Cómputo Estadístico

October 26, 2024

Godinez Bravo Diego

Tarea 4 - Evaluación y Selección de Modelos

Centro de Investigación en Matemáticas

Maestría en Cómputo Estadístico

0.1 Problema 1

Generación de datos simulados y aplicación de los métodos de selección de subconjuntos

- Usa una función en R para generar una variable predictora X de longitud $n = 100$, así como un vector de ruido ϵ de tamaño $n = 100$.
- Genera un vector de respuesta Y de longitud $n = 100$ de acuerdo al modelo:

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

  donde $\beta_0, \beta_1, \beta_2, \beta_3$ son constantes de tu elección.

- Utiliza la función `regsubsets()` para realizar la selección de los mejores subconjuntos con el fin de elegir el mejor modelo que contenga los predictores $X, X^2, X^3, \ldots, X^{10}$. ¿Cuál es el mejor modelo obtenido según el **AIC**, **BIC** y el R^2 ajustado? Muestra algunas gráficas que proporcionen evidencia de tu respuesta y reporta los coeficientes del mejor modelo obtenido.
- Repite (c) usando la selección *forward stepwise* y *backward stepwise*. ¿Cómo se comparan tus respuestas con los resultados obtenidos en (c)?

0.2 Resultados

[1]:
```r
library(leaps) # loading library leaps for model selection functions
```

[56]:
```r
set.seed(80)
X <- rnorm(100) # X variable with n = 100
noise <- runif(100, -1, 1) # random noise
```

[3]:
```r
length(X) # predictor variable lenght
```

100

[4]:
```
beta_0 <- 1.0
beta_1 <- 0.20
beta_2 <- 0.45
beta_3 <- 0.15 # set parameters
```

[5]:
```
Y <- beta_0 + beta_1*X + beta_2*X^2 + beta_3*X^3 + noise # response vector Y
```

[6]:
```
length(Y) # respose vector lenght
```

100

[7]:
```
df <- data.frame(Y, X, X^2, X^3, X^4, X^5, X^6, X^7, X^8, X^9, X^10) # data␣
↪frame containing both Y and X
```

[10]:
```
results <- regsubsets(Y~., data = df, nvmax = 10) # maximum size of subsets to␣
↪examine nvmax = 10
summary(results)
```

```
Subset selection object
Call: regsubsets.formula(Y ~ ., data = df, nvmax = 10)
10 Variables  (and intercept)
     Forced in Forced out
X        FALSE      FALSE
X.2      FALSE      FALSE
X.3      FALSE      FALSE
X.4      FALSE      FALSE
X.5      FALSE      FALSE
X.6      FALSE      FALSE
X.7      FALSE      FALSE
X.8      FALSE      FALSE
X.9      FALSE      FALSE
X.10     FALSE      FALSE
1 subsets of each size up to 10
Selection Algorithm: exhaustive
          X   X.2 X.3 X.4 X.5 X.6 X.7 X.8 X.9 X.10
1  ( 1 )  " " " " "*" " " " " " " " " " " " " " "
2  ( 1 )  " " "*" "*" " " " " " " " " " " " " " "
3  ( 1 )  "*" "*" "*" " " " " " " " " " " " " " "
4  ( 1 )  "*" "*" "*" " " " " " " " " " " " " "*"
5  ( 1 )  "*" "*" " " " " "*" " " "*" " " " " "*"
6  ( 1 )  "*" "*" "*" " " "*" " " "*" " " "*" " "
7  ( 1 )  "*" "*" "*" " " "*" " " "*" " " "*" "*"
8  ( 1 )  "*" "*" "*" " " "*" "*" "*" "*" "*" " "
9  ( 1 )  "*" " " "*" "*" "*" "*" "*" "*" "*" "*"
10  ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
```

[11]:
```
rsummary <- summary(results)
```

[12]:
```
rsummary$bic # based on BIC criterion the optimal model has 3 variables: X, X^2
↪and X^3
```

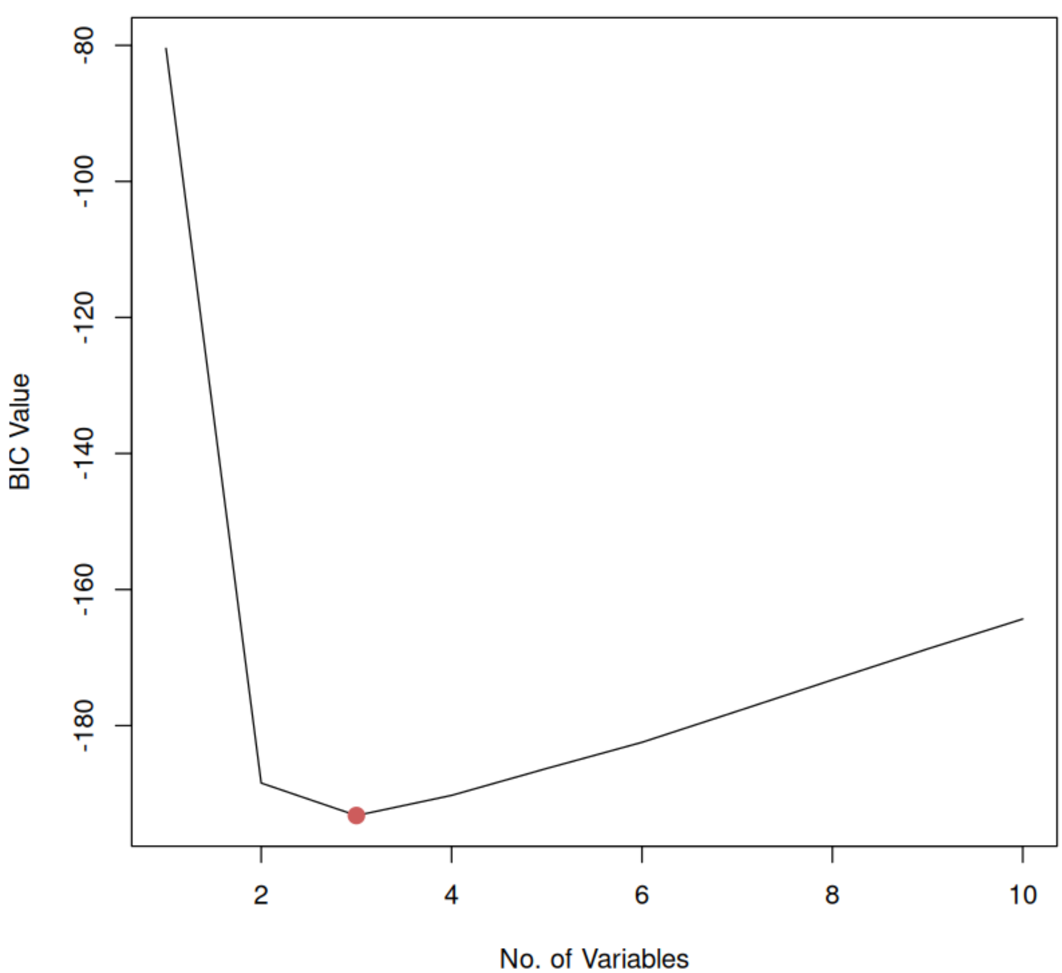
1. -80.4433660633648 2. -188.44818257233 3. -193.219244069958 4. -190.253957967214 5. -186.294927233416 6. -182.453775148901 7. -177.881976198719 8. -173.277779290899 9. -168.727274744861 10. -164.328234764821

Coeficientes estimados para el mejor modelo de acuerdo al **criterio BIC**.

[13]:
```
coef(results, 3) # coefficient estimates associated with this model
```

(Intercept) 1.02446352781608 **X** 0.280233605379258 **X.2** 0.456030224765697 **X.3** 0.147061525512855

[14]:
```
plot(rsummary$bic, xlab = 'No. of Variables', ylab = 'BIC Value', type = 'l')
points(3, rsummary$bic[3], col = 'indianred', cex = 2, pch = 20)
```



El valor más bajo del criterio **BIC** se observa en el modelo con 3 variables.

[15]:
```
rsummary$adjr2 # based on adjusted R-squared criterion the optimal model has 4␣
  ↪variables: X, X^2, X^3 and X^10
```

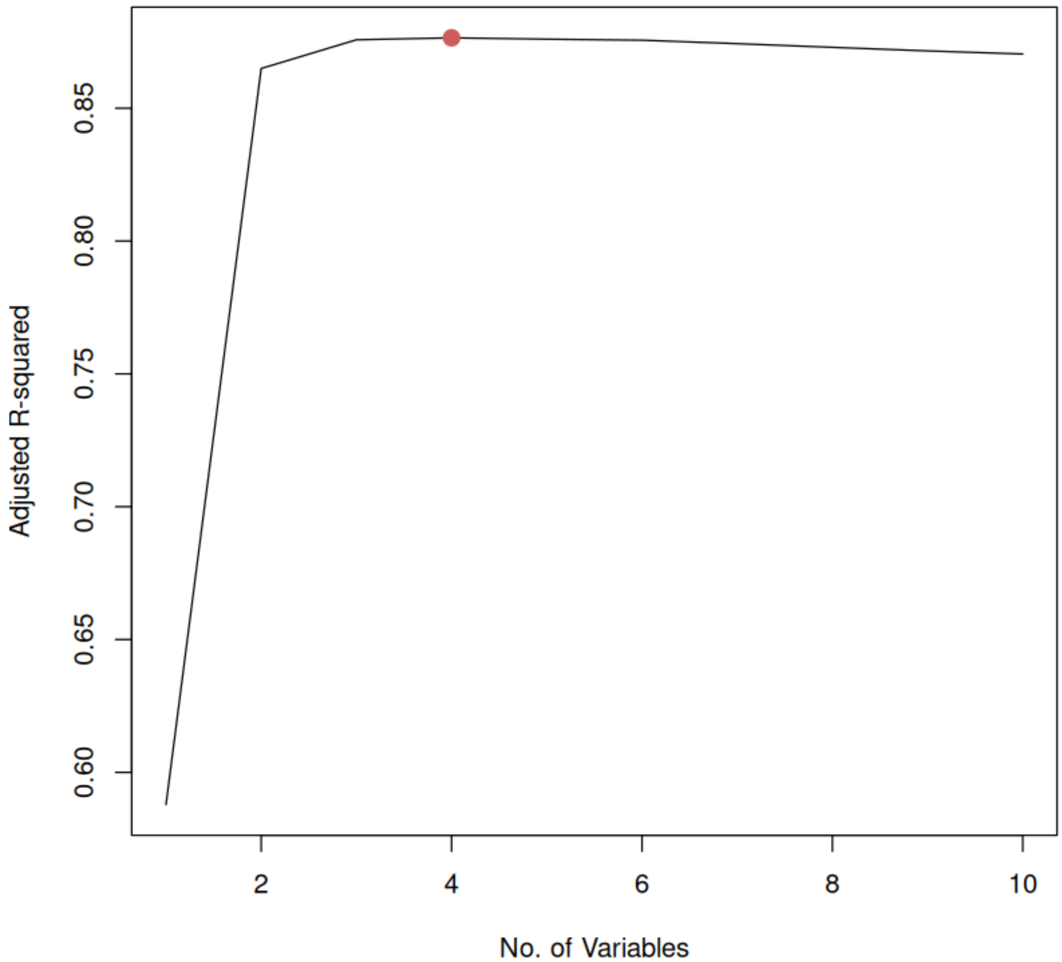
1. 0.587856913620807 2. 0.864965916180391 3. 0.875770869198963 4. 0.8765050643253 5. 0.875995126983469 6. 0.875615699390191 7. 0.874305648880689 8. 0.872925629068109 9. 0.871583910281548 10. 0.870408435731653

Coeficientes estimados para el mejor modelo de acuerdo al valor **Adjusted R-squared**.

[16]:
```
coef(results, 4) # coefficient estimates associated with this model
```

(Intercept) 1.05138982268519 **X** 0.309830304036517 **X.2** 0.41718960967728 **X.3** 0.136446865571499 **X.10** 7.06877270374999e-06

[17]:
```
plot(rsummary$adjr2, xlab = 'No. of Variables', ylab = 'Adjusted R-squared',␣
  ↪type = 'l')
points(4, rsummary$adjr2[4], col = 'indianred', cex = 2, pch = 20)
```



El valor más alto de **Adjusted R-squared** se observa en el modelo con 4 variables. Sin embargo, a partir del modelo con 3 variables, los valores subsecuentes muestran poca variación entre sí.

Dado que la penalización utilizada por el criterio BIC es `log(n)d`, esta se resta de los valores BIC obtenidos previamente. Los cuales se ajustan considerando la penalización correspondiente al criterio AIC, que es de `2d`.

[18]:
```
n <- length(df$Y) # no. of observations
d <- apply(rsummary$which, 1, sum) # no. of predictors for each model
```

[19]:
```
aic <- rsummary$bic - log(n) * d + 2 * d # AIC values
```

[20]:
```
aic # based on AIC criterion the optimal model has 3 variables: X, X^2 and X^3
```

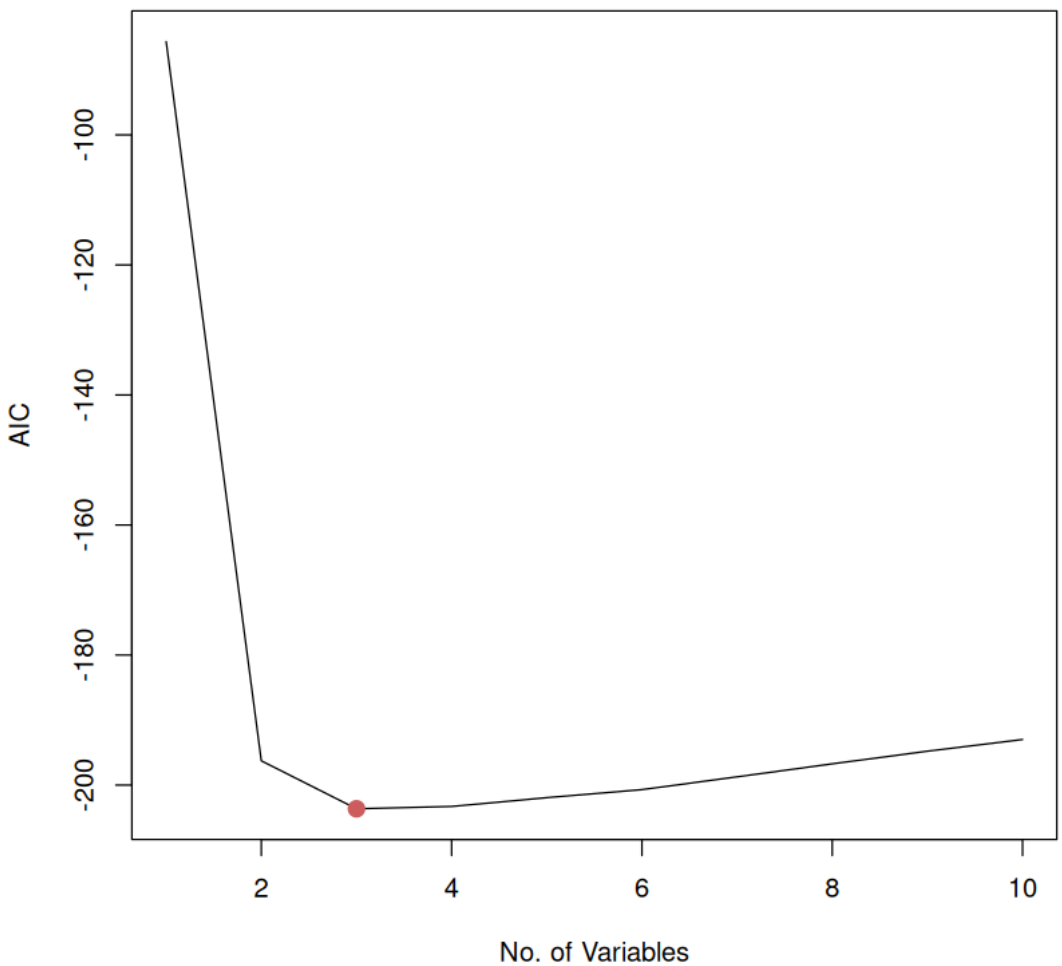
1 -85.653706435341 **2** -196.263693130295 **3** -203.63992481391 **4** -203.279808897154 **5** -201.925948349344 **6** -200.689966450817 **7** -198.723337686624 **8** -196.724310964792 **9** -194.778976604742 **10** -192.98510681069

Coeficientes estimados para el mejor modelo de acuerdo al **criterio AIC**.

[21]:
```
coef(results, 3) # coefficient estimates associated with this model
```

(Intercept) 1.02446352781608 **X** 0.280233605379258 **X.2** 0.456030224765697 **X.3** 0.147061525512855

[22]:
```
plot(aic, xlab = 'No. of Variables', ylab = 'AIC', type = 'l')
points(3, aic[3], col = 'indianred', cex = 2, pch = 20)
```



Al igual que en el criterio anterior, el valor más bajo del **AIC** se obtiene en el modelo con 3 variables.

De acuerdo con los criterios **AIC** y **BIC**, el mejor modelo está compuesto por 3 variables: X, X^2, X^3. Sin embargo, basándonos en el criterio **Adjusted R-squared**, el modelo óptimo incluye 4 variables, que, de acuerdo con el resumen de la función `regsubsets`, corresponden a X, X^2, X^3, X^{10}.

Considerando la parsimonia del modelo, en este caso optaríamos por el de 3 variables. Al utilizar menos predictores, se logra un equilibrio entre simplicidad y poder explicativo. Este es un principio fundamental, ya que los modelos más simples suelen ser preferibles por su facilidad de interpretación y menor riesgo de sobreajuste.

0.2.1 Selección *Forward Stepwise*

[23]:
```
results_forward_method <- regsubsets(Y~., data = df, nvmax = 10, method = 'forward') # maximum size of subsets to examine nvmax = 10
summary(results_forward_method)
```

```
Subset selection object
Call: regsubsets.formula(Y ~ ., data = df, nvmax = 10, method = "forward")
10 Variables  (and intercept)
     Forced in Forced out
X        FALSE      FALSE
X.2      FALSE      FALSE
X.3      FALSE      FALSE
X.4      FALSE      FALSE
X.5      FALSE      FALSE
X.6      FALSE      FALSE
X.7      FALSE      FALSE
X.8      FALSE      FALSE
X.9      FALSE      FALSE
X.10     FALSE      FALSE
1 subsets of each size up to 10
Selection Algorithm: forward
          X   X.2 X.3 X.4 X.5 X.6 X.7 X.8 X.9 X.10
1  ( 1 )  " " " " "*" " " " " " " " " " " " " " " 
2  ( 1 )  " " "*" "*" " " " " " " " " " " " " " " 
3  ( 1 )  "*" "*" "*" " " " " " " " " " " " " " " 
4  ( 1 )  "*" "*" "*" " " " " " " " " " " " " "*" 
5  ( 1 )  "*" "*" "*" " " " " " " " " " " "*" "*" 
6  ( 1 )  "*" "*" "*" " " "*" " " " " " " "*" "*" 
7  ( 1 )  "*" "*" "*" " " "*" " " "*" " " "*" "*" 
8  ( 1 )  "*" "*" "*" "*" "*" " " "*" " " "*" "*" 
9  ( 1 )  "*" "*" "*" "*" "*" "*" "*" " " "*" "*" 
10  ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" "*" 
```

[24]:
```
rsummary_forward_method <- summary(results_forward_method)
```

[25]:
```
rsummary_forward_method$bic # based on BIC criterion the optimal model has 3␣
↪variables: X, X^2 and X^3
```

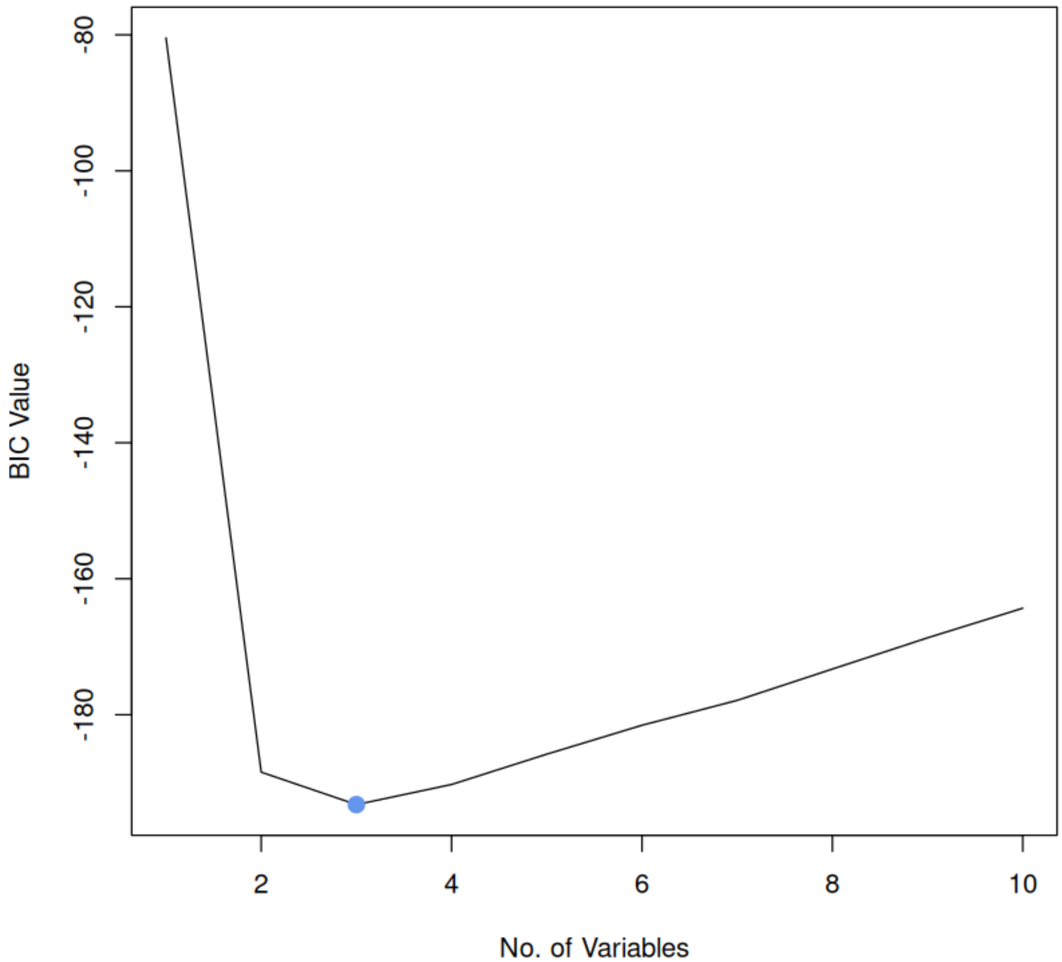
1. -80.4433660633647 2. -188.44818257233 3. -193.219244069958 4. -190.253957967214 5. -185.793616429043 6. -181.54471999583 7. -177.881976198719 8. -173.277242368175 9. -168.683834791522 10. -164.328234764821

Coeficientes estimados para el mejor modelo de acuerdo al **criterio BIC**.

[26]:
```
coef(results_forward_method, 3) # coefficient estimates associated with this␣
↪model
```

(Intercept) 1.02446352781608 **X** 0.280233605379258 **X.2** 0.456030224765697 **X.3** 0.147061525512855

[27]:
```
plot(rsummary_forward_method$bic, xlab = 'No. of Variables', ylab = 'BIC␣
↪Value', type = 'l')
points(3, rsummary_forward_method$bic[3], col = 'cornflowerblue', cex = 2, pch␣
↪= 20)
```



[28]:
```
rsummary_forward_method$adjr2 # based on adjusted R-squared criterion the
↪optimal model has 4 variables: X, X^2, X^3 and X^10
```

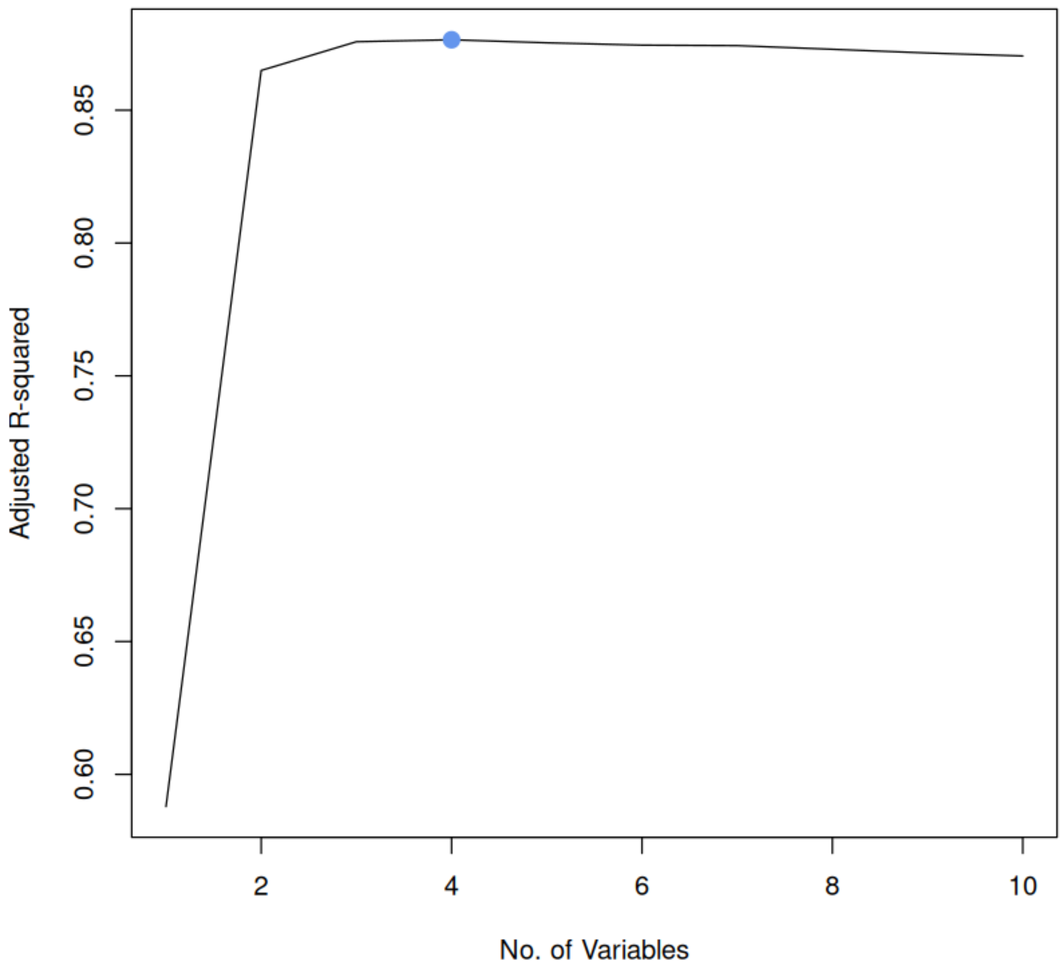
1. 0.587856913620807 2. 0.864965916180391 3. 0.875770869198963 4. 0.8765050643253 5. 0.875371916351144 6. 0.874479822444149 7. 0.874305648880689 8. 0.872924946775104 9. 0.871528114274092 10. 0.870408435731653

Coeficientes estimados para el mejor modelo de acuerdo al valor **Adjusted R-squared**.

[29]:
```
coef(results_forward_method, 4) # coefficient estimates associated with this
↪model
```

(Intercept) 1.05138982268519 **X** 0.309830304036517 **X.2** 0.417189609677279 **X.3** 0.136446865571498 **X.10** 7.06877270375006e-06

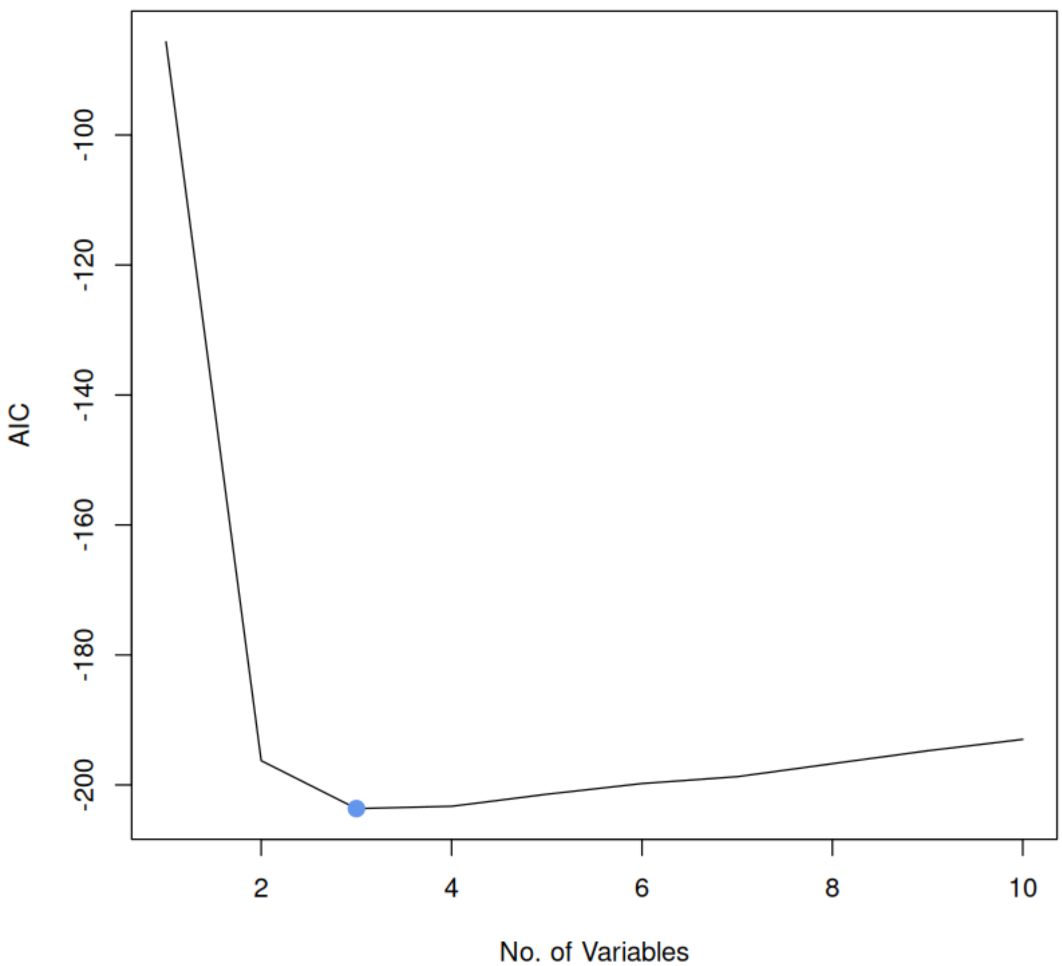
[30]:
```
plot(rsummary_forward_method$adjr2, xlab = 'No. of Variables', ylab = 'Adjusted␣
↪R-squared', type = 'l')
points(4, rsummary_forward_method$adjr2[4], col = 'cornflowerblue', cex = 2,␣
↪pch = 20)
```



[31]:
```
d <- apply(rsummary_forward_method$which, 1, sum) # no. of predictors for each␣
↪model
aic <- rsummary_forward_method$bic - log(n) * d + 2 * d # AIC values
aic # based on AIC criterion the optimal model has 3 variables: X, X^2 and X^3
```

1 -85.6537064353409 **2** -196.263693130295 **3** -203.63992481391 **4** -203.279808897154 **5** -201.424637544972 **6** -199.780911297747 **7** -198.723337686624 **8** -196.723774042068 **9** -194.735536651403 **10** -192.98510681069

[32]:
```
plot(aic, xlab = 'No. of Variables', ylab = 'AIC', type = 'l')
points(3, aic[3], col = 'cornflowerblue', cex = 2, pch = 20)
```



0.2.2 Selección *Backward Stepwise*

[33]:
```
results_backward_method <- regsubsets(Y~., data = df, nvmax = 10, method =␣
↪'backward') # maximum size of subsets to examine nvmax = 10
summary(results_backward_method)
```

```
Subset selection object
Call: regsubsets.formula(Y ~ ., data = df, nvmax = 10, method = "backward")
10 Variables  (and intercept)
     Forced in Forced out
```

```
X        FALSE      FALSE
X.2      FALSE      FALSE
X.3      FALSE      FALSE
X.4      FALSE      FALSE
X.5      FALSE      FALSE
X.6      FALSE      FALSE
X.7      FALSE      FALSE
X.8      FALSE      FALSE
X.9      FALSE      FALSE
X.10     FALSE      FALSE
1 subsets of each size up to 10
Selection Algorithm: backward
          X   X.2 X.3 X.4 X.5 X.6 X.7 X.8 X.9 X.10
1  ( 1 )  " " " " " " "*" " " " " " " " " " " " "
2  ( 1 )  "*" " " " " "*" " " " " " " " " " " " "
3  ( 1 )  "*" " " " " "*" "*" " " " " " " " " " "
4  ( 1 )  "*" " " " " "*" "*" "*" " " " " " " " "
5  ( 1 )  "*" " " " " "*" "*" "*" " " "*" " " " "
6  ( 1 )  "*" " " " " "*" "*" "*" "*" "*" " " " "
7  ( 1 )  "*" " " " " "*" "*" "*" "*" "*" " " "*"
8  ( 1 )  "*" " " " " "*" "*" "*" "*" "*" "*" "*"
9  ( 1 )  "*" " " "*" "*" "*" "*" "*" "*" "*" "*"
10  ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
```

[34]:
```
rsummary_backward_method <- summary(results_backward_method)
```

[35]:
```
rsummary_backward_method$bic # BIC values using the backward stepwise method
```

1. -51.4861148223593 2. -140.839135357172 3. -170.320800924082 4. -180.26184764569 5. -178.148104999865 6. -180.756135111871 7. -177.093112222387 8. -172.817982458597 9. -168.727274744861 10. -164.328234764821

En este caso, el modelo con el valor más bajo de **BIC** corresponde al que incluye 6 variables $(X, X^4, X^5, X^6, X^7, X^8)$. Sin embargo, a partir del modelo con 4 variables, los valores de **BIC** no varían significativamente. Por lo tanto, elegir el modelo con 6 variables sería contraproducente, ya que su valor de BIC es apenas inferior al del modelo con 4 variables, lo que podría aumentar innecesariamente la complejidad del modelo.

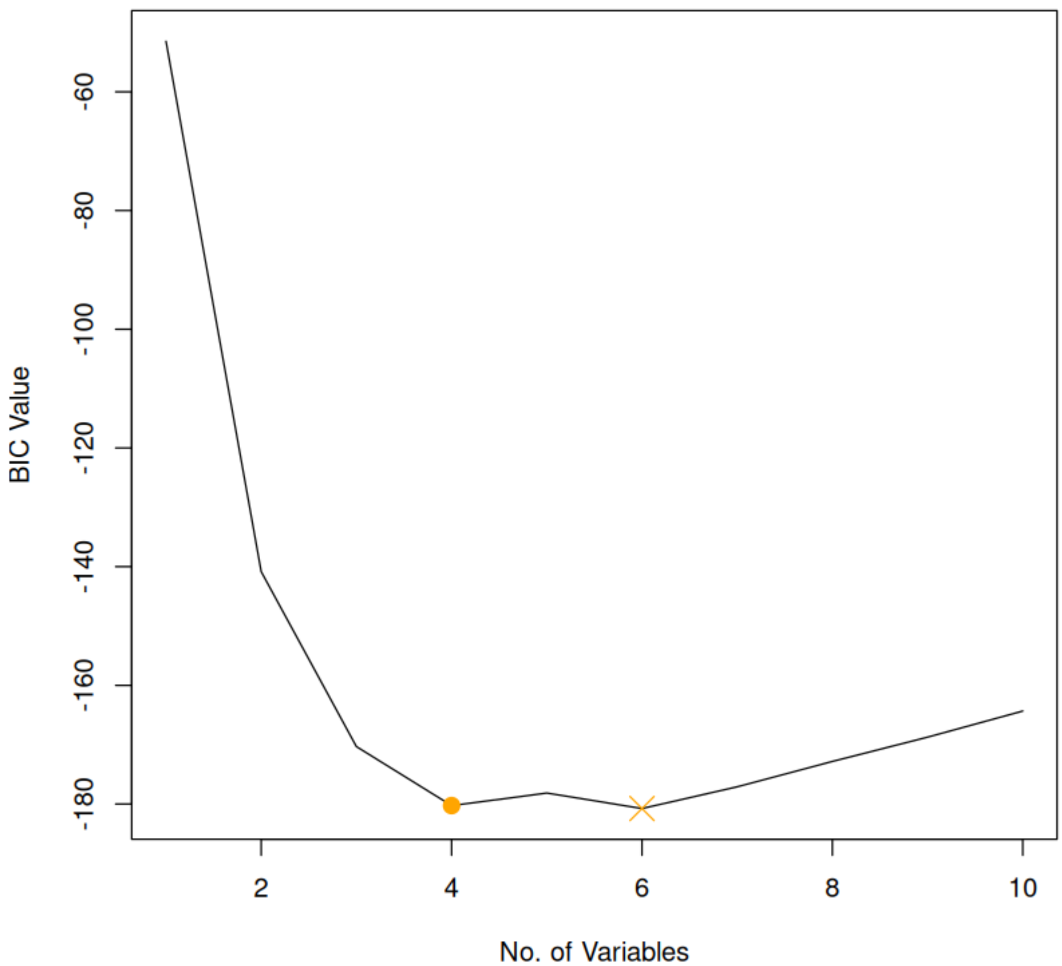
Coeficientes estimados para el mejor modelo de acuerdo al **criterio BIC**.

[36]:
```
coef(results_backward_method, 4) # coefficient estimates associated with this␣
↪model
```

(Intercept) 1.20772010243288 **X** 0.501665215149097 **X.4** 0.120701793250172 **X.5** 0.0145471519260609 **X.6** -0.00794040335623914

[37]:
```
plot(rsummary_backward_method$bic, xlab = 'No. of Variables', ylab = 'BIC␣
↪Value', type = 'l')
points(4, rsummary_backward_method$bic[4], col = 'orange', cex = 2, pch = 20)
```

```
points(6, rsummary_backward_method$bic[6], col = 'orange', cex = 2, pch = 4)
```



[38]:
```
rsummary_backward_method$adjr2 # Adjusted R-squared values using the backward␣
    ↪stepwise method
```

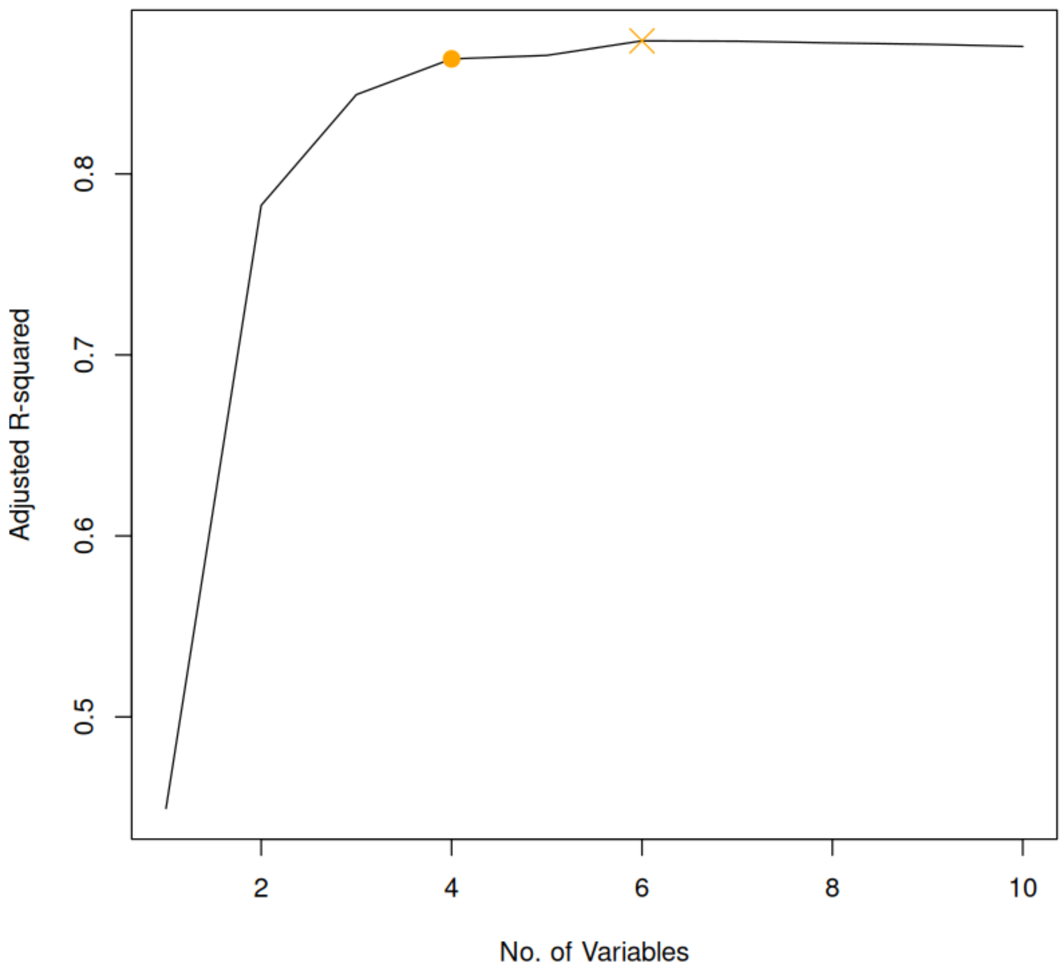
1. 0.449436059928791 2. 0.782626365374272 3. 0.843803923138197 4. 0.863527756208704 5. 0.865469749344636 6. 0.873486076181022 7. 0.873310170100112 8. 0.872339999815267 9. 0.871583910281548 10. 0.870408435731653

El modelo con el valor más bajo de **Adjusted R-squared** corresponde al que incluye 6 variables.

[39]:
```
coef(results_backward_method, 7) # coefficient estimates associated with this␣
    ↪model
```

(Intercept) 1.1131485203165 **X** 0.423118375048814 **X.4** 0.384167428991017 **X.5** 0.0333580719056403 **X.6** -0.113769494581084 **X.7** -0.00209432030398532 **X.8** 0.0131550963330699 **X.10** -0.000506832437163779

[40]:
```
plot(rsummary_backward_method$adjr2, xlab = 'No. of Variables', ylab =
    'Adjusted R-squared', type = 'l')
points(4, rsummary_backward_method$adjr2[4], col = 'orange', cex = 2, pch = 20)
points(6, rsummary_backward_method$adjr2[6], col = 'orange', cex = 2, pch = 4)
```



En este caso, al observar la curva, se observa que el modelo con 6 variables tiene el valor más alto, que difiere significativamente del modelo con 4 variables. Sin embargo, como se mencionó anteriormente, los modelos más simples suelen ser preferibles por su facilidad de interpretación y menor riesgo de sobreajuste. Por lo tanto, se recomendaría optar por el modelo con el menor grado de complejidad.

[41]:
```
d <- apply(rsummary_backward_method$which, 1, sum) # no. of predictors for each
↪model
aic <- rsummary_backward_method$bic - log(n) * d + 2 * d # AIC values
aic # based on AIC criterion the optimal model has 6 variables: X, X^4, X^5,
↪X^6, X^7, and X^8
```

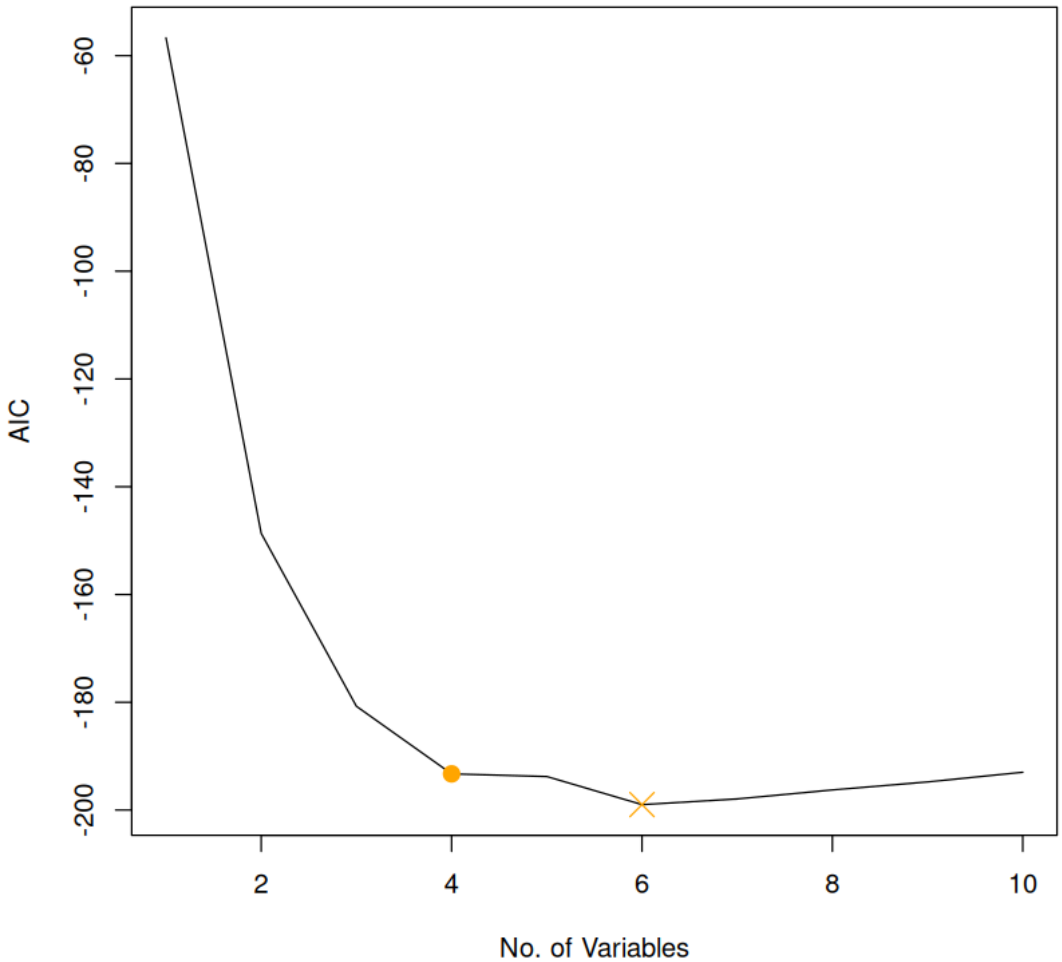
1 -56.6964551943355 **2** -148.654645915137 **3** -180.741481668034 **4** -193.28769857563 **5** -193.779126115793 **6** -198.992326413788 **7** -197.934473710292 **8** -196.264514132489 **9** -194.778976604742 **10** -192.98510681069

Coeficientes estimados para el mejor modelo de acuerdo al **criterio AIC**.

[42]:
```
coef(results_backward_method, 6) # coefficient estimates associated with this
↪model
```

(Intercept) 1.13435945382632 **X** 0.403532570106159 **X.4** 0.280452157601234 **X.5** 0.0419936203671447 **X.6** -0.0542412937591682 **X.7** -0.00331969222588616 **X.8** 0.00316685323833655

[43]:
```
plot(aic, xlab = 'No. of Variables', ylab = 'AIC', type = 'l')
points(6, aic[6], col = 'orange', cex = 2, pch = 4)
points(4, aic[4], col = 'orange', cex = 2, pch = 20)
```

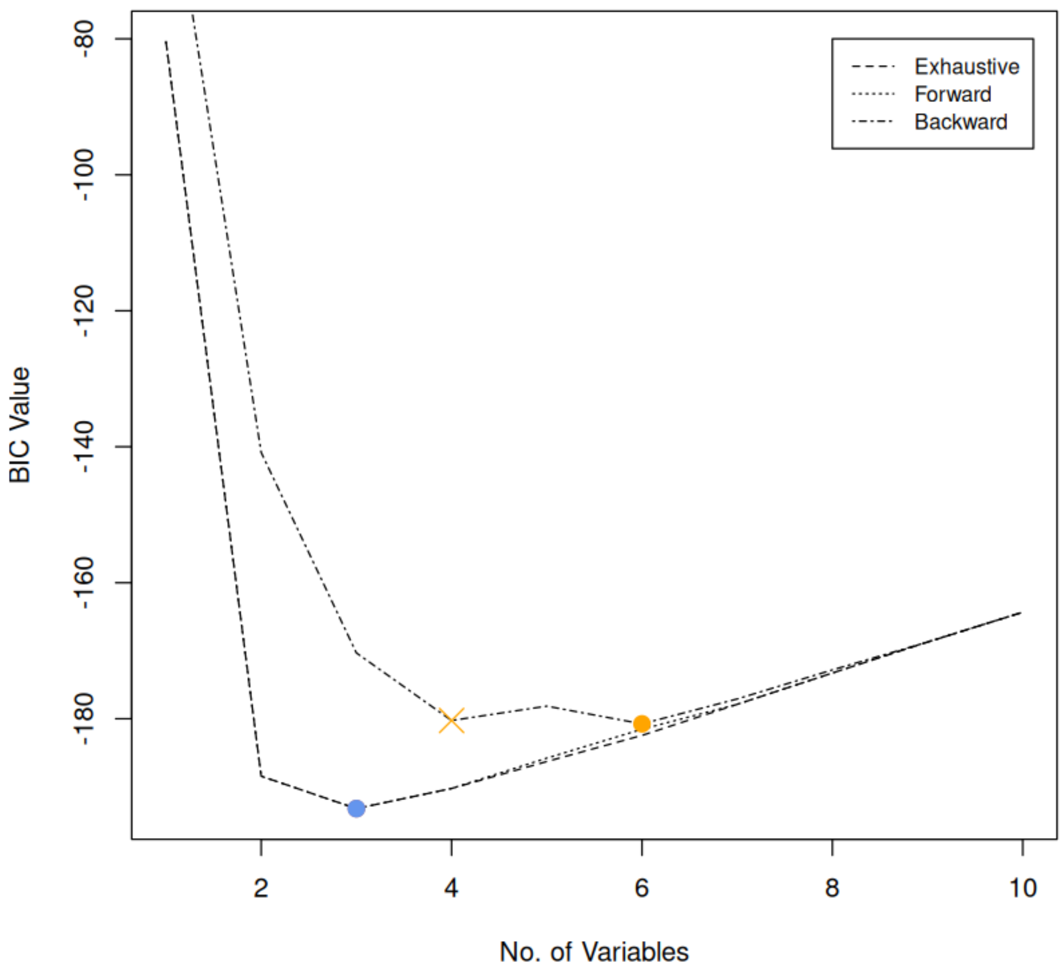


0.2.3 Comparación de los Métodos de Selección

Al aplicar el método *forward stepwise*, los resultados son similares a los obtenidos con el método *exhaustive*. Según los criterios **BIC** y **AIC**, el mejor modelo incluye las variables X, X^2, X^3. Sin embargo, al basarnos en el criterio **Adjusted R-squared**, el modelo óptimo incorpora las variables X, X^2, X^3, X^{10}. En ambos casos, es preferible optar por el modelo más parsimonioso.

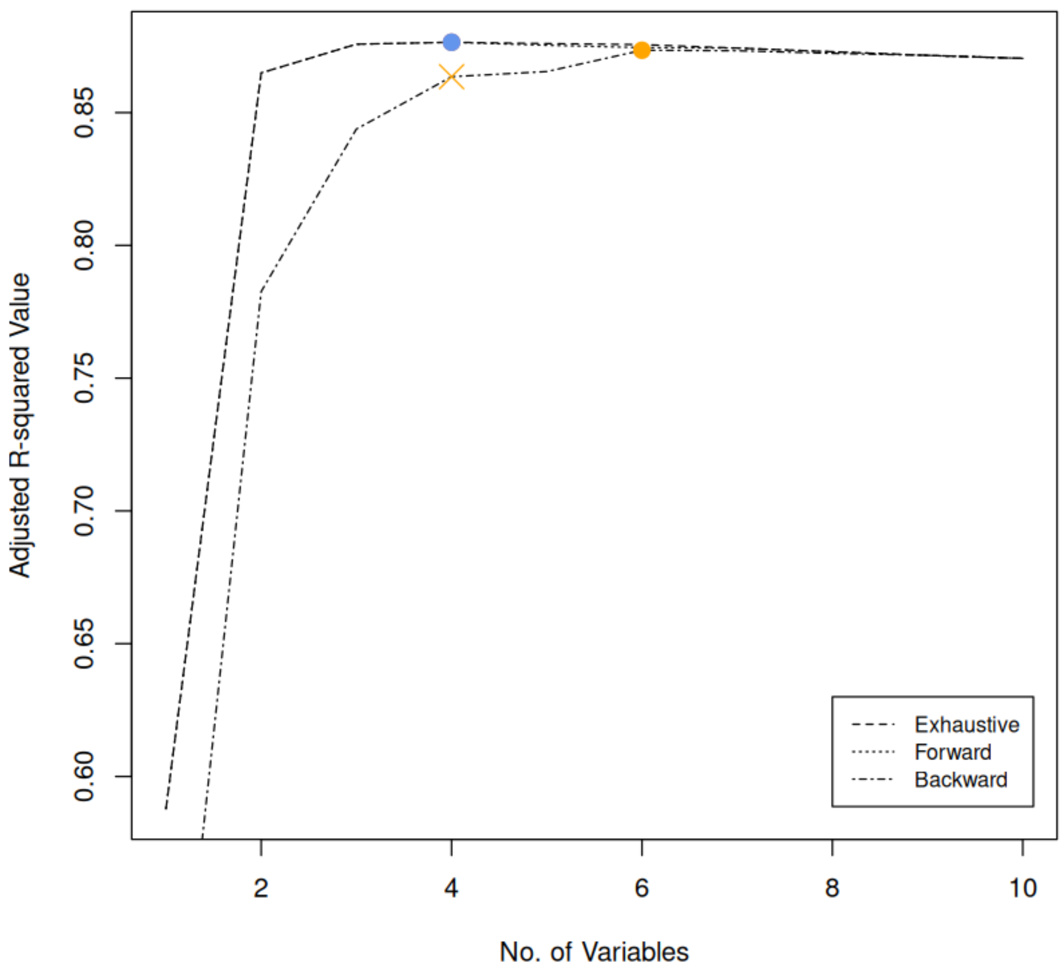
Por otro lado, los resultados del método *backward stepwise* difieren en gran medida respecto a los otros enfoques. Los valores más bajos de los criterios se obtienen cuando se seleccionan 6 variables. Sin embargo, es recomendable elegir el modelo con 4 variables, ya que, aunque existe una diferencia entre ambos, esta no es lo suficientemente grande como para justificar un aumento en la complejidad del modelo.

[46]:
```
plot(rsummary$bic, xlab = 'No. of Variables', ylab = 'BIC Value', type = 'l',␣
↪lty = 2) # BIC values exhaustive stepwise method
lines(rsummary_forward_method$bic, lty = 3) # BIC values forward stepwise method
lines(rsummary_backward_method$bic, lty = 4) # BIC values backward stepwise␣
↪method
points(3, rsummary$bic[3], col = 'indianred', cex = 2, pch = 20)
points(3, rsummary_forward_method$bic[3], col = 'cornflowerblue', cex = 2, pch␣
↪= 20)
points(4, rsummary_backward_method$bic[4], col = 'orange', cex = 2, pch = 4)
points(6, rsummary_backward_method$bic[6], col = 'orange', cex = 2, pch = 20)
legend(8, -80, legend = c('Exhaustive', 'Forward', 'Backward'),
       lty = 2:4, cex = 0.8) # add legend
```



[55]:

```
plot(rsummary$adjr2, xlab = 'No. of Variables', ylab = 'Adjusted R-squared␣
 ↪Value', type = 'l', lty = 2) # BIC values exhaustive stepwise method
lines(rsummary_forward_method$adjr2, lty = 3) # BIC values forward stepwise␣
 ↪method
lines(rsummary_backward_method$adjr2, lty = 4) # BIC values backward stepwise␣
 ↪method
points(4, rsummary$adjr2[4], col = 'indianred', cex = 2, pch = 20)
points(4, rsummary_forward_method$adjr2[4], col = 'cornflowerblue', cex = 2,␣
 ↪pch = 20)
points(4, rsummary_backward_method$adjr2[4], col = 'orange', cex = 2, pch = 4)
points(6, rsummary_backward_method$adjr2[6], col = 'orange', cex = 2, pch = 20)
legend(8, 0.63, legend = c('Exhaustive', 'Forward', 'Backward'),
       lty = 2:4, cex = 0.8) # add legend
```



0.3 Problema 2

Se ha visto que a medida que aumenta el número de características de un modelo, el error de entrenamiento disminuirá necesariamente, pero el error de prueba no. Explorar esto con datos simulados,

- Genera un conjunto de datos con $p = 20$ características, $n = 1000$ observaciones y un vector de respuesta cuantitativo generado de acuerdo con el modelo

$$Y = X\beta + \epsilon$$

donde β tiene algunos elementos que son exactamente iguales a cero.

- Divide tu conjunto de datos en un conjunto de entrenamiento que contenga 100 observaciones y un conjunto de pruebas que contenga 900 observaciones.
- Realiza la selección del *mejor subconjunto* sobre el conjunto de entrenamiento y grafica el error de entrenamiento MSE asociado con el mejor modelo en cada tamaño.
- Grafica el error de prueba MSE asociado con el mejor modelo de cada tamaño.
- Para qué tamaño de modelo el error de prueba MSE toma su valor mínimo? Comenta tus resultados. Si toma su valor mínimo en un modelo que sólo contiene una interceptación o un modelo que contenga todas las características, entonces juega con la forma en la que estás generando los datos en (a) hasta que aparezca un escenario en el que el error de prueba MSE se minimiza para un tamaño de modelo intermedio.
- Cómo se compara el modelo con el que se minimiza el error de prueba con el modelo verdadero utilizado para generar los datos? Comenta sobre los valores de los coeficientes.

0.4 Resultados

[1]:
```
library(leaps) # loading library leaps for model selection functions
```

[2]:
```
set.seed(614)
n <- 1000 # no. of observations
p <- 20 # no. of features
```

[3]:
```
error <- rnorm(n) # error terms
```

[4]:
```
X <- matrix(runif(n * p, -1, 1), n, p) # create a matrix with 1000 rows and 20␣
↪columns
```

[5]:
```
b <- rnorm(p) # vector of beta values
b[3] = b[4] = b[7] = b[8] = b[12] = b[15] = 0 # randomly assign 0 to some betas
```

[6]:
```
Y <- X %*% b + error # response vector
```

```
[7]: index <- sample(1:n, 100, replace = FALSE) # indices to be chosen
X_train <- X[index,]
y_train <- Y[index,] # training set contains 100 observations
X_test <- X[-index,]
y_test <- Y[-index,] # test set contains 900 observations
```

0.4.1 Calculo del Error Cuadrático Medio de Entrenamiento

```
[8]: train_df = data.frame(y = y_train, x = X_train) # training dataframe
```

Selección de Subconjuntos utilizando *Método Exhaustivo*

```
[9]: results <- regsubsets(y~., data = train_df, nvmax = p) # model selection using␣
  ↪exhaustive method
```

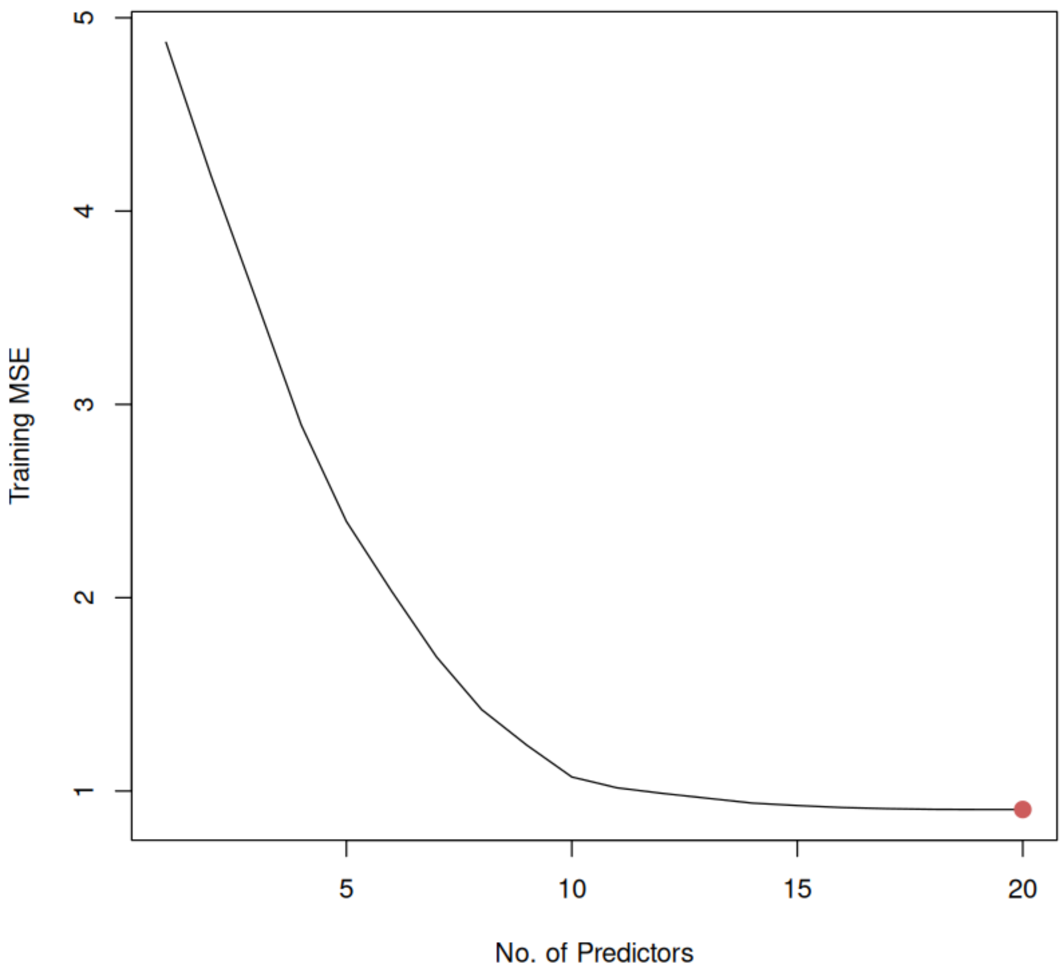
$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(\hat{x_i}))^2$$

```
[10]: training_MSE_values <- c() # empty vector to store MSE values
for(i in 1:p){
    predictions <- model.matrix(y~., data = train_df, nvmax = p)[,␣
  ↪names(coef(results, i))] %*% coef(results, i)
    training_MSE_values[i] = mean((predictions - y_train)^2)
} # compute training MSE values for each model with i predictors
```

```
[11]: cat('Index of the smallest MSE value:', which.min(training_MSE_values),␣
  ↪'\nSmallest MSE value', training_MSE_values[20]) #
```

```
Index of the smallest MSE value: 20
Smallest MSE value 0.9041558
```

```
[12]: plot(training_MSE_values, xlab = 'No. of Predictors', ylab = 'Training MSE',␣
  ↪type = 'l')
points(20, training_MSE_values[20], col = 'indianred', cex = 2, pch = 20)
```



El valor más bajo del MSE de entrenamiento se obtiene cuando el modelo incluye todos los predictores disponibles, obteniendo un valor cercano a 0.90.

0.4.2 Calculo del Error Cuadrático Medio de Prueba

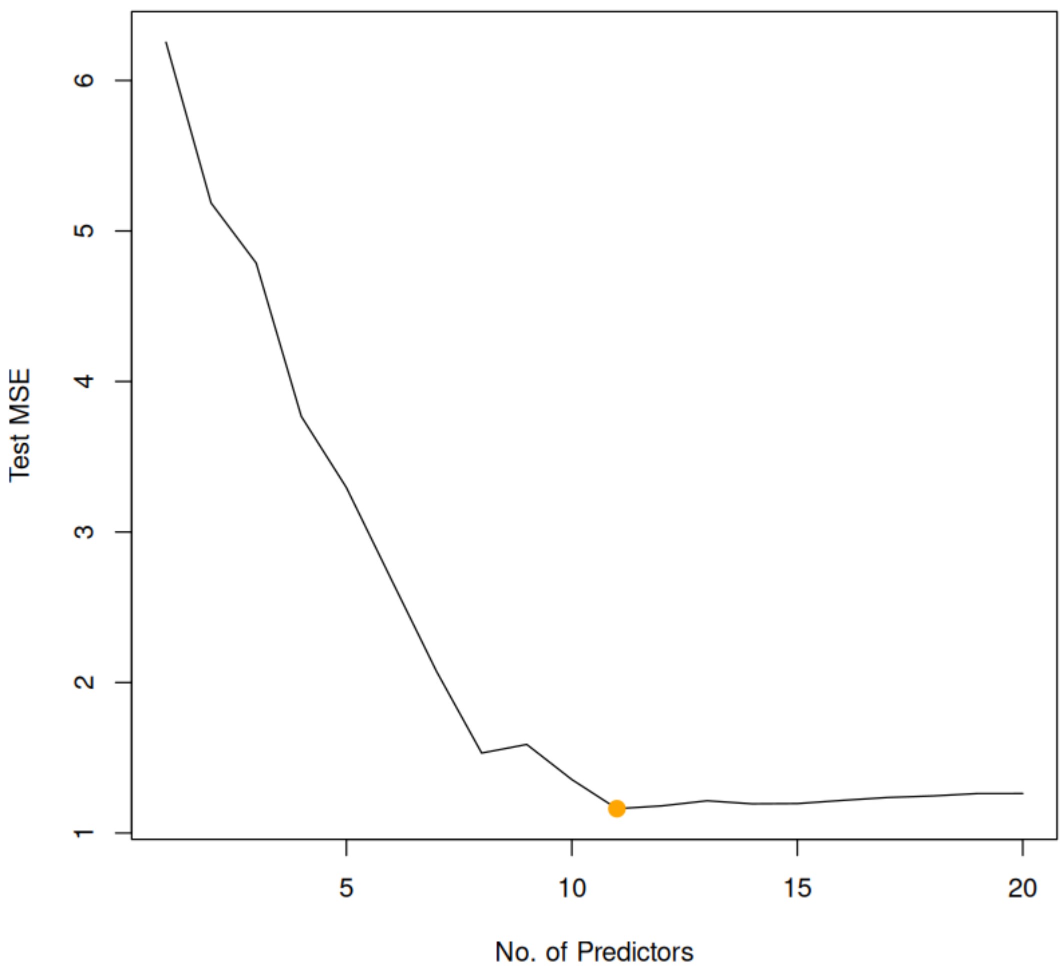
[13]:
```
test_df = data.frame(y = y_test, x = X_test) # testing dataframe
```

[14]:
```
testing_MSE_values <- c() # empty vector to store MSE values
for(i in 1:p){
    predictions <- model.matrix(y~., data = test_df, nvmax = p)[, 
 ↪names(coef(results, i))] %*% coef(results, i)
    testing_MSE_values[i] = mean((predictions - y_test)^2)
} # compute training MSE values for each model with i predictors
```

[15]:
```
cat('Index of the smallest MSE value:', which.min(testing_MSE_values),
'\nSmallest MSE value', testing_MSE_values[11]) #
```

```
Index of the smallest MSE value: 11
Smallest MSE value 1.161698
```

[17]:
```
plot(testing_MSE_values, xlab = 'No. of Predictors', ylab = 'Test MSE', type =
'l')
points(11, testing_MSE_values[11], col = 'orange', cex = 2, pch = 20)
```



A diferencia del caso anterior, el valor más bajo del MSE de prueba se obtiene al considerar un número intermedio de predictores, específicamente 11, lo que resulta en un valor aproximado de 1.16.

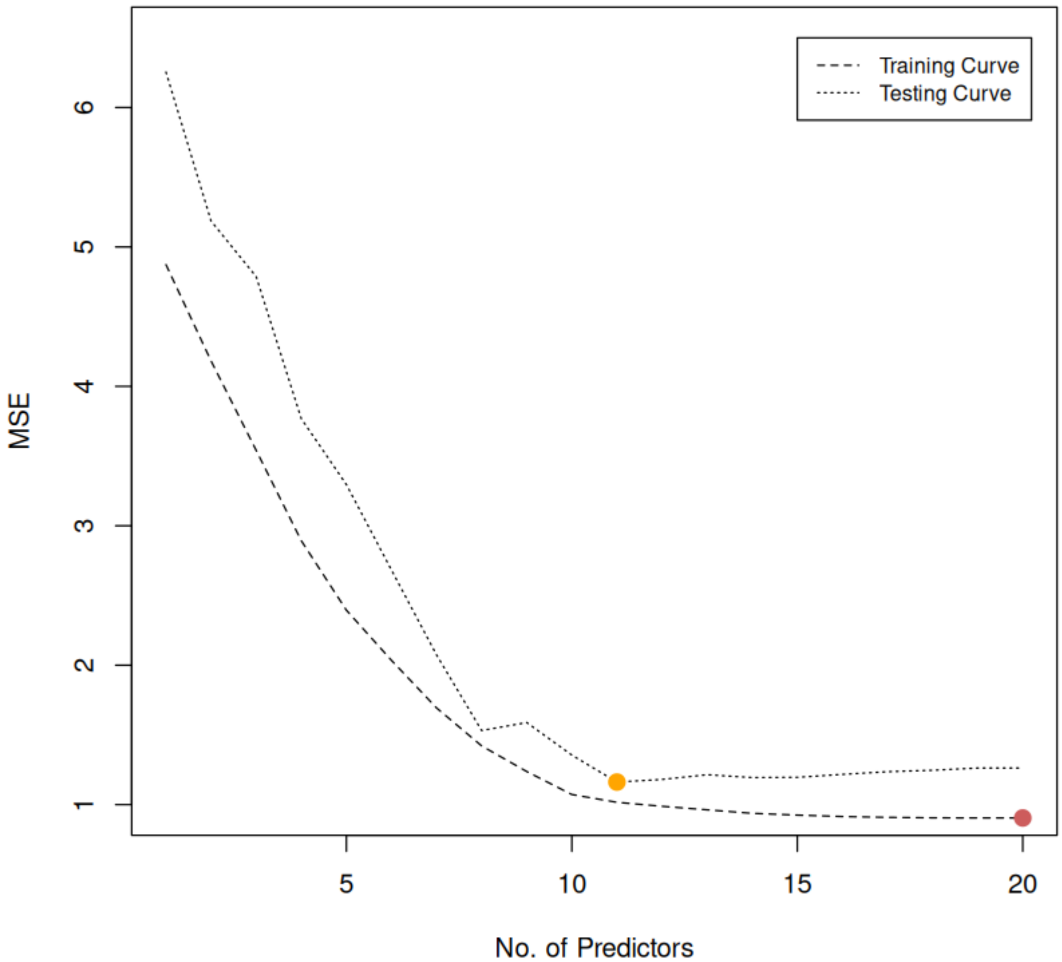
Es importante notar que $min\ MSE_{test} > min\ MSE_{train}$.

0.4.3 Relación entre la Flexibilidad del Modelo y el Comportamiento del Error Cuadrático Medio

A medida que aumenta la flexibilidad del modelo, el error cuadrático medio (MSE) en el conjunto de entrenamiento tiende a disminuir, alcanzando su valor más bajo al considerar todas las características disponibles. Sin embargo, este patrón no se observa en el MSE del conjunto de prueba, donde el valor mínimo se alcanza al incluir un número intermedio de características, en este caso 11 de las 20 disponibles.

En conclusión, no se garantiza que el modelo con el MSE más bajo en el conjunto de entrenamiento también presente el menor MSE en el conjunto de prueba. Independientemente de la presencia de sobreajuste, se espera que el MSE del conjunto de entrenamiento sea siempre menor que el del conjunto de prueba, dado que la mayoría de los métodos de aprendizaje estadístico están diseñados para minimizar explícita o implícitamente el MSE en el entrenamiento.

[18]:
```
plot(training_MSE_values, xlab = 'No. of Predictors', ylab = 'MSE', ylim = c(1,␣
↪6.5), type = 'l', lty = 2)
lines(testing_MSE_values, lty = 3)
points(20, training_MSE_values[20], col = 'indianred', cex = 2, pch = 20) #␣
↪smallest training MSE value
points(11, testing_MSE_values[11], col = 'orange', cex = 2, pch = 20) #␣
↪smallest testing MSE value
legend(15, 6.5, legend = c('Training Curve', 'Testing Curve'),
       lty = 2:3, cex = 0.8) # add legend
```



Coeficientes del modelo que minimizan el MSE de prueba.

[21]:
```
coef(results, which.min(testing_MSE_values))
```

(Intercept) -0.00691515825461912 **x.1** 1.69218569971264 **x.2** 0.763848141887755 **x.5** -1.27672010856663 **x.10** -1.25683301441883 **x.11** -1.11725007220012 **x.13** 0.454343790042969 **x.14** 2.8168577913396 **x.16** 1.66425832869807 **x.17** -1.16789109659159 **x.18** 0.994048348291716 **x.20** 0.750161846492621

Se inicializaron los valores de los coeficientes $\beta_3 = \beta_4 = \beta_7 = \beta_8 = \beta_{12} = \beta_{15} = 0$. En el modelo que minimiza el error cuadrático medio (MSE) de prueba, estos valores son consistentes, lo que implica que las variables asociadas a estos no forman parte del modelo propuesto.