채널 G
2016125077 최재혁
2015125043 어성준
2015125080 표정진

데이터 사이언스

- 10주차 팀 과제 (회귀분석을 통한 예측, 설명모델 생성)

목 차

데이터 셋 선택, 속성

- 데이터 셋 결정
  - 이번에는 교수님께서 정해주신 Boston housing data set을 선택
  - Kaggle site에서 데이터를 수집
- 데이터 속성 조사

```
> df_train <- read.csv(file = "C:/Users/user/Desktop/3학년2학기/데이터사이언스/과제/10주/boston_data.cs
v",header=TRUE, fileEncoding = "UTF-8")
> str(df_train)
'data.frame':   404 obs. of  14 variables:
 $ crim   : num  0.1588 0.1033 0.3494 2.734 0.0434 ...
 $ zn     : num  0 25 0 0 21 45 22 0 0 0 ...
 $ indus  : num  10.81 5.13 9.9 19.58 5.64 ...
 $ chas   : num  0 0 0 0 0 0 0 0 0 1 ...
 $ nox    : num  0.413 0.453 0.544 0.871 0.439 0.437 0.431 0.544 0.584 0.871 ...
 $ rm     : num  5.96 5.93 5.97 5.6 6.12 ...
 $ age    : num  17.5 47.2 76.7 94.9 63 38.9 17.5 82.8 94.3 96 ...
 $ dis    : num  5.29 6.93 3.1 1.53 6.81 ...
 $ rad    : num  4 8 4 5 4 5 7 4 24 5 ...
 $ tax    : num  305 284 304 403 243 398 330 304 666 403 ...
 $ ptratio: num  19.2 19.7 18.4 14.7 16.8 15.2 19.1 18.4 20.2 14.7 ...
 $ black  : num  377 397 396 352 394 ...
 $ lstat  : num  9.88 9.22 9.97 21.45 9.43 ...
 $ medv   : num  21.7 19.6 20.3 15.4 20.5 34.9 26.2 21.6 14.1 17 ...
>
```

<사진 1> 객체 탐색

crim - 마을 별 1인당 범죄율
zn - 25000 평방 피트 이상의 부지에 구역화 된 주거용 토지 비율
indus - 도시 당 비 소매 사업 에이커의 비율
chas - Charles River 더미 변수 (지역이 강 경계면 = 1, 그렇지 않으면 0)
nox - 질소 산화물 농도 (1000 만분 율)
rm - 주거 당 평균 방 수
age - 1940 년 이전에 지어진 소유주가 소유 한 주택의 비율
dis - 5 개의 보스턴 고용 센터까지의 가중 평균 거리
rad - 방사형 고속도로에 대한 접근성 지수
tax - $10,000 당 전체 가치 재산 세율
ptratio - 도시 별 학생-교사 비율
black - 1000 (Bk-0.63) ^ 2 (여기서 Bk는 도시 별 흑인 비율)
lstat - 인구의 낮은 지위(%)
medv - 소유주가 거주하는 주택의 중간 가치 ($1000)

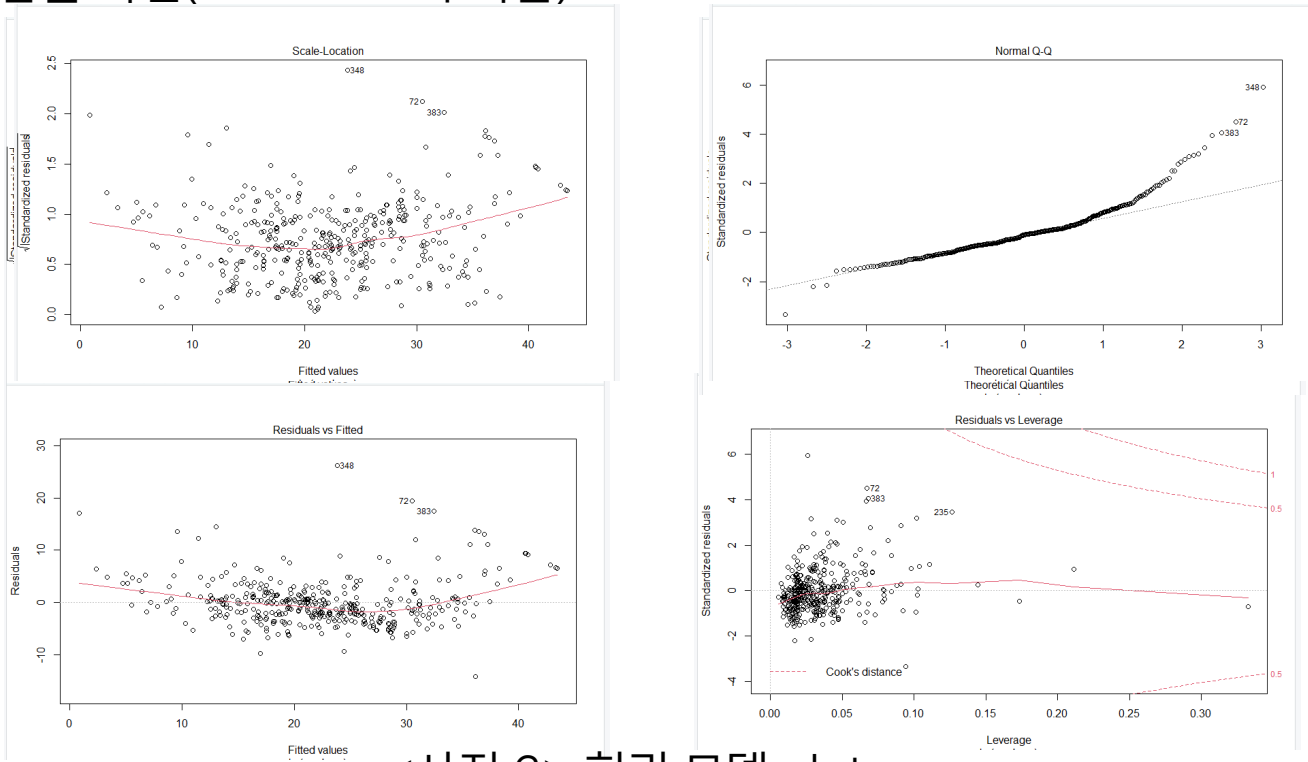
<사진 2> 속성 값에 대한 설명

설명 모델 생성

- 모델 생성
  - 회귀 분석을 실시, lm 함수를 이용한 다변량 회귀 모델 생성
  - 전체 데이터를 통해서 유의한 변수를 파악한다.
  - 또한 R-sqared 값을 통해 모델의 설명력을 확인할 수 있으며 (0.74로 비교적 높은 값 도출)
  - P-value 값이 0.05이하이기 때문에 변수들이 유의함을 확인(2.2e-16보다 작음)

```
> model <- lm(medv~.,data = df_train)
> summary(model)

Call:
lm(formula = medv ~ ., data = df_train)

Residuals:
     Min       1Q   Median       3Q      Max
-14.2637  -2.5392  -0.4509   1.5086  26.1848

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  33.740005   5.347397   6.310 7.60e-10 ***
crim         -0.116648   0.032276  -3.614 0.000341 ***
zn            0.039847   0.015512   2.569 0.010579 *
indus         0.012978   0.065082   0.199 0.842040
chas          1.612301   0.906347   1.779 0.076035 .
nox         -15.237963   3.929783  -3.878 0.000124 ***
rm            3.989207   0.443925   8.986  < 2e-16 ***
age          -0.004293   0.013716  -0.313 0.754458
dis          -1.335055   0.207485  -6.434 3.64e-10 ***
rad           0.275248   0.071390   3.856 0.000135 ***
tax          -0.012861   0.004029  -3.192 0.001525 **
ptratio      -0.916149   0.139499  -6.567 1.64e-10 ***
black         0.008367   0.002776   3.014 0.002748 **
lstat        -0.511245   0.054176  -9.437  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.473 on 390 degrees of freedom
Multiple R-squared:  0.7521,    Adjusted R-squared:  0.7438
F-statistic:     91 on 13 and 390 DF,  p-value: < 2.2e-16
```

<사진 1> 회귀 분석의 결과



<사진 2> 회귀 모델 plot

예측 모델 생성

- 데이터 partitioning
  - Caret package에 있는 createdatapartition함수를 이용하여 훈련데이터와 테스트데이터 파티셔닝 (75:25)
- 모델 생성
  - 앞서 설명 모델에서 생성한 유의한 회귀 변수들을 가지고 예측 모델을 생성한다

```
> indexes <- createDataPartition(y = df_train$medv, p = .75, list = FALSE)
> train <- df_train[indexes, ]
> test <- df_train[-indexes, ]
```

<사진 1> 훈련, 테스트 데이터 분리

```
> set.seed(100)
> model <- lm(medv~crim+zn+chas+nox+rm+dis+rad+tax+ptratio+black+lstat,data = train)
```

<사진 2> 예측 모델 생성

```
> summary (model)

Call:
lm(formula = medv ~ crim + zn + chas + nox + rm + dis + rad +
    tax + ptratio + black + lstat, data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-12.7280  -2.5786  -0.3944   1.7015  26.5728

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  36.739513   6.232846   5.895 1.04e-08 ***
crim         -0.132480   0.036376  -3.642 0.000320 ***
zn            0.052879   0.018880   2.801 0.005437 **
chas          1.019315   1.041294   0.979 0.328443
nox         -16.308294   4.139357  -3.940 0.000102 ***
rm            3.886285   0.512045   7.590 4.33e-13 ***
dis          -1.400868   0.232506  -6.025 5.08e-09 ***
rad           0.299656   0.080580   3.719 0.000240 ***
tax          -0.014634   0.004276  -3.422 0.000710 ***
ptratio      -1.012431   0.165779  -6.107 3.23e-09 ***
black         0.008722   0.003223   2.706 0.007215 **
lstat        -0.472800   0.058490  -8.083 1.69e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.443 on 292 degrees of freedom
Multiple R-squared:  0.7602,    Adjusted R-squared:  0.7512
F-statistic: 84.17 on 11 and 292 DF,  p-value: < 2.2e-16
```

<사진 3> 예측 모델

모델 예측

- 모델에 테스트 데이터 예측
  - Predict()함수를 이용하여 Partitioning한 데이터를 모델에 예측
- 모델의 정확도
  - 실제 레이블링한 데이터와 모델을 통해 예측한 값을 비교하여 정확도를 알아본다
  - Actuals는 실제 값, Predicteds는 예측 모델을 통한 predicted 값

```
> pred <- predict(model, test)
> pred
       14        17        18        24        25        29        32        39        40
24.203052 24.733020 10.670882 30.715147 22.263482 24.798009 28.056924 12.200332 35.695741
       48        49        53        58        61        69        71        77        83
18.933327 37.830119 35.284521 30.682435 15.803068 18.886062 30.038708 27.501842 20.090379
       87        90        97       107       111       113       116       118       122
25.259693  9.392222 19.640745 28.437722 21.477290 35.724949 21.130166 25.478374 20.439389
      123       125       129       134       135       142       144       147       151
13.658585  9.667054 31.199215 28.203426 17.779435 16.140400  5.200812 20.409497 22.102183
      159       169       171       175       187       188       190       192       195
13.759656 14.504698  4.885007 23.426951 28.993271 25.138630 26.627871 19.590045 35.561094
      201       203       218       221       225       230       231       232       237
43.919230 33.419163 16.509811 18.527139 31.463171 27.891820 20.842219 17.005230 20.029887
      239       241       244       245       246       253       260       261       263
34.302945 21.174559 26.503400 14.291345 30.670147 17.143758 25.217044 17.490457 21.077671
      266       268       271       272       277       284       285       286       288
22.348599 17.637062 16.368305 16.772124 22.567594 24.138024 23.236358  3.938430 32.269127
      290       291       293       294       300       301       305       313       320
15.047816 14.126883 25.012197 22.591332 23.171061 27.847405  9.333162 15.294448 30.163626
      322       330       332       338       351       353       364       366       370
30.877725 18.397507 14.682604 33.820266 31.114643 36.744536 12.969683 16.203966 24.520006
      374       382       383       386       387       388       392       395       396
24.437907 30.561955 31.197961 21.064975 19.409939 15.783215 28.075720 25.880008 22.512132
      400
16.628325
```

<사진 1> 모델에 예측

```
> actuals_preds <- data.frame(cbind(actuals=test$medv, predicteds=pred))
> actuals_preds
    actuals predicteds
14      22.2  24.203052
17      22.2  24.733020
18      12.0  10.670882
24      29.4  30.715147
25      16.5  22.263482
29      21.9  24.798009
32      36.2  28.056924
39      12.7  12.200332
40      35.1  35.695741
48      15.6  18.933327
49      43.1  37.830119
53      31.0  35.284521
58      34.9  30.682435
61      10.2  15.803068
69      17.1  18.886062
71      24.1  30.038708
77      22.0  27.501842
83      20.6  20.090379
87      21.6  25.259693
90      23.1   9.392222
97      18.4  19.640745
107     24.5  28.437722
111     19.0  21.477290
113     34.6  35.724949
116     20.1  21.130166
118     23.8  25.478374
122     21.8  20.439389
123     13.9  13.658585
125     11.8   9.667054
```

<사진 2> 실제 데이터와 비교

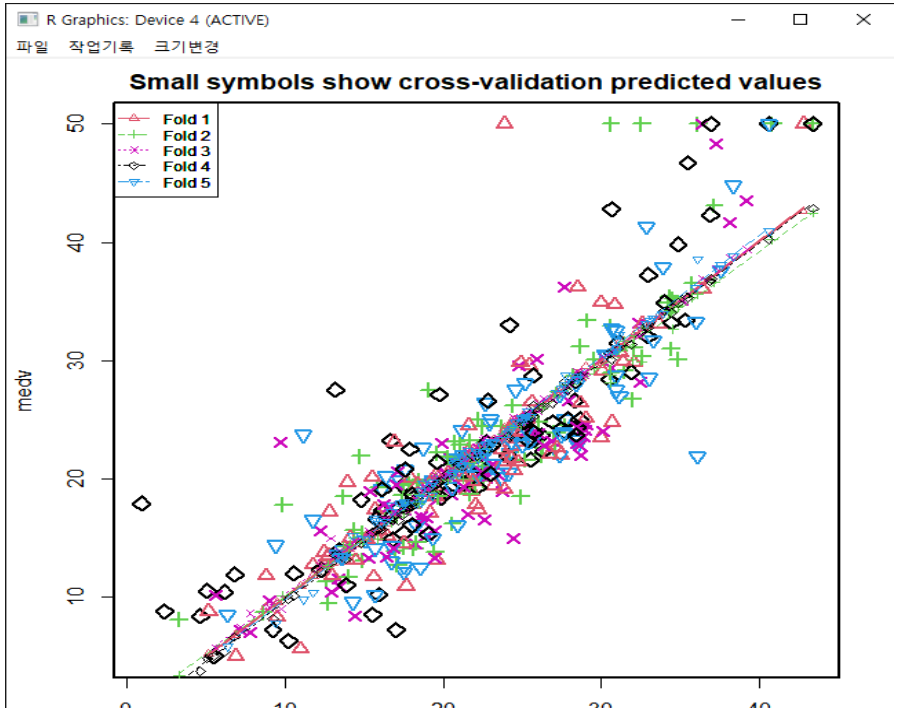
```
> correlation_accuracy <- cor(actuals_preds)
> correlation_accuracy
              actuals predicteds
actuals     1.0000000  0.8471642
predicteds  0.8471642  1.0000000
```

<사진 3> 예측 정확도

예측 모델의 성능 측정

- CV를 이용
  - 회귀 모형에 Cross-validation을 적용하기 위하여 Cvlm()함수를 이용한다
  - K-fold를 이용하는데 k-fold는 데이터 셋을 k개의 같은 크기로(함수에서 m) 나눈 다음 하나의 부분 씩 test set으로 사용하여 k개의 test performance를 평균내는 것을 의미함
  - 여기서 RMSE(평균 제곱오차)를 확인, MSE(평균 오차)도 확인 할 수 있음

```
> windows()
> cvResults <- suppressWarnings(
+   CVlm(data = df_train,
+        form.lm=medv~crim+zn+chas+nox+rm+dis+rad+tax+ptratio+black+lstat,
+        m=5,
+        dots=FALSE,
+        seed=100,
+        legend.pos="topleft",
+        printit=TRUE
+        ));
Analysis of Variance Table

Response: medv
           Df Sum Sq Mean Sq F value  Pr(>F)
crim        1   5060    5060   254.1 < 2e-16 ***
zn          1   2519    2519   126.5 < 2e-16 ***
chas        1    545     545    27.4 2.7e-07 ***
nox         1   1411    1411    70.9 7.3e-16 ***
rm          1   9039    9039   453.9 < 2e-16 ***
dis         1    722     722    36.2 4.0e-09 ***
rad         1    292     292    14.7 0.00015 ***
tax         1    546     546    27.4 2.6e-07 ***
ptratio     1   1029    1029    51.7 3.3e-12 ***
black       1    429     429    21.5 4.7e-06 ***
lstat       1   2073    2073   104.1 < 2e-16 ***
Residuals 392   7806      20
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



<사진 1> 모델에 예측

<사진 2> Cross validation을 이용하여 예측한 값들의 선형 그래프

예측 모델의 성능 측정

- RMSE와 MSE
  - 예측한 모델로 부터 사용한 평균 제곱오차와 제곱오차를 구해본다

```
> attr(cvResults, 'ms')
[1] 21
> #평균 제곱 오차(MSE)
> sqrt(attr(cvResults,'ms'))
[1] 4.58
```

<사진 1> MSE,RMSE

- 예측 값과 성능 측정
  - 마지막으로 이전에 예측 했던 회귀 모델과 동일하게 교차검증을 통해 만든 예측 모델에 테스트 데이터를 예측함
  - 최대 정확도 (min_max_accuracy)와 평균 백분율 오차를 확인해본다
  - 기존 모델보다 성능이 향상한 것을 확인 (min_max_accuracy와 매우 근접)

	Predicted	cvpred
1	21.73	21.82
2	21.20	20.93
3	23.49	23.48
4	14.92	14.88
5	24.12	23.87
6	34.21	33.91
7	24.40	24.35
8	25.49	25.58
9	18.13	18.00
10	21.51	22.41
11	6.15	5.36
12	28.40	28.43
13	21.18	21.48
14	24.19	24.27
15	8.59	8.77
16	26.21	25.98
17	24.22	24.05
18	10.56	10.13
19	20.96	21.30

<사진 3> 모든 관찰을 통한 예측 값과 교차 검증을 통한 예측 값

```
> correlation_accuracy <- cor(actuals_preds)
> correlation_accuracy
           actuals predicteds
actuals      1.000      0.867
predicteds   0.867      1.000
> min_max_accuracy <- mean(apply(actuals_preds, 1, min) / apply(actuals_preds, 1, max))
> min_max_accuracy
[1] 0.867
> mape <- mean(abs((actuals_preds$predicteds - actuals_preds$actuals))/actuals_preds$actuals)
> mape
[1] 0.153
```

<사진 5> 성능 확인

```
> cvResults
      crim   zn indus chas   nox   rm   age   dis rad tax ptratio  black lstat medv
1   0.1588  0.0 10.81    0 0.413 5.96  17.5  5.29   4 305    19.2 376.94  9.88 21.7
2   0.1033 25.0  5.13    0 0.453 5.93  47.2  6.93   8 284    19.7 396.90  9.22 19.6
3   0.3494  0.0  9.90    0 0.544 5.97  76.7  3.10   4 304    18.4 396.24  9.97 20.3
4   2.7340  0.0 19.58    0 0.871 5.60  94.9  1.53   5 403    14.7 351.85 21.45 15.4
5   0.0434 21.0  5.64    0 0.439 6.12  63.0  6.81   4 243    16.8 393.97  9.43 20.5
6   0.0837 45.0  3.44    0 0.437 7.18  38.9  4.57   5 398    15.2 396.90  5.39 34.9
7   0.1907 22.0  5.86    0 0.431 6.72  17.5  7.83   7 330    19.1 393.74  6.56 26.2
```

<사진 2> cvResult로 확인해본 계수 값들

설명 모델의 성능 측정

- 설명 모델
  - 설명 모델은 예측변수가 목표변수에 미치는 영향을 분석하기위 한 것이기 때문에 테스트셋이 따로 필요하지 않다.
- 모델의 성능 측정
  - 모델 F-test의 p-value : 2.2e-16 즉, 측정 불가의 0 수렴 값

  -> 하나 이상의 예측변수가 설명력 증가에 기여한다고 해석가능함.
  - Adjust-R : 0.7338 즉, 실무적으로 유의미한 모델이라고 해석가능함.

  - 변수 T-test의 p-value를 0.001 이하로 가지는 변수(중요도 순)

RM very close to 0 (2.2e-16)
LSTAT very close to 0 (2.2e-16)
DIS 0.000000000000602
PTRATIO 0.00000000000127
NOX 0.00000412
RAD 0.00000519
B 0.000507
ZN 0.000784

====p-value > 0.001====
TAX 0.001118
CRIM 0.001126
CHAS 0.001912
INDUS 0.734597
AGE 0.954686

```
> summary(lm)

Call:
lm(formula = MEDV ~ ., data = bt)

Residuals:
     Min       1Q   Median       3Q      Max
-15.5795  -2.7256  -0.5165   1.7831  26.1887

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.649e+01  5.104e+00   7.149 3.18e-12 ***
CRIM        -1.072e-01  3.271e-02  -3.276 0.001126 **
ZN           4.640e-02  1.373e-02   3.380 0.000784 ***
INDUS        2.086e-02  6.150e-02   0.339 0.734597
CHAS         2.689e+00  8.616e-01   3.120 0.001912 **
NOX         -1.780e+01  3.821e+00  -4.658 4.12e-06 ***
RM           3.805e+00  4.180e-01   9.102  < 2e-16 ***
AGE          7.511e-04  1.321e-02   0.057 0.954686
DIS         -1.476e+00  1.995e-01  -7.398 6.02e-13 ***
RAD          3.057e-01  6.633e-02   4.608 5.19e-06 ***
TAX         -1.233e-02  3.761e-03  -3.278 0.001118 **
PTRATIO     -9.535e-01  1.308e-01  -7.287 1.27e-12 ***
B            9.392e-03  2.684e-03   3.500 0.000507 ***
LSTAT       -5.255e-01  5.069e-02 -10.366  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.746 on 492 degrees of freedom
Multiple R-squared:  0.7406,    Adjusted R-squared:  0.7338
F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

<사진 1> F-test, T-test

부분선택 알고리즘

- 후방향 제거

  후방향 제거 출력 결과 p-value가 0.734597, 0.954686로 매우 높은 값을 가지는 INDUS, AGE 변수가 제거되었음.

- 각 변수의 회귀계수의 의미
  - 회귀식의 기울기를 나타내는 통계량으로 실제 회귀식을 작성할 때 사용하는 값
  - 데이터 단위가 그대로 남아있기 때문에 변수의 영향력을 알 수 없음
  - 따라서 이를 표준화 시켜 사용함

```
Step:  AIC=1585.83
MEDV ~ CRIM + ZN + CHAS + NOX + RM + DIS + RAD + TAX + PTRATIO +
    B + LSTAT

          Df Sum of Sq   RSS    AIC
<none>                  11083 1585.8
- CHAS     1    227.65 11311 1594.1
- CRIM     1    243.80 11327 1594.8
- ZN       1    257.40 11340 1595.5
- TAX      1    272.99 11356 1596.2
- B        1    276.15 11359 1596.3
- RAD      1    499.63 11583 1606.1
- NOX      1    542.98 11626 1608.0
- PTRATIO  1   1207.85 12291 1636.2
- DIS      1   1449.70 12533 1646.0
- RM       1   1958.21 13041 1666.2
- LSTAT    1   2732.93 13816 1695.4
```

<사진 1> 후방향 제거

	Estimate
(Intercept)	3.649e+01
CRIM	-1.072e-01
ZN	4.640e-02
INDUS	2.086e-02
CHAS	2.689e+00
NOX	-1.780e+01
RM	3.805e+00
AGE	7.511e-04
DIS	-1.476e+00
RAD	3.057e-01
TAX	-1.233e-02
PTRATIO	-9.535e-01
B	9.392e-03
LSTAT	-5.255e-01

<사진 2> 각 변수의 회귀 계수

고찰

- 다양한 변수 선택법

    1) 전진 선택법
    2) 후방 제거법
    3) 단계별 선택법
    4) 모든 가능한 회귀접근법

고찰

- 1) 전진 선택법

```
forward <- step(linear_model_description, direction = "forward")
summary(forward)
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  33.740005   5.347397   6.310 7.60e-10 ***
crim         -0.116648   0.032276  -3.614 0.000341 ***
zn            0.039847   0.015512   2.569 0.010579 *
indus         0.012978   0.065082   0.199 0.842040
chas          1.612301   0.906347   1.779 0.076035 .
nox         -15.237963   3.929783  -3.878 0.000124 ***
rm            3.989207   0.443925   8.986  < 2e-16 ***
age          -0.004293   0.013716  -0.313 0.754458
dis          -1.335055   0.207485  -6.434 3.64e-10 ***
rad           0.275248   0.071390   3.856 0.000135 ***
tax          -0.012861   0.004029  -3.192 0.001525 **
ptratio      -0.916149   0.139499  -6.567 1.64e-10 ***
black         0.008367   0.002776   3.014 0.002748 **
lstat        -0.511245   0.054176  -9.437  < 2e-16 ***
---
```

- P-value > 0.05인 변수 Indus, chas, age는 유의하지 않음

고찰

- 2) 후방 제거법

```
backward <- step(linear_model_description, direction = "backward")
summary(backward)
```

```
Step:  AIC=1220.32
medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio +
    black + lstat

          Df Sum of Sq    RSS    AIC
<none>                 7805.6 1220.3
- chas     1     64.42 7870.1 1221.6
- zn       1    137.61 7943.3 1225.4
- black    1    179.26 7984.9 1227.5
- tax      1    240.32 8046.0 1230.6
- crim     1    263.09 8068.7 1231.7
- rad      1    315.66 8121.3 1234.3
- nox      1    359.74 8165.4 1236.5
- ptratio  1    874.29 8679.9 1261.2
- dis      1    917.87 8723.5 1263.2
- rm       1   1663.01 9468.6 1296.3
- lstat    1   2073.26 9878.9 1313.5
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  33.856310   5.306720   6.380 5.00e-10 ***
crim         -0.116978   0.032182  -3.635 0.000315 ***
zn            0.039833   0.015152   2.629 0.008904 **
chas          1.619126   0.900194   1.799 0.072845 .
nox         -15.389961   3.620776  -4.250 2.67e-05 ***
rm            3.954358   0.432703   9.139  < 2e-16 ***
dis          -1.325033   0.195163  -6.789 4.18e-11 ***
rad           0.273031   0.068574   3.982 8.16e-05 ***
tax          -0.012542   0.003610  -3.474 0.000570 ***
ptratio      -0.917009   0.138391  -6.626 1.14e-10 ***
black         0.008274   0.002758   3.000 0.002868 **
lstat        -0.515966   0.050566 -10.204  < 2e-16 ***
```

- indus, age 변수가 제거됨

고찰

- 3) 단계별 선택법

```
both <- step(linear_model_description, direction = "both")
summary(both)
```

```
Step:  AIC=1220.32
medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio +
    black + lstat

          Df Sum of Sq    RSS    AIC
<none>                 7805.6 1220.3
- chas     1     64.42 7870.1 1221.6
+ age      1      2.04 7803.6 1222.2
+ indus    1      0.87 7804.8 1222.3
- zn       1    137.61 7943.3 1225.4
- black    1    179.26 7984.9 1227.5
- tax      1    240.32 8046.0 1230.6
- crim     1    263.09 8068.7 1231.7
- rad      1    315.66 8121.3 1234.3
- nox      1    359.74 8165.4 1236.5
- ptratio  1    874.29 8679.9 1261.2
- dis      1    917.87 8723.5 1263.2
- rm       1   1663.01 9468.6 1296.3
- lstat    1   2073.26 9878.9 1313.5
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  33.856310   5.306720   6.380 5.00e-10 ***
crim         -0.116978   0.032182  -3.635 0.000315 ***
zn            0.039833   0.015152   2.629 0.008904 **
chas          1.619126   0.900194   1.799 0.072845 .
nox         -15.389961   3.620776  -4.250 2.67e-05 ***
rm            3.954358   0.432703   9.139  < 2e-16 ***
dis          -1.325033   0.195163  -6.789 4.18e-11 ***
rad           0.273031   0.068574   3.982 8.16e-05 ***
tax          -0.012542   0.003610  -3.474 0.000570 ***
ptratio      -0.917009   0.138391  -6.626 1.14e-10 ***
black         0.008274   0.002758   3.000 0.002868 **
lstat        -0.515966   0.050566 -10.204  < 2e-16 ***
```

- indus, age 변수가 제거됨

고찰

- 4) 모든 가능한 회귀 접근법

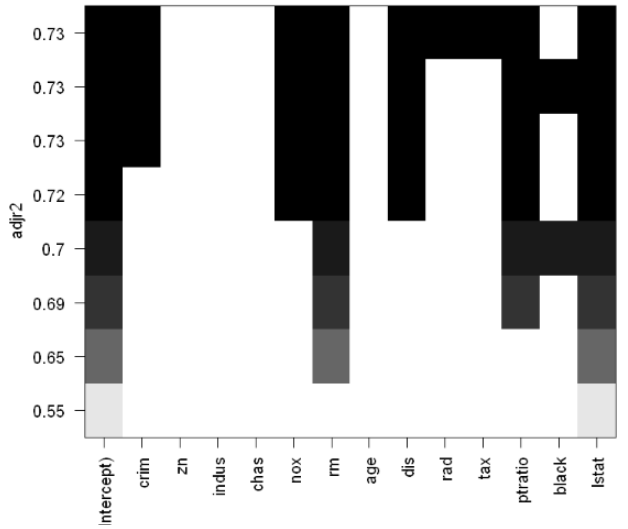
```
all_model <- leaps::regsubsets(medv~., data=boston_data)
summary(all_model)
```

```
          crim zn  indus chas nox rm  age dis rad tax ptratio black lstat
1  ( 1 ) " "  " " " "   " "  " " " " " " " " " " " " " "     " "   "*"
2  ( 1 ) " "  " " " "   " "  " " "*" " " " " " " " " " "     " "   "*"
3  ( 1 ) " "  " " " "   " "  " " "*" " " " " " " " " "*"     " "   "*"
4  ( 1 ) " "  " " " "   " "  " " "*" " " " " " " " " "*"     "*"   "*"
5  ( 1 ) " "  " " " "   " "  "*" "*" " " "*" " " " " "*"     " "   "*"
6  ( 1 ) "*"  " " " "   " "  "*" "*" " " "*" " " " " "*"     " "   "*"
7  ( 1 ) "*"  " " " "   " "  "*" "*" " " "*" " " " " "*"     "*"   "*"
8  ( 1 ) "*"  " " " "   " "  "*" "*" " " "*" "*" "*" "*"     " "   "*"
```

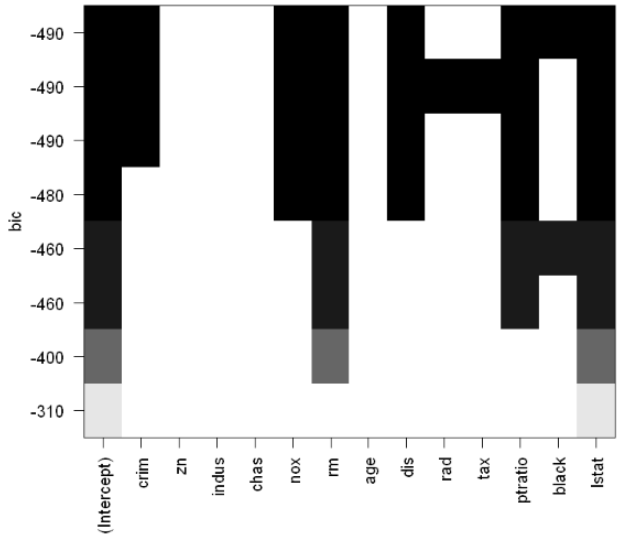
- 수정 결정 계수 : crim, nox, rm, dis, rad, tax, ptratio, black, lstat

- Bayes Information Criterion : crim, nox, rm, dis, ptratio, black, lstat

을 포함한 모델이 최적 모델




```
plot(all_model, scale="adjr2")
```




```
plot(all_model, scale="bic")
```