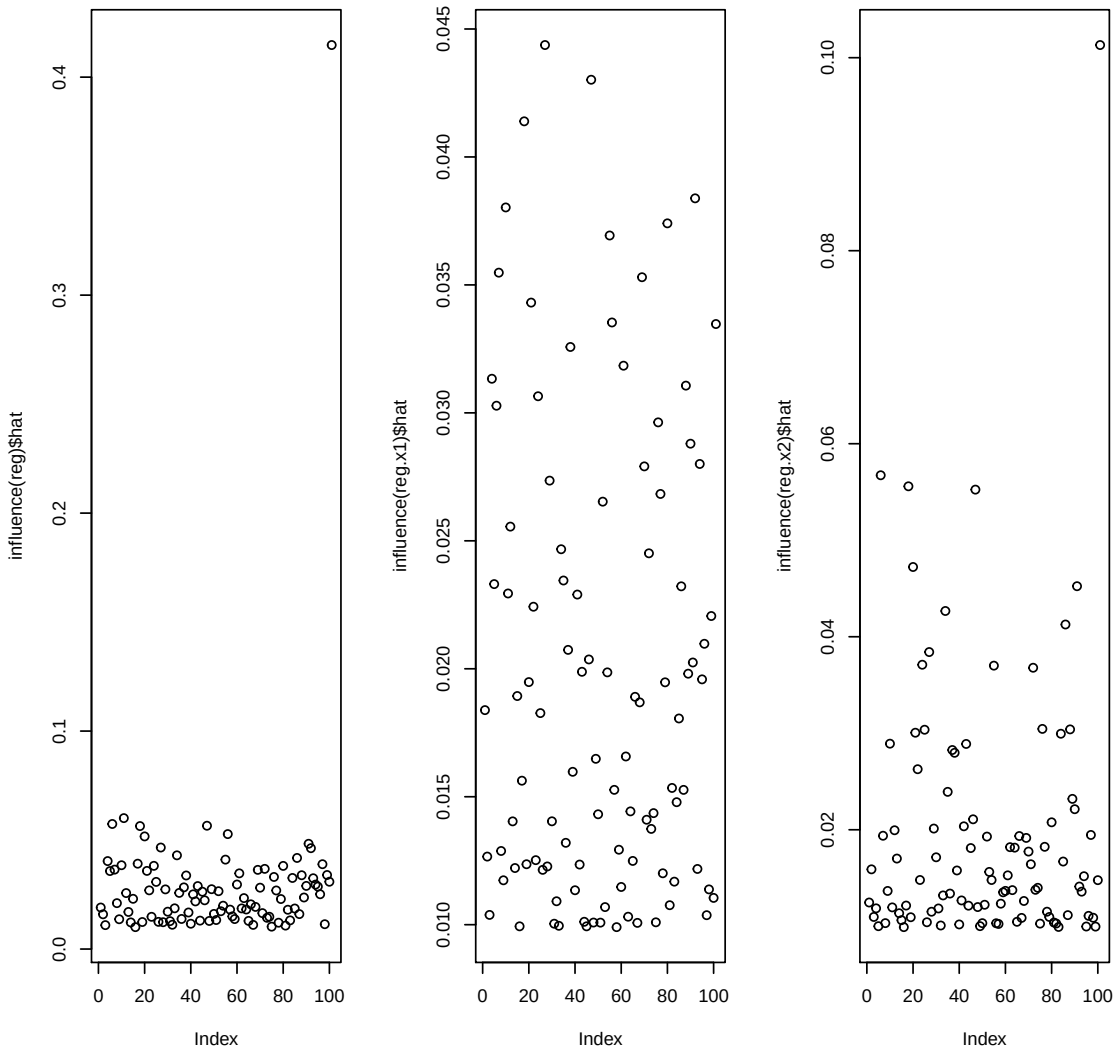
```
# plot
par(mfrow = c(2,2))
plot(x1,x2)
points(x = 0.1, y = 0.8, col = 2)
plot(x1,y)
points(x = 0.1, y = 6, col = 2)
plot(x2,y)
points(x = 0.8, y = 6, col = 2)
# outlier
par(mfrow = c(1,3))
```




```
t.reg <- rstudent(reg)
t1.reg <- rstudent(reg.x1)
t2.reg <- rstudent(reg.x2)
plot(t.reg); plot(t1.reg); plot(t2.reg)
```




```
# influential
par(mfrow = c(1,3))
plot(influence(reg)$hat)
plot(influence(reg.x1)$hat)
plot(influence(reg.x2)$hat)
```



It is not an outlier or high-leverage point in model (d) and (e). But it is both an outlier and influential case in model (c) and (d). The estimated slope of model (d) and (e) does not change much, but the estimated slopes of model (c) change very much.

(h) What are standard errors of estimated regression slopes in (a), (d), and (e)? Which models produce more stable and therefore, more reliable estimates?

model	$Var(\beta_1)$	$Var(\beta_2)$
(c)	0.7212	1.1337
(d)	0.3963	
(e)		0.6330

Model (d) and (e) produce more stable estimates than model (c).

(i) Compute both VIF in question (a) and relate them to your answer to question (h).

```
car::vif(reg)

##       x1       x2
## 2.204867 2.204867
```

The collinearity betwenn x_1 and x_2 cause the inflation of variance in model (c), so we have vif of x_1 and x_2 1.76 greater than 1. Removing any of them can reduce the variance of model (c). Thus model (d) and (e) have smaller variance than (c).

2. **(Chap. 6, # 2, p.259)** Consider three methods of fitting a linear regression model - (a) lasso, (b) ridge regression, and (c) fitting nonlinear trends. For each method, choose the right answer, comparing it with the least squares regression:

   i. The method is more flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

   ii. The method is more flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

   iii. The method is less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

   iv. The method is less flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

   (a) iii.

   (b) iii.

   (c) ii.

3. **(Chap. 6, ≈# 6, p.261)** Ridge regression minimizes

$$\sum_{i=1}^{n}(Y_i - \beta_0 - X_{i1}\beta_1 - \cdots - X_{ip}\beta_p)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 \tag{1}$$

whereas lasso minimizes

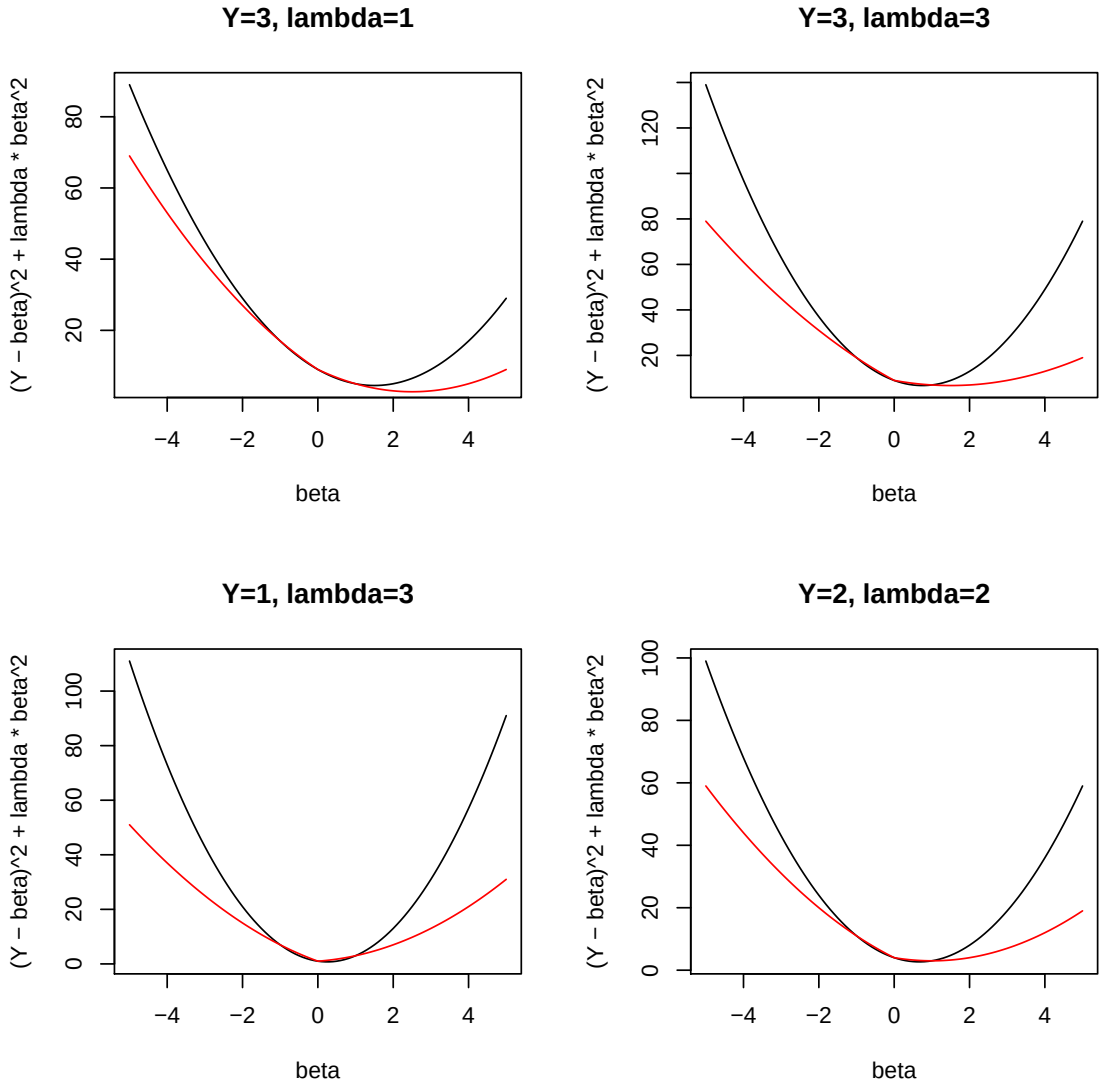
$$\sum_{i=1}^{n}(Y_i - \beta_0 - X_{i1}\beta_1 - \cdots - X_{ip}\beta_p)^2 + \lambda\sum_{j=1}^{p}|\beta_j| \tag{2}$$

Consider a "toy" example, where $n = p = 1$, $X = 1$, and the intercept is omitted from the model. Then RSS reduces to $RSS = (Y - \beta)^2$.

(a) Choose some Y and λ, plot (1) and (2) as functions of β, and find their minima on these graphs. Verify that these minima are attained at

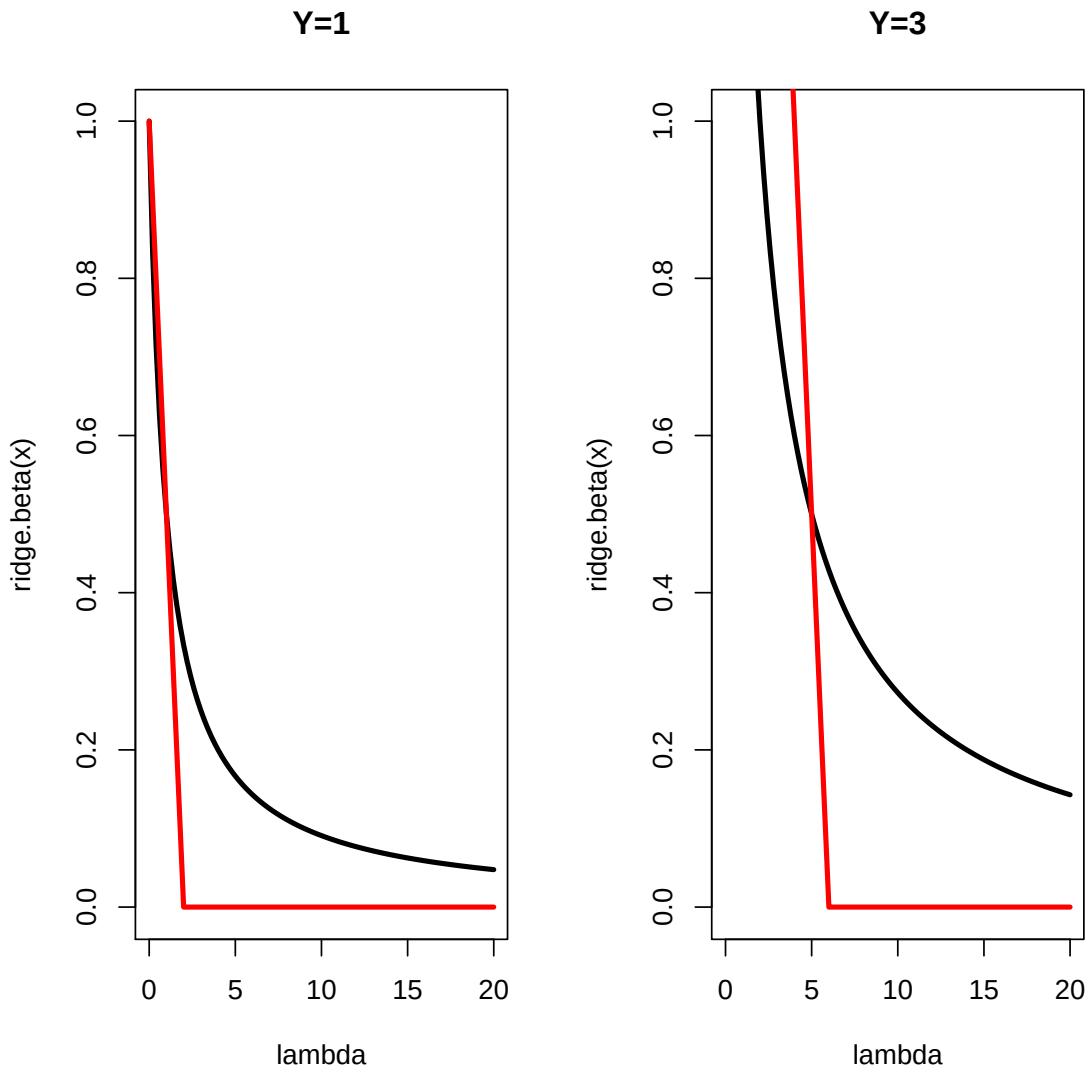
$$\hat{\beta}_{ridge} = \frac{Y}{1+\lambda} \quad \text{and} \quad \hat{\beta}_{lasso} = \begin{cases} Y - \lambda/2 & \text{if} \quad Y > \lambda/2 \\ Y + \lambda/2 & \text{if} \quad Y < -\lambda/2 \\ 0 & \text{if} \quad |Y| < \lambda/2 \end{cases} \tag{3}$$

```
X <- 1
par(mfrow = c(2,2))
Y <- 3
lambda <- 1
curve((Y - beta)^2 + lambda*beta^2, -5, 5, col =1, xname = "beta", xlab="beta", ma
curve((Y - beta)^2 + lambda*abs(beta), -5, 5, col =2, xname = "beta", xlab="beta",
Y <- 3
lambda <- 3
curve((Y - beta)^2 + lambda*beta^2, -5, 5, col =1,xname = "beta", xlab="beta", mai
curve((Y - beta)^2 + lambda*abs(beta), -5, 5, col =2, xname = "beta", xlab="beta",
Y <- 1
lambda <- 3
curve((Y - beta)^2 + lambda*beta^2, -5, 5, col =1,xname = "beta", xlab="beta", mai
curve((Y - beta)^2 + lambda*abs(beta), -5, 5, col =2, xname = "beta", xlab="beta",
Y <- 2
lambda <- 2
curve((Y - beta)^2 + lambda*beta^2, -5, 5, col =1, xname = "beta", xlab="beta", ma
curve((Y - beta)^2 + lambda*abs(beta), -5, 5, col =2, xname = "beta", xlab="beta",
```



(b) Now choose some value of Y and plot ridge regression and lasso solutions (3) on the same axes, as functions of λ. Observe how ridge regression keeps a slope whereas lasso sends the slope to 0 when the penalty term is high.

```
ridge.beta <- function(l){return( Y/(1+l) )}
lasso.beta <- function(l){ return( (Y-l/2)*(Y > l/2) + (Y+l/2)*(Y < -l/2)) + 0*(ab
# plot
par(mfrow = c(1,2))
Y <- 1
curve( ridge.beta, 0, 20, col=1, lwd=3, ylim = c(-0.001,1), xlab="lambda", main="Y
curve( lasso.beta, 0, 20, col=2, lwd=3, add=TRUE)
Y <- 3
curve( ridge.beta, 0, 20, col=1, lwd=3, ylim = c(-0.001,1), xlab="lambda", main="Y
curve( lasso.beta, 0, 20, col=2, lwd=3, add=TRUE)
```




```
dev.off()

## null device
##           1
```

4. **(Simulation project - Chap. 6, # 8, p.262)**
In this exercise, we will generate simulated data, and will then use this data to perform best subset selection.

   (a) Use the `rnorm()` function to generate a predictor X and a noise vector ε of length $n = 100$ (you can refer to our lab "First steps in R" for this command).

```
set.seed(666)
x <- rnorm(100)
```

```
epsi <- rnorm(100)
```

(b) Generate a response vector Y according to the model

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \varepsilon$$

where β_0, β_1, β_2, and β_3 are constants of your choice.

```
b.0 <- b.1 <- b.2 <- b.3 <- 2
y <- b.0 + b.1*x + b.2^2*x + b.3^3*x + epsi
```

(c) Use stepwise selection with `step` for variable selection. How does your answer compare to the results in (c)?

```
df <- data.frame(x,y)
null <- lm(y~1, data =df)
full <- lm(y ~ x + I(x^2)+ I(x^3) + I(x^4)+ I(x^5)+
             I(x^6)+ I(x^7)+ I(x^8)+ I(x^9)+ I(x^10), data = df)
step( null,scope=list(lower=null, upper=full), direction="forward")

## Start:  AIC=538.01
## y ~ 1
##
##           Df Sum of Sq     RSS    AIC
## + x        1   21151.5   123.4  25.05
## + I(x^3)   1   13969.5  7305.5 433.12
## + I(x^5)   1    7012.7 14262.3 500.02
## + I(x^7)   1    3964.6 17310.3 519.39
## + I(x^9)   1    2782.2 18492.7 526.00
## + I(x^10)  1    1698.5 19576.4 531.69
## + I(x^8)   1    1498.0 19777.0 532.71
## + I(x^6)   1    1108.5 20166.4 534.66
## + I(x^4)   1     531.8 20743.1 537.48
## <none>                 21274.9 538.01
## + I(x^2)   1     131.4 21143.6 539.39
##
## Step:  AIC=25.05
## y ~ x
##
##           Df Sum of Sq    RSS    AIC
## + I(x^3)   1    7.0862 116.34 21.134
## + I(x^5)   1    6.1138 117.31 21.966
## + I(x^7)   1    5.4299 118.00 22.547
## + I(x^6)   1    5.4248 118.00 22.552
## + I(x^8)   1    5.3370 118.09 22.626
## + I(x^10)  1    5.1960 118.23 22.745
```

```
## + I(x^9)   1    5.1257 118.30 22.805
## + I(x^4)   1    5.0708 118.35 22.851
## + I(x^2)   1    3.4906 119.93 24.177
## <none>                 123.42 25.046
##
## Step:  AIC=21.13
## y ~ x + I(x^3)
##
##           Df Sum of Sq    RSS    AIC
## <none>                 116.34 21.134
## + I(x^2)   1   1.13273 115.21 22.155
## + I(x^4)   1   0.72746 115.61 22.506
## + I(x^6)   1   0.37907 115.96 22.807
## + I(x^8)   1   0.19718 116.14 22.964
## + I(x^10)  1   0.11515 116.22 23.035
## + I(x^5)   1   0.02569 116.31 23.111
## + I(x^9)   1   0.00447 116.33 23.130
## + I(x^7)   1   0.00081 116.34 23.133
##
## Call:
## lm(formula = y ~ x + I(x^3), data = df)
##
## Coefficients:
## (Intercept)            x       I(x^3)
##      1.9169      13.8510       0.1183
```

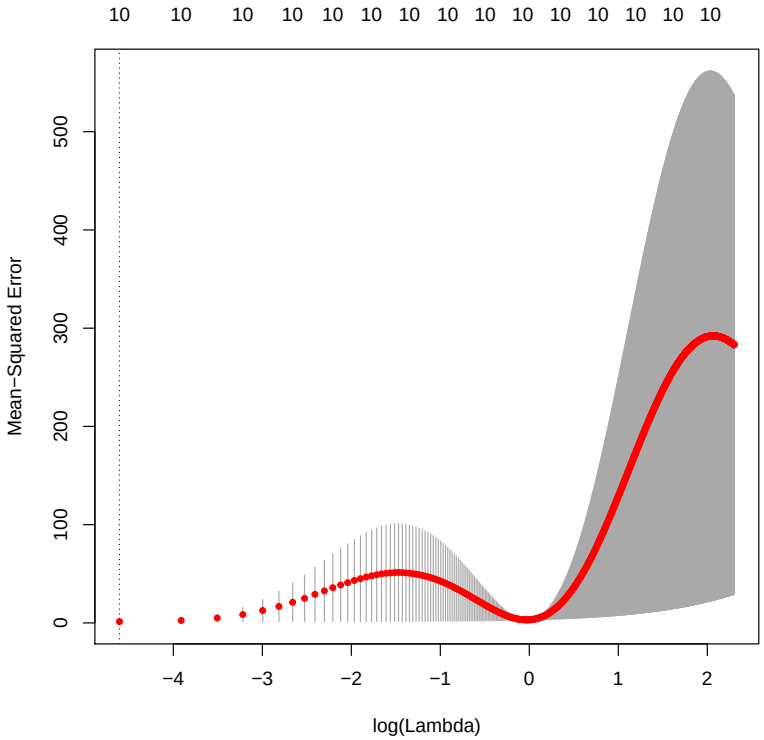
The second order term x^2 is removed from the model. But we keep x_2 as long as we want to keep the higher order term x_3.

(d) Now fit a lasso model with the same predictors. Use cross-validation to select the optimal value of λ. Create plots of the cross-validation error as a function of λ. Report the resulting coefficient estimates, and discuss the results obtained. Which predictors got eliminated by lasso?

```
x.mat <- model.matrix(full)
lasso <- glmnet::cv.glmnet(x.mat,y,alpha = 0, lambda = seq(0,10,0.01))
lasso$lambda.min
```

```
## [1] 0
```

```
plot(lasso)
```




```
predict( lasso, lasso$lambda.min, type="coefficients")

## 12 x 1 sparse Matrix of class "dgCMatrix"
##                            1
## (Intercept)  1.9970539230
## (Intercept)  .
## x           13.5607612237
## I(x^2)      -0.1083057380
## I(x^3)       0.3896222751
## I(x^4)       0.0438761238
## I(x^5)      -0.0293403843
## I(x^6)      -0.0042655290
## I(x^7)      -0.0035394067
## I(x^8)      -0.0010009796
## I(x^9)      -0.0002439069
## I(x^10)     -0.0001101012
```

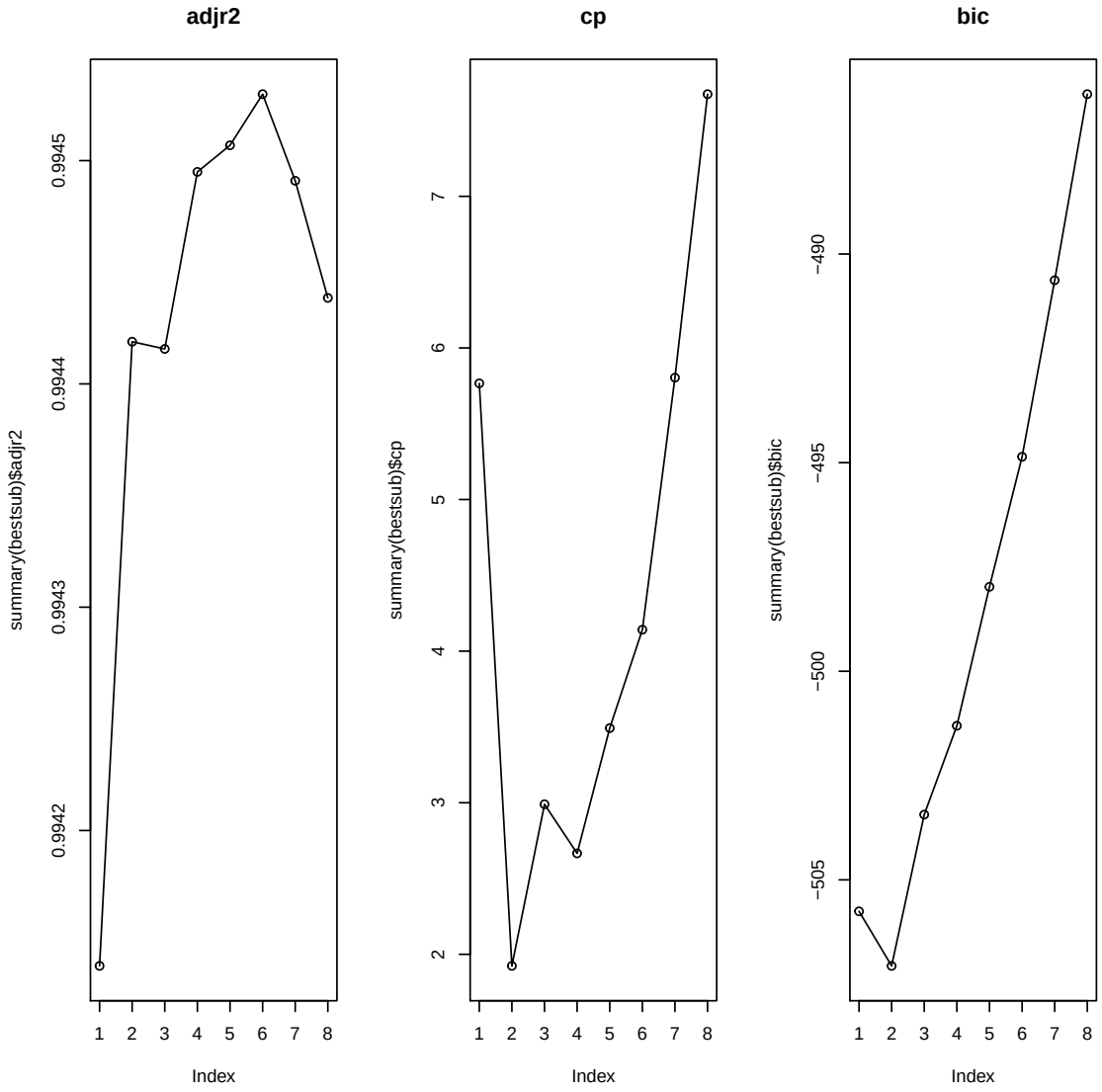
The best λ derived from cross validation is 0, so no predictors got eliminated by lasso.

(e) Now generate a response vector Y according to the model

$$Y = \beta_0 + \beta_7 X^7 + \varepsilon$$

and perform best subset selection and the lasso. Discuss the results.

```
b.7 <- 2
y <- b.0 + b.7^7*x + epsi
bestsub <- leaps::regsubsets(y ~ x + I(x^2)+ I(x^3) + I(x^4)+ I(x^5)+ I(x^6)
                             + I(x^7)+ I(x^8)+ I(x^9)+ I(x^10), data = df)
par(mfrow = c(1,3))
plot(summary(bestsub)$adjr2, main = "adjr2")
lines(summary(bestsub)$adjr2)
plot(summary(bestsub)$cp, main = "cp")
lines(summary(bestsub)$cp)
plot(summary(bestsub)$bic, main = "bic")
lines(summary(bestsub)$bic)
```



```r
which.max(summary(bestsub)$adjr2)
```

```
## [1] 6
```

```r
which.min(summary(bestsub)$cp)
```

```
## [1] 2
```

```r
which.min(summary(bestsub)$bic)
```

```
## [1] 2
```

```r
summary(lm(y ~ x + I(x^2)+ I(x^3) + I(x^4)+ I(x^5)+ I(x^6),data =df))
```

```
##
## Call:
## lm(formula = y ~ x + I(x^2) + I(x^3) + I(x^4) + I(x^5) + I(x^6),
##     data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.62261 -0.74327  0.01143  0.80274  2.82978
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.05283    0.18606  11.033   <2e-16 ***
## x           13.49669    0.35343  38.188   <2e-16 ***
## I(x^2)      -0.33301    0.43557  -0.765    0.446
## I(x^3)       0.54101    0.33154   1.632    0.106
## I(x^4)       0.18609    0.21612   0.861    0.391
## I(x^5)      -0.08400    0.06413  -1.310    0.193
## I(x^6)      -0.03102    0.02908  -1.067    0.289
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.102 on 93 degrees of freedom
## Multiple R-squared:  0.9947,Adjusted R-squared:  0.9943
## F-statistic:  2902 on 6 and 93 DF,  p-value: < 2.2e-16
```

```r
summary(lm(y ~ x + I(x^2),data =df))
```

```
##
## Call:
## lm(formula = y ~ x + I(x^2), data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -2.74071 -0.70677  0.02491  0.70566  2.45316
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.05946    0.13876   14.84   <2e-16 ***
## x           14.18762    0.10880  130.40   <2e-16 ***
## I(x^2)      -0.13265    0.07895   -1.68   0.0961 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.112 on 97 degrees of freedom
## Multiple R-squared:  0.9944,Adjusted R-squared:  0.9942
## F-statistic:  8555 on 2 and 97 DF,  p-value: < 2.2e-16
```

```
# y ~ x^2
reg <- lm(y ~ x + I(x^2),data =df)
x.mat <- model.matrix(reg)
lasso <- glmnet::cv.glmnet(x.mat,y,alpha = 0, lambda = seq(0,10,0.01))
predict( lasso, lasso$lambda.min, type="coefficients")
```

```
## 4 x 1 sparse Matrix of class "dgCMatrix"
##                        1
## (Intercept)   2.0529777
## (Intercept)   .
## x           127.8161495
## I(x^2)       -0.1500017
```

The best subset selection algorithm yields the model $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2^2 + \varepsilon$. The cross validation for lasso suggests no to drop any estimator of the best subset model.