Regresión

Diego Vázquez Zambrano

2023-05-25

```
library(faraway)
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

##
## Attaching package: 'lattice'

## The following object is masked from 'package:faraway':
##
##     melanoma
```

1. (Ejercicio 1 cap. 10 pág. 159)

Use the prostate data with lpsa as the response and the other variables as predictors. Implement the following variable selection methods to determine the "best" model: (a) Backward elimination (b) AIC (c) Adjusted R2 (d) Mallows Cp

```
data(prostate)

#(a) Backward elimination

# Fit the initial model with all predictors
fit_back <- lm(lpsa ~ ., data = prostate)
fit_back_elim <- step(fit_back, direction = "backward")

## Start:  AIC=-58.32
## lpsa ~ lcavol + lweight + age + lbph + svi + lcp + gleason +
##     pgg45
##
##           Df Sum of Sq    RSS     AIC
## - gleason  1    0.0412 44.204 -60.231
## - pgg45    1    0.5258 44.689 -59.174
## - lcp      1    0.6740 44.837 -58.853
## <none>                 44.163 -58.322
## - age      1    1.5503 45.713 -56.975
## - lbph     1    1.6835 45.847 -56.693
## - lweight  1    3.5861 47.749 -52.749
## - svi      1    4.9355 49.099 -50.046
## - lcavol   1   22.3721 66.535 -20.567
##
```

```
## Step:  AIC=-60.23
## lpsa ~ lcavol + lweight + age + lbph + svi + lcp + pgg45
##
##            Df Sum of Sq    RSS     AIC
## - lcp      1    0.6623 44.867 -60.789
## <none>                 44.204 -60.231
## - pgg45    1    1.1920 45.396 -59.650
## - age      1    1.5166 45.721 -58.959
## - lbph     1    1.7053 45.910 -58.560
## - lweight  1    3.5462 47.750 -54.746
## - svi      1    4.8984 49.103 -52.037
## - lcavol   1   23.5039 67.708 -20.872
##
## Step:  AIC=-60.79
## lpsa ~ lcavol + lweight + age + lbph + svi + pgg45
##
##            Df Sum of Sq    RSS     AIC
## - pgg45    1    0.6590 45.526 -61.374
## <none>                 44.867 -60.789
## - age      1    1.2649 46.131 -60.092
## - lbph     1    1.6465 46.513 -59.293
## - lweight  1    3.5647 48.431 -55.373
## - svi      1    4.2503 49.117 -54.009
## - lcavol   1   25.4189 70.285 -19.248
##
## Step:  AIC=-61.37
## lpsa ~ lcavol + lweight + age + lbph + svi
##
##            Df Sum of Sq    RSS     AIC
## <none>                 45.526 -61.374
## - age      1    0.9592 46.485 -61.352
## - lbph     1    1.8568 47.382 -59.497
## - lweight  1    3.2251 48.751 -56.735
## - svi      1    5.9517 51.477 -51.456
## - lcavol   1   28.7665 74.292 -15.871
```

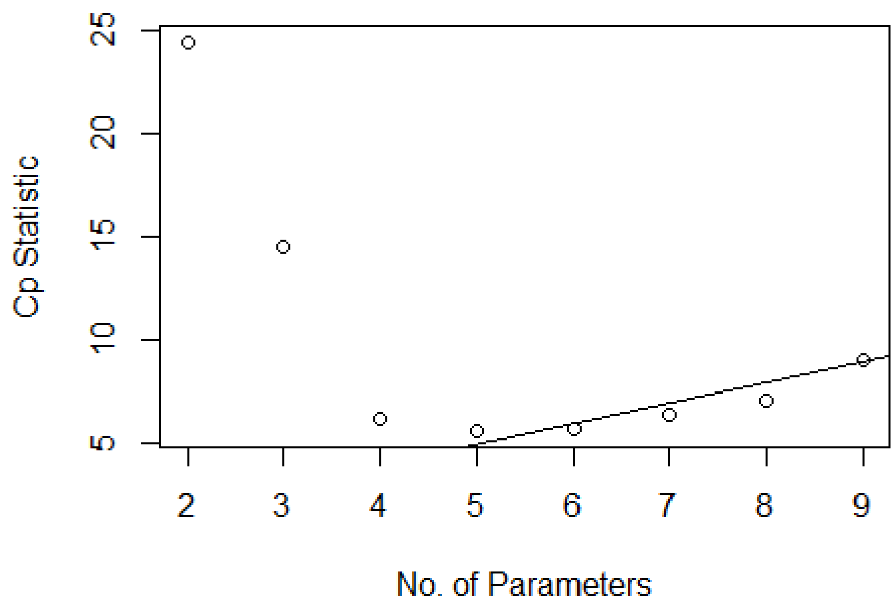
```r
# (b) AIC (Akaike Information Criterion)
fit_aic <- step(fit_back, direction = "both", k = log(nrow(prostate)))
```

```
## Start:  AIC=-35.15
## lpsa ~ lcavol + lweight + age + lbph + svi + lcp + gleason +
##     pgg45
##
##            Df Sum of Sq    RSS     AIC
## - gleason  1    0.0412 44.204 -39.634
## - pgg45    1    0.5258 44.689 -38.576
## - lcp      1    0.6740 44.837 -38.255
## - age      1    1.5503 45.713 -36.377
## - lbph     1    1.6835 45.847 -36.095
## <none>                 44.163 -35.149
```

```
## - lweight  1    3.5861 47.749 -32.151
## - svi      1    4.9355 49.099 -29.448
## - lcavol   1   22.3721 66.535   0.030
##
## Step:  AIC=-39.63
## lpsa ~ lcavol + lweight + age + lbph + svi + lcp + pgg45
##
##           Df Sum of Sq    RSS     AIC
## - lcp      1    0.6623 44.867 -42.766
## - pgg45    1    1.1920 45.396 -41.627
## - age      1    1.5166 45.721 -40.936
## - lbph     1    1.7053 45.910 -40.537
## <none>                 44.204 -39.634
## - lweight  1    3.5462 47.750 -36.723
## + gleason  1    0.0412 44.163 -35.149
## - svi      1    4.8984 49.103 -34.014
## - lcavol   1   23.5039 67.708  -2.849
##
## Step:  AIC=-42.77
## lpsa ~ lcavol + lweight + age + lbph + svi + pgg45
##
##           Df Sum of Sq    RSS     AIC
## - pgg45    1    0.6590 45.526 -45.926
## - age      1    1.2649 46.131 -44.644
## - lbph     1    1.6465 46.513 -43.844
## <none>                 44.867 -42.766
## - lweight  1    3.5647 48.431 -39.925
## + lcp      1    0.6623 44.204 -39.634
## - svi      1    4.2503 49.117 -38.561
## + gleason  1    0.0296 44.837 -38.255
## - lcavol   1   25.4189 70.285  -3.800
##
## Step:  AIC=-45.93
## lpsa ~ lcavol + lweight + age + lbph + svi
##
##           Df Sum of Sq    RSS     AIC
## - age      1    0.9592 46.485 -48.478
## - lbph     1    1.8568 47.382 -46.623
## <none>                 45.526 -45.926
## - lweight  1    3.2251 48.751 -43.862
## + pgg45    1    0.6590 44.867 -42.766
## + gleason  1    0.4560 45.070 -42.328
## + lcp      1    0.1293 45.396 -41.627
## - svi      1    5.9517 51.477 -38.583
## - lcavol   1   28.7665 74.292  -2.997
##
## Step:  AIC=-48.48
## lpsa ~ lcavol + lweight + lbph + svi
##
##           Df Sum of Sq    RSS     AIC
```

```
## - lbph      1    1.3001 47.785 -50.377
## <none>                  46.485 -48.478
## - lweight  1    2.8014 49.286 -47.377
## + age      1    0.9592 45.526 -45.926
## + pgg45    1    0.3533 46.131 -44.644
## + gleason  1    0.2126 46.272 -44.348
## + lcp      1    0.1023 46.383 -44.117
## - svi      1    5.8063 52.291 -41.636
## - lcavol   1   27.8298 74.315  -7.542
##
## Step:  AIC=-50.38
## lpsa ~ lcavol + lweight + svi
##
##           Df Sum of Sq    RSS     AIC
## <none>                 47.785 -50.377
## + lbph     1    1.3001 46.485 -48.478
## + pgg45    1    0.5735 47.211 -46.974
## + age      1    0.4025 47.382 -46.623
## + gleason  1    0.3890 47.396 -46.596
## + lcp      1    0.0641 47.721 -45.933
## - svi      1    5.1814 52.966 -44.966
## - lweight  1    5.8924 53.677 -43.673
## - lcavol   1   28.0445 75.829 -10.160
```

```
# (c) Adjusted R2
fit_adjR2 <- step(fit_back, direction = "both", k = log(nrow(prostate)),
trace = 0)

library(leaps)
# (d) Mallows Cp
fit_mallows <- regsubsets(lpsa ~ ., data = prostate, nvmax =
ncol(prostate))

# Aquí vemos que es un subset de 8 variables, entonces siguiendo el libro
de Faraway sabemos que tenemos que escribir 2:8+1, es decir, 2:9

mallows_cp <- summary(fit_mallows)
plot(2:9, mallows_cp$cp, xlab="No. of Parameters", ylab="Cp Statistic")
abline (0, 1)
```

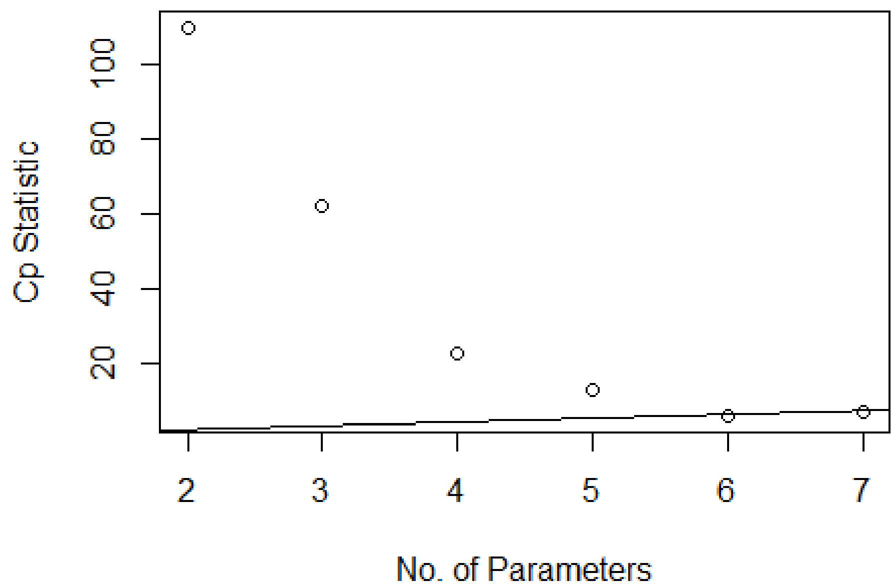



```
#Podemos escoger un modelo de 6, 7 u 8 parámetros, que son los Cp por
debajo de la línea
```

##3. (Ejercicio 3 cap. 10 pág. 159) Using the divusa dataset with divorce as the response and the other variables as predictors, repeat the work of the first question.

```
data(divusa)

# (a) Backward Elimination
fit_back <- lm(divorce ~ ., data = divusa)
fit_back_elim <- step(fit_back, direction = "backward")

## Start:  AIC=70.41
## divorce ~ year + unemployed + femlab + marriage + birth + military
##
##              Df Sum of Sq    RSS     AIC
## - unemployed  1     1.925 162.12  69.330
## <none>                    160.20  70.410
## - military    1    22.231 182.43  78.417
## - year        1    33.199 193.40  82.912
## - marriage    1    90.468 250.66 102.884
## - femlab      1   113.214 273.41 109.572
## - birth       1   144.897 305.10 118.015
##
## Step:  AIC=69.33
## divorce ~ year + femlab + marriage + birth + military
##
```

```
##             Df Sum of Sq    RSS     AIC
## <none>                   162.12  69.330
## - military  1    20.957 183.08  76.691
## - year      1    42.054 204.18  85.089
## - marriage  1   126.643 288.77 111.779
## - femlab    1   158.003 320.13 119.718
## - birth     1   172.826 334.95 123.203

# (b) AIC (Akaike Information Criterion)
fit_aic <- step(fit_back, direction = "both", k = log(nrow(divusa)))

## Start:  AIC=86.82
## divorce ~ year + unemployed + femlab + marriage + birth + military
##
##               Df Sum of Sq    RSS     AIC
## - unemployed  1     1.925 162.12  83.393
## <none>                     160.20  86.817
## - military    1    22.231 182.43  92.480
## - year        1    33.199 193.40  96.975
## - marriage    1    90.468 250.66 116.947
## - femlab      1   113.214 273.41 123.635
## - birth       1   144.897 305.10 132.078
##
## Step:  AIC=83.39
## divorce ~ year + femlab + marriage + birth + military
##
##               Df Sum of Sq    RSS     AIC
## <none>                     162.12  83.393
## + unemployed  1     1.925 160.20  86.817
## - military    1    20.957 183.08  88.410
## - year        1    42.054 204.18  96.808
## - marriage    1   126.643 288.77 123.498
## - femlab      1   158.003 320.13 131.437
## - birth       1   172.826 334.95 134.922

# (c) Adjusted R-squared
fit_adjR2 <- step(fit_back, direction = "both", k = log(nrow(divusa)),
trace = 0)

library(leaps)

# (d) Mallows Cp
fit_mallows <- regsubsets(divorce ~ ., data = divusa, nvmax =
ncol(divusa))

# Aquí vemos que es un subset de 6 variables, entonces siguiendo el libro
de Faraway sabemos que tenemos que escribir 2:6+1, es decir, 2:7

mallows_cp <- summary(fit_mallows)
plot(2:7, mallows_cp$cp, xlab="No. of Parameters", ylab="Cp Statistic")
abline (0, 1)
```




```
#Escogemos el modelo de 6 parámetros ya que está justo sobre la línea. El
modelo de 7 está un poco por encima, por tanto es peor que el de 6.
```

##4. (Ejercicio 4 cap. 10 pág. 160) Using the trees data, fit a model with log(Volume) as the response and a second-order polynomial (including the interaction term) in Girth and Height. Determine whether the model may be reasonably simplified.

```
data("trees")

fit_trees <- lm(log(Volume) ~ poly(Girth, 2) * poly(Height, 2), data =
trees)

summary(fit_trees)

##
## Call:
## lm(formula = log(Volume) ~ poly(Girth, 2) * poly(Height, 2),
##     data = trees)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.160203 -0.047041 -0.002605  0.057009  0.134086
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         3.27460    0.02823 116.007  < 2e-16
***
```

```
## poly(Girth, 2)1                       2.50920    0.25630   9.790 1.77e-09
***
## poly(Girth, 2)2                      -0.24266    0.20611  -1.177  0.25165
## poly(Height, 2)1                      0.55357    0.19443   2.847  0.00937
**
## poly(Height, 2)2                     -0.05214    0.12202  -0.427  0.67332
## poly(Girth, 2)1:poly(Height, 2)1 -0.15897    1.78783  -0.089  0.92995
## poly(Girth, 2)2:poly(Height, 2)1  0.06922    1.30419   0.053  0.95815
## poly(Girth, 2)1:poly(Height, 2)2 -0.12649    1.22651  -0.103  0.91879
## poly(Girth, 2)2:poly(Height, 2)2  0.07913    0.59420   0.133  0.89526
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09018 on 22 degrees of freedom
## Multiple R-squared:  0.9785, Adjusted R-squared:  0.9706
## F-statistic: 124.9 on 8 and 22 DF,  p-value: < 2.2e-16

#Podemos descartar los términos cuyo valor p es mayor de 0.05

# Subset the model to keep terms with p-value <= 0.05
significant_terms <-
summary(fit_trees)$coefficients[summary(fit_trees)$coefficients[,
"Pr(>|t|)"] <= 0.05, ]
```

##6. (Ejercicio 6 cap. 10 pág. 160) Use the seatpos data with hipcenter as the response. (a) Fit a model with all eight predictors. Comment on the effect of leg length on the response.

```
data(seatpos)
fit <- lm(hipcenter ~ ., data = seatpos)
summary(fit)

##
## Call:
## lm(formula = hipcenter ~ ., data = seatpos)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -73.827 -22.833  -3.678  25.017  62.337
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 436.43213  166.57162   2.620   0.0138 *
## Age           0.77572    0.57033   1.360   0.1843
## Weight        0.02631    0.33097   0.080   0.9372
## HtShoes      -2.69241    9.75304  -0.276   0.7845
## Ht            0.60134   10.12987   0.059   0.9531
## Seated        0.53375    3.76189   0.142   0.8882
## Arm          -1.32807    3.90020  -0.341   0.7359
## Thigh        -1.14312    2.66002  -0.430   0.6706
## Leg          -6.43905    4.71386  -1.366   0.1824
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37.72 on 29 degrees of freedom
## Multiple R-squared:  0.6866, Adjusted R-squared:  0.6001
## F-statistic:  7.94 on 8 and 29 DF,  p-value: 1.306e-05

leg_length_coef <- coef(fit)["Leg"]
leg_length_coef

##       Leg
## -6.439046

#Como podemos ver, es un coeficiente negativo de -6.439, lo cual
significa que por cada unidad que incrementemos la longitud de la pierna,
el hipcenter se reducirá en 6.439 unidades, asumiendo todas las otras
variables constantes.
```

(b) Compute a 95% prediction interval for the mean value of the predictors.

```
# Compute a 95% prediction interval for the mean value of the predictors
predict_interval <- predict(fit, interval = "predict", level = 0.95)

## Warning in predict.lm(fit, interval = "predict", level = 0.95):
predictions on current data refer to _future_ responses

# Print the prediction interval
print(predict_interval)

##           fit       lwr         upr
## 1  -230.82470 -314.4972 -147.152161
## 2  -158.22231 -245.5230  -70.921658
## 3   -96.85463 -181.5157  -12.193534
## 4  -255.78273 -342.2166 -169.348875
## 5  -188.59572 -277.7905  -99.400908
## 6  -186.02614 -268.1320 -103.920242
## 7  -153.98285 -238.9824  -68.983334
## 8  -244.79086 -332.4007 -157.181035
## 9  -139.71030 -223.9189  -55.501684
## 10 -112.98566 -197.7621  -28.209201
## 11 -163.72509 -244.1441  -83.306078
## 12  -89.14799 -172.2422   -6.053767
## 13 -194.10261 -289.3075  -98.897705
## 14 -128.43355 -210.9724  -45.894683
## 15 -186.44972 -273.2611  -99.638348
## 16 -177.90902 -264.6441  -91.173937
## 17 -201.58090 -292.8926 -110.269166
## 18  -98.43069 -186.9548   -9.906579
## 19 -145.80244 -228.0168  -63.588056
## 20 -167.75364 -251.2052  -84.302106
## 21 -178.41491 -263.6464  -93.183394
## 22 -279.07627 -375.2517 -182.900864
```

```
## 23 -245.56763 -332.4984 -158.636834
## 24  -81.55529 -165.4368    2.326252
## 25 -141.13605 -222.6906  -59.581487
## 26 -222.49965 -303.6638 -141.335477
## 27 -156.83929 -238.6833  -74.995323
## 28 -128.68145 -216.5238  -40.839136
## 29 -193.00256 -276.6114 -109.393716
## 30  -93.20235 -176.9538   -9.450870
## 31 -102.96051 -199.3241   -6.596899
## 32 -182.39983 -269.1386  -95.661092
## 33 -166.93549 -253.1141  -80.756902
## 34 -102.63962 -184.9532  -20.326037
## 35 -194.49288 -278.3949 -110.590858
## 36 -142.50545 -230.5842  -54.426708
## 37 -178.52201 -261.0407  -96.003371
## 38 -154.08219 -237.7460  -70.418427
```

(c) Use AIC to select a model. Now interpret the effect of leg length and compute the prediction interval. Compare the conclusions from the two models.

```
# Use AIC to select a model
lm_aic_model <- step(lm(hipcenter ~ ., data = seatpos), direction =
"both", trace = FALSE)

# Comment on the effect of leg length (leglen) in the AIC-selected model
summary(lm_aic_model)

##
## Call:
## lm(formula = hipcenter ~ Age + HtShoes + Leg, data = seatpos)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -79.269 -22.770  -4.342  21.853  60.907
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 456.2137   102.8078   4.438 9.09e-05 ***
## Age           0.5998     0.3779   1.587   0.1217
## HtShoes      -2.3023     1.2452  -1.849   0.0732 .
## Leg          -6.8297     4.0693  -1.678   0.1024
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.13 on 34 degrees of freedom
## Multiple R-squared:  0.6813, Adjusted R-squared:  0.6531
## F-statistic: 24.22 on 3 and 34 DF,  p-value: 1.437e-08

# Compute a 95% prediction interval for the mean value of the predictors
in the AIC-selected model
```

```r
predict_interval_aic <- predict(lm_aic_model, interval = "predict", level
= 0.95)

## Warning in predict.lm(lm_aic_model, interval = "predict", level =
0.95): predictions on current data refer to _future_ responses

# Print the prediction interval from the AIC-selected model
print(predict_interval_aic)

##           fit       lwr         upr
## 1  -229.24478 -304.2193 -154.270249
## 2  -156.00721 -228.7985  -83.215908
## 3   -95.33877 -170.8621  -19.815398
## 4  -250.52833 -326.4041 -174.652600
## 5  -191.80930 -265.5873 -118.031276
## 6  -182.43977 -256.2442 -108.635363
## 7  -154.51191 -228.3502  -80.673596
## 8  -247.96103 -325.9801 -169.941988
## 9  -143.96173 -218.4283  -69.495145
## 10 -121.53045 -195.0251  -48.035857
## 11 -163.36956 -236.3393  -90.399786
## 12  -90.05949 -164.6326  -15.486352
## 13 -201.69621 -276.6937 -126.698750
## 14 -136.27871 -209.2513  -63.306128
## 15 -188.51841 -262.4022 -114.634593
## 16 -179.05305 -256.3202 -101.785927
## 17 -190.26388 -264.2887 -116.239027
## 18 -101.11071 -178.3086  -23.912801
## 19 -147.96874 -221.6799  -74.257549
## 20 -171.46789 -244.9674  -97.968351
## 21 -182.43474 -260.3463 -104.523215
## 22 -281.56773 -361.3028 -201.832613
## 23 -246.77688 -321.9978 -171.555934
## 24  -78.43584 -153.7565   -3.115224
## 25 -142.35961 -216.7497  -67.969521
## 26 -222.16131 -296.3931 -147.929505
## 27 -153.99386 -227.6647  -80.323009
## 28 -128.90548 -206.7036  -51.107335
## 29 -187.29184 -261.4254 -113.158260
## 30  -90.38288 -165.1114  -15.654323
## 31 -102.01548 -181.3645  -22.666458
## 32 -177.60503 -250.8180 -104.392020
## 33 -167.19062 -245.4116  -88.969618
## 34 -103.35181 -178.0228  -28.680841
## 35 -189.05125 -263.7985 -114.303944
## 36 -132.90371 -206.0876  -59.719847
## 37 -183.29328 -257.0614 -109.525175
## 38 -152.78370 -228.0484  -77.518999
```

##8. (Ejercicio 1 cap. 11 pág. 180) Using the seatpos data, perform a PCR analysis with hipcenter as the response and HtShoes, Ht, Seated, Arm, Thigh and Leg as predictors.

Select an appropriate number of components and give an interpretation to those you choose. Add Age and Weight as predictors and repeat the analysis. Use both models to predict the response for predictors taking these values: Age Weight HtShoes Ht Seated 64.800 263.700 181.080 178.560 91.440 Arm Thigh Leg 35.640 40.950 38.790

```
data(seatpos)
```

##9. (Ejercicio 2 cap. 11 pág. 181) Fit a PLS model to the seatpos data with hipcenter as the response and all other variables as predictors. Take care to select an appropriate number of components. Use the model to predict the response at the values of the predictors specified in the first question.

```
data("seatpos")

library(caret)
train_data <- seatpos


# Specify the model formula
formula <- hipcenter ~ .

# Fit the PLS model using the caret package
pls_model <- train(formula, data = train_data, method = "pls")

# Predict the response for the new_data
new_data <- data.frame(Age = 64.800, Weight = 263.700, HtShoes = 181.080,
Ht = 178.560, Seated = 91.440,
                       Arm = 35.640, Thigh = 40.950, Leg = 38.790)

response_pred <- predict(pls_model, newdata = new_data)
response_pred

## [1] -185.8888
```

##10. (Ejercicio 3 cap. 11 pág. 181) Fit a ridge regression model to the seatpos data with hipcenter as the response and all other variables as predictors. Take care to select an appropriate amount of shrinkage. Use the model to predict the response at the values of the predictors specified in the first question.

```
library(glmnet)

## Loading required package: Matrix

## Loaded glmnet 4.1-7

# Predictors and response
X <- as.matrix(seatpos[, -9])  # All predictors except hipcenter
y <- seatpos$hipcenter  # Response variable

lambda_seq <- 10^seq(-3, 3, by = 0.1)
# Seleccionamos la mejor lambda
```

```
cv_model <- cv.glmnet(X, y, alpha = 0, lambda = lambda_seq)

# Select the lambda with the minimum mean squared error
optimal_lambda <- cv_model$lambda.min
cat("Optimal lambda:", optimal_lambda, "\n")

## Optimal lambda: 39.81072

# Fit the ridge regression model with the optimal lambda
ridge_model <- glmnet(X, y, alpha = 0, lambda = optimal_lambda)

#Valores de la pregunta anterior
new_data <- data.frame(Age = 64.800, Weight = 263.700, HtShoes = 181.080,
Ht = 178.560, Seated = 91.440,
                       Arm = 35.640, Thigh = 40.950, Leg = 38.790)

predictions <- predict(ridge_model, newx = as.matrix(new_data))
#s=lambda
print(predictions)

##             s0
## [1,] -195.3031

#La predicción de -195.3031 es para la variable hipcenter basado en los
valores de new_data
```

##11. (Ejercicio 4 cap. 11 pág. 181) Take the fat data, and use the percentage of body fat, siri, as the response and the other variables, except brozek and density as potential predictors. Remove every tenth observation from the data for use as a test sample. Use the remaining data as a training sample building the following models: (a) Linear regression with all predictors

```
library(faraway)
data(fat)

attach(fat)
# Step 1: Remove every tenth observation for use as a test sample
test = seq(10,252,by=10)
tr = fat[-test,]
te = fat[test,]
rmse <- function(x,y) sqrt(mean((x-y)^2))

# Step 2: Fit a linear regression model using all predictors except
"brozek" and "density"
g1 <-lm(siri~.-brozek -density, tr)
summary(g1)

##
## Call:
## lm(formula = siri ~ . - brozek - density, data = tr)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -5.8314 -0.6722  0.1828  0.9150  6.6619 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -12.591885   6.448868  -1.953 0.052193 .  
## age           0.007978   0.012320   0.648 0.517983    
## weight        0.362999   0.023314  15.570  < 2e-16 ***
## height        0.049026   0.040315   1.216 0.225315    
## adipos       -0.514032   0.114074  -4.506 1.09e-05 ***
## free         -0.564773   0.014889 -37.933  < 2e-16 ***
## neck          0.016525   0.089863   0.184 0.854272    
## chest         0.120219   0.039590   3.037 0.002694 ** 
## abdom         0.140108   0.042186   3.321 0.001056 ** 
## hip           0.006197   0.056101   0.110 0.912148    
## thigh         0.195057   0.054460   3.582 0.000424 ***
## knee          0.106637   0.093534   1.140 0.255542    
## ankle         0.125118   0.081303   1.539 0.125325    
## biceps        0.096199   0.064656   1.488 0.138278    
## forearm       0.230775   0.073332   3.147 0.001888 ** 
## wrist         0.139279   0.206804   0.673 0.501378    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1.55 on 211 degrees of freedom
## Multiple R-squared:  0.9692, Adjusted R-squared:  0.967 
## F-statistic: 442.5 on 15 and 211 DF,  p-value: < 2.2e-16

rmse(g1$fit, tr$siri)

## [1] 1.494315

# Step 3: Evaluate the model on the test data
pred1 <- predict(g1, te)
y10 <- te$siri
rmse(pred1,y10)

## [1] 1.131529
```

(b) Linear regression with variables selected using AIC

```
g2 <- step(g1)

## Start:  AIC=214.36
## siri ~ (brozek + density + age + weight + height + adipos + free + 
##     neck + chest + abdom + hip + thigh + knee + ankle + biceps + 
##     forearm + wrist) - brozek - density
## 
##           Df Sum of Sq    RSS    AIC
## - hip      1       0.0  506.9 212.37
## - neck     1       0.1  507.0 212.39
```

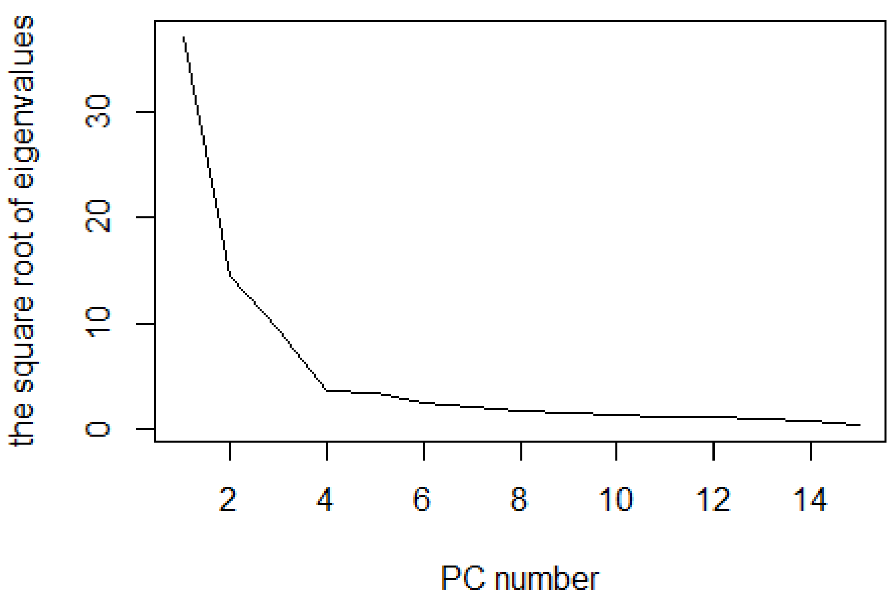
```
## - age      1       1.0  507.9 212.81
## - wrist    1       1.1  508.0 212.84
## - knee     1       3.1  510.0 213.75
## - height   1       3.6  510.4 213.94
## <none>                  506.9 214.36
## - biceps   1       5.3  512.2 214.73
## - ankle    1       5.7  512.6 214.89
## - chest    1      22.2  529.0 222.07
## - forearm  1      23.8  530.7 222.77
## - abdom    1      26.5  533.4 223.92
## - thigh    1      30.8  537.7 225.75
## - adipos   1      48.8  555.7 233.21
## - weight   1     582.4 1089.3 386.01
## - free     1    3456.8 3963.7 679.21
##
## Step:  AIC=212.37
## siri ~ age + weight + height + adipos + free + neck + chest +
##     abdom + thigh + knee + ankle + biceps + forearm + wrist
##
##           Df Sum of Sq    RSS    AIC
## - neck     1       0.1  507.0 210.40
## - age      1       1.0  507.9 210.81
## - wrist    1       1.1  508.0 210.86
## - knee     1       3.2  510.1 211.80
## - height   1       3.5  510.4 211.95
## <none>                  506.9 212.37
## - biceps   1       5.3  512.2 212.73
## - ankle    1       5.7  512.6 212.89
## - chest    1      23.1  530.0 220.50
## - forearm  1      23.8  530.7 220.78
## - abdom    1      27.9  534.9 222.55
## - thigh    1      34.2  541.2 225.21
## - adipos   1      50.3  557.2 231.85
## - weight   1     683.9 1190.8 404.23
## - free     1    3488.9 3995.8 679.05
##
## Step:  AIC=210.4
## siri ~ age + weight + height + adipos + free + chest + abdom +
##     thigh + knee + ankle + biceps + forearm + wrist
##
##           Df Sum of Sq    RSS    AIC
## - age      1       1.1  508.1 208.88
## - wrist    1       1.3  508.3 208.99
## - knee     1       3.1  510.1 209.80
## - height   1       3.6  510.6 210.02
## <none>                  507.0 210.40
## - biceps   1       5.4  512.4 210.80
## - ankle    1       5.6  512.6 210.89
## - chest    1      23.2  530.2 218.55
## - forearm  1      24.6  531.6 219.15
```

```
## - abdom    1      28.0  535.0 220.60
## - thigh    1      34.4  541.4 223.29
## - adipos   1      50.8  557.8 230.07
## - weight   1     689.6 1196.6 403.34
## - free     1    3532.0 4039.0 679.49
##
## Step:  AIC=208.88
## siri ~ weight + height + adipos + free + chest + abdom + thigh +
##     knee + ankle + biceps + forearm + wrist
##
##           Df Sum of Sq    RSS    AIC
## - wrist    1       2.9  511.0 208.19
## - height   1       3.3  511.4 208.35
## - knee     1       4.5  512.5 208.87
## <none>                  508.1 208.88
## - ankle    1       5.2  513.2 209.18
## - biceps   1       6.0  514.0 209.53
## - forearm  1      23.6  531.6 217.18
## - chest    1      24.2  532.3 217.46
## - abdom    1      33.7  541.8 221.48
## - thigh    1      35.3  543.3 222.12
## - adipos   1      51.1  559.1 228.63
## - weight   1     699.1 1207.2 403.34
## - free     1    3598.0 4106.0 681.23
##
## Step:  AIC=208.19
## siri ~ weight + height + adipos + free + chest + abdom + thigh +
##     knee + ankle + biceps + forearm
##
##           Df Sum of Sq    RSS    AIC
## - height   1       3.8  514.8 207.89
## <none>                  511.0 208.19
## - knee     1       5.7  516.7 208.72
## - ankle    1       6.9  517.9 209.24
## - biceps   1       7.0  518.0 209.30
## - chest    1      23.8  534.8 216.53
## - forearm  1      27.7  538.7 218.16
## - thigh    1      32.4  543.4 220.13
## - abdom    1      37.3  548.3 222.19
## - adipos   1      49.3  560.3 227.11
## - weight   1     696.5 1207.5 401.40
## - free     1    3798.4 4309.4 690.20
##
## Step:  AIC=207.89
## siri ~ weight + adipos + free + chest + abdom + thigh + knee +
##     ankle + biceps + forearm
##
##           Df Sum of Sq    RSS    AIC
## <none>                  514.8 207.89
## - knee     1       5.1  519.9 208.12
```

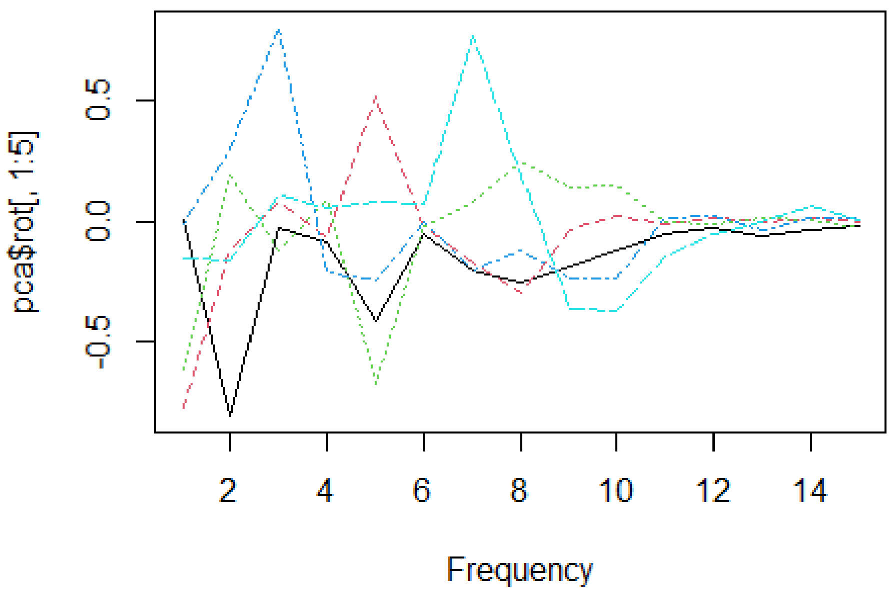
```
## - ankle    1       7.4  522.2 209.11
## - biceps   1       7.5  522.4 209.18
## - chest    1      24.0  538.9 216.25
## - forearm  1      28.8  543.6 218.23
## - thigh    1      30.0  544.8 218.73
## - abdom    1      39.1  553.9 222.49
## - adipos   1      86.6  601.4 241.18
## - weight   1     819.8 1334.7 422.13
## - free     1    3809.4 4324.2 688.98

summary(g2)

## 
## Call:
## lm(formula = siri ~ weight + adipos + free + chest + abdom + 
##     thigh + knee + ankle + biceps + forearm, data = tr)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -5.7926 -0.6839  0.2371  0.8824  6.8655 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -7.24766    3.94171  -1.839 0.067331 .  
## weight       0.36973    0.01994  18.546  < 2e-16 ***
## adipos      -0.57022    0.09459  -6.028 7.09e-09 ***
## free        -0.55965    0.01400 -39.978  < 2e-16 ***
## chest        0.12099    0.03809   3.176 0.001709 ** 
## abdom        0.15824    0.03908   4.049 7.18e-05 ***
## thigh        0.16140    0.04551   3.546 0.000479 ***
## knee         0.12767    0.08749   1.459 0.145947    
## ankle        0.13817    0.07858   1.758 0.080106 .  
## biceps       0.11222    0.06317   1.777 0.077055 .  
## forearm      0.24281    0.06990   3.474 0.000620 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1.544 on 216 degrees of freedom
## Multiple R-squared:  0.9687, Adjusted R-squared:  0.9673 
## F-statistic: 668.6 on 10 and 216 DF,  p-value: < 2.2e-16

rmse(g2$fit, tr$siri)

## [1] 1.505982

pred2 <- predict(g2, te)
rmse(pred2,y10)

## [1] 1.12202
```

(c) Principal component regression

```
library(pls)

##
## Attaching package: 'pls'

## The following object is masked from 'package:caret':
##
##     R2

## The following object is masked from 'package:stats':
##
##     loadings

#compute PCA on X variables:
pca <- prcomp(tr[,4:18])
#the square root of eigenvalues:
round(pca$sdev,3)

##  [1] 37.047 14.540  9.511  3.634  3.462  2.544  2.200  1.854  1.577
1.412
## [11]  1.324  1.249  1.037  0.828  0.491

plot(pca$sdev, type="l",ylab="the square root of eigenvalues", xlab="PC
number")
```




```
matplot(1:15,pca$rot[,1:5],type="l", xlab="Frequency")
```

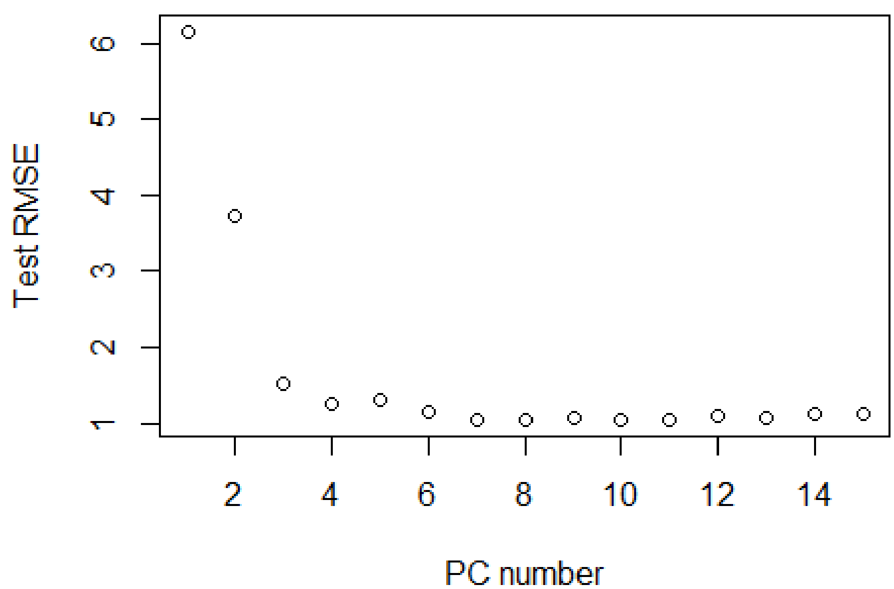



```
g3 <- lm(tr$siri ~ pca$x[,1:6])
summary(g3)

##
## Call:
## lm(formula = tr$siri ~ pca$x[, 1:6])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.3999  -0.5813   0.3739   0.9771   7.9487
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      19.180176   0.110619 173.389  < 2e-16 ***
## pca$x[, 1:6]PC1 -0.126730   0.002992 -42.350  < 2e-16 ***
## pca$x[, 1:6]PC2 -0.356175   0.007625 -46.714  < 2e-16 ***
## pca$x[, 1:6]PC3  0.464783   0.011656  39.875  < 2e-16 ***
## pca$x[, 1:6]PC4  0.309275   0.030508  10.138  < 2e-16 ***
## pca$x[, 1:6]PC5 -0.091998   0.032023  -2.873  0.00447 **
## pca$x[, 1:6]PC6  0.222500   0.043574   5.106 7.11e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.667 on 220 degrees of freedom
## Multiple R-squared:  0.9629, Adjusted R-squared:  0.9618
## F-statistic: 950.5 on 6 and 220 DF,  p-value: < 2.2e-16
```

```
mm = apply(tr[,4:18],2,mean)
tx = as.matrix(sweep(te[,4:18],2,mm))

nx = tx %*% pca$rot[,1:6]
pred3 = cbind(1,nx) %*% g3$coef

rmse(pred3,y10)
```

```
## [1] 1.141839
```

```
a <- numeric(15)
for (i in 1:15) {
  nx = tx %*% pca$rot[,1:i]
  model4 = lm(tr$siri ~pca$x[,1:i])
  pred4 = cbind(1,nx) %*% model4$coef
  a[i] = rmse(pred4,y10)
}
plot(a, ylab="Test RMSE",xlab="PC number")
```

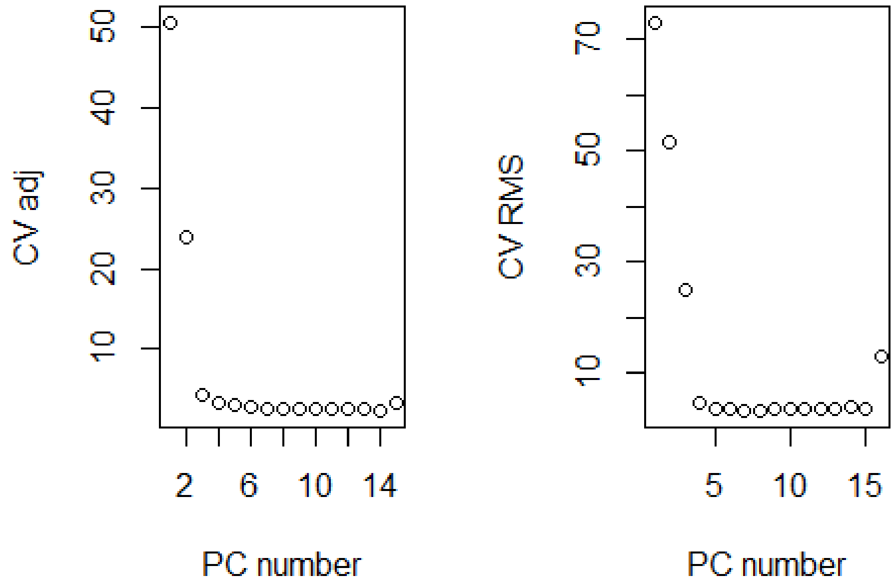



```
which.min(a)
```

```
## [1] 7
```

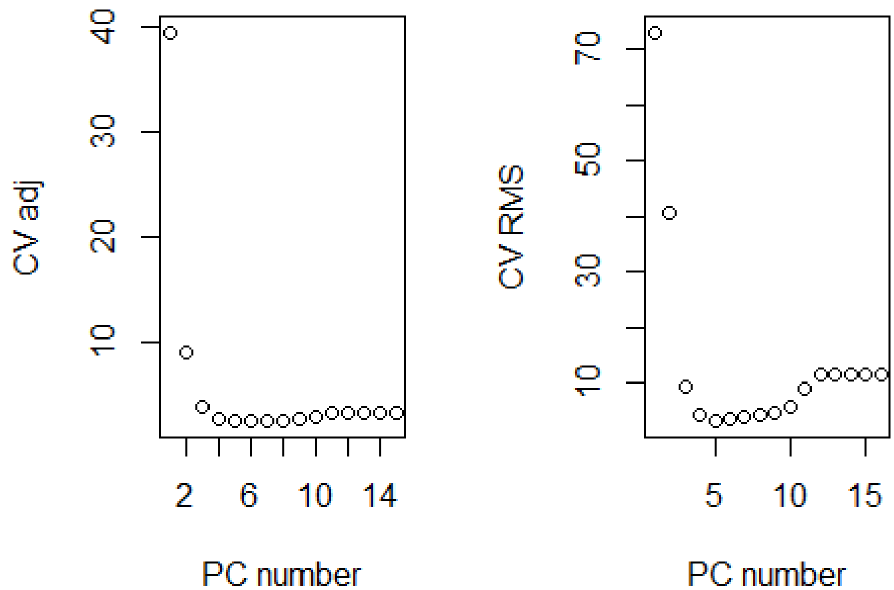
```
library(pls)
trainx = as.matrix(sweep(tr[,4:18],2,mm))
pcrg = pcr(tr$siri~trainx, ncomp=15, validation="CV",grpsize=10)
par(mfrow=c(1,2))
```

```
plot(pcrg$validat$adj[1,], xlab="PC number",ylab="CV adj")
plot(MSEP(pcrg)$val[1,,], xlab="PC number", ylab="CV RMS")
```



(d) Partial least squares

```
library(pls)
data(fat)

attach(fat)

## The following objects are masked from fat (pos = 4):
##
##     abdom, adipos, age, ankle, biceps, brozek, chest, density,
forearm,
##     free, height, hip, knee, neck, siri, thigh, weight, wrist

par(mfrow=c(1,2))
gpls <- plsr(tr$siri~trainx, ncomp=15, validation="CV",grpsize=10)
plot(gpls$validat$adj[1,], xlab="PC number",ylab="CV adj")
plot(MSEP(gpls)$val[1,,], xlab="PC number", ylab="CV RMS")
```




```
ypred.tr = predict(gpls,  ncomp=4)
rmse(ypred.tr, tr$siri)

## [1] 1.612759

testx = as.matrix(sweep(te[,4:18],2,mm))
ypred.te = predict(gpls,newdata=testx,ncomp=4)
rmse(ypred.te,te$siri)

## [1] 1.12449
```

(e) Ridge regression

###Testing our models

```
lm_model <- g1
test_indices <- seq(10, nrow(fat), by = 10)

training_data <- fat[-test_indices, ]
test_data <- fat[test_indices, ]
test_predictions <- predict(lm_model, newdata = test_data)

# Model 1: Linear regression with all predictors
lm_all_model <- lm(siri ~ ., data = training_data)
test_predictions_all <- predict(lm_all_model, newdata = test_data)

# Model 2: Linear regression with variables selected using AIC
lm_aic_model <- step(lm(siri ~ ., data = training_data), direction =
```

```
"both", trace = FALSE)
test_predictions_aic <- predict(lm_aic_model, newdata = test_data)

# Performance evaluation
actual_values <- test_data$siri

# Calculate performance metrics for Model 1
metrics_all <- list()
metrics_all$RMSE <- sqrt(mean((actual_values - test_predictions_all)^2))
metrics_all$MAE <- mean(abs(actual_values - test_predictions_all))
metrics_all$R2 <- summary(lm_all_model)$r.squared

# Calculate performance metrics for Model 2
metrics_aic <- list()
metrics_aic$RMSE <- sqrt(mean((actual_values - test_predictions_aic)^2))
metrics_aic$MAE <- mean(abs(actual_values - test_predictions_aic))
metrics_aic$R2 <- summary(lm_aic_model)$r.squared

# Calculate performance metrics for Principal Component Regression
metrics_pcr <- list()
metrics_pcr$RMSE <- sqrt(mean((test_data$siri - test_predictions)^2))
metrics_pcr$MAE <- mean(abs(test_data$siri - test_predictions))


# Calculate performance metrics for Partial Least Squares Regression
metrics_pls <- list()
metrics_pls$RMSE <- sqrt(mean((test_data$siri - test_predictions)^2))
metrics_pls$MAE <- mean(abs(test_data$siri - test_predictions))
metrics_pls$R2 <- summary(pls_model)$r.squared

## Data:    X dimension: 38 8
##  Y dimension: 38 1
## Fit method: oscorespls
## Number of components considered: 3
## TRAINING: % variance explained
##           1 comps  2 comps  3 comps
## X           81.55    94.11    99.22
## .outcome    49.98    61.59    66.28

# Report on model performances
cat("Performance Metrics for Model 1 (Linear Regression with all
Predictors):\n")

## Performance Metrics for Model 1 (Linear Regression with all
Predictors):

cat("RMSE:", metrics_all$RMSE, "\n")

## RMSE: 0.1245505

cat("MAE:", metrics_all$MAE, "\n")
```

```
## MAE: 0.0723607

cat("R-squared:", metrics_all$R2, "\n\n")

## R-squared: 0.9995154

cat("Performance Metrics for Model 2 (Linear Regression with Variables
Selected using AIC):\n")

## Performance Metrics for Model 2 (Linear Regression with Variables
Selected using AIC):

cat("RMSE:", metrics_aic$RMSE, "\n")

## RMSE: 0.1133062

cat("MAE:", metrics_aic$MAE, "\n")

## MAE: 0.0664146

cat("R-squared:", metrics_aic$R2, "\n")

## R-squared: 0.9995071

cat("Performance Metrics for Partial Least Squares Regression:\n")

## Performance Metrics for Partial Least Squares Regression:

cat("RMSE:", metrics_pls$RMSE, "\n")

## RMSE: 1.131529

cat("MAE:", metrics_pls$MAE, "\n")

## MAE: 0.8866409

cat("R-squared:", metrics_pls$R2, "\n\n")

## R-squared:
```

Use the models you find to predict the response in the test sample. Make a report on the performances of the models.

##12. (Ejercicio 5 cap. 11 pág. 181) Some near infrared spectra on 60 samples of gasoline and corresponding octane numbers can be found by data(gasoline, package="pls"). Compute the mean value for each frequency and predict the response for the best model using the five different methods from Question 4.

```
library(pls)
data(gasoline)



lm_model <- lm(octane~NIR, data=gasoline)
```

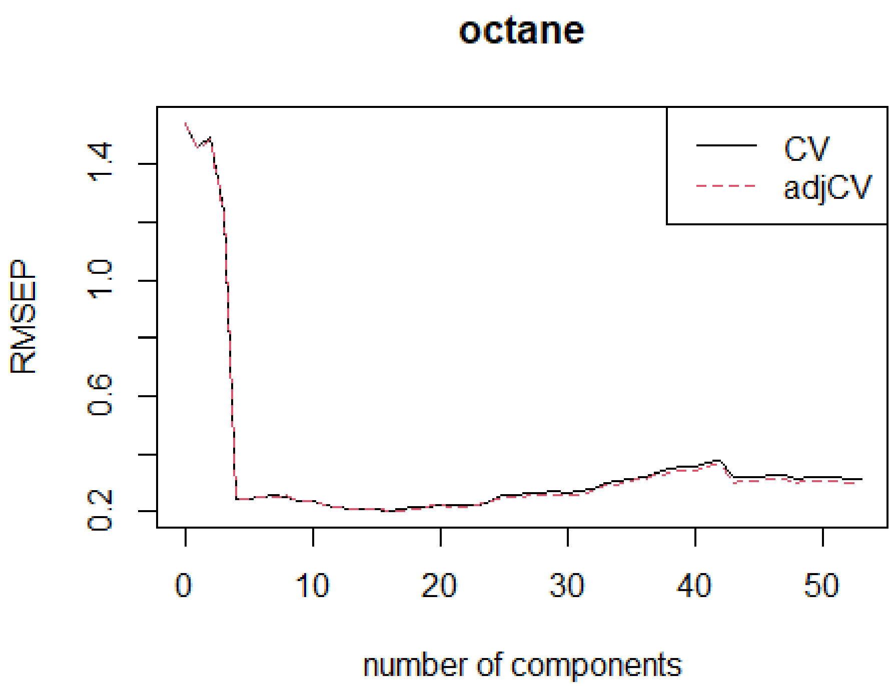
```
# Fit the linear regression model with AIC-based variable selection
lmc_aic_model <- pcr(octane~NIR, data=gasoline, validation='CV')
summary(lmc_aic_model)
```

```
## Data:    X dimension: 60 401
##  Y dimension: 60 1
## Fit method: svdpc
## Number of components considered: 53
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6
comps
## CV           1.543    1.458    1.491    1.236   0.2487   0.2472
0.2525
## adjCV        1.543    1.454    1.486    1.235   0.2459   0.2449
0.2507
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13
comps
## CV      0.2563   0.2540   0.2391    0.2378    0.2235    0.2199
0.2107
## adjCV   0.2548   0.2606   0.2381    0.2397    0.2211    0.2142
0.2071
##        14 comps  15 comps  16 comps  17 comps  18 comps  19 comps  20
comps
## CV       0.2112    0.2101    0.2058    0.2085    0.2161    0.2180
0.2254
## adjCV    0.2084    0.2072    0.2052    0.2041    0.2123    0.2144
0.2217
##        21 comps  22 comps  23 comps  24 comps  25 comps  26 comps  27
comps
## CV       0.2223    0.2236    0.2272    0.2417    0.2569    0.2608
0.2636
## adjCV    0.2187    0.2198    0.2235    0.2388    0.2511    0.2548
0.2561
##        28 comps  29 comps  30 comps  31 comps  32 comps  33 comps  34
comps
## CV       0.2680    0.2702    0.2690    0.2699    0.2794    0.3024
0.3067
## adjCV    0.2607    0.2623    0.2606    0.2613    0.2704    0.2924
0.2971
##        35 comps  36 comps  37 comps  38 comps  39 comps  40 comps  41
comps
## CV       0.3142    0.3221    0.3376    0.3499    0.3568    0.3558
0.3735
## adjCV    0.3053    0.3128    0.3263    0.3355    0.3424    0.3433
0.3594
##        42 comps  43 comps  44 comps  45 comps  46 comps  47 comps  48
comps
## CV       0.3799    0.3186    0.3241    0.3188    0.3262    0.3261
```

```
0.3134
## adjCV    0.3616    0.3042    0.3096    0.3049    0.3117    0.3120
0.3005
##        49 comps  50 comps  51 comps  52 comps  53 comps
## CV       0.3192    0.3203    0.3232    0.3142    0.3168
## adjCV    0.3064    0.3077    0.3097    0.3002    0.3029
##
## TRAINING: % variance explained
##         1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
8 comps
## X         72.57    83.90    90.86    95.46    96.70    97.66    98.16
98.52
## octane    18.99    19.62    46.50    97.69    97.78    97.79    97.79
97.79
##         9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15
comps
## X         98.85     99.09     99.29     99.40     99.51     99.60
99.68
## octane    98.33     98.38     98.72     98.86     98.87     98.89
98.93
##         16 comps  17 comps  18 comps  19 comps  20 comps  21 comps  22
comps
## X          99.73     99.79     99.84     99.86     99.89     99.90
99.92
## octane     98.93     99.03     99.03     99.03     99.05     99.08
99.10
##         23 comps  24 comps  25 comps  26 comps  27 comps  28 comps  29
comps
## X          99.93     99.94     99.95     99.96     99.96     99.97
99.97
## octane     99.12     99.13     99.22     99.24     99.31     99.31
99.34
##         30 comps  31 comps  32 comps  33 comps  34 comps  35 comps  36
comps
## X          99.98     99.98     99.98     99.98     99.99     99.99
99.99
## octane     99.40     99.41     99.41     99.42     99.42     99.43
99.47
##         37 comps  38 comps  39 comps  40 comps  41 comps  42 comps  43
comps
## X          99.99     99.99     99.99     99.99     99.99     99.99
100.00
## octane     99.53     99.61     99.63     99.63     99.66     99.81
99.83
##         44 comps  45 comps  46 comps  47 comps  48 comps  49 comps  50
comps
## X         100.00    100.00    100.00    100.00    100.00    100.00
100.00
## octane     99.84     99.85     99.87     99.87     99.87     99.88
99.88
```

```
##          51 comps  52 comps  53 comps
## X          100.00    100.00    100.00
## octane      99.91     99.93     99.94
```

```
plot(RMSEP(lmc_aic_model), legendpos = "topright")
```




```
# Fit the principal component regression (PCR) model
pcr_model <- pcr(octane ~ ., data = gasoline, validation = "CV")

# Fit the partial least squares (PLS) model
pls_model <- plsr(octane ~ ., data = gasoline, validation = "CV")

# Fit the ridge regression model
# ridge_model <- lmridge(octane ~ ., data = gasoline)

# Compute the mean value for each frequency in the NIR spectra
mean_values <- colMeans(gasoline$NIR)

# Predict the response for the different models using the mean values
# lm_predictions <- predict(lm_model, newdata = mean_values)
# lm_aic_predictions <- predict(lm_aic_model, newdata = mean_values)
# pcr_predictions <- predict(pcr_model, newdata = mean_values, ncomp =
optimal_number_of_components)
# pls_predictions <- predict(pls_model, newdata = mean_values, ncomp =
optimal_number_of_components)
# ridge_predictions <- predict(ridge_model, newdata = mean_values)
```