hw7

Hoang Chu

2023-11-12

```r
library(faraway)
library(MASS)
```

10.1

```r
# Load the prostate data
data(prostate)

# Full model with lpsa as the response and all other variables as predictors
full_model <- lm(lpsa ~ ., data = prostate)

# (a) Backward elimination
backward_model <- stepAIC(full_model, 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
```

```
# (b) AIC
aic_model <- stepAIC(full_model, direction = "both")
```

```
## 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
```

```
## + gleason  1    0.0412 44.163 -58.322
## - 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
## + lcp      1    0.6623 44.204 -60.231
## - age      1    1.2649 46.131 -60.092
## - lbph     1    1.6465 46.513 -59.293
## + gleason  1    0.0296 44.837 -58.853
## - 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
## + pgg45    1    0.6590 44.867 -60.789
## + gleason  1    0.4560 45.070 -60.351
## + lcp      1    0.1293 45.396 -59.650
## - 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
# (c) Adjusted R-squared
# The model with the highest adjusted R-squared is considered the best
adj_r_squared <- summary(full_model)$adj.r.squared

# (d) Mallows Cp
# Compute the residuals and the predicted values
residuals <- resid(full_model)
predicted <- predict(full_model)

# Compute the error sum of squares
sse <- sum(residuals^2)

# Compute the total sum of squares
sst <- sum((predicted - mean(predicted))^2)

# Compute Cp
p <- length(coef(full_model)) - 1
n <- length(predicted)
mse <- sse / n
cp <- (sse + 2 * p * mse) / n
```

10.3

```r
# Load the divusa data
data(divusa)

# Full model with divorce as the response and all other variables as predictors
full_model <- lm(divorce ~ ., data = divusa)

# (a) Backward elimination
backward_model <- stepAIC(full_model, 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
```

```r
# (b) AIC
aic_model <- stepAIC(full_model, direction = "both")
```

```
## 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
## + unemployed  1     1.925 160.20  70.410
## - 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
```

```r
# (c) Adjusted R-squared
# The model with the highest adjusted R-squared is considered the best
adj_r_squared <- summary(full_model)$adj.r.squared

# (d) Mallows Cp
# Compute the residuals and the predicted values
residuals <- resid(full_model)
predicted <- predict(full_model)

# Compute the error sum of squares
sse <- sum(residuals^2)

# Compute the total sum of squares
sst <- sum((predicted - mean(predicted))^2)

# Compute Cp
p <- length(coef(full_model)) - 1
n <- length(predicted)
mse <- sse / n
cp <- (sse + 2 * p * mse) / n
```

11.2

```r
# Load the libraries
library(pls)
```

```
##
## Attaching package: 'pls'

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

```r
# Load the seatpos data
data(seatpos)

# Fit a PLS model with hipcenter as the response and all other variables as predictors
pls.model <- plsr(hipcenter ~ ., data = seatpos, ncomp = 2)

# Print a summary of the model
summary(pls.model)
```

```
## Data:    X dimension: 38 8
##  Y dimension: 38 1
## Fit method: kernelpls
## Number of components considered: 2
## TRAINING: % variance explained
##            1 comps  2 comps
## X            81.55    94.11
## hipcenter    49.98    61.59
```

```r
# Predict the response at the values of the predictors specified in the first question
# Assuming 'predictors' is a data frame containing the predictor values
predictors <- data.frame(seatpos[1, -which(names(seatpos) == "hipcenter")])
predictions <- predict(pls.model, predictors)

# Print the predictions
print(predictions)
```

```
## , , 1 comps
##
##   hipcenter
## 1 -200.5626
##
## , , 2 comps
##
##   hipcenter
## 1 -204.0235
```

11.4

```r
# Load necessary libraries
library(faraway)
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

```r
library(pls)

# Load the data
data(fat)

# Remove brozek and density columns
fat <- fat[, !(names(fat) %in% c("brozek", "density"))]

# Split the data into training and test sets
train_indices <- seq(1, nrow(fat), by = 10)
train_data <- fat[-train_indices, ]
test_data <- fat[train_indices, ]
```

```r
# (a) Linear regression with all predictors
model_a <- lm(siri ~ ., data = train_data)

# (b) Linear regression with variables selected using AIC
model_b <- step(lm(siri ~ ., data = train_data), trace = 0)

# (c) Principal component regression
model_c <- pcr(siri ~ ., data = train_data, validation = "CV")

# (d) Partial least squares
model_d <- plsr(siri ~ ., data = train_data, validation = "CV")

# (e) Ridge regression
x <- model.matrix(siri ~ . - 1, data = train_data)
y <- train_data$siri
model_e <- cv.glmnet(x, y, alpha = 0)

# Use the models to predict the response in the test sample
pred_a <- predict(model_a, newdata = test_data)
pred_b <- predict(model_b, newdata = test_data)
pred_c <- predict(model_c, newdata = test_data, ncomp = model_c$ncomp)
pred_d <- predict(model_d, newdata = test_data, ncomp = model_d$ncomp)
pred_e <- predict(model_e, newx = model.matrix(siri ~ . - 1, data = test_data), s = model_e$lambda.min)

# Report on the performances of the models
models <- list("Linear regression with all predictors" = pred_a,
               "Linear regression with variables selected using AIC" = pred_b,
               "Principal component regression" = pred_c,
               "Partial least squares" = pred_d,
               "Ridge regression" = pred_e)

for (name in names(models)) {
  cat(name, ": ", sqrt(mean((test_data$siri - models[[name]])^2)), "\n")
}
```

```
## Linear regression with all predictors :  1.946023
## Linear regression with variables selected using AIC :  1.98911
## Principal component regression :  1.946023
## Partial least squares :  1.946023
## Ridge regression :  2.575652
```

11.5

```r
# Load the data
data(gasoline, package="pls")

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

# Split the data into training and test sets
train_indices <- seq(1, nrow(gasoline), by = 10)
```

```r
train_data <- gasoline[-train_indices, ]
test_data <- gasoline[train_indices, ]

# (a) Linear regression with all predictors
model_a <- lm(octane ~ ., data = train_data)

# AIC is infinity for this model since there are very few observations compared to predictors -> overfi
# model_b <- step(lm(octane ~ ., data = train_data), trace = 0)

# (c) Principal component regression
model_c <- pcr(octane ~ ., data = train_data, validation = "CV")

# (d) Partial least squares
model_d <- plsr(octane ~ ., data = train_data, validation = "CV")

# (e) Ridge regression
x <- model.matrix(octane ~ . - 1, data = train_data)
y <- train_data$octane
model_e <- cv.glmnet(x, y, alpha = 0)

# Use the models to predict the response in the test sample
pred_a <- predict(model_a, newdata = test_data)
```

```
## Warning in predict.lm(model_a, newdata = test_data): prediction from
## rank-deficient fit; attr(*, "non-estim") has doubtful cases
```

```r
# pred_b <- predict(model_b, newdata = test_data)
pred_c <- predict(model_c, newdata = test_data, ncomp = model_c$ncomp)
pred_d <- predict(model_d, newdata = test_data, ncomp = model_d$ncomp)
pred_e <- predict(model_e, newx = model.matrix(octane ~ . - 1, data = test_data), s = model_e$lambda.mi

# Report on the performances of the models
models <- list("Linear regression with all predictors" = pred_a,
               "Principal component regression" = pred_c,
               "Partial least squares" = pred_d,
               "Ridge regression" = pred_e)

for (name in names(models)) {
  cat(name, ": ", sqrt(mean((test_data$octane - models[[name]])^2)), "\n")
}
```

```
## Linear regression with all predictors :  74.62211
## Principal component regression :  0.1901717
## Partial least squares :  0.175098
## Ridge regression :  0.3793098
```

There is a warning for a full linear regression model. This is due to the same reason as AIC (too few observations compared to predictors) and multi-collinearity. Hence the output value for the linear regression with all predictors is unreliable.