

Capítulo 6. Linear Model Selection and Regularization

Diego Felipe Bobadilla Restrepo
Yosef Shmuel Guevara Salamanca

June 15, 2021

Ejercicios

1. We perform best subset, forward stepwise, and backward stepwise selection on a single data set. For each approach, we obtain $p + 1$ models, containing $0, 1, 2, \dots, p$ predictors. Explain your answers:

- (a) Which of the three models with k predictors has the smallest training RSS ?

El modelo cuyo RSS sea el más pequeño sera el que se desempeñe mejor, inclusive puede que tanto la aproximación forward como backward lleguen al seleccionar el mismo subset para los datos de entrenamiento,

- (b) Which of the three models with k predictors has the smallest test RSS ?

Una vez se selecciona el subset de variables predictoras que conforman un modelo que minimiza el RSS para los datos de entrenamiento, no es posible afirmar con seguridad que este sea modelo sea el que mejor se ajusta para los datos de prueba aunque esto no es del todo descartable. El modelo resultante puede generar un sobre ajuste para los datos de prueba sobre todo en aquellos casos donde $p > n$ o donde n no sea lo suficientemente grande que p .

- (c) True or False:

- i. The predictors in the k -variable model identified by forward stepwise are a subset of the predictors in the $(k + 1)$ -variable model identified by forward stepwise selection.

Verdadero. El modelo con $(k + 1)$ -variables es igual al modelo con k -variables, pero con una variable predictora adicional.

- ii. The predictors in the k -variable model identified by backward stepwise are a subset of the predictors in the $(k + 1)$ -variable model identified by backward stepwise selection.

Verdadero. El modelo de k -variables es igual al modelo con $(k + 1)$ -variables pero con un predictor menos.

- iii. The predictors in the k -variable model identified by backward stepwise are a subset of the predictors in the $(k + 1)$ -variable model identified by forward stepwise selection.

Falso. Debido a que los caminos que toman ambos procesos para seleccionar el subset de variables predictoras es diferente, el modelo foward stepwise comienza haciendo uso del modelo nulo, mientras el bacwkard stepwise usa el modelo completo con todas las variables predictoras en su intento para encontrar el RSS más pequeño.

- iv. The predictors in the k -variable model identified by forward stepwise are a subset of the predictors in the $(k + 1)$ -variable model identified by backward stepwise selection.

Falso. Al igual que en el caso anterior los caminos que toman ambos métodos son diferentes por lo que el subset de variables predictoras que conforman el modelo puede ser diferente.

- v. The predictors in the k -variable model identified by best subset are a subset of the predictors in the $(k + 1)$ -variable model identified by best subset selection.

Falso. No existe garantía absoluta que nos permite afirmar que las variables predictoras seleccionadas por el modelo de k -variables sea el mismo que para el modelo con $(k + 1)$ -variables.

2. For parts (a) through (c), indicate which of i. through iv. is correct. Justify your answer.

- (a) The lasso, relative to least squares, is:

- i. More flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.
- ii. More flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.
- iii. Less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.
- iv. Less flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

La Respuesta es iii. LASSO es menos flexible, por lo tanto, dará una mayor precisión en la predicción cuando el aumento en el sesgo es menor que la disminución en la varianza. Esto dado que la selección LASSO selecciona los $\hat{\beta}$ que minimizan a $RSS + \lambda \sum_{i=1}^p |\beta_i|$, pues intentará que estas estimaciones sean cercanas a cero, reduciendo la varianza de las predicciones a costa de un pequeño aumento en el sesgo.

- (b) Repeat (a) for ridge regression relative to least squares.

La Respuesta es iii. Al igual que en el punto anterior la regresión ridge es menos flexible, por lo tanto, dará una mayor precisión en la predicción cuando el aumento en el sesgo es menor que la disminución en la varianza, sin embargo a diferencia del método LASSO no reducirá los coeficientes de las variables menos útiles a cero.

- (c) Repeat (a) for non-linear methods relative to least squares.

La Respuesta es ii. Más flexible y, por lo tanto, dará una mayor precisión de la predicción cuando el aumento en la varianza sea menor que la disminución en el sesgo. Dado que los métodos no lineales son más flexibles y se ajustan mejor cuando no existe linealidad, pues no se asume directamente una relación entre las variables predictoras y la variable respuesta por lo tanto una disminución en el sesgo compensa el aumento en la varianza.

3. Suppose we estimate the regression coefficients in a linear regression model by minimizing

$$\sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 \text{ subject to } \sum_{i=1}^p |\beta_j| \leq s$$

for a particular value of s . For parts (a) through (e), indicate which of i. through v. is correct. Justify your answer.

- (a) As we increase s from 0, the training RSS will:

- i. Increase initially, and then eventually start decreasing in an inverted U shape.
- ii. Decrease initially, and then eventually start increasing in a U shape.
- iii. Steadily increase.
- iv. Steadily decrease.
- v. Remain constant.

La Respuesta es iv. Cuando incrementamos s desde 0 el RSS disminuirá constantemente, a medida que añadimos más parametro el RSS de entrenamiento se hará más pequeño pues $\sum_{k=1}^p |\beta_j| \leq s$. Hay cada vez menos restricciones y los coeficientes aumentan a sus estimaciones de mínimos cuadrados haciendo el modelo más flexible y se tendrá cada vez un mejor error de entrenamiento.

- (b) Repeat (a) for test RSS.

La Respuesta es ii. Disminuye inicialmente y luego, eventualmente, comience a aumentar en forma de U. El error de prueba disminuye hasta cierto punto y luego aumenta a medida que las restricciones de los coeficientes son cada vez menores y el modelo tiende a sobreajustarse.

- (c) Repeat (a) for variance.

La Respuesta es iii. Aumenta constantemente. Cada vez que las restricciones sobre los coeficientes son menores, la varianza tiene a incrementar de forma constante.

- (d) Repeat (a) for (squared) bias.

La Respuesta es iv. Disminuye constantemente. El comportamiento del sesgo es inversamente proporcional a la flexibilidad del modelo. A medida que aumenta dicha flexibilidad, el sesgo disminuye constantemente.

- (e) Repeat (a) for the irreducible error.

La Respuesta es v. Permanece constante. el error irreducible es un valor constante, no relacionado con la selección del modelo y por lo tanto no se encuentra relacionado con el modelo seleccionado.

4. Suppose we estimate the regression coefficients in a linear regression model by minimizing

$$\sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

for a particular value of λ . For parts (a) through (e), indicate which of i. through v. is correct. Justify your answer.

- (a) As we increase λ from 0, the training RSS will:

- i. Increase initially, and then eventually start decreasing in an inverted U shape.
- ii. Decrease initially, and then eventually start increasing in a U shape.
- iii. Steadily increase.
- iv. Steadily decrease.
- v. Remain constant.

La Respuesta es iii. Cuando $\lambda = 0$, la regresión ridge $\hat{\beta}$ sera igual que el estimado β para el metodo de minimos cuadrados, puesto que el termino de contracción es removido, lo que reducira el *RSS* de entrenamiento. A medida que se incrementa λ , el *RSS* de entrenamiento solo se incrementará y lo hará a medida que la contracción aumenta.

- (b) Repeat (a) for test RSS.

La Respuesta es ii. Decresera inicialmente, para posteriormente incrementar dibujando una forma de U. Al incrementar λ , el modelo se hace menos flexible, por lo que el aumento del sesgo superara a la disminución de la varianza a medida que se restringen los coeficientes β_j

- (c) Repeat (a) for variance.

La Respuesta es iv. Disminuye de manera constante. Al incrementar λ disminuye la flexibilidad del modelo al restringir los coeficientes β_j reduciendo de manera constante la varianza, y por ende acercando los coeficientes estimados $\hat{\beta}$ a cero aproximándonos al modelo nulo.

- (d) Repeat (a) for (squared) bias.

La Respuesta es iii. Incrementa de manera constante, pues al igual que en la pregunta anterior al incrementar λ acercando los coeficientes estimados $\hat{\beta}$ a cero aproximándonos al modelo nulo.

- (e) Repeat (a) for the irreducible error.

La Respuesta es v. Dado que el error irreducible es independiente del modelo. Permanece constante independientemente de la flexibilidad del modelo, ya que puede haber variables no medidas que no estén entre las predictoras y que serían necesarias para explicarlo, o una variación no medible en la variables respuesta que no puede predecirse con las variables predictoras.

5. It is well-known that ridge regression tends to give similar coefficient values to correlated variables, whereas the lasso may give quite different coefficient values to correlated variables. We will now explore this property in a very simple setting. Suppose that $n = 2$, $p = 2$, $x_{11} = x_{12}$, $x_{21} = x_{22}$. Furthermore, suppose that $y_1 + y_2 = 0$ and $x_{11} + x_{21} = 0$ and $x_{12} + x_{22} = 0$, so that the estimate for the intercept in a least squares, ridge regression, or lasso model is zero: $\hat{\beta}_0 = 0$.

- (a) Write out the ridge regression optimization problem in this setting.

$$(y_1 - \hat{\beta}_1 x_1 - \hat{\beta}_2 x_2)^2 + (y_2 - \hat{\beta}_1 x_2 - \hat{\beta}_2 x_1)^2 + \lambda(\hat{\beta}_1^2 + \hat{\beta}_2^2)$$

- (b) Argue that in this setting, the ridge coefficient estimates satisfy $\hat{\beta}_1 = \hat{\beta}_2$

$$(y_1 - \hat{\beta}_1 x_1 - \hat{\beta}_2 x_1)^2 + (y_2 - \hat{\beta}_1 x_2 - \hat{\beta}_2 x_2)^2 + \lambda(\hat{\beta}_1^2 + \hat{\beta}_2^2)$$

$$= (y_1^2 + \hat{\beta}_1^2 x_1^2 + \hat{\beta}_2^2 x_1^2 - 2\hat{\beta}_1 x_1 y_1 - 2\hat{\beta}_2 x_1 y_1 + 2\hat{\beta}_1 \hat{\beta}_2 x_1^2)$$

$$+ (y_2^2 + \hat{\beta}_1^2 x_2^2 + \hat{\beta}_2^2 x_2^2 - 2\hat{\beta}_1 x_2 y_2 - 2\hat{\beta}_2 x_2 y_2 + 2\hat{\beta}_1 \hat{\beta}_2 x_2^2) + \lambda \hat{\beta}_1^2 + \lambda \hat{\beta}_2^2$$

Ahora derivando parcialmente con respecto a $\hat{\beta}_1$

$$\frac{\partial}{\partial \hat{\beta}_1} = (2\hat{\beta}_1 x_1^2 - 2x_1 y_1 + 2\hat{\beta}_2 x_1^2) + (2\hat{\beta}_1 x_2^2 - 2x_2 y_2 + 2\hat{\beta}_2 x_2^2) + 2\lambda \hat{\beta}_1 = 0$$

Dividiendo por dos toda la ecuación y despejando $\hat{\beta}_1$ y $\hat{\beta}_2$:

$$(\hat{\beta}_1 x_1^2 - x_1 y_1 + \hat{\beta}_2 x_1^2) + (\hat{\beta}_1 x_2^2 - x_2 y_2 + \hat{\beta}_2 x_2^2) + \lambda \hat{\beta}_1 = 0$$

$$\hat{\beta}_1(x_1^2 + x_2^2) + \hat{\beta}_2(x_1^2 + x_2^2) + \lambda \hat{\beta}_1 = x_1 y_1 + x_2 y_2$$

De forma similar tomamos la derivada parcial respecto a $\hat{\beta}_2$

$$\hat{\beta}_1(x_1^2 + x_2^2) + \hat{\beta}_2(x_1^2 + x_2^2) + \lambda \hat{\beta}_2 = x_1 y_1 + x_2 y_2$$

Al restar las dos ecuaciones se obtiene que: $\hat{\beta}_1 = \hat{\beta}_2$

- (c) Write out the lasso optimization problem in this setting.

En Lasso se debe minimizar la siguiente expresión:

$$(y_1 - \hat{\beta}_1 x_1 - \hat{\beta}_2 x_2)^2 + (y_2 - \hat{\beta}_1 x_2 - \hat{\beta}_2 x_1)^2 + \lambda(|\hat{\beta}_1| + |\hat{\beta}_2|)$$

- (d) Argue that in this setting, the lasso coefficients $\hat{\beta}_1$ and $\hat{\beta}_2$ are not unique—in other words, there are many possible solutions to the optimization problem in (c). Describe these solutions.

Tomando en cuenta la ecuación para las derivadas parciales correspondientes a la coeficientes de las variables explicativas, tenemos:

$$\frac{\partial}{\partial \hat{\beta}}(\lambda|\beta|) : \lambda \frac{|\beta|}{\beta}$$

Además como $\hat{\beta}_1 = \hat{\beta}_2$,

$$\lambda \frac{|\beta_1|}{\beta_1} = \lambda \frac{|\beta_2|}{\beta_2}$$

Por lo tanto, el problema de optimización del lazo tiene un conjunto completo de soluciones en lugar de una única respuesta, solo requiere que los β_i sean distintos de cero.

6. We will now explore (6.12) and (6.13) further.
- (a) Consider (6.12) with $p = 1$. For some choice of y_1 and $\lambda > 0$, plot (6.12) as a function of β_1 . Your plot should confirm that (6.12) is solved by (6.14).

(6.12)

$$\sum_{i=1}^p (y_j - \beta_j)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

```
y <- 5
lambda <- 3
beta <- seq(-5, 7.4, 0.01)
ridge <- (y - beta)^2 + lambda * beta^2 ## Ecuación (6.12)
```

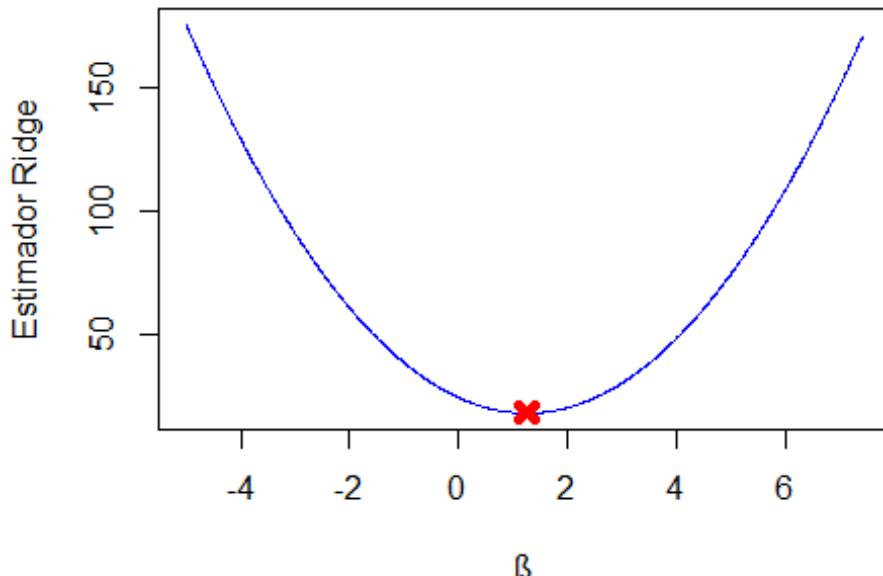
(6.14)

$$\hat{\beta}_j^R = y_j / (1 + \lambda)$$

```
beta.est <- y / (1 + lambda) ## Ecuación (6.14)
rigde.est <- (y - beta.est)^2 + lambda * beta.est^2

plot(beta, ridge, type="l", xlab = "\beta", ylab = "Estimador Ridge", col = "blue",
     main = "Optimización Ridge")
points(beta.est, rigde.est, col = "red", pch = 4, lwd = 5)
```

Optimización Ridge



- (b) Consider (6.13) with $p = 1$. For some choice of y_1 and $\lambda > 0$, plot (6.13) as a function of β_1 . Your plot should confirm that (6.13) is solved by (6.15).

(6.13)

$$\sum_{i=1}^p (y_j - \beta_j)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

```
y <- 5
lambda <- 3
beta <- seq(-5, 13.5, 0.01)
lasso <- (y - beta)^2 + lambda * abs(beta) ## Ecuación (6.13)
```

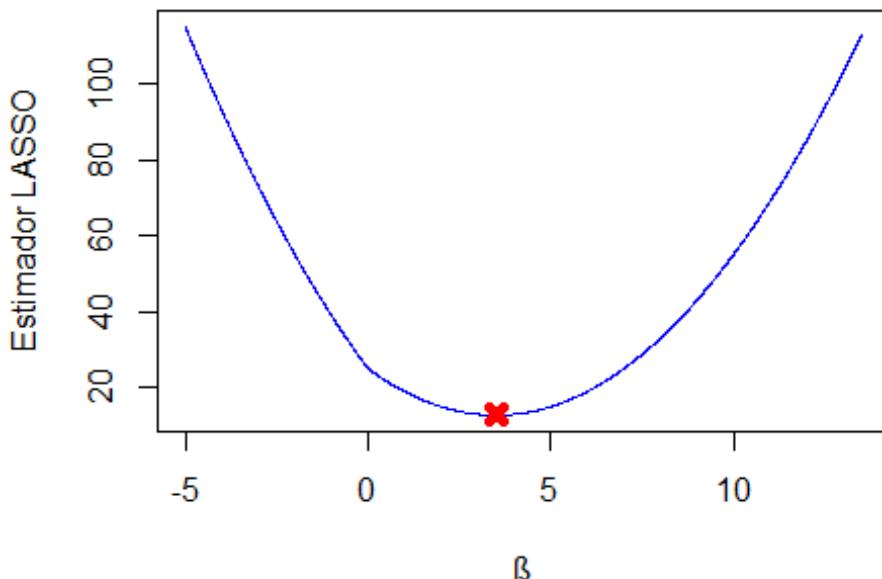
(6.15)

$$\hat{\beta}_j^L = \begin{cases} y_j - \lambda/2 & \text{if } y_j > \lambda/2; \\ y_j + \lambda/2 & \text{if } y_j < -\lambda/2; \\ 0 & \text{if } |y_j| \leq \lambda/2. \end{cases}$$

```
beta.est <- y - lambda/2
lasso.est <- (y - beta.est)^2 + lambda * abs(beta.est)

plot(beta, lasso, type="l", xlab = "\beta", ylab = "Estimador LASSO", main = "Optimización LASSO ", col="blue")
points(beta.est, lasso.est, col = "red", pch = 4, lwd = 5)
```

Optimización LASSO



7. We will now derive the Bayesian connection to the lasso and ridge regression discussed in Section 6.2.2.

- (a) Suppose that $y_i = \beta_0 + \sum_{j=1}^p x_{ij}\beta_j + \varepsilon_i$ where $\varepsilon_1, \dots, \varepsilon_n$ are independent and identically distributed from a $N(0, \sigma^2)$ distribution. Write out the likelihood for the data.

La verosimilitud para nuestros datos esta dada por:

$$\begin{aligned} L(\theta | \beta) &= p(\beta | \theta) \\ &= p(\beta_1 | \theta) \times \cdots \times p(\beta_n | \theta) \\ &= \prod_{i=1}^n p(\beta_i | \theta) \\ &= \prod_{i=1}^n \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{Y_i - (\beta_0 + \sum_{j=1}^p \beta_j X_{ij})}{2\sigma^2}\right) \\ &= \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n \left[Y_i - (\beta_0 + \sum_{j=1}^p \beta_j X_{ij})\right]^2\right) \end{aligned}$$

- (b) Assume the following prior for $\beta : \beta_1, \dots, \beta_p$ are independent and identically distributed according to a double-exponential distribution with mean 0 and common scale parameter b: i.e. $p(\beta) = \frac{1}{2b} \exp(-|\beta|/b)$. Write out the posterior for β in this setting.

Teniendo en cuenta que:

$$f(\beta | X, Y) \propto f(Y | X, \beta)p(\beta | X) = f(Y | X, \beta)p(\beta)$$

Sustituyendo nuestros valores en (a) y nuestra función de densidad tenemos que:

$$\begin{aligned} f(Y | X, \beta)p(\beta) &= \\ &\left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n \left[Y_i - (\beta_0 + \sum_{j=1}^p \beta_j X_{ij})\right]^2\right) \left(\frac{1}{2b} \exp(-|\beta|/b)\right) \\ &= \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n \left(\frac{1}{2b}\right) \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n \left[Y_i - (\beta_0 + \sum_{j=1}^p \beta_j X_{ij})\right]^2 - \frac{|\beta|}{b}\right) \end{aligned}$$

- (c) Argue that the lasso estimate is the mode for β under this posterior distribution.

Haciendo uso de la ecuación (6.7) y haciendo uso de la verosimilitud, podemos demostrar que la estimación es la moda para β es igual que mostrar que β esta dada por un λ determinado para el método LASSO.

(6.7)

$$\sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

Al tomar el logaritmo a ambos lado de la ecuación, tenemos que:

$$\begin{aligned} & \log f(Y | X, \beta) p(\beta) = \\ & \log \left[\left(\frac{1}{\sigma \sqrt{2\pi}} \right)^n \left(\frac{1}{2b} \right) \exp \left(-\frac{1}{2\sigma^2} \sum_{i=1}^n \left[Y_i - (\beta_0 + \sum_{j=1}^p \beta_j X_{ij}) \right]^2 - \frac{|\beta|}{b} \right) \right] \\ & = \log \left[\left(\frac{1}{\sigma \sqrt{2\pi}} \right)^n \left(\frac{1}{2b} \right) \right] - \left(\frac{1}{2\sigma^2} \sum_{i=1}^n \left[Y_i - (\beta_0 + \sum_{j=1}^p \beta_j X_{ij}) \right]^2 + \frac{|\beta|}{b} \right) \end{aligned}$$

Queremos maximizar el posterior para β , esto significa:

$$\begin{aligned} & \arg \max_{\beta} f(\beta | X, Y) = \\ & \arg \max_{\beta} \log \left[\left(\frac{1}{\sigma \sqrt{2\pi}} \right)^n \left(\frac{1}{2b} \right) \right] - \left(\frac{1}{2\sigma^2} \sum_{i=1}^n \left[Y_i - (\beta_0 + \sum_{j=1}^p \beta_j X_{ij}) \right]^2 + \frac{|\beta|}{b} \right) \end{aligned}$$

Como estamos tomando la diferencia de dos valores, el máximo de este valor es el equivalente a tomar la diferencia del segundo valor en términos de β .

$$\begin{aligned} & = \arg \min_{\beta} \frac{1}{2\sigma^2} \sum_{i=1}^n \left[Y_i - (\beta_0 + \sum_{j=1}^p \beta_j X_{ij}) \right]^2 + \frac{|\beta|}{b} \\ & = \arg \min_{\beta} \frac{1}{2\sigma^2} \sum_{i=1}^n \left[Y_i - (\beta_0 + \sum_{j=1}^p \beta_j X_{ij}) \right]^2 + \frac{1}{b} \sum_{j=1}^p |\beta_j| \\ & = \arg \min_{\beta} \frac{1}{2\sigma^2} \left(\sum_{i=1}^n \left[Y_i - (\beta_0 + \sum_{j=1}^p \beta_j X_{ij}) \right]^2 + \frac{2\sigma^2}{b} \sum_{j=1}^p |\beta_j| \right) \end{aligned}$$

De este modo cuando $\lambda = \frac{2\sigma^2}{b}$:

$$\begin{aligned} & = \arg \min_{\beta} \sum_{i=1}^n \left[Y_i - (\beta_0 + \sum_{j=1}^p \beta_j X_{ij}) \right]^2 + \lambda \sum_{j=1}^p |\beta_j| \\ & = \arg \min_{\beta} \text{RSS} + \lambda \sum_{j=1}^p |\beta_j| \end{aligned}$$

Cuando el LASSO proviene de una distribución de Laplace con media cero y escala común b , la moda par β esta dada por $\lambda = \frac{2\sigma^2}{b}$.

- (d) Now assume the following prior for $\beta : \beta_1, \dots, \beta_p$ are independent and identically distributed according to a normal distribution with mean zero and variance c . Write out the posterior for β in this setting.

La posterior β distribuida según la distribución Normal con media 0 y varianza c es:

$$f(\beta | X, Y) \propto f(Y | X, \beta)p(\beta | X) = f(Y | X, \beta)p(\beta)$$

Nuestra función de distribución de probabilidad es:

$$p(\beta) = \prod_{i=1}^p p(\beta_i) = \prod_{i=1}^p \frac{1}{\sqrt{2c\pi}} \exp\left(-\frac{\beta_i^2}{2c}\right) = \left(\frac{1}{\sqrt{2c\pi}}\right)^p \exp\left(-\frac{1}{2c} \sum_{i=1}^p \beta_i^2\right)$$

Sustituyendo(a) y nuestra función de densidad nos da:

$$\begin{aligned} f(Y | X, \beta)p(\beta) &= \\ &\left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n \left[Y_i - (\beta_0 + \sum_{j=1}^p \beta_j X_{ij})\right]^2\right) \left(\frac{1}{\sqrt{2c\pi}}\right)^p \exp\left(-\frac{1}{2c} \sum_{i=1}^p \beta_i^2\right) \\ &= \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n \left(\frac{1}{\sqrt{2c\pi}}\right)^p \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n \left[Y_i - (\beta_0 + \sum_{j=1}^p \beta_j X_{ij})\right]^2 - \frac{1}{2c} \sum_{i=1}^p \beta_i^2\right) \end{aligned}$$

- (e) Argue that the ridge regression estimate is both the *mode* and the *mean* for β under this posterior distribution.

Al igual que en la parte c, mostrar que la estimación de la regresión Ridge para β es la moda y la media bajo esta distribución posterior es lo mismo que mostrar que el valor más probable para β viene dado por la solución del lazo con una determinada λ .

Podemos hacer esto tomando nuestra verosimilitud y posterior y mostrando que se puede reducir a la ecuación canónica de regresión Ridge 6.5 del libro.

(6.5)

$$\sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij}\right)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

Al tomar el logaritmo de ambos lados para simplificarlo:

$$\begin{aligned} & \log f(Y | X, \beta) p(\beta) = \\ & \left(\frac{1}{\sigma \sqrt{2\pi}} \right)^n \left(\frac{1}{\sqrt{2c\pi}} \right)^p \exp \left(-\frac{1}{2\sigma^2} \sum_{i=1}^n \left[Y_i - (\beta_0 + \sum_{j=1}^p \beta_j X_{ij}) \right]^2 - \frac{1}{2c} \sum_{i=1}^p \beta_i^2 \right) \\ & = \log \left[\left(\frac{1}{\sigma \sqrt{2\pi}} \right)^n \left(\frac{1}{\sqrt{2c\pi}} \right)^p \right] - \left(\frac{1}{2\sigma^2} \sum_{i=1}^n \left[Y_i - (\beta_0 + \sum_{j=1}^p \beta_j X_{ij}) \right]^2 + \frac{1}{2c} \sum_{i=1}^p \beta_i^2 \right) \end{aligned}$$

Queremos maximizar el posterior, esto significa:

$$\begin{aligned} & \arg \max_{\beta} f(\beta | X, Y) = \\ & \arg \max_{\beta} \log \left[\left(\frac{1}{\sigma \sqrt{2\pi}} \right)^n \left(\frac{1}{\sqrt{2c\pi}} \right)^p \right] - \left(\frac{1}{2\sigma^2} \sum_{i=1}^n \left[Y_i - (\beta_0 + \sum_{j=1}^p \beta_j X_{ij}) \right]^2 + \frac{1}{2c} \sum_{i=1}^p \beta_i^2 \right) \end{aligned}$$

Como estamos tomando la diferencia de dos valores, el máximo de este valor es el equivalente a tomar la diferencia del segundo valor en términos de β .

$$\begin{aligned} & = \arg \min_{\beta} \left(\frac{1}{2\sigma^2} \sum_{i=1}^n \left[Y_i - (\beta_0 + \sum_{j=1}^p \beta_j X_{ij}) \right]^2 + \frac{1}{2c} \sum_{i=1}^p \beta_i^2 \right) \\ & = \arg \min_{\beta} \left(\frac{1}{2\sigma^2} \right) \left(\sum_{i=1}^n \left[Y_i - (\beta_0 + \sum_{j=1}^p \beta_j X_{ij}) \right]^2 + \frac{\sigma^2}{c} \sum_{i=1}^p \beta_i^2 \right) \end{aligned}$$

Dejando que $\lambda = \frac{2}{c}$, finalmente tenemos que:

$$\begin{aligned} & = \arg \min_{\beta} \left(\frac{1}{2\sigma^2} \right) \left(\sum_{i=1}^n \left[Y_i - (\beta_0 + \sum_{j=1}^p \beta_j X_{ij}) \right]^2 + \lambda \sum_{i=1}^p \beta_i^2 \right) \\ & = \arg \min_{\beta} \text{RSS} + \lambda \sum_{i=1}^p \beta_i^2 \end{aligned}$$

Es decir que cuando el posterior proviene de la distribución $N(0, c)$ la moda esta dada por $\lambda = \frac{\sigma^2}{c}$.

8. 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 of length n = 100, as well as a noise vector of length n = 100.

```
library(leaps)

set.seed(123)
x <- rnorm(100)
epsilon <- rnorm(100)
```

- (b) Generate a response vector Y of length n = 100 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.

```
#Se definen los beta
B0 <- 2
B1 <- 4
B2 <- 6
B3 <- 8

# Calculando y
y <- B0 + B1*x + B2*x^2 + B3*x^3 + epsilon
head(y)

## [1] -0.4760224 1.5565023 52.8615188 1.9671238 1.6831130 66.8221142
```

- (c) Use the regsubsets() function to perform best subset selection in order to choose the best model containing the predictors X, X_2, \dots, X_{10} . What is the best model obtained according to C_p , BIC , and adjusted R^2 ? Show some plots to provide evidence for your answer, and report the coefficients of the best model obtained. Note you will need to use the data.frame() function to create a single data set containing both X and Y .

```
regfit.full <- regsubsets(y~poly(x,10,raw=T), data=data.frame(y,x), nvmax=10)
reg.summary <- summary(regfit.full)

par(mfrow=c(1,3))

#Cp
min.cp <- which.min(reg.summary$cp)
plot(reg.summary$cp, xlab="Número de variables", ylab="Cp", type="l")
points(min.cp, reg.summary$cp[min.cp], col="red", pch=4, lwd=5)

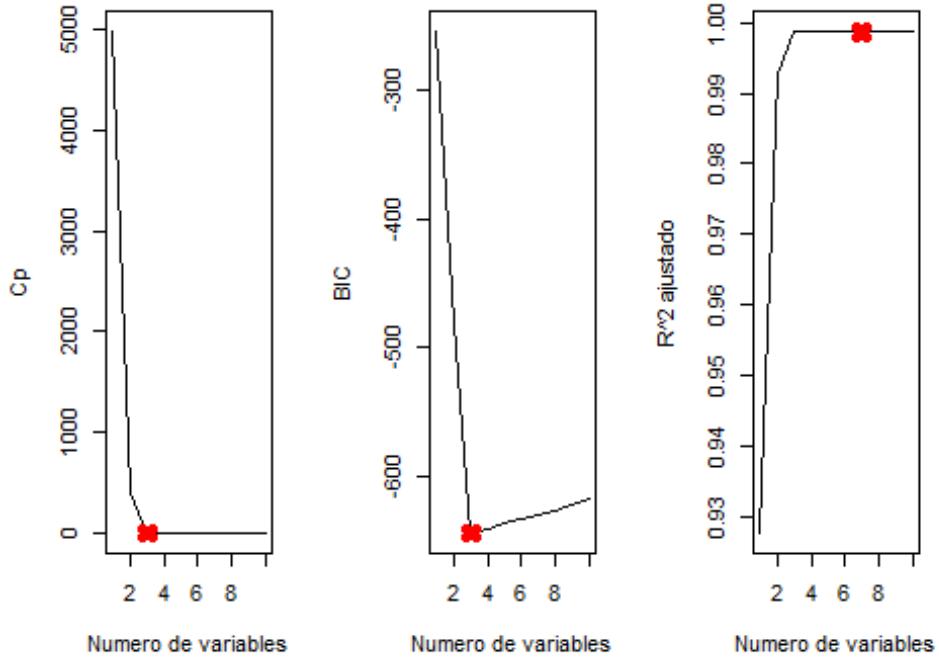
#BIC
min.bic <- which.min(reg.summary$bic)
plot(reg.summary$bic, xlab="Número de variables", ylab="BIC", type="l")
points(min.bic, reg.summary$bic[min.bic], col="red", pch=4, lwd=5)

#R^2 ajustado
```

```

max.adjr2 <- which.max(reg.summary$adjr2)
plot(reg.summary$adjr2, xlab="Numero de variables", ylab="R^2 ajustado", type
="l")
points(max.adjr2, reg.summary$adjr2[max.adjr2], col="red", pch=4, lwd=5)

```



Vemos que los criterios C_p y BIC seleccionan el modelo con 3 variables, mientras que el R^2 ajustado selecciona el modelo con 7 variables.

Según el C_p y el BIC los coeficientes para las variables son.

```

coef(regfit.full, id = 3 )

##           (Intercept) poly(x, 10, raw = T)1 poly(x, 10, raw = T)2
##             1.970394          3.920446          5.908457
## poly(x, 10, raw = T)3
##             8.020436

```

Según el R^2 ajustado los coeficientes para las variables son.

```

coef(regfit.full, id=7)

##           (Intercept) poly(x, 10, raw = T)1 poly(x, 10, raw = T)2
##             1.7931750          3.8907484          8.2758848
## poly(x, 10, raw = T)3 poly(x, 10, raw = T)4 poly(x, 10, raw = T)6
##             8.0530514          -4.0243410          2.2138377
## poly(x, 10, raw = T)8 poly(x, 10, raw = T)10
##            -0.4870719          0.0373538

```

- (d) Repeat (c), using forward stepwise selection and using backwards stepwise selection. How does your answer compare to the results in (c)?

```
# forward stepwise
regfit.fwd <- regsubsets(y~poly(x,10,raw=T), data=data.frame(y,x), nvmax=10)
fwd.summary <- summary(regfit.fwd)

# backward stepwise
regfit.bwd <- regsubsets(y~poly(x,10,raw=T), data=data.frame(y,x), nvmax=10)
bwd.summary <- summary(regfit.bwd)
```

Selección mediante criterio C_p para backward y forward

```
cpb <- which.min(bwd.summary$cp)
cpf <- which.min(fwd.summary$cp)

cat("Número de variables según  $C_p$ : \n")
## Número de variables según  $C_p$ :
cat("bwd\t", "fwd\n")

## bwd    fwd
cat(cpb, "\t", cpf)
## 3      3
```

Selección mediante criterio BIC para backward y forward

```
bpb <- which.min(bwd.summary$bic)
bpf <- which.min(fwd.summary$bic)

cat("Número de variables según  $BIC$ : \n")
## Número de variables según  $BIC$ :
cat("bwd\t", "fwd\n")

## bwd    fwd
cat(bpb, "\t", bpf)
## 3      3
```

Selección mediante criterio R^2 ajustado para backward y forward

```
rpb <- which.max(bwd.summary$adjr2)
rpf <- which.max(fwd.summary$adjr2)

cat("Número de variables según  $R^2$  ajustado: \n")
## Número de variables según  $R^2$  ajustado:
```

```

cat("bwd\t","fwd\n")
## bwd    fwd

cat(rpb, "\t",rpf)
## 7      7

par(mfrow=c(3,2))

#Cp forward
plot(fwd.summary$cp, xlab="Numero de variables", ylab="Cp", type="l", main="Cp Forward stepwise")
points(cpf, fwd.summary$cp[cpf], col="red", pch=4, lwd=5)

#Cp backward
plot(bwd.summary$cp, xlab="Numero de variables", ylab="Cp", type="l", main="Cp Backward stepwise")
points(cpb, bwd.summary$cp[cpb], col="red", pch=4, lwd=5)

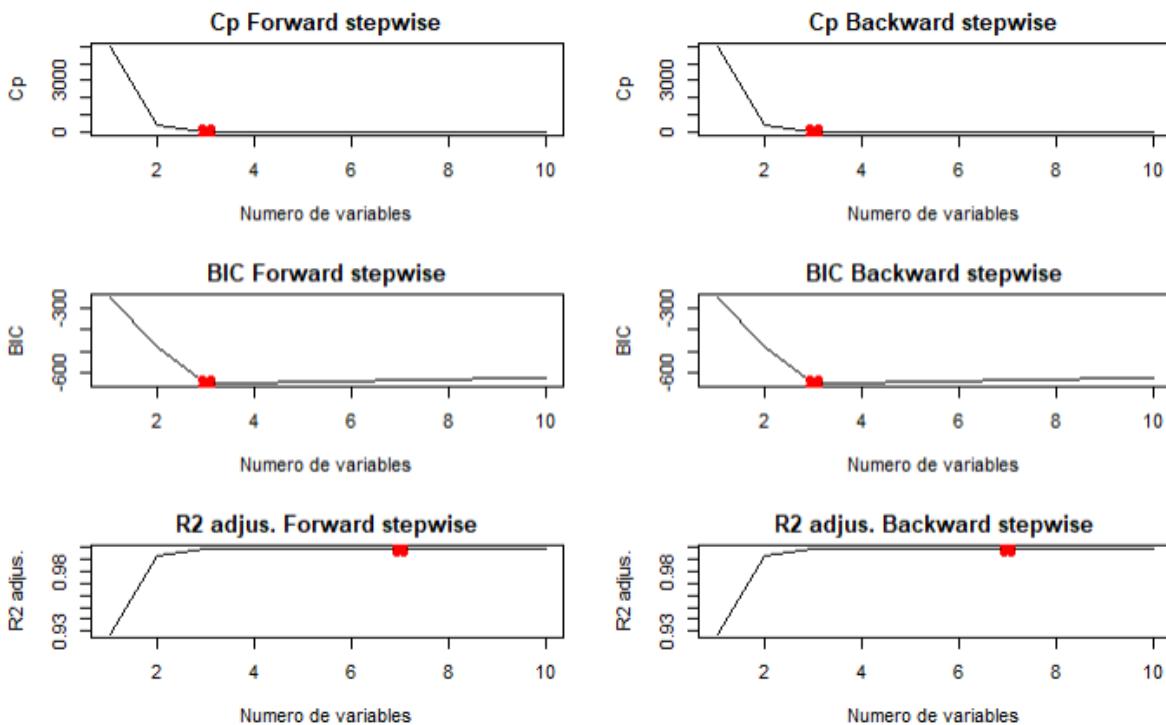
#BIC forward
plot(fwd.summary$bic, xlab="Numero de variables", ylab="BIC", type="l", main="BIC Forward stepwise")
points(bpf, fwd.summary$bic[bpf], col="red", pch=4, lwd=5)

#BIC backward
plot(bwd.summary$bic, xlab="Numero de variables", ylab="BIC", type="l", main="BIC Backward stepwise")
points(bp, bwd.summary$bic[bp], col="red", pch=4, lwd=5)

#R2 adjus forward
plot(fwd.summary$adjr2, xlab="Numero de variables", ylab="R2 adjus.", type="l", main="R2 adjus. Forward stepwise")
points(rpf, fwd.summary$adjr2[rpf], col="red", pch=4, lwd=5)

#R2 adjus backward
plot(bwd.summary$adjr2, xlab="Numero de variables", ylab="R2 adjus.", type="l", main="R2 adjus. Backward stepwise")
points(rpb, bwd.summary$adjr2[rpb], col="red", pch=4, lwd=5)

```



Según el CP y el BIC mediante el **forward stepwise** los coeficientes son:

```
coefficients(regfit.fwd, 3)
##              (Intercept) poly(x, 10, raw = T)1 poly(x, 10, raw = T)2
##            1.970394          3.920446          5.908457
## poly(x, 10, raw = T)3
##            8.020436
```

Según el CP y el BIC mediante el **backward stepwise** los coeficientes son:

```
coefficients(regfit.bwd, 3)
##              (Intercept) poly(x, 10, raw = T)1 poly(x, 10, raw = T)2
##            1.970394          3.920446          5.908457
## poly(x, 10, raw = T)3
##            8.020436
```

Según el R^2 ajustado mediante el **backward stepwise** y el **forward stepwise** los coeficientes son:

```
coefficients(regfit.bwd, 7)
##              (Intercept) poly(x, 10, raw = T)1 poly(x, 10, raw = T)2
##            1.7931750          3.8907484          8.2758848
## poly(x, 10, raw = T)3 poly(x, 10, raw = T)4 poly(x, 10, raw = T)6
##            8.0530514          -4.0243410          2.2138377
## poly(x, 10, raw = T)8 poly(x, 10, raw = T)10
##           -0.4870719          0.0373538
```

- (e) Now fit a lasso model to the simulated data, again using X, X_2, \dots, X_{10} as 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.

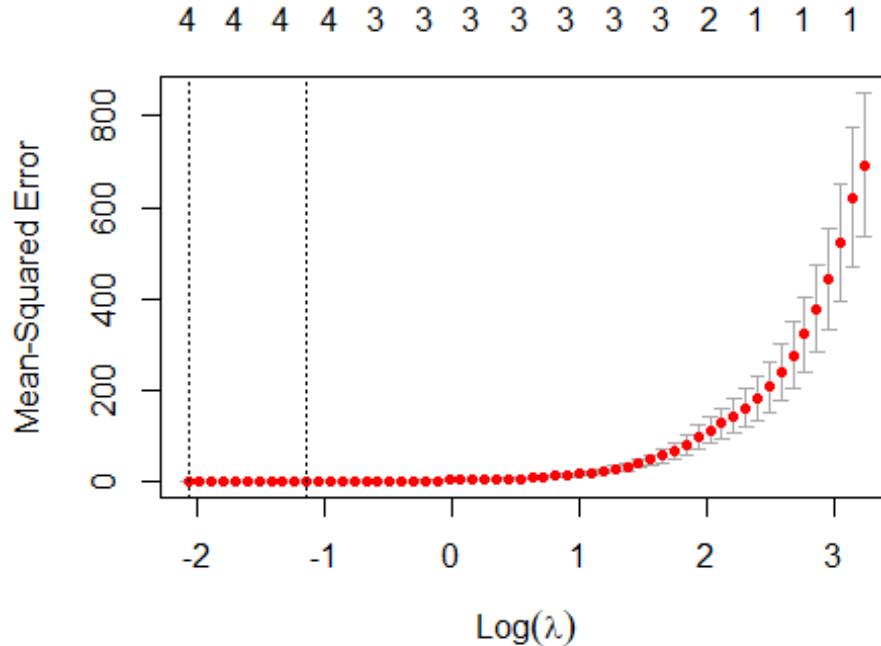
```
library(glmnet)

## Loading required package: Matrix

## Loaded glmnet 4.1-1

xmat = model.matrix(y ~ poly(x, 10, raw = T), data = data.frame(y,x))[, -1]
mod.lasso = cv.glmnet(xmat, y, alpha = 1)

plot(mod.lasso)
```



El mejor λ para nuestro modelo es:

```
best.lambda = mod.lasso$lambda.min
best.lambda

## [1] 0.1256765
```

Se ajusta el modelo para todos los datos usando el mejor λ .

```
best.model = glmnet(xmat, y, alpha = 1)
predict(best.model, s = best.lambda, type = "coefficients")

## 11 x 1 sparse Matrix of class "dgCMatrix"
##                                     1
## (Intercept)      2.10602812
## poly(x, 10, raw = T)1 3.85015134
## poly(x, 10, raw = T)2 5.67123875
## poly(x, 10, raw = T)3 8.00222022
## poly(x, 10, raw = T)4 0.03683589
## poly(x, 10, raw = T)5 .
## poly(x, 10, raw = T)6 .
## poly(x, 10, raw = T)7 .
## poly(x, 10, raw = T)8 .
## poly(x, 10, raw = T)9 .
## poly(x, 10, raw = T)10 .
```

El metodo LASSO selecciona las variables X, X^2, X^3, X^4 para nuestro modelo.

- (f) 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 obtained.

```
B7 = 10
y = B0 + B7 * x^7 + epsilon

mod.full = regsubsets(y ~ poly(x, 10, raw = T), data = data.frame(y,x), nvmax = 10)
mod.summary = summary(mod.full)

CP <- which.min(mod.summary$cp)
BIC <- which.min(mod.summary$bic)
R2a <- which.max(mod.summary$adjr2)

cat("CP\t", "BIC\t", "R2 Adjust\n")

## CP      BIC      R2 Adjust
cat(CP, "\t", BIC, "\t", R2a)

## 1      1      6
```

Tanto el CP como el BIC seleccionan el modelo con menos una variable

```
coefficients(mod.full, id = 1)

## (Intercept) poly(x, 10, raw = T)7
## 1.893251      9.999704
```

Generando el modelo LASSO.

```
xmat = model.matrix(y ~ poly(x, 10, raw = T), data = data.frame(y,x))[, -1]
mod.lasso = cv.glmnet(xmat, y, alpha = 1)
```

Cuyo mejor *lambda* es:

```
best.lambda = mod.lasso$lambda.min
best.lambda

## [1] 15.28987

best.model = glmnet(xmat, y, alpha = 1)
predict(best.model, s = best.lambda, type = "coefficients")

## 11 x 1 sparse Matrix of class "dgCMatrix"
##           1
## (Intercept) 2.680362
## poly(x, 10, raw = T)1 .
## poly(x, 10, raw = T)2 .
## poly(x, 10, raw = T)3 .
## poly(x, 10, raw = T)4 .
## poly(x, 10, raw = T)5 .
## poly(x, 10, raw = T)6 .
## poly(x, 10, raw = T)7 9.708208
## poly(x, 10, raw = T)8 .
## poly(x, 10, raw = T)9 .
## poly(x, 10, raw = T)10 .
```

Se evidencia que también se selecciona el modelo con 1-variable siendo esta X^7

9. In this exercise, we will predict the number of applications received using the other variables in the College data set.

```
library(ggplot2)
library(dplyr)
library(ISLR)
library(corrplot)
library(RColorBrewer)
library(car)
library(class)
library(MASS)
library(boot)
library(glmnet)
library(pls)
```

- (a) Split the data set into a training set and a test set.

```
#rm(list = ls())
data(College)
set.seed(1)
trainid <- sample(1:nrow(College), nrow(College)/2)
train <- College[trainid,]
test <- College[-trainid,]
str(College)

## 'data.frame':    777 obs. of  18 variables:
## $ Private      : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 ...
## $ Apps         : num  1660 2186 1428 417 193 ...
## $ Accept       : num  1232 1924 1097 349 146 ...
## $ Enroll       : num  721 512 336 137 55 158 103 489 227 172 ...
## $ Top10perc   : num  23 16 22 60 16 38 17 37 30 21 ...
## $ Top25perc   : num  52 29 50 89 44 62 45 68 63 44 ...
## $ F.Undergrad: num  2885 2683 1036 510 249 ...
## $ P.Undergrad: num  537 1227 99 63 869 ...
## $ Outstate     : num  7440 12280 11250 12960 7560 ...
## $ Room.Board   : num  3300 6450 3750 5450 4120 ...
## $ Books        : num  450 750 400 450 800 500 500 450 300 660 ...
## $ Personal     : num  2200 1500 1165 875 1500 ...
## $ PhD          : num  70 29 53 92 76 67 90 89 79 40 ...
## $ Terminal     : num  78 30 66 97 72 73 93 100 84 41 ...
## $ S.F.Ratio   : num  18.1 12.2 12.9 7.7 11.9 9.4 11.5 13.7 11.3 11.5 ...
## $ perc.alumni: num  12 16 30 37 2 11 26 37 23 15 ...
## $ Expend       : num  7041 10527 8735 19016 10922 ...
## $ Grad.Rate   : num  60 56 54 59 15 55 63 73 80 52 ...
```

Se realiza la división entre el conjunto de datos de entrenamiento y de prueba tomando la mitad de la base de datos para cada uno de los conjuntos.

- (b) Fit a linear model using least squares on the training set, and report the test error obtained.

```
mod.lm <- lm(Apps~., data=train)
pred.lm <- predict(mod.lm, test)
```

```

err.lm <- mean((test$Apps - pred.lm)^2)
err.lm

## [1] 1135758

```

El error de prueba MSE es 1135758

- (c) Fit a ridge regression model on the training set, with λ chosen by cross-validation. Report the test error obtained.

```

xmat.train <- model.matrix(Apps~., data=train)[,-1]
xmat.test <- model.matrix(Apps~., data=test)[,-1]
mod.ridge <- cv.glmnet(xmat.train, train$Apps, alpha=0)
lambda <- mod.ridge$lambda.min
pred.ridge <- predict(mod.ridge, s=lambda, newx=xmat.test)
err.ridge <- mean((test$Apps - pred.ridge)^2)
err.ridge

## [1] 976261.5

```

La prueba MSE es menor para la ‘Ridge’ regresión que para mínimos cuadrados.

- (d) Fit a lasso model on the training set, with λ chosen by crossvalidation. Report the test error obtained, along with the number of non-zero coefficient estimates.

```

require(glmnet)
xmat.train <- model.matrix(Apps~., data=train)[,-1]
xmat.test <- model.matrix(Apps~., data=test)[,-1]
mod.lasso <- cv.glmnet(xmat.train, train$Apps, alpha=1)
lambda <- mod.lasso$lambda.min
pred.lasso <- predict(mod.lasso, s=lambda, newx=xmat.test)
err.lasso <- mean((test$Apps - pred.lasso)^2)
coef.lasso <- predict(mod.lasso, type="coefficients", s=lambda)[1:ncol(College),]
coef.lasso[coef.lasso != 0]

## (Intercept) PrivateYes Accept Enroll Top10perc
## -7.688896e+02 -3.127034e+02 1.762718e+00 -1.318195e+00 6.482356e+01
## Top25perc F.Undergrad P.Undergrad Outstate Room.Board
## -2.081406e+01 7.119149e-02 1.246161e-02 -1.049091e-01 2.088305e-01
## Books Personal PhD Terminal S.F.Ratio
## 2.926466e-01 3.955068e-03 -1.455463e+01 5.395858e+00 2.171398e+01
## perc.alumni Expend Grad.Rate
## 5.088260e-01 4.824455e-02 7.036148e+00

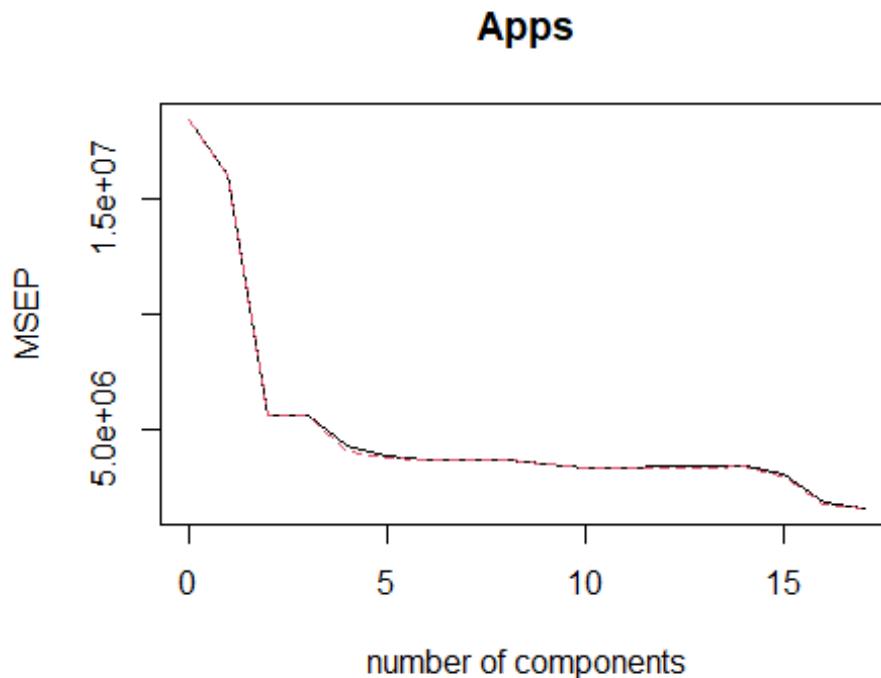
length(coef.lasso[coef.lasso != 0])
## [1] 18

```

Estos son los coeficientes obtenidos del modelo Lasso distintos de cero.

- (e) Fit a PCR model on the training set, with M chosen by crossvalidation. Report the test error obtained, along with the value of M selected by cross-validation.

```
set.seed(1)
mod.pcr <- pcr(Apps~., data=train, scale=TRUE, validation="CV")
validationplot(mod.pcr, val.type="MSEP")
```



```
summary(mod.pcr)

## Data: X dimension: 388 17
## Y dimension: 388 1
## Fit method: svdpc
## Number of components considered: 17
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##          (Intercept) 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps
## CV        4288     4006    2373    2372    2069    1961    1919
## adjCV    4288     4007    2368    2369    1999    1948    1911
##          7 comps 8 comps 9 comps 10 comps 11 comps 12 comps 13 comps
## CV        1919     1921    1876    1832    1832    1836    1837
## adjCV    1912     1915    1868    1821    1823    1827    1827
##          14 comps 15 comps 16 comps 17 comps
## CV        1853     1759    1341    1270
## adjCV    1850     1733    1326    1257
##
## TRAINING: % variance explained
##          1 comps 2 comps 3 comps 4 comps 5 comps 6 comps 7 comps 8 com
```

```

ps
## X      32.20    57.78    65.31    70.99    76.37    81.27    84.8     87.
85
## Apps   13.44    70.93    71.07    79.87    81.15    82.25    82.3     82.
33
##      9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
## X      90.62    92.91    94.98    96.74    97.79    98.72    99.42
## Apps   83.38    84.76    84.80    84.84    85.11    85.14    90.55
##      16 comps 17 comps
## X      99.88   100.00
## Apps   93.42    93.89

pred.pcr <- predict(mod.pcr, test, ncomp=16)
err.pcr <- mean((test$Apps - pred.pcr)^2)
err.pcr

## [1] 1137877

```

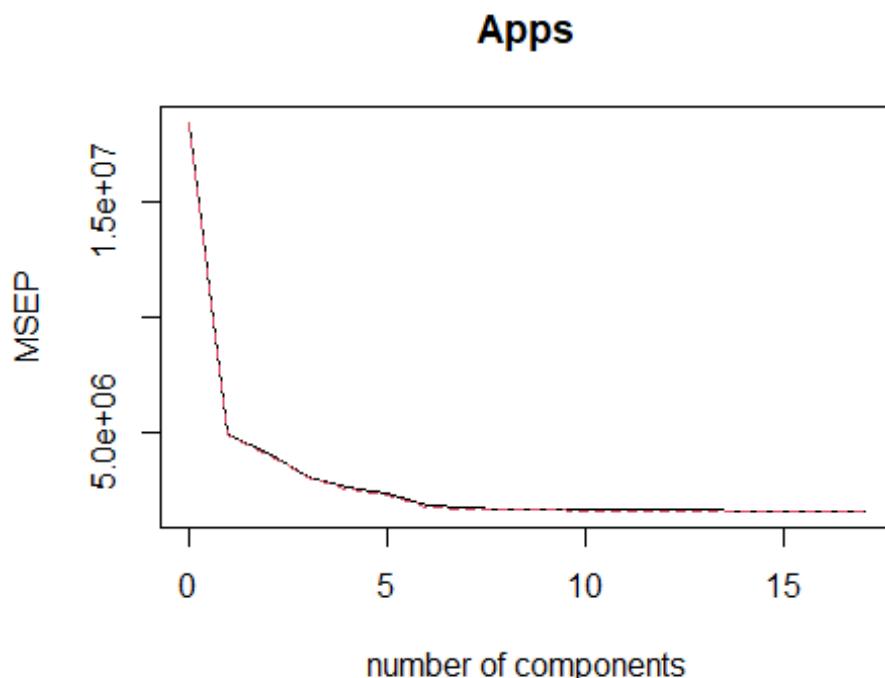
En este caso el error si es ligeramente mayor que para MSE, a diferencia de Lasso donde fue menor.

- (f) Fit a PLS model on the training set, with M chosen by crossvalidation. Report the test error obtained, along with the value of M selected by cross-validation.

```

set.seed(1)
mod.pls <- plsr(Apps~., data=train, scale=TRUE, validation="CV")
validationplot(mod.pls, val.type="MSEP")

```



```

summary(mod.pls)

## Data: X dimension: 388 17
## Y dimension: 388 1
## Fit method: kernelpls
## Number of components considered: 17
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##          (Intercept) 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps
## CV        4288     2217    2019    1761    1630    1533    1347
## adjCV    4288     2211    2012    1749    1605    1510    1331
##          7 comps 8 comps 9 comps 10 comps 11 comps 12 comps 13 comps
## CV        1309     1303    1286    1283    1283    1277    1271
## adjCV    1296     1289    1273    1270    1270    1264    1258
##          14 comps 15 comps 16 comps 17 comps
## CV        1270     1270    1270    1270
## adjCV    1258     1257    1257    1257
##
## TRAINING: % variance explained
##          1 comps 2 comps 3 comps 4 comps 5 comps 6 comps 7 comps 8 comps
## X        27.21    50.73   63.06   65.52   70.20   74.20   78.62   80.
## Apps    75.39    81.24   86.97   91.14   92.62   93.43   93.56   93.
##          9 comps 10 comps 11 comps 12 comps 13 comps 14 comps 15 comps
## X        83.29    87.17   89.15   91.37   92.58   94.42   96.98
## Apps    93.76    93.79   93.83   93.86   93.88   93.89   93.89
##          16 comps 17 comps
## X        98.78    100.00
## Apps    93.89    93.89

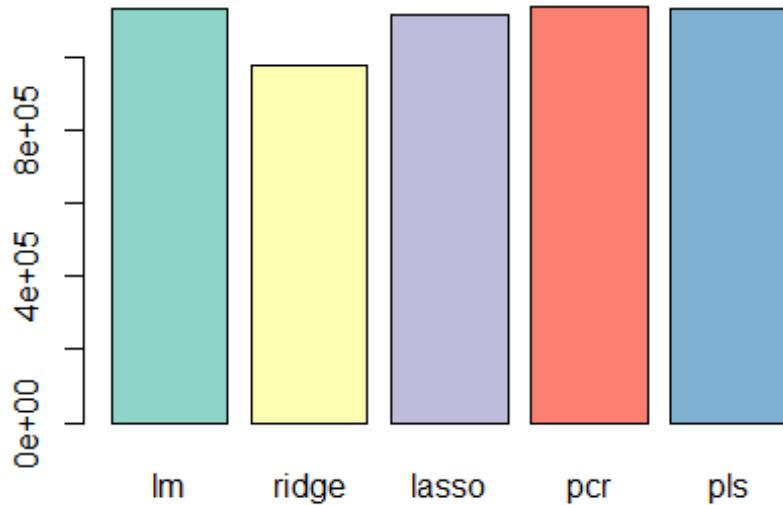
pred.pls <- predict(mod.pls, test, ncomp=10)
err.pls <- mean((test$Apps - pred.pls)^2)
err.pls

## [1] 1131661

```

- (g) Comment on the results obtained. How accurately can we predict the number of college applications received? Is there much difference among the test errors resulting from these five approaches?

```
err.all <- c(err.lm, err.ridge, err.lasso, err.pcr, err.pls)
names(err.all) <- c("lm", "ridge", "lasso", "pcr", "pls")
barplot(err.all, col = brewer.pal(5, "Set3"))
```



Como se puede observar las diferencias no son evidentes salvo, en el modelo 'ridge', que evidentemente resulta ser el que menos error presenta. El segundo modelo con menor error es Lasso, mientras que PCR y PLS no son muy convenientes en este caso al tener errores relativamente altos con respecto a los otros modelos.

10. We have seen that as the number of features used in a model increases, the training error will necessarily decrease, but the test error may not. We will now explore this in a simulated data set.

```
library(ggplot2)
library(dplyr)
library(ISLR)
library(corrplot)
library(RColorBrewer)
library(car)
library(class)
library(MASS)
library(boot)
library(glmnet)
library(pls)
library(leaps)
```

- (a) Generate a data set with $p = 20$ features, $n = 1,000$ observations, and an associated quantitative response vector generated according to the model.

where β has some elements that are exactly equal to zero.

```
set.seed(1)
eps <- rnorm(1000)
xmat <- matrix(rnorm(1000*20), ncol=20)
betas <- sample(-5:5, 20, replace=TRUE)
betas[c(3,6,7,10,13,17)] <- 0
betas

## [1] -3  5  0 -5  2  0  0  4  5  0  4  1  0  5 -3  4  0  4 -2  1

y <- xmat %*% betas + eps
```

- (b) Split your data set into a training set containing 100 observations and a test set containing 900 observations.

```
train <- sample(seq(1000), 100, replace = FALSE)
test <- -train
x.train <- xmat[train, ]
x.test <- xmat[test, ]
y.train <- y[train]
y.test <- y[test]
```

El conjunto de datos de entrenamiento ahora tiene 100 observaciones y el de prueba cuenta con las 900 observaciones requeridas.

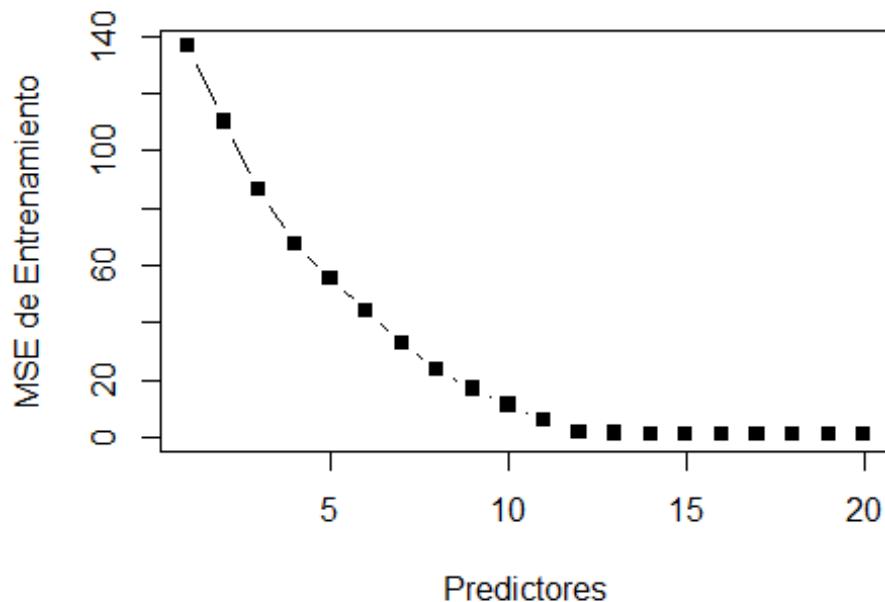
- (c) Perform best subset selection on the training set and plot the training set MSE associated with the best model of each size.

```
data.train <- data.frame(y = y.train, x = x.train)
mod.full <- regsubsets(y ~ ., data = data.train, nvmax = 20)
train.mat <- model.matrix(y ~ ., data = data.train, nvmax = 20)
val.errors <- rep(NA, 20)
```

```

for (i in 1:20) {
  coefi <- coef(mod.full, id = i)
  pred <- train.mat[, names(coefi)] %*% coefi
  val.errors[i] <- mean((pred - y.train)^2)
}
plot(val.errors, xlab = "Predictores", ylab = "MSE de Entrenamiento", pch = 15, type = "b")

```

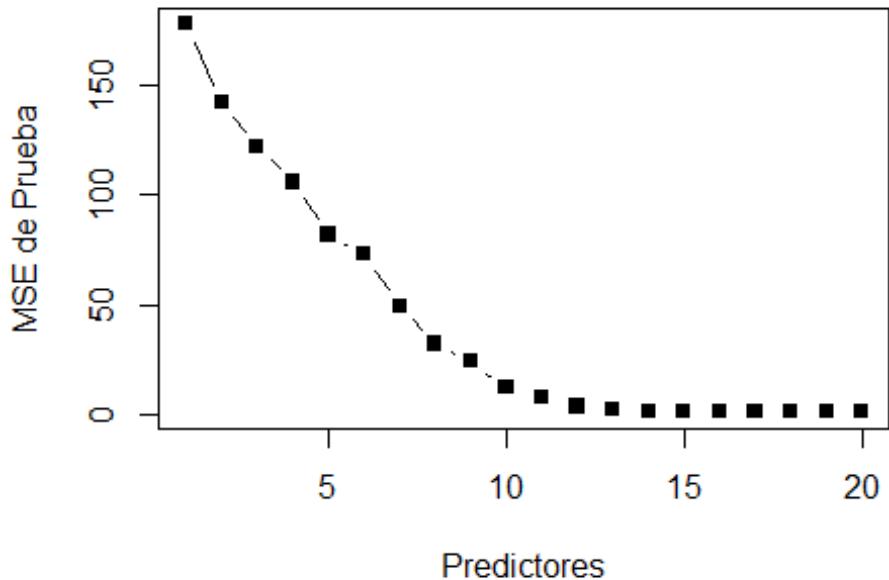


(d) Plot the test set MSE associated with the best model of each size.

```

data.test <- data.frame(y = y.test, x = x.test)
test.mat <- model.matrix(y ~ ., data = data.test, nvmax = 20)
val.errors <- rep(NA, 20)
for (i in 1:20) {
  coefi <- coef(mod.full, id = i)
  pred <- test.mat[, names(coefi)] %*% coefi
  val.errors[i] <- mean((pred - y.test)^2)
}
plot(val.errors, xlab = "Predictores", ylab = "MSE de Prueba", pch = 15, type = "b")

```



- (e) For which model size does the test set MSE take on its minimum value? Comment on your results. If it takes on its minimum value for a model containing only an intercept or a model containing all of the features, then play around with the way that you are generating the data in (a) until you come up with a scenario in which the test set MSE is minimized for an intermediate model size.

Mediante la función ‘Which.min()’ se determina el mínimo del vector asociado al error para saber con cuantas de las variables dicho error disminuye.

```
which.min(val.errors)
## [1] 14
```

El menor error se obtiene del modelo con 14 variables o predictores.

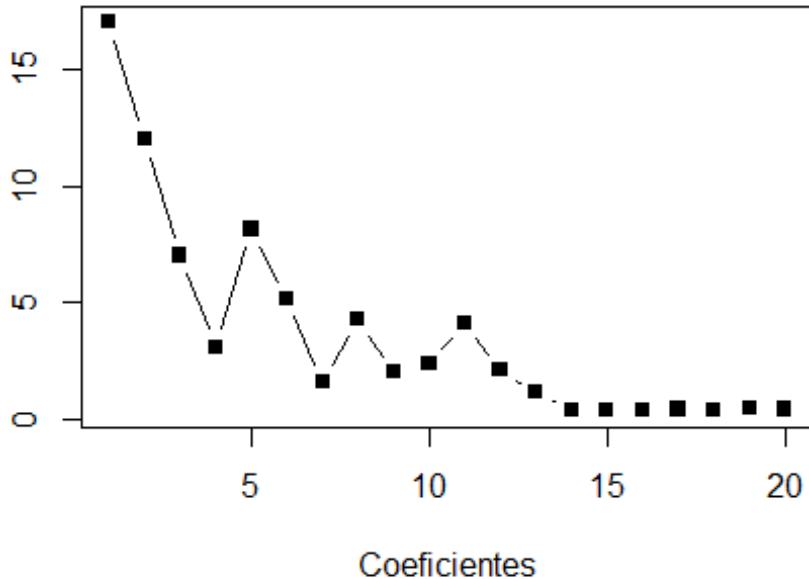
- (f) How does the model at which the test set MSE is minimized compare to the true model used to generate the data? Comment on the coefficient values.

```
coef(mod.full, which.min(val.errors))

## (Intercept)          x.1          x.2          x.4          x.5          x.8 
##  0.06571078 -3.12611061  5.04237610 -5.09237175  2.10378116  4.05270359 
##          x.9          x.11          x.12          x.14          x.15          x.16 
##  5.03049156  3.85962346  0.99202378  4.81505371 -3.01753210  4.02529521 
##          x.18          x.19          x.20
##  4.02534373 -2.04856037  0.95878618
```

- (g) Create a plot displaying for a range of values of r, where $\hat{\beta}_j$ is the jth coefficient estimate for the best model containing r coefficients. Comment on what you observe. How does this compare to the test MSE plot from (d)?

```
val.errors <- rep(NA, 20)
x_cols = colnames(xmat, do.NULL = FALSE, prefix = "x.")
for (i in 1:20) {
  coefi <- coef(mod.full, id = i)
  val.errors[i] <- sqrt(sum((betas[x_cols %in% names(coefi)] - coefi[names(coefi) %in% x_cols])^2) + sum(betas[!(x_cols %in% names(coefi))]^2))
}
plot(val.errors, xlab = "Coeficientes", ylab = "", pch = 15, type = "b")
```



Se logra evidenciar nuevamente que el modelo con 14 variables es el que minimiza el error, pero además los modelos con más variables, también resultan ser convenientes ya que el error es muy pequeño.

11. We will now try to predict per capita crime rate in the Boston data set.

```
library(ggplot2)
library(dplyr)
library(ISLR)
library(corrplot)
library(RColorBrewer)
library(car)
library(class)
library(MASS)
library(boot)
library(glmnet)
library(pls)
library(leaps)
```

- (a) Try out some of the regression methods explored in this chapter, such as best subset selection, the lasso, ridge regression, and PCR. Present and discuss results for the approaches that you consider.

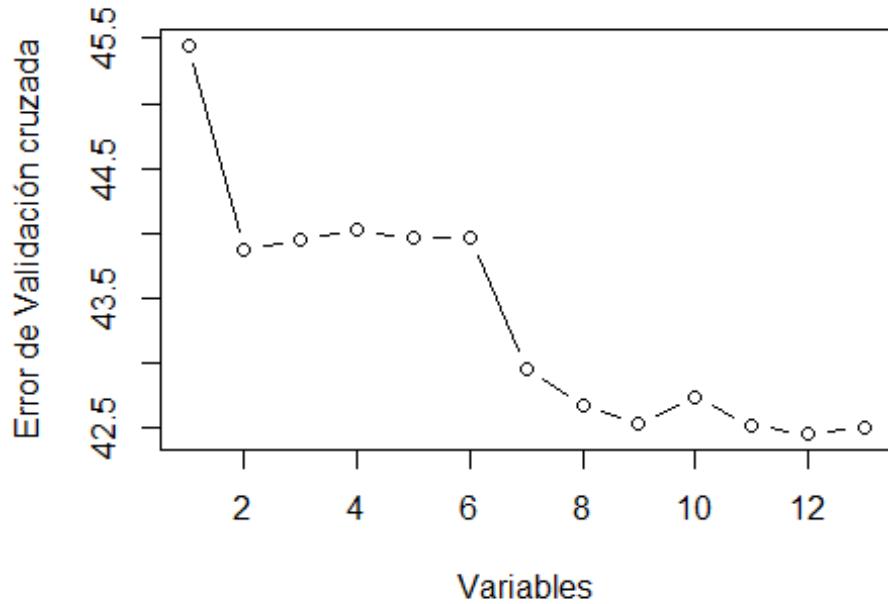
```
data(Boston)
set.seed(1)

predict.regsubsets <- function(object, newdata, id, ...) {
  form <- as.formula(object$call[[2]])
  mat <- model.matrix(form, newdata)
  coefi <- coef(object, id = id)
  xvars <- names(coefi)
  mat[, xvars] %*% coefi
}

k = 10
folds <- sample(1:k, nrow(Boston), replace = TRUE)
cv.errors <- matrix(NA, k, 13, dimnames = list(NULL, paste(1:13)))
for (j in 1:k) {
  best.fit <- regsubsets(crim ~ ., data = Boston[folds != j, ], nvmax = 13)
  for (i in 1:13) {
    pred <- predict(best.fit, Boston[folds == j, ], id = i)
    cv.errors[j, i] <- mean((Boston$crim[folds == j] - pred)^2)
  }
}
mean.cv.errors <- apply(cv.errors, 2, mean)
mean.cv.errors

##          1          2          3          4          5          6          7          8
## 45.44573 43.87260 43.94979 44.02424 43.96415 43.96199 42.96268 42.66948
##          9         10         11         12         13
## 42.53822 42.73416 42.52367 42.46014 42.50125

plot(mean.cv.errors, type = "b", xlab = "Variables", ylab = "Error de Validación cruzada")
```



El menor error para la validación cruzada es encuentra tomando 12 variables y tiene un valor de 42.46014. Pasamos a analizar Lasso.

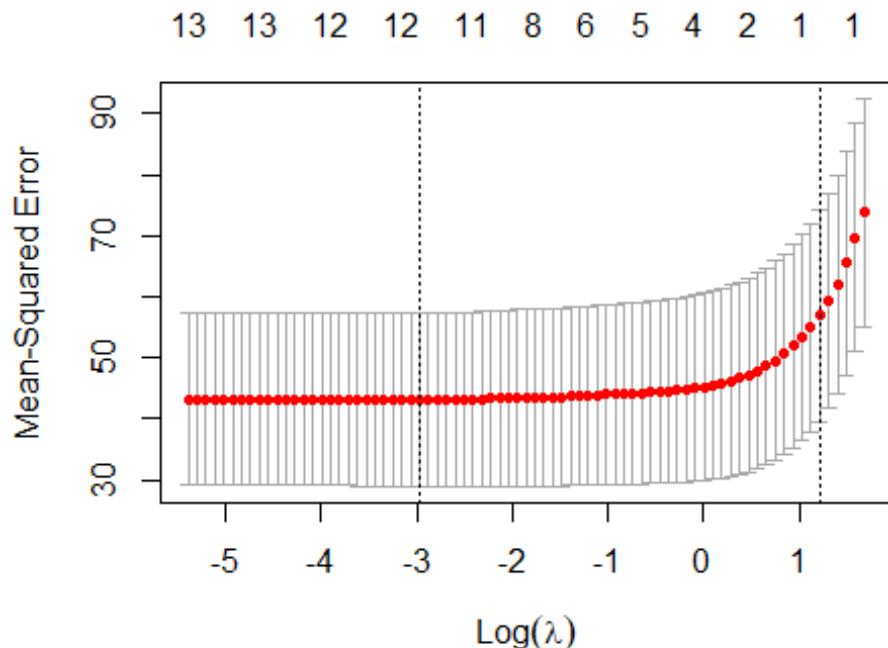
```

x <- model.matrix(crim ~ ., Boston)[, -1]
y <- Boston$crim
cv.out <- cv.glmnet(x, y, alpha = 1, type.measure = "mse")
cv.out

##
## Call: cv.glmnet(x = x, y = y, type.measure = "mse", alpha = 1)
##
## Measure: Mean-Squared Error
##
##      Lambda Index Measure     SE Nonzero
## min  0.051     51   43.11 14.16      11
## 1se  3.376      6   56.89 17.29       1

plot(cv.out)

```

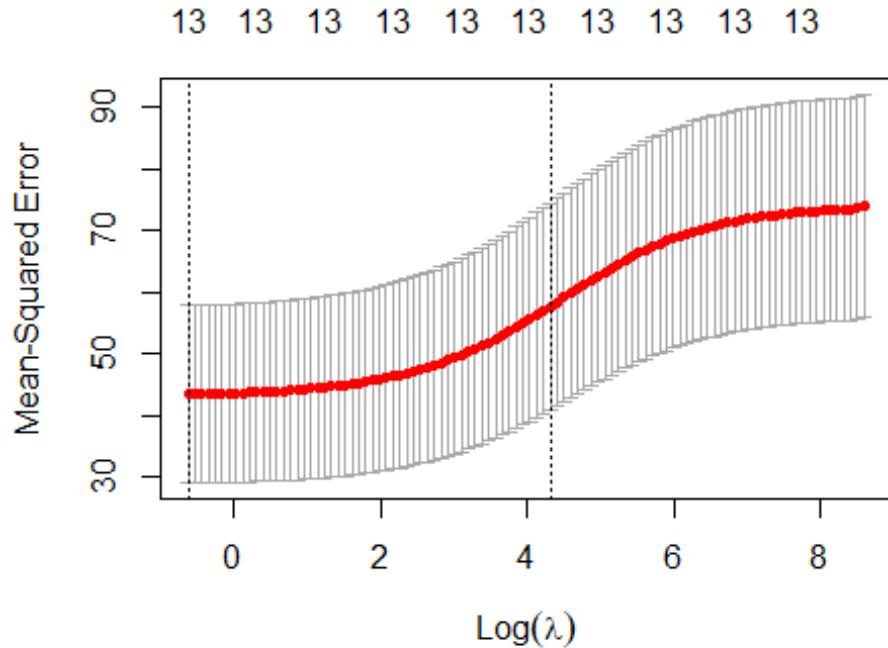


El Lambda seleccionado es igual a 0.051 para un error de validación cruzada igual a 43.11. Pero solo se están considerando 11 variables, una reducción importante. Ahora veremos que sucede con la regresión 'ridge'.

```
cv.out <- cv.glmnet(x, y, alpha = 0, type.measure = "mse")
cv.out

##
## Call: cv.glmnet(x = x, y = y, type.measure = "mse", alpha = 0)
##
## Measure: Mean-Squared Error
##
##      Lambda Index Measure     SE Nonzero
## min    0.54    100  43.48 14.33       13
## 1se   74.44     47  57.69 16.76       13

plot(cv.out)
```



En este caso se obtiene un error para la validación cruzada de 42.61 para una lambda igual a 0.54.

```
pqr.mod <- pqr(crim ~ ., data = Boston, scale = TRUE, validation = "CV")
pqr.mod

## Principal component regression , fitted with the singular value decomposition algorithm.
## Cross-validated using 10 random segments.
## Call:
## pqr(formula = crim ~ ., data = Boston, scale = TRUE, validation = "CV")

summary(pqr.mod)

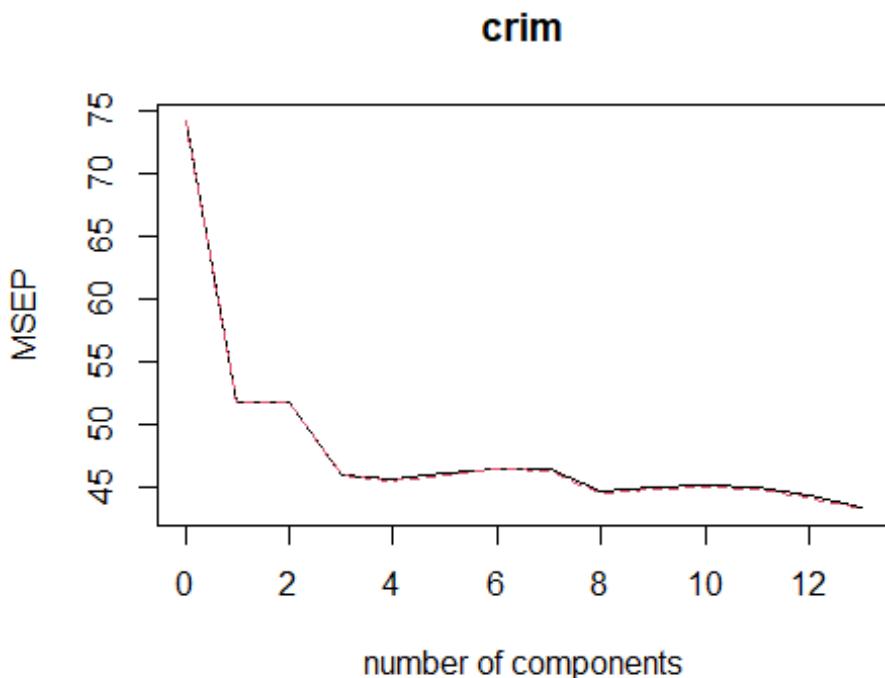
## Data: X dimension: 506 13
## Y dimension: 506 1
## Fit method: svdpc
## Number of components considered: 13
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##          (Intercept) 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps
## CV          8.61    7.198   7.198   6.786   6.762   6.790   6.821
## adjCV       8.61    7.195   7.195   6.780   6.753   6.784   6.813
##          7 comps 8 comps 9 comps 10 comps 11 comps 12 comps 13 comps
## CV          6.822   6.689   6.712    6.720   6.712   6.664   6.593
## adjCV       6.812   6.679   6.701    6.708   6.700   6.651   6.580
##
## TRAINING: % variance explained
```

```

##          1 comps   2 comps   3 comps   4 comps   5 comps   6 comps   7 comps   8 com
ps
## X      47.70     60.36    69.67    76.45    82.99    88.00    91.14    93.
45
## crim  30.69     30.87    39.27    39.61    39.61    39.86    40.14    42.
47
##          9 comps   10 comps   11 comps   12 comps   13 comps
## X      95.40     97.04    98.46    99.52    100.0
## crim  42.55     42.78    43.04    44.13    45.4

validationplot(pcr.mod, val.type = "MSEP")

```

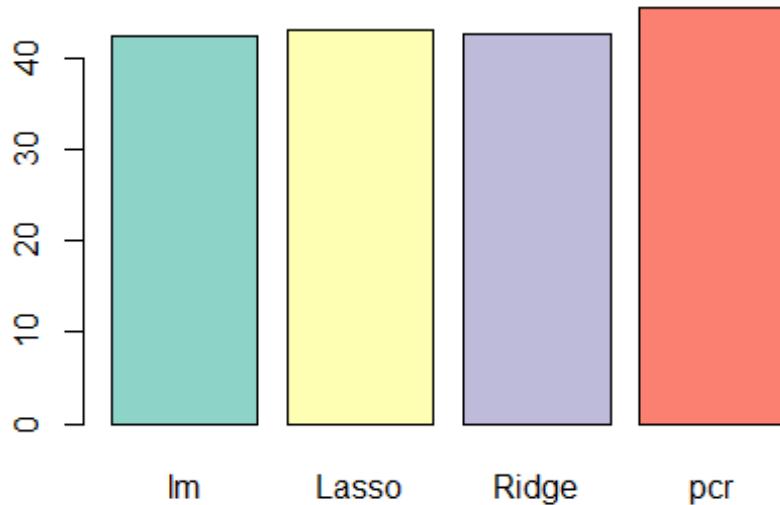


La validación cruzada selecciona 13 variables para un error de 45.4.

- (b) Propose a model (or set of models) that seem to perform well on this data set and justify your answer. Make sure that you are evaluating model performance using validation set error, crossvalidation, or some other reasonable alternative, as opposed to using training error.

Vemos entonces que el menor error de todos los presentados fue la primera (selección de subconjuntos), con un valor de 42.46, sin embargo, todas se encuentran muy cercanos a este valor. Como lo vemos en la siguiente gráfica.

```
err.all_f <- c(42.46, 43.11, 42.61, 45.40)
names(err.all_f) <- c("lm", "Lasso", "Ridge", "pcr")
barplot(err.all_f, col = brewer.pal(5, "Set3"))
```



A pesar de que en la selección de subconjuntos se tienen un error más bajo, es en el modelo Lasso donde se presenta una mayor reducción de variables, haciendo el modelo más sencillo y fácil de interpretar y el error solamente incrementa 0.65. Por este motivo el modelo Lasso puede ser más conveniente.

- (c) Does your chosen model involve all of the features in the data set? Why or why not?

El modelo Lasso solamente consideró 11 variables. Los que fueron excluidos no resultaron significativos dentro del modelo.