# 15. 회귀분석

회귀분석은 최근 통계분야에서 가장 중요한 역할을 담당하고 있는 분석 도구로서 하나 또는 여러 개의 설명변수(혹은 독립변수)를 이용하여 하나의 반응변수(혹은 종속변수)를 예측하는 방법론에 대한 지칭이다. 일반적으로 회귀분석은 반응변수와 연관되어있는 설명변수들을 선택하고, 변수들 사이의 관계를 정립하며, 반응변수를 예측하는 작업에 사용된다고 할 수 있다.

본 장에서는 R을 이용하여 회귀분석을 실시하는 다양한 기법들을 살펴볼 것이다. 우선 설명변수가 하나인 단순선형회귀모형 및 설명변수가 2개 이상인 다중회귀모형의 적합 및 결과 해석 방법에서 대해서 알아볼 것이다. 이어서 추정된 모형에 있을 수 있는 다양한 문제를 진단하고 대안을 제시하는 방법을 알아볼 것이며, 또한 어떤 설명변수를 모형에 포함시키는 것이 최상의 결과를 얻을 수 있게 되는지에 대한 것도 살펴볼 것이다.

본 장에서 다루고 있지 않은 회귀분석의 이론적 배경 및 R에 의한 고급 회귀분석 기법에 대해서는 Faraway(2005) 혹은 이와 거의 비슷한 내용의 원고로 "Practical Regression and ANOVA Using R"이 CRAN 사이트에(https://cran.r-project.org/doc/contrib/Faraway-PRA.pdf) PDF 파일로 올라와 있으니 참고하기 바란다.

## 15.1 단순선형회귀모형

반응변수 $Y$와 설명변수 $X$ 사이에 다음과 같은 선형관계가 존재한다고 가정하자.

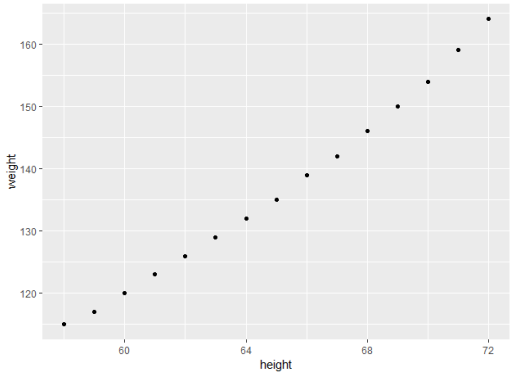
$$\mathrm{Y_i} = \beta_0 + \beta_1 X_i + \varepsilon_i, \qquad i = 1, \cdots, n$$

일차적인 관심은 회귀계수 $\beta_0$와 $\beta_1$의 추정이라고 할 수 있다. 물론 회귀계수의 추정 및 적절한 해석을 위해서 오차항 $\varepsilon_i$이 서로 독립이며, $\varepsilon_i \sim N(0, \sigma^2)$이라는 가정이 필요하다.

예를 들어 데이터 프레임 women에 있는 변수 height를 설명변수로 하고 변수 weight를 반응변수로 하는 단순선형회귀모형을 설정한다고 할 때 회귀계수 $\beta_0$와 $\beta_1$을 추정해보자. 회귀계수를 추정하기 전에 두 변수의 산점도를 작성하여 두 변수의 관계를 확인해보자.

그림 15.1의 산점도를 보면 두 변수 사이에 양의 선형관계가 있는 것으로 보인다. 선형회귀모형의 회귀계수 추정은 함수 lm()으로 할 수 있다.

```
> fit <- lm(weight ~ height, women)
> fit

Call:
lm(formula = weight ~ height, data = women)

Coefficients:
(Intercept)       height
    -87.52         3.45
```




<그림 15.1> 변수 height와 weight의 산점도

추정된 회귀계수는 $\hat{\beta}_0 = -87.52$ , $\hat{\beta}_1 = 3.45$ 임을 알 수 있다. 함수 lm()은 선형회귀분석을 실행하는 함수로서 주요 인자는 y~x의 형태를 취하고 있는 모형공식이다. 물결표(~)의 왼쪽에는 반응변수가 있고 오른쪽에는 설명변수가 있다. 함수 lm()으로 생성된 객체 fit을 단순하게 출력시키면 위에서 볼 수 있듯이 추정된 회귀계수만이 화면에 나타난다. 하지만 사실 객체 fit에는 다음과 같이 많은 양의 정보가 담겨 있는 리스트이다.

```
> names(fit)
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
 [9] "xlevels"       "call"          "terms"         "model"
```

사용자마다 필요한 정보가 서로 다를 수 있기 때문에 모든 것을 한번에 출력시키는 대신 각자 선택하게끔 하는 방식이라고 하겠다. 필요한 정보를 추출하는 방법은 다음 절에서부터 찾아볼 수 있다.

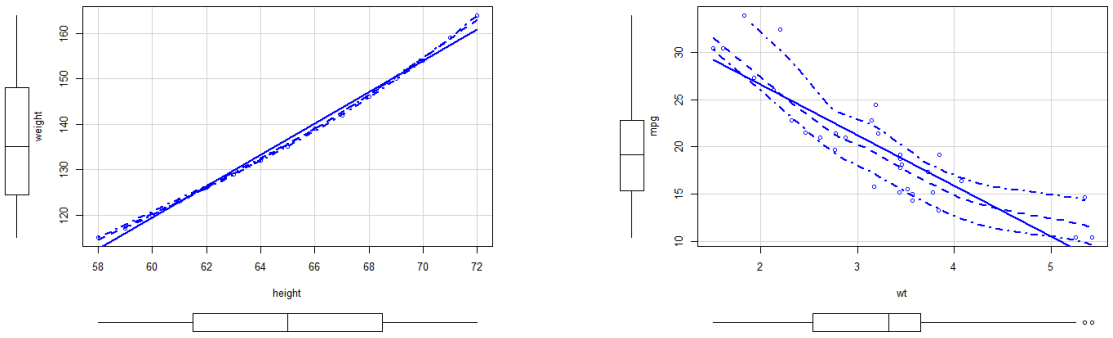
패키지 car에 있는 함수 scatterplot()을 사용하면 두 변수의 산점도 및 단순선형회귀모형의 적합결과를 효과적으로 표현하는 그래프를 작성할 수 있다. 데이터 프레임 women에 있는 변수 weight와 height를 대상으로 함수 scatterplot()을 적용시켜보자.

```
> library(car)

> # 그림 15.2 (a)
> scatterplot(weight ~ height, women)
```

함수 scatterplot()은 두 변수의 산점도, 추정된 회귀직선과 비모수 회귀곡선을 작성한다. 또한 자료의 조건부 변동량을 나타내는 두 개의 비모수 회귀곡선 및 두 변수의 상자그림을 하나의 그래프에 매우 효과적으로 작성한다. 데이터 프레임 women의 경우에 비모수 회귀곡선의 형태를 보면, 두 변수의 관계는 선형보다 2차가 더 적합한 것으로 보인다. 또한 분산의 추정과 관련된 두 개의 비모수 회귀곡선이 거의 겹쳐져 있어, 의미 파악이 조금 어려워 보인다. 분산을

추정한 비모수 회귀곡선의 형태를 보기 위해 데이터 프레임 mtcars에서 변수 mpg를 반응변수로, wt를 설명변수로 하여 함수 scatterplot()을 실행시켜 보자.

```
> # 그림 15.2 (b)
> scatterplot(mpg ~ wt, mtcars)
```




(a) (b)

<그림 15.2> 함수 scatterplot()에 의한 두 변수의 산점도

회귀직선은 lty=1의 실선으로, 비모수 회귀곡선은 lty=2의 dashed line으로 표시되었다. 자료의 변량을 나타내는 두 개의 선은 lty=4의 선으로 나타나 있으며, 두 선의 간격으로 자료의 조건부 변량을 표현하고 있다.

## 15.2 다중선형회귀모형

반응변수 Y와 k개의 설명변수 $X_1, \cdots, X_k$ 사이에 다음과 같은 선형관계가 존재한다고 가정하자.

$$\mathrm{Y_i} = \beta_0 + \beta_1 X_{1i} + \cdots + \beta_k X_{ki} + \varepsilon_i, \qquad i = 1, \cdots, n$$

오차항 $\varepsilon_i$에 대해서는 단순선형회귀모형의 경우와 동일하게 서로 독립이며 $\varepsilon_i \sim N(0, \sigma^2)$이라는 가정이 필요하다. 선형성의 가정 및 오차항의 가정은 회귀모형의 추정과 추론의 정당성을 보장하기 위한 것으로 만일 가정이 만족되지 않는 경우라면 검정결과 및 신뢰구간 추정결과가 부정확하게 나올 수도 있다.

### 15.2.1 회귀모형의 적합

회귀모형의 적합이란 회귀계수의 추정을 의미하는 것으로 일반적인 최소제곱법(ordinary least squares; OLS)에 의한 회귀모형의 적합은 함수 lm()으로 할 수 있으며, 일반적인 사용법은 다음과 같다.

```
lm(formula, data, subset, weights, ...)
```

구성인자 formula는 설정된 회귀모형을 나타내는 모형 공식이고, data는 회귀분석에 사용될 데이터가 들어있는 데이터 프레임을 지정하는 것이다. 또한 weights는 각 관찰값에 가중치를 부여하고자 하는 경우 사용되는 것으로 가중치를 나타내는 숫자형 벡터를 입력하면 되며, subset은 데이터의 일부분만을 이용하여 회귀분석을 실시하고자 하는 경우 사용된다. 예를 들어 처음 100개의 데이터만을 이용하여 변수 y에 대한 변수 x의 회귀모형을 적합하고자 하는 경우에 대한 표현식은 다음과 같다.

```
> lm(y ~ x, subset=1:100)
```

또한 만일 변수 z의 값이 0 이상이 되는 케이스에 대해서만 회귀분석을 실시하고자 한다면 다음과 같이 실행시키면 된다.

```
> lm(y ~ x, subset=z>=0)
```

다양한 형태의 회귀모형을 표현하기 위하여 구성인자 formula에는 몇 가지 기호가 사용되는데, 자세한 사용법이 표 15.1에 설명되어 있다.

<표 15.1> R 공식에서 사용되는 기호

| 기호 | 사용법 |
|---|---|
| 물결표<br>( ~ ) | 반응변수와 설명변수의 구분<br>물결표의 왼쪽에는 반응변수, 오른쪽에는 설명변수를 둔다. |
| 플러스<br>( + ) | 모형에 포함된 설명변수의 구분<br>반응변수 y와 설명변수 x1, x2, x3의 회귀모형은 y ~ x1 + x2 + x3로 표현된다. |
| 콜론<br>( : ) | 설명변수 사이의 상호작용 표현<br>반응변수 y와 설명변수 x1, x2 그리고 x1과 x2의 상호작용이 포함된 모형은 y ~ x1 + x2 + x1: x2로 표현된다. |
| 별표<br>( * ) | 모든 가능한 상호작용 표현<br>y ~ x1 * x2 * x3는 y ~ x1 + x2 + x3 + x1: x2 + x1: x3 + x2: x3 + x1: x2: x3를 의미한다. |
| 윗꺽쇠<br>( ^ ) | 지정된 차수까지의 상호작용 표현<br>y ~ (x1 + x2 + x3)^2는 y ~ x1 + x2 + x3 + x1: x2 + x1: x3 + x2: x3을 의미한다. |
| 마침표<br>( . ) | 반응변수를 제외한 데이터 프레임에 있는 모든 변수<br>만일 데이터 프레임에 y, x1, x2, x3가 있다면 y ~ . 은 y ~ x1 + x2 + x3을 의미한다. |

| | |
|---|---|
| 마이너스<br>( - ) | 회귀모형에서 제외되는 변수<br>y ~ (x1 + x2 + x3)^2 – x2: x3는 y ~ x1 + x2 + x3 + x1: x2 + x1: x3을 의미한다. |
| - 1 또는<br>+ 0 | 원점을 지나는 회귀모형<br>y ~ x – 1 혹은 y ~ x + 0은 절편이 0인 원점을 지나는 회귀모형을 의미한다. |
| I( ) | 괄호 안의 연산자를 수학 연산자로 인식<br>y ~ x1 + I(x2+x3)는 $y = \beta_0 + \beta_1 x_1 + \beta_2 (x_2 + x_3) + \varepsilon$ 의 모형을 의미한다. |
| poly(x, n) | 변수 x의 n차 다항회귀모형 설정<br>y ~ poly(x , 2, raw=TRUE)는 $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \varepsilon$ 의 모형을 의미한다. |

- 예제 1: 다중선형회귀모형 적합

행렬 state.x77에는 미국 50개 주와 관련된 8개 변수의 데이터가 들어있다. 그 중 변수 Murder를 반응변수로 하고 변수 Population, Illiteracy, Income, Frost를 설명변수로 하는 회귀모형을 적합시켜보자.

함수 lm()을 이용하기 위해서는 자료가 데이터 프레임이어야 하는데, state.x77은 행렬이므로 필요한 변수만을 선택하여 데이터 프레임으로 전환해 보자.

```
> library(tidyverse)
> states <- as.data.frame(state.x77) %>%
    select(Murder, Population, Illiteracy, Income, Frost)

> head(states, n=3)
           Murder Population Illiteracy Income Frost
Alabama      15.1       3615        2.1   3624    20
Alaska       11.3        365        1.5   6315   152
Arizona       7.8       2212        1.8   4530    15
```

자료에 적합한 회귀모형을 설정하기 전에 모형에 포함시킬 변수들의 관계를 살펴보는 것이 반드시 필요하다. 여기에서 변수들의 관계는 두 변수씩 짝을 지어 살펴보는 것으로써, 연속형 변수의 경우에는 상관계수와 산점도 행렬을 이용하게 된다.

상관계수는 함수 cor()로 계산할 수 있으며, 사용법은 다음과 같다.

```
cor(x, y = NULL , use = "everything", method = c("pearson", "kendall",
                                                 "spearman") )
```

x와 y 모두 벡터, 행렬 혹은 데이터 프레임이 가능한데, x만 주어지면 x에 포함된 모든 변수들 사이의 상관계수를 계산하게 되고, y도 주어지면 x에 속한 변수와 y에 속한 변수들을 하나씩

짝을 지어 상관계수를 계산하게 된다.

옵션 use는 결측값의 처리방식에 대한 것으로 선택할 수 있는 것은 디폴트인 “everything”과 "all.obs", "complete.obs", "pairwise.complete.obs"이 있으며, 해당 문자의 약칭으로도 사용이 가능하다. 결측값이 존재하면 use=“everything”인 경우에는 NA가 계산결과로 출력되고, use=”all”인 경우에는 오류가 발생한다. 또한 use="complete"의 경우에는 NA가 있는 케이스는 모두 제거된 상태에서 상관계수가 계산되고, use="pairwise"의 경우에는 상관계수가 계산되는 변수들만을 대상으로 NA가 있는 케이스를 제거한다.

옵션 method는 계산하는 상관계수의 종류를 선택한다. 디폴트인 “pearson”은 Pearson 상관계수를 지정하는 것으로 두 변수 사이의 선형관계 정도를 표현하는 가장 일반적으로 많이 사용되는 상관계수를 계산한다. 두 번째 방법인 “kendall”은 Kendall의 순위상관계수 혹은 Kendall의 $\tau$를 지정하는 것으로 concordant pair와 discordant pair를 이용하여 정의되는 비모수 상관계수를 계산한다. 순서형 자료에 주로 적용되는 방법이다. 세 번째 방법인 “spearman”은 Spearman의 순위상관계수 혹은 Spearman의 $\rho$를 지정하는 것으로 두 변수 사이의 관계가 단조증가 혹은 단조감소 함수로 얼마나 잘 설명될 수 있는지를 표현하는 비모수 상관계수를 계산한다. 정규성 가정이 어긋나는 경우 많이 이용되고 있다.

데이터 프레임 states를 구성하고 있는 5개 변수의 상관계수를 구해보자.

```
> cor(states)
              Murder Population Illiteracy     Income       Frost
Murder      1.0000000  0.3436428  0.7029752 -0.2300776 -0.5388834
Population  0.3436428  1.0000000  0.1076224  0.2082276 -0.3321525
Illiteracy  0.7029752  0.1076224  1.0000000 -0.4370752 -0.6719470
Income     -0.2300776  0.2082276 -0.4370752  1.0000000  0.2262822
Frost      -0.5388834 -0.3321525 -0.6719470  0.2262822  1.0000000
```

함수 cor()로 생성되는 상관계수 행렬은 변수의 개수가 많아지면 변수 사이의 관계를 파악하는 것이 어려워진다. 이러한 경우 상관계수 행렬을 그래프로 나타내는 패키지 GGally의 함수 ggcorr()을 이용하는 것이 좋은 대안이 된다. 함수 ggcorr()의 기본적인 사용법은 다음과 같다.

```
ggcorr(data, method=c(“pairwise”, “pearson”), label=FALSE,
                        label_round=1, ...)
```

옵션 method는 두 개의 문자로 이루어졌는데, 첫 번째는 결측값 처리 방법에 대한 것이고 두 번째는 상관계수의 종류에 대한 것이다. 함수 cor()의 옵션 use와 method를 하나의 문자형 벡터로 합친 것이 된다. 차이점은 결측값 처리 디폴트가 함수 cor()에서 “everything”이지만, 함수 ggcorr()에서는 “pairwise”라는 점이다. 옵션 label은 그래프에 상관계수를 표시할지 여부를 지정하는 것이고, label_round는 상관계수의 반올림 자릿수이다.

데이터 프레임 states의 5개 변수의 상관계수를 그래프로 나타내보자. 숫자로만 구성되어 있는

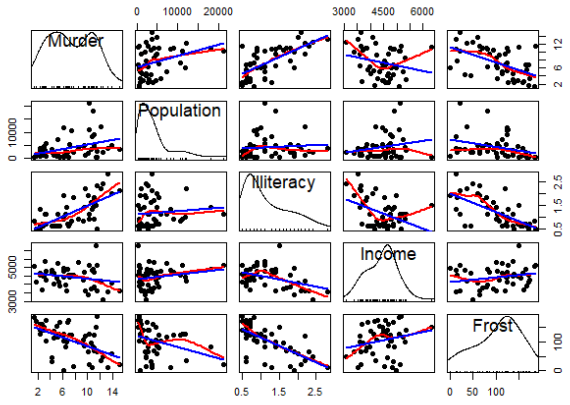
상관계수 행렬보다 훨씬 간편하게 변수 사이의 관계를 파악할 수 있다.

```
> library(GGally)
> # 그림 15.3
> ggcorr(states, label=TRUE, label_round=2)
```


<그림 15.3> 함수 ggcorr()에 의한 상관계수 그래프

상관계수는 두 변수 사이의 선형관계만을 측정할 수 있는 측도이다. 변수 사이에 존재하는 '있는 그대로'의 관계를 확인하는 가장 좋은 방법은 산점도 행렬이 될 것이다. 산점도 행렬은 13장에서 살펴본 함수 pairs() 혹은 패키지 GGally의 함수 ggpairs()로 작성할 수 있다. 또한 패키지 car의 함수 scatterplotMatrix()로도 작성할 수 있다.

```
> library(car)
> # 그림 15.4
> scatterplotMatrix(states, col="black", pch=19,
      regLine=list(lty=1, col="blue"),
      smooth=list(spread=FALSE, lty.smooth=1, col.smooth="red"))
```




<그림 15.4> 함수 scatterplotMatrix()에 의한 산점도 행렬

함수 scatterplotMatrix()에는 산점도 행렬을 작성하고자 하는 변수로 이루어진 행렬 혹은 데이터 프레임을 입력한다. 산점도의 점 색과 모양은 col과 pch로 조절하며, 각 패널에 나타나는

회귀직선의 선 색 및 종류 등은 옵션 regLine에 리스트로 지정한다. 또한 패널에 나타나는 비모수 회귀곡선에 대한 조절은 옵션 smooth에 리스트로 지정하는데, spread=FALSE를 지정하면 분산 추정이 이루어지지 않는다.

이제 반응변수 Murder에 대한 설명변수 Population, Illiteracy, Income, Frost의 다중회귀모형을 적합시켜보자.

```
> fit <- lm(Murder ~ Population + Illiteracy + Income + Frost, states)
> fit

Call:
lm(formula = Murder ~ Population + Illiteracy + Income + Frost,
    data = states)

Coefficients:
(Intercept)   Population   Illiteracy       Income        Frost
  1.235e+00    2.237e-04    4.143e+00    6.442e-05    5.813e-04
```

함수 lm()으로 생성된 객체 fit을 단순히 출력시키면 회귀계수의 추정결과만이 나타나게 되는 것은 이미 단순회귀모형에서 살펴보았다. 객체 fit에 담겨 있는 다양한 정보를 획득하기 위해서는 객체 fit에 몇 가지 함수를 적용시켜야 하는데, 그 함수의 목록이 표 15.2에 정리되어 있다.

<표 15.2> 회귀분석 결과의 획득에 유용하게 사용되는 함수

| 함수 | 산출 결과 |
|---|---|
| anova( ) | 추정된 회귀모형의 분산분석표 혹은 두 개 이상의 추정된 모형을 비교하기 위한 분산분석표 |
| coefficients( ) | 추정된 회귀계수. coef( )도 가능. |
| confint ( ) | 회귀계수의 신뢰구간. 95% 신뢰구간이 디폴트. |
| deviance( ) | 잔차제곱합(residual sum of squares; RSS), $\sum\left(Y_i-\hat{Y}_i\right)^2=\sum e_i^2$ |
| fitted( ) | 반응변수의 적합값, $\hat{Y}_i$ |
| residuals( ) | 회귀모형의 잔차, $e_i$. resid( )도 가능. |
| summary( ) | 회귀모형의 다양한 적합 결과 |

표 15.2에 정리되어 있는 함수를 이용하여 함수 lm()에 의하여 생성된 객체에서 필요한 통계량의 값을 계산하는 방법 및 해석에 대해서는 다음 절에서 살펴보겠다.

- 예제 2: 다항회귀모형의 적합

데이터 프레임 women의 두 변수 weight와 height의 관계는 그림 15.2에서 볼 수 있듯이 선형보다는 2차가 더 적합한 것으로 보인다. 이러한 경우 설명변수 height의 제곱을 모형에 추가하는 2차 다항회귀모형을 사용할 수 있다. 다항회귀모형의 적합은 함수 poly()를 이용하는 것이 효율적인데, 일반적인 사용법을 함수 lm()과 함께 살펴보면 다음과 같다.

```
lm(y ~ poly(x , degree=1 , raw=FALSE))
```

구성인자 degree는 다항회귀모형의 차수를 지정하는 것으로 디폴트 값은 1차이고, raw는 직교다항회귀(orthogonal polynomial regression)의 사용여부를 선택하는 것으로 디폴트 값인 FALSE는 직교다항회귀에 의한 적합이 된다. 따라서 단순 다항회귀모형을 사용하고자 한다면 반드시 raw에 TRUE를 입력시켜야 한다.

이제 반응변수 weight에 설명변수 height의 2차 다항회귀모형을 적합시켜보자.

```
> lm(weight ~ poly(height, degree=2, raw=TRUE), women)

Call:
lm(formula = weight ~ poly(height, degree = 2, raw = TRUE), data = women)

Coefficients:
                          (Intercept)  poly(height, degree=2, raw=TRUE)1
                            261.87818                            -7.34832
poly(height, degree=2, raw=TRUE)2
                              0.08306
```

추정된 회귀모형식은 다음과 같다.

$$\hat{Y}_i = 261.87 - 7.34X_i + 0.08X_i^2$$

다항회귀모형의 적합은 함수 I()를 사용하여 다음과 같이 실행할 수도 있다.

```
> lm(weight ~ height + I(height^2), women)

Call:
lm(formula = weight ~ height + I(height^2), data = women)

Coefficients:
(Intercept)       height  I(height^2)
  261.87818    -7.34832      0.08306
```

- 예제 3: 질적 변수를 설명변수로 사용

질적 변수를 설명변수에 포함시켜야 할 때가 종종 있다. 이런 경우에는 질적 변수를 나타내는 지시변수 혹은 가변수(dummy variable)를 회귀모형에 포함시키면 된다. 함수 lm()은 요인을 설명변수로 포함시키면 요인의 수준 개수에 따라 필요한 가변수를 자동으로 모형에 포함시킨다. 패키지 carData에 있는 데이터 프레임 Leinhardt는 1970년대 105개 나라의 신생아 사망률과 소득, 지역 및 원유 수출 여부를 조사한 자료이다. 패키지 carData는 패키지 car를 설치하거나 로딩하면 따라오는 종속된 패키지이다. 신생아 사망률(infant)을 반응변수로 하고, 소득(income)과 지역(region), 원유 수출 여부(oil)을 설명변수로 하는 회귀모형을 적합하고자 한다. 여기에서 변수 region은 4개의 수준을 갖고 있는 요인이고, 변수 oil은 두 개의 수준을 갖고 있는 요인이다. 우선 변수 income과 region만을 설명변수로 하여 회귀모형을 적합해 보자.

```
> lm(infant ~ income + region, data=Leinhardt)

Call:
lm(formula = infant ~ income + region, data = Leinhardt)

Coefficients:
   (Intercept)         income  regionAmericas     regionAsia
     1.432e+02     -3.458e-03      -8.473e+01     -4.480e+01
  regionEurope
    -1.135e+02
```

요인 region의 수준은 Africa, Americas, Asia, Europe의 네 가지이지만 모형에 포함된 가변수는 세 개이다. 이것은 절편이 모형에 포함된 상태에서 요인의 수준 개수만큼 가변수를 포함시키면 회귀계수 전체를 추정할 수 없는 문제가 발생하기 때문이다. 제거된 가변수는 알파벳 순서에서 첫 번째 범주인 Africa에 대한 것이며, 이 범주가 기준 범주가 된다. 나머지 세 범주를 나타내는 가변수의 모수는 기준 범주와 해당 범주의 차이를 나타낸 것으로써, 예를 들어 변수 regionAmericas는 Africa와 Americas의 차이를, regionAsia는 Africa와 Asia의 차이를 각각 나타낸 것이다.

만일 요인의 수준 개수만큼 가변수를 모형에 포함시키고자 한다면 절편을 제거해야 한다. 절편을 제거하는 방법은 모형 공식에 + 0 또는 - 1을 포함시키면 된다.

```
> lm(infant ~ income + region + 0, data=Leinhardt)

Call:
lm(formula = infant ~ income + region + 0, data = Leinhardt)

Coefficients:
       income    regionAfrica  regionAmericas      regionAsia
    -0.003458      143.235952       58.504549       98.440309
  regionEurope
     29.767837
```

절편이 제거되고 요인의 수준 개수만큼 가변수가 포함되면 각 가변수의 모수는 해당 범주의

효과를 나타내는 것이 된다. 이와 같이 절편을 제거하는 방법은 모형에 설명변수로 포함되는 질적 변수가 하나인 경우에는 가능하지만, 두 개 이상의 질적 변수가 포함되는 경우라면 적용하기 어려운 방법이 된다. 요인 oil을 추가하는 경우에는 절편이 포함된 다음의 모형으로 적합해야 한다.

```
> lm(infant ~ income + region + oil, data=Leinhardt)

Call:
lm(formula = infant ~ income + region + oil, data = Leinhardt)

Coefficients:
   (Intercept)         income  regionAmericas     regionAsia
     136.82468       -0.00529       -83.64943       -45.88540
  regionEurope         oilyes
    -101.48624       78.33508
```

### 15.2.2 회귀모형의 추론

회귀모형의 적합은 함수 lm()으로 이루어진다. 적합된 회귀모형의 추론은 함수 lm()으로 생성된 객체에 표 15.2에 정리되어 있는 함수를 적용시켜 생성된 결과물을 근거로 진행된다. 가장 빈번하게 사용되는 함수는 summary()로서, 프레임 states을 예제로 하여 함수 summary()를 적용시켜보자.

```
> fit1 <- lm(Murder ~ Population + Illiteracy + Income + Frost, states)
> summary(fit1)

Call:
lm(formula = Murder ~ Population + Illiteracy + Income + Frost,
    data = states)

① Residuals:
    Min      1Q  Median      3Q     Max
-4.7960 -1.6495 -0.0811  1.4815  7.6210

② Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.235e+00  3.866e+00   0.319   0.7510
Population  2.237e-04  9.052e-05   2.471   0.0173 *
Illiteracy  4.143e+00  8.744e-01   4.738 2.19e-05 ***
Income      6.442e-05  6.837e-04   0.094   0.9253
Frost       5.813e-04  1.005e-02   0.058   0.9541
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

③ Residual standard error: 2.535 on 45 degrees of freedom

④ Multiple R-squared:  0.567,    Adjusted R-squared:  0.5285

⑤ F-statistic: 14.73 on 4 and 45 DF,  p-value: 9.133e-08
```

함수 lm()으로 생성된 객체에 함수 summary()를 적용시켜 얻은 결과물은 SAS나 SPSS에서 볼

수 있는 결과물과는 형식에서 많은 차이가 있으나 많은 정보를 매우 효과적으로 보여주는 방식이라고 할 수 있다. 결과물을 하나씩 살펴보자.

먼저 ①에는 잔차의 분포를 엿볼 수 있는 기술통계량의 값이 계산되어 있다. 가정이 만족된다면 잔차는 평균이 0인 정규분포를 하게 되는데, 잔차의 기술통계량 값으로 대략적인 판단을 할 수 있게 된다.

②에는 회귀계수의 추정값과 표준오차가 계산되어 있다. 또한 개별 회귀계수의 유의성 검정인 $H_0: \beta_i = 0,\ H_1: \beta_i \neq 0$에 대한 검정통계량의 값과 p-값이 계산되어 있다.

③에 있는 Residual standard error는 오차항 $\varepsilon$ 의 표준편차인 $\sigma$ 의 추정값으로 $\sqrt{MSE}$ , 즉 잔차의 평균제곱합의 제곱근이다.

④에는 회귀모형의 결정계수 $R^2$ 및 수정결정계수의 값이 계산되어 있다.

⑤에는 모든 회귀계수가 0이라는 가설, 즉 $H_0: \beta_1 = \beta_2 = \cdots = \beta_k = 0$에 대한 검정통계량의 값과 자유도, 그리고 p-값이 계산되어 있다.

이와 같이 함수 summary()로 얻어지는 결과물로 회귀모형에 대한 중요한 추론이 가능함을 알 수 있다. 한 가지 SAS나 SPSS에 익숙한 사용자들에게 아쉬울 수 있는 점은 아마도 회귀모형의 분산분석표가 출력되지 않는다는 것일 것인데, 이것은 함수 anova()를 사용함으로써 해결할 수 있다.

```
> anova(fit1)
Analysis of Variance Table

Response: Murder
           Df  Sum Sq Mean Sq F value    Pr(>F)
Population  1  78.854  78.854 12.2713  0.001052 **
Illiteracy  1 299.646 299.646  46.6307 1.83e-08 ***
Income      1   0.057   0.057   0.0089 0.925368
Frost       1   0.021   0.021   0.0033 0.954148
Residuals  45 289.167   6.426
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

함수 anova()는 회귀모형의 분산분석표 계산뿐만 아니라 두 회귀모형을 비교할 때에도 사용되는 함수이다. 예를 들어 확장된 회귀모형 Ω와 축소된 회귀모형 ω를 비교한다고 하자. 이 때 축소모형 ω의 설명변수들은 확장모형 Ω에 있는 설명변수들의 부분집합이어야 한다. 만일 축소모형의 잔차제곱합인 $RSS_\omega$와 확장모형의 잔차제곱합인 $RSS_\Omega$의 차이인 $RSS_\omega - RSS_\Omega$가 매우 적다면 축소모형이 확장모형만큼 좋다는 것을 의미하는 것이므로 모수절약의 원칙에 의하여 축소모형을 선택할 수 있게 된다.

이 문제는 회귀모형 $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \varepsilon$을 확장모형으로 하고, 다음의 가설이

만족될 때의 모형을 축소모형으로 설정하는 것과 동일한 것이 된다.

$$\mathrm{H}_0\colon\ \beta_q = \beta_{q+1} = \cdots = \beta_r = 0, \qquad q < r \le k$$

위 가설은 2개 이상의 회귀계수가 모두 0이라는 가설로서, 검정은 anova(fit2,fit1)으로 실시할 수 있다. 단, fit1은 확장모형에 대한 함수 lm()의 적합 객체이고, fit2는 축소모형에 대한 적합 객체가 된다.

예를 들어 데이터 프레임 states에서 변수 Income과 Frost의 회귀계수가 모두 0이라는 가설의 검정을 실시해 보자.

```
> fit1 <- lm(Murder ~ ., states)

> fit2 <- lm(Murder ~ Population + Illiteracy, states)

> anova(fit2, fit1)
Analysis of Variance Table

Model 1: Murder ~ Population + Illiteracy
Model 2: Murder ~ Population + Illiteracy + Income + Frost
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     47 289.25
2     45 289.17  2   0.078505 0.0061 0.9939
```

두 모형의 RSS가 거의 동일한 값을 갖고 있는 것을 알 수 있다. 이것은 두 변수 Income과 Frost가 모형에 기여하는 바가 거의 없다는 것을 의미하는 것으로, 1에 가까운 p-값이 계산되었다.

회귀계수의 신뢰구간은 함수 confint()로 계산할 수 있다. 95% 신뢰구간이 디폴트로 계산되며, 만일 신뢰수준을 변경하고자 한다면 옵션 level에 원하는 신뢰수준을 입력하면 된다. 객체 fit1에 포함되어 있는 회귀계수의 95% 신뢰구간은 다음과 같이 계산된다.

```
> confint(fit1)
                    2.5 %       97.5 %
(Intercept) -6.552191e+00 9.0213182149
Population   4.136397e-05 0.0004059867
Illiteracy   2.381799e+00 5.9038743192
Income      -1.312611e-03 0.0014414600
Frost       -1.966781e-02 0.0208304170
```

회귀계수 $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \cdots, \hat{\beta}_k)^T$이 추정되면 새롭게 주어진 설명변수의 자료에 대한 반응변수의 값을 예측할 수 있다. 즉, 새롭게 주어진 설명변수의 값을 $\boldsymbol{x}_o = (1, X_{10}, \cdots, X_{ko})^T$라고 한다면 반응변수의 값은 $\hat{Y}_o = \boldsymbol{x}_o^T\hat{\boldsymbol{\beta}}$ 으로 예측된다. 여기에서 반응변수의 예측은 두 가지로 구분되는데, 하나는 반응변수의 평균값에 대한 예측이고 다른 하나는 반응변수의 개별 관찰값에 대한 예측이다. 예를 들어 주택의 매매 가격( $Y$ )와 주택의 특성과 관련된 k개 설명변수 $X_1, \cdots, X_k$ 로 이루어진 회귀모형을 추정하였다고 하자. 새롭게 설명변수의 값이 $\boldsymbol{x}_o$로 주어졌을 경우, 주택의 특성이 $\boldsymbol{x}_o$에

해당되는 모든 주택의 평균 매매 가격을 예측하는 것이 반응변수의 평균값에 대한 예측이 되고, 주택의 특성이 $\boldsymbol{x}_o$인 어느 특정 주택의 매매 가격을 예측하는 것이 반응변수의 개별 관찰값에 대한 예측이 된다. 두 가지 예측에서 예측 결과는 $\hat{Y}_o = \boldsymbol{x}_o^T\hat{\boldsymbol{\beta}}$으로 동일하지만 예측 오차는 개별 관찰값에 대한 예측의 경우가 더 큰 값을 갖게 된다.

반응변수의 예측은 함수 predict()로 할 수 있으며, 일반적인 사용법은 다음과 같다.

```
predict(object, newdata, interval=c("none", "confidence", "prediction"),
                                                    level=0.95)
```

구성인자 object는 함수 lm()으로 생성된 객체를 의미하며, newdata는 새롭게 주어지는 설명변수의 값으로서 반드시 데이터 프레임으로 입력되어야 한다. 옵션 interval은 예측에 대한 신뢰구간을 계산하는 것으로 반응변수의 평균값에 대한 예측인 경우에는 "confidence", 반응변수의 개별 관찰값에 대한 예측인 경우에는 "prediction"을 선택해야 하며, 디폴트는 "none"이 된다.

예를 들어, 앞에서 적합된 모형 fit2에 대하여 두 변수 (Population, Illiteracy)에 대한 새로운 관찰값이 각각 (15000, 0.8), (10000, 1.5), (5000, 2.5)로 주어졌을 때 반응변수 Murder의 평균값 및 개별 관찰값에 대한 예측은 다음과 같이 계산할 수 있다.

```
> x0 <- data.frame(Population=c(15000,10000,5000),
                   Illiteracy=c(0.8,1.5,2.5))

> predict(fit2, newdata=x0, interval="confidence")
       fit       lwr      upr
1  8.278931  6.321113 10.23675
2 10.014516  8.820475  11.20856
3 12.974322 11.265395 14.68325

> predict(fit2, newdata=x0, interval="prediction")
       fit       lwr      upr
1  8.278931  2.918002 13.63986
2 10.014516 4.883020 15.14601
3 12.974322 7.699197 18.24945
```

반응변수의 예측은 회귀분석의 큰 목적 중의 하나라고 할 수 있다. R에서는 예측구간을 포함한 반응변수의 예측을 무척 간단하게 할 수 있다. 물론 이러한 예측이 의미를 갖기 위해서는 설정된 회귀모형의 정당성이 확보되어야 할 것이다. 모형의 정당성은 가정의 만족여부를 확인함으로써 부여할 수 있다. 다음 절에서는 회귀진단을 통하여 가정의 만족여부 및 특이한 관찰값을 발견하는 절차를 살펴보겠다.

## 15.3 회귀진단 및 대안탐색

지금까지 우리는 함수 lm()으로 회귀모형을 추정하고 함수 summary() 등을 이용하여 다양한

추론을 할 수 있음을 살펴보았다. 그러나 이러한 회귀모형의 적합 및 추론은 회귀모형 가정의 만족 정도에 따라 매우 신빙성이 있는 결과가 되거나 혹은 전혀 믿을 수 없는 결과가 되기도 한다. 따라서 회귀모형을 추정하면 반드시 추정된 모형이 가정사항을 모두 만족시키고 있는지를 확인하는 과정이 필요한데, 이것을 회귀진단이라고 한다.

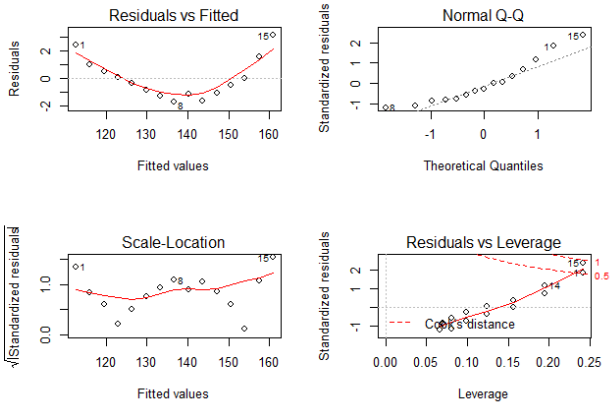
회귀진단은 일반적으로 회귀모형에 대한 진단과 관찰값에 대한 진단으로 구분해서 진행된다. 본 절에서도 두 가지의 진단이 R에서 어떻게 실행될 수 있는지 구분해서 살펴보도록 하겠다.

### 15.3.1 회귀모형의 가정 만족여부 확인

추정된 회귀모형의 가정 만족여부를 확인하는 가장 기본적인 절차는 함수 `lm()`으로 생성된 객체를 함수 `plot()`에 적용시키는 것이다. 여섯 개의 그래프가 작성되지만 디폴트로 네 개의 그래프가 차례로 그려진다. 하나의 Plots 창에 네 개의 그래프를 모두 그리기 위해서는 함수 `par()`를 사용하여 미리 그래프 영역을 분할시켜야 한다.

예제로 데이터 프레임 `women`에 있는 변수 `weight`와 `height`의 단순회귀모형을 적합시키고 가정의 만족여부를 함수 `plot()`으로 확인해보자.

```
> fit_w <- lm(weight ~ height, women)

> # 그림 15.5
> par(mfrow=c(2,2))
> plot(fit_w)
> par(mfrow=c(1,1))
```




<그림 15.5> 함수 `plot()`에 의한 회귀진단

그림 15.5에는 네 개의 그래프가 한꺼번에 작성되었는데 함수 `plot()`에 옵션 `which`를 추가하면 각 그래프를 개별적으로 작성할 수 있다.

```
> # 왼쪽 위 그래프
```

```
> plot(fit_w, which=1)

> # 오른쪽 위 그래프
> plot(fit_w, which=2)

> # 왼쪽 아래 그래프
> plot(fit_w, which=3)

> # 오른쪽 아래 그래프
> plot(fit_w, which=5)
```

그림 15.5의 왼쪽 위 패널에 있는 'Residuals vs Fitted'라는 제목의 그래프는 일반적으로 가장 많이 사용되는 잔차 산점도 그래프로서, 잔차 $\mathrm{e_i} = Y_i - \hat{Y}_i$와 $\hat{Y}_i$의 산점도이다. 오른쪽 위 패널에 있는 'Normal Q-Q'라는 제목의 그래프는 표준화 잔차의 정규 분위수-분위수 그래프이다. 왼쪽 아래 패널에 있는 'Scale-Location'이라는 제목의 그래프는 표준화 잔차의 절대값에 제곱근을 적용시켜 얻은 값과 $\hat{Y}_i$의 산점도로서 동일 분산 가정의 만족여부를 확인하는 그래프이다. 마지막으로 오른쪽 아래 패널에 있는 'Residuals vs Leverage'라는 제목의 그래프는 관찰값의 진단에 사용되는 그래프로, 자세한 설명은 15.3.2절에서 하겠다.

회귀모형의 가정은 대부분 오차항과 관련된 것으로 잔차를 이용하여 만족여부를 확인하게 된다. 그림 15.5의 네 개 그래프도 모두 잔차가 사용된 그래프인데, 이 중 세 개의 그래프에는 잔차가 아닌 표준화(standardized) 잔차가 사용되었다. 잔차의 분산이 오차항의 분산과는 다르게 $\mathrm{Var}(\mathrm{e_i}) = \sigma^2(1-h_i)$가 됨을 이용하여 표준화 잔차 $\mathrm{r_i}$는 다음과 같이 정의된다.

$$\mathrm{r_i} = \frac{e_i}{\sqrt{MSE(1-h_i)}}$$

단, $\mathrm{h_i}$는 모자행렬(hat matrix)의 대각 원소로서 레버리지(leverage)라고 불리는데, 이것은 $i$ 번째 관찰값이 $X$ 변수의 공간에서 자료의 중심으로부터 떨어져 있는 거리를 표현하는 통계량이다. 위에 정의된 표준화 잔차는 내적 스튜던트화(internally studentized) 잔차라고도 불린다.

표준화 잔차 외에 회귀진단에서 많이 사용되는 잔차로 스튜던트화(studentized) 잔차가 있다. 통계량 $\hat{Y}_{(i)}$를 $i$ 번째 관찰값을 제외하고 나머지 $n-1$개의 데이터로 수립된 회귀모형을 이용하여 제외된 $\mathrm{Y_i}$를 예측한 통계량이라고 할 때 스튜던트화 잔차 $\mathrm{t_i}$는 다음과 같이 정의된다.

$$\mathrm{t_i} = \frac{Y_i - \hat{Y}_{(i)}}{\sqrt{\widehat{Var}\left(Y_i - \hat{Y}_{(i)}\right)}}$$

스튜던트화 잔차의 값이 크다는 것은 $i$ 번째 관찰값이 이상값으로 분류될 가능성이 높다는 것을 의미한다. R에서 표준화 잔차 $\mathrm{r_i}$와 스튜던트화 잔차 $\mathrm{t_i}$는 함수 `lm()`으로 생성된 객체를 각각 함수 `rstandard()`와 `rstudent()`에 적용하면 계산할 수 있다.

회귀모형의 가정은 다음과 같이 네 가지로 구분된다.

1) 오차항 $\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n$의 분산이 모두 동일하다.

2) 오차항 $\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n$의 분포는 정규분포이다.

3) 오차항 $\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n$은 서로 독립이다.

4) 반응변수와 설명변수의 관계는 선형이다.

네 가지 가정의 만족여부를 확인하는 방법을 차례로 살펴보자. 예제 데이터로 데이터 프레임 women과 행렬 state.x77에서 5개 변수만을 선택하여 생성한 데이터 프레임 states를 대상으로 각각 객체 fit_w과 fit_s를 다음과 같이 생성하여 두 회귀모형의 가정 만족여부를 확인해 보자.

```
> fit_w <- lm(weight ~ height, women)
> fit_s <- lm(Murder ~ ., states)
```

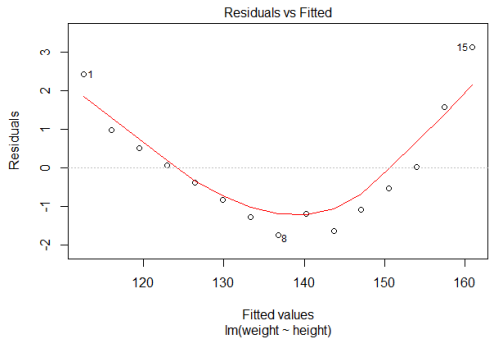
**1) 오차항의 동일 분산 가정**

동일 분산 가정의 만족 여부를 확인하는 기본적인 방법은 함수 plot()으로 생성되는 그래프 중 옵션 which=1과 which=3로 생성되는 두 그래프를 살펴보는 것이다. 또한 패키지 car에 있는 함수 spreadLevelPlot()에 의해 생성되는 그래프를 이용하는 방법과 패키지 car에 있는 함수 ncvTest()로 실행되는 score 검정을 이용하는 방법도 있다.

함수 plot()으로 생성되는 두 그래프는 이미 그림 15.5에서 살펴본 것으로, 옵션 which=1으로 생성되는 잔차 $e_i$와 반응변수의 추정값 $\hat{Y}_i$의 산점도에서는 동일 분산과 선형관계를 확인할 수 있으며, 옵션 which=3로 생성되는 $\sqrt{|r_i|}$와 $\hat{Y}_i$의 산점도에서는 동일 분산의 가정을 확인할 수 있다. 또한 패키지 car의 함수 spreadLevelPlot()은 $|t_i|$와 $\hat{Y}_i$의 산점도를 작성하면서 분산 안정화를 위한 반응변수의 변환 지수, 즉 $Y^p$의 p 값을 제안한다. 패키지 car에 있는 함수 ncvTest()는 동일 분산에 대하여 Breusch and Pagan(1979)와 Cook and Weisberg(1983)이 제안한 score 검정을 실시한다.

데이터 프레임 women의 회귀모형 fit_w에 대한 동일 분산 가정을 확인해 보자. 먼저 함수 plot()에서 옵션 which=1으로 작성된 그래프는 가장 일반적으로 볼 수 있는 잔차 산점도이며, 추가된 곡선은 함수 panel.smooth()로 작성된 로버스트 국소다항회귀 곡선이다. 추가된 곡선이 2차 형태를 보이고 있어 선형관계에는 문제가 있는 것으로 보이지만 동일 분산에 대해서는 확인할 수 있는 것이 없어 보인다. 그래프에 표시된 숫자는 잔차가 가장 극단적인 값을 세 관찰값의 케이스 번호이다.

```
> # 그림 15.6
> plot(fit_w, which=1)
```

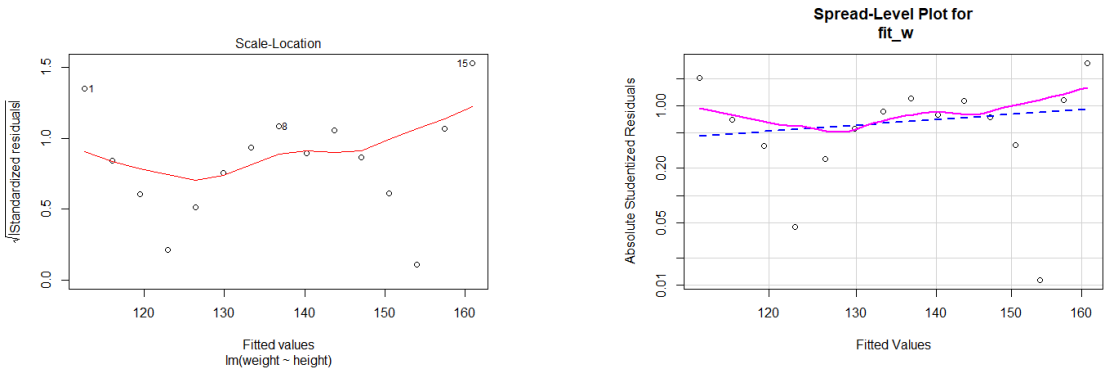



<그림 15.6> 함수 plot()에 의한 잔차 산점도

함수 plot()에서 옵션 which=3으로 작성되는 그래프는 'Scale-Location' 그래프라고 불리는 것으로써 표준화 잔차가 사용된다. 이 그래프에서는 점들이 체계적으로 증가하거나 감소하는 경향이 있는지를 확인해야 한다. 추가된 국소다항회귀 곡선을 참조하여 판단하는 것이 좋다.

패키지 car의 함수 spreadLevelPlot()으로 작성된 그래프에서는 스튜던트화 잔차가 사용되며, 점들이 체계적으로 증가하거나 감소하는 경향이 있으면 동일 분산을 의심해야 한다. 추가된 파란 직선은 로버스트 회귀직선이고 빨간 곡선은 국소다항회귀 곡선이다. 또한 이 함수에서 제안한 변환 지수가 1과 큰 차이가 있는지도 확인해야 한다.

```
> # 그림 15.7 (a)
> plot(fit_w, which=3)

> library(car)
> # 그림 15.7 (b)
> spreadLevelPlot(fit_w)

Suggested power transformation:  -0.8985826
```




(a) (b)

<그림 15.7> 동일 분산 확인을 위한 Scale-Location 그래프

패키지 car의 함수 ncvTest()는 동일분산에 대한 score 검정 혹은 Breusch-Pagan 검정을 실시한다. 귀무가설은 오차의 분산이 일정하다가 된다.

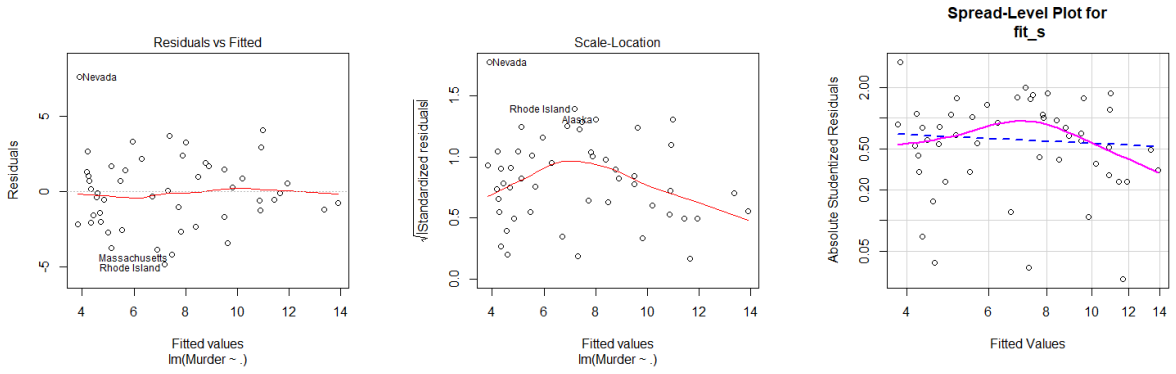
```
> ncvTest(fit_w)
```

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 0.8052115    Df = 1     p = 0.3695398
```

회귀모형 fit_w에 대해서는 그래프에서 동일 분산의 가정이 어긋난 증거를 찾기 어려워 보이며, 제안된 변환 지수가 -0.89로 1과 큰 차이가 없고, 또한 score 검정의 p-값이 0.369로 분산이 같다는 귀무가설을 기각할 수 없다.

이제 데이터 프레임 states에 대한 회귀모형 fit_s의 동일 분산 가정을 확인해보자.

```
> # 그림 15.8 (a)
> plot(fit_s, which=1)

> # 그림 15.8 (b)
> plot(fit_s, which=3)

> # 그림 15.8 (c)
> spreadLevelPlot(fit_s)

Suggested power transformation:  1.209626

> ncvTest(fit_s)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 1.746514    Df = 1     p = 0.1863156
```




(a) (b) (c)

<그림 15.8> 회귀모형 fit_s의 동일 분산 가정 확인을 위한 그래프

회귀모형 fit_s도 동일분산의 가정에는 문제가 없는 것으로 보인다.

### 2) 오차항의 정규분포 가정

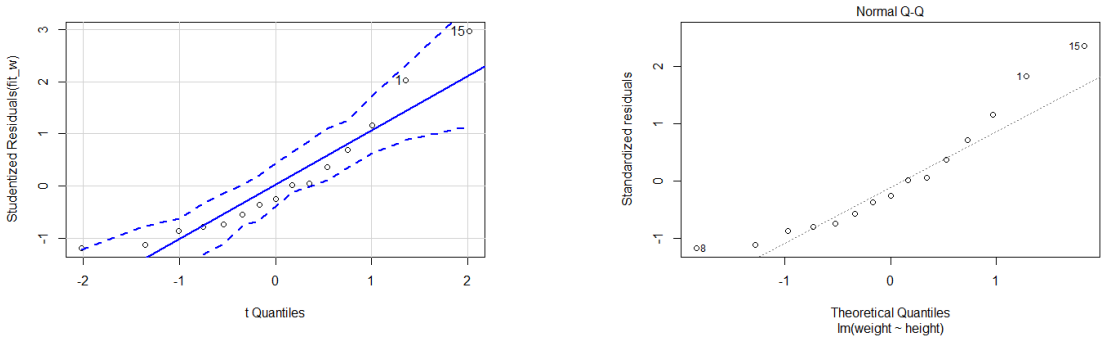
회귀분석에서 이루어지는 검정 및 신뢰구간은 오차항이 정규분포를 한다는 가정에 근거를 두고 이루어진다. 그러나 오차항의 분포가 정규분포의 형태에서 약간 벗어나는 것은 큰 문제를 유발하지 않으며 표본 크기가 클수록 비정규성은 덜 문제가 된다고 할 수 있다. 그러나 오차항의

분포가 Cauchy 분포와 같이 꼬리가 긴 형태의 분포임이 확인된다면 최소제곱법에 의한 회귀계수의 추정보다는 로버스트 선형회귀를 이용하는 것이 더 효과적일 것이다.

정규성의 확인은 주로 그래프를 이용하게 되는데, 사용하게 되는 그래프로는 함수 plot()에 옵션 which=2로 생성되는 표준화 잔차 $r_i$에 대한 정규 분위수-분위수 그래프와 패키지 car에 있는 함수 qqPlot()으로 생성되는 스튜던트화 잔차 $t_i$에 대한 t-분포의 분위수-분위수 그래프가 있다. 스튜던트화 잔차 $t_i$에 대하여 t-분포의 분위수-분위수 그래프를 작성하는 이유는 회귀모형의 모든 가정사항이 만족되는 경우 스튜던트화 잔차 $t_i$는 자유도가 $n-k-2$인 t-분포를 하기 때문이다. 단, $k$는 회귀모형에 포함된 설명변수의 개수다. 정규성 확인을 위한 검정으로는 Shapiro-Wilk 검정이 있으며, R에서는 함수 shapiro.test( )로 할 수 있다.

데이터 프레임 women에 대한 회귀모형 fit_w의 정규성 가정을 확인해보자.

```
> library(car)

> # 그림 15.9 (a)
> qqPlot(fit_w)
[1]  1 15

> # 그림 15.9 (b)
> plot(fit_w, which=2)

> shapiro.test(residuals(fit_w))

        Shapiro-Wilk normality test

data:  residuals(fit_w)
W = 0.91909, p-value = 0.1866
```




(a) (b)

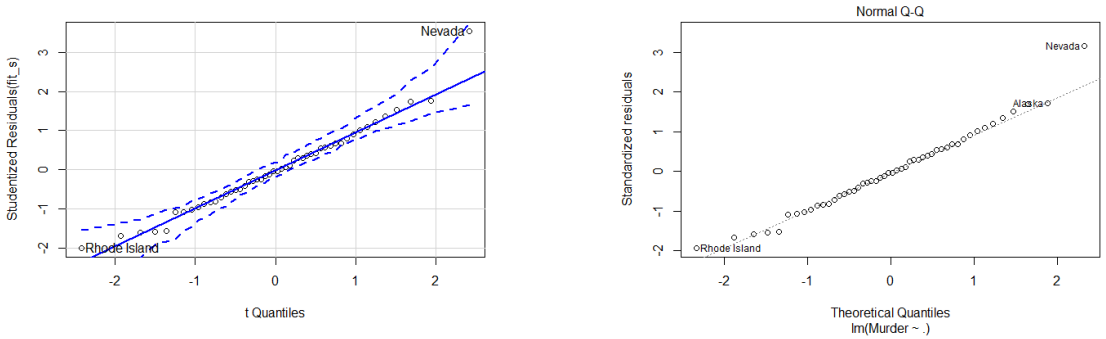
<그림 15.9> 회귀모형 fit_w의 정규성 확인을 위한 그래프

함수 qqPlot()으로 생성되는 그래프에는 기준선 위아래로 점선이 그려져 있는데, 두 점선은 모수적 부스트랩(bootstrap)으로 계산된 95% 신뢰영역을 나타내고 있다. 점들이 기준선 근처에 있으면서 신뢰영역 안에 있으면 정규성에는 큰 문제가 없는 것으로 판단할 수 있다. 회귀모형 fit_w의 경우, 기준선에서 벗어난 점들이 있기는 하지만 모두 신뢰영역 안에 위치하고 있다.

Shapiro-Wilk 검정의 p-값도 0.186으로 정규분포라는 귀무가설을 기각할 수 없어 정규성에는 큰 문제가 없는 것으로 볼 수 있다.

데이터 프레임 states의 회귀모형 fit_s의 정규성 가정을 확인해보자

```
> # 그림 15.10 (a)
> qqPlot(fit_s)
      Nevada Rhode Island
          28           39

> # 그림 15.10 (b)
> plot(fit_s, which=2)

> shapiro.test(residuals(fit_s))

        Shapiro-Wilk normality test

data:  residuals(fit_s)
W = 0.98264, p-value = 0.6672
```




(a) (b)

<그림 15.10> 회귀모형 fit_s의 정규성 확인을 위한 그래프

함수 qqPlot()은 스튜던트화 잔차의 절대값이 가장 큰 두 케이스의 라벨을 그래프에 표시한다. 오른쪽 위에 있는 하나의 점을 제외하고는 모든 점들이 기준선 근처에 있음을 알 수 있다. 문제의 점은 95% 신뢰영역의 경계선에 있음을 알 수 있는데, 그 점이 Nevada 주의 데이터임을 알 수 있다. Nevada 주의 관찰값은 앞으로 자세히 살펴봐야 할 데이터이다. 또한 Shapiro-Wilk 검정의 p-값이 0.66으로 귀무가설을 기각할 수 없다. 회귀모형 fit_s의 경우에도 정규성에는 문제가 없는 것으로 나타났다.

### 3) 오차항의 독립성 가정

수집된 자료가 시간적 혹은 공간적으로 서로 연관되어 있는 경우에는 오차항의 독립성 가정이 만족되지 않을 수 있다. 시간에 흐름에 따라 관측된 시계열 자료나 공간에 따라 관측된 공간

자료를 대상으로 회귀분석을 하는 경우에는 반드시 확인해야 할 가정이 된다.

독립성 가정은 여러 행태로 위반될 수 있는데, 우선 $\varepsilon_i$가 $\varepsilon_{i-1}$과 연관되어 있는지 여부, 즉 1차 자기상관 관계만을 확인하려면 Durbin-Watson 검정을 실시하면 된다. Durbin-Watson 검정은 패키지 car에 있는 함수 durbinWatsonTest()로 할 수 있다. 오차항이 1차 자기상관이 없다고 해서 바로 독립이라 할 수는 없는 것이고, 조금 더 일반적인 형태의 위반 여부를 확인해야 할 것이다. 조금 더 일반적인 형태란 오차항이 p차 자기회귀모형, 즉 AR(p) 모형을 따르는지 여부를 확인하는 것인데, 이것은 Breusch-Godfrey 검정으로 확인할 수 있으며, 패키지 forecast의 함수 checkresiduals()로 할 수 있다.

데이터 프레임 women과 states는 모두 시간적 혹은 공간적으로 연관을 갖기 어려운 방식으로 수집되어 있기 때문에 오차항의 독립성 가정에는 큰 문제가 없는 경우라고 할 수 있다. 대신 살펴볼 자료는 패키지 carData에 있는 데이터 프레임 Hartnagel이다. 패키지 carData는 패키지 car를 설치하거나 로딩하면 따라오는 종속된 패키지이다. 데이터 프레임 Hartnagel은 1931년부터 1968년까지의 캐나다 범죄율에 대한 자료이다. 여성 범죄율(fconvict)을 반응변수로 하고 출산율(tfr)과 여성 고용률(partic)을 설명변수로 하는 회귀모형 fit_h의 독립성 가정을 확인해 보자.

```
> library(car)
> fit_h <- lm(fconvict ~ tfr + partic, Hartnagel)
```

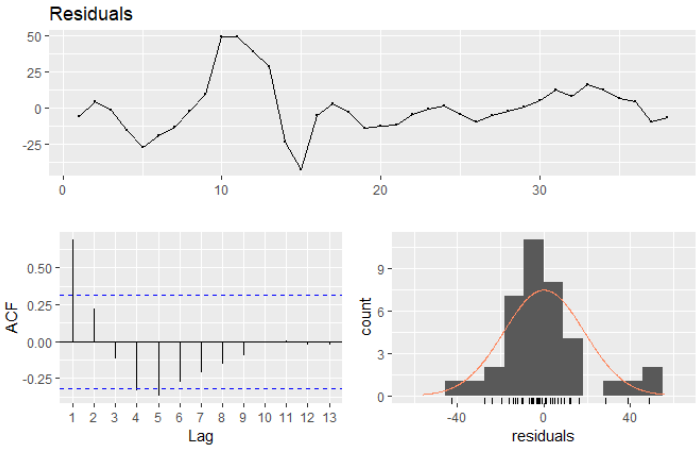
오차항의 1차 자기상관 여부를 확인하는 Durbin-Watson 검정을 실시해 보자.

```
> durbinWatsonTest(fit_h)
 lag Autocorrelation D-W Statistic p-value
   1       0.6894045     0.6148353       0
 Alternative hypothesis: rho != 0
```

1차 자기상관계수는 0.689로 추정되었고, 검정 결과 p-값은 0으로 계산되어 1차 자기상관계수가 0이라는 귀무가설을 기각할 수 있다.

오차항이 AR(p) 모형을 따르지는 여부를 확인해 보자.

```
> library(forecast)

> # 그림 15.11
> checkresiduals(fit_h)

        Breusch-Godfrey test for serial correlation of order up to 7

data:  Residuals
LM test = 24.701, df = 7, p-value = 0.0008572
```




<그림 15.11> 오차의 독립성 가정 확인

검정의 귀무가설은 1 시차부터 7 시차까지의 자기상관계수가 모두 0이라는 것이며, 매우 작은 p-값으로 귀무가설을 기각할 수 있다.

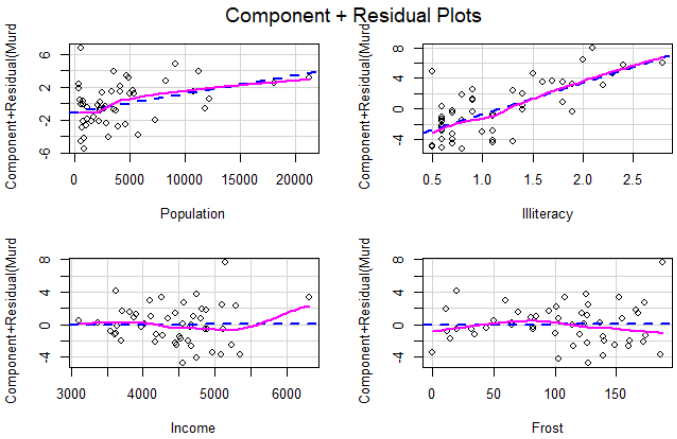
함수 `checkresiduals()`는 오차의 독립성 만족 여부에 대한 검정뿐만이 아니라 잔차에 대한 몇 가지 그래프를 작성한다. 위 패널에 작성된 것은 잔차의 시계열 그래프이고, 왼쪽 아래 패널에는 각 시차(lag)별 잔차의 표본 자기상관도표가 작성되어 있으며 오른쪽 아래 패널에는 잔차의 히스토그램이 작성되어 있다.

#### 4) 선형관계

단순회귀모형의 경우 반응변수와 설명변수의 선형관계는 두 변수의 산점도로 충분히 확인할 수 있다. 그러나 다중회귀모형의 경우에는 $X_i$와 Y의 산점도 혹은 $X_i$와 잔차 e의 산점도가 큰 의미를 갖지 못하게 되는데, 이것은 회귀모형에 포함된 다른 변수의 영향력을 확인할 수 없기 때문이다. 이러한 경우 부분잔차(partial residual)가 매우 유용하게 사용될 수 있는데, 변수 $X_i$의 부분잔차란 반응변수 Y에서 모형에 포함된 다른 설명변수의 영향력이 제거된 잔차를 의미하는 것으로 $Y - \sum_{j \neq i} \hat{\beta}_j X_j$로 정의된다. 또한 $Y = \hat{Y} + e$가 되기 때문에 부분잔차는 $\hat{Y} + e - \sum_{j \neq i} \hat{\beta}_j X_j = e + \hat{\beta}_i X_i$로 정리된다. 따라서 다중회귀모형에서 $X_i$와 Y가 선형관계에 있는지를 확인하기 위해서는 변수 $X_i$의 부분잔차인 $e + \hat{\beta}_i$와 $X_i$의 산점도를 작성하면 된다.

데이터 프레임 `states`의 회귀모형 `fit_s`에 대하여 모형에 포함된 네 개의 설명변수가 반응변수와 선형관계에 있는지를 부분잔차 산점도를 작성하여 확인해 보자. 부분잔차 산점도는 패키지 `car`에 있는 함수 `crPlots()`로 작성할 수 있는데, 이 그래프는 Component + Residual Plots라고도 불린다.

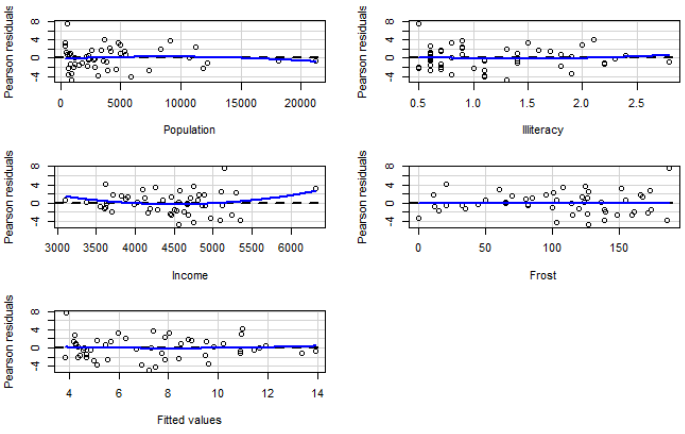
```
> library(car)

> # 그림 15.12
> crPlots(fit_s)
```




<그림 15.12> 선형 가정 확인을 위한 부분잔차 산점도

그림 15.12에는 회귀모형 fit_s에 포함된 네 개의 설명변수 각각에 대한 부분잔차의 산점도가 작성되어 있다. 각 그래프에는 각 설명변수와 부분잔차에 대한 회귀직선과 국소다항회귀곡선이 추가되어 있는데, 부분잔차의 산점도에서 비선형 관계가 탐지된다면 적절한 변환이 이루어져야 할 것이다. 회귀모형 fit_s에 대해서는 선형관계에 큰 문제가 없는 것으로 확인된다. 또한 변수 Income과 Frost의 영향력이 매우 미미하다는 것도 확인할 수 있다.

선형관계를 확인하는 다른 방법으로써 패키지 car의 함수 residualPlots()를 이용할 수 있다. 이 함수는 회귀모형에 포함된 설명변수 $X_i$와 잔차의 산점도 및 $\hat{Y}$과 잔차의 산점도를 작성하고 국소다항회귀 곡선을 추가한다. 또한 각 설명변수의 제곱 $X_i^2$을 하나씩 모형에 포함시켰을 때의 유의성을 검정하고, Tukey의 nonadditivity 검정을 실시한다.




<그림 15.13> 함수 residualPlots()에 의한 선형관계 확인

```
> # 그림 15.13
> residualPlots(fit_s)
           Test stat  Pr(>|Test stat|)
Population   -0.5218           0.60446
Illiteracy    0.4625           0.64601
```

```
Income        1.8355           0.07319 .
Frost         0.0507           0.95981
Tukey test    0.3190           0.74971
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

#### 5) 다중공선성

다중공선성은 회귀모형의 가정과 직접적인 연관이 있는 것이 아니지만 회귀모형의 추정결과를 해석하는 과정에 큰 영향을 미칠 수 있는 문제가 된다. 즉, 설명변수들 사이에 강한 선형관계가 존재하는 다중공선성의 문제가 생기면 회귀계수 추정량의 분산이 크게 증가하게 되어 결과적으로 회귀계수의 신뢰구간 설정 및 해석에 큰 어려움이 발생하게 된다.

다중공선성은 분산팽창계수(variance inflation factor)를 계산해 보면 확인할 수 있다. 변수 $X_j$의 분산팽창계수는 $\frac{1}{1-R_j^2}$로 계산되는데, 여기에서 $R_j^2$는 변수 $X_j$를 종속변수로 하고 나머지 설명변수를 독립변수로 하는 회귀모형의 결정계수가 된다. 예를 들어 분산팽창계수의 값이 4 이상이 된다는 것은 $R_j^2$의 값이 0.75 이상이 된다는 것이므로 설명변수 사이에 강한 상관관계가 존재한다는 것을 의미한다고 하겠다.

분산팽창계수의 계산은 패키지 car에 있는 함수 vif()로 할 수 있다.

```
> library(car)
> vif(fit_s)
Population Illiteracy     Income      Frost
  1.245282   2.165848   1.345822   2.082547
```

회귀모형 fit_s에 있는 네 설명변수의 분산팽창계수가 모두 큰 값이 아닌 것으로 계산되었고, 따라서 다중공선성의 문제는 없는 것으로 보인다.

### 15.3.2 특이한 관찰값 탐지

회귀모형의 가정사항 만족여부를 확인하는 것과 더불어 특이한 관찰값의 존재유무를 확인하는 것도 중요한 회귀진단 항목이 된다. 특이한 관찰값이란 이상값 또는 영향력이 큰 관찰값 등을 의미하는 것으로 회귀계수의 추정에 과도하게 큰 영향을 미치는 관찰값이나 추정된 회귀모형으로는 설명이 잘 안 되는 관찰값이라고 할 수 있다.

영향력이 큰 관찰값을 발견하는데 필요한 통계량으로는 DFBETAS, DFFITS, Covariance ratio, Cook's distance와 Leverage 등이 있는데, 이러한 통계량 값의 출력은 함수 influence.measures()로 할 수 있다. 각 통계량에 대한 자세한 설명은 회귀분석과 관련된 문헌들을 참고하기 바란다.

데이터 프레임 women에 대한 회귀모형 fit_w에서 사용된 관찰값 중 특이한 관찰값이 있는지를

함수 influence.measures()를 이용하여 살펴보자.

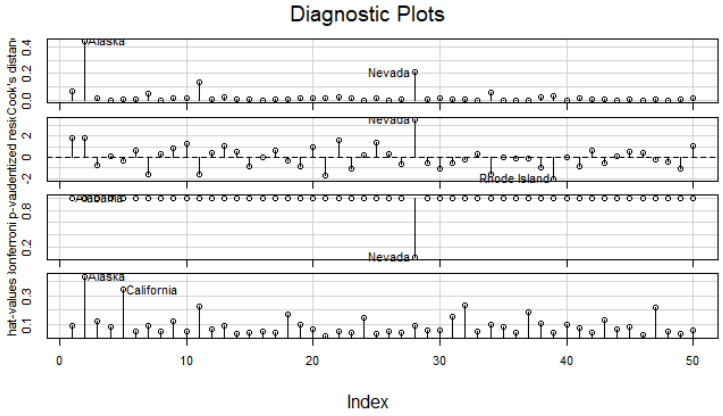
```
> influence.measures(fit_w)
Influence measures of
       lm(formula = weight ~ height, data = women) :

    dfb.1_  dfb.hght    dffit  cov.r   cook.d    hat inf
1   1.0106 -9.73e-01  1.14329 0.860 5.28e-01 0.2417   *
2   0.2893 -2.77e-01  0.34099 1.348 6.06e-02 0.1952
3   0.1222 -1.16e-01  0.15310 1.362 1.26e-02 0.1560
4   0.0123 -1.15e-02  0.01687 1.339 1.54e-04 0.1238
5  -0.0527  4.82e-02 -0.08447 1.288 3.84e-03 0.0988
6  -0.0789  6.91e-02 -0.16460 1.214 1.43e-02 0.0810
7  -0.0688  5.36e-02 -0.23753 1.119 2.88e-02 0.0702
8  -0.0212 -4.53e-16 -0.31959 1.004 4.94e-02 0.0667
9   0.0350 -4.92e-02 -0.21800 1.140 2.45e-02 0.0702
10  0.1203 -1.41e-01 -0.33506 1.044 5.50e-02 0.0810
11  0.1252 -1.39e-01 -0.24336 1.193 3.07e-02 0.0988
12  0.0854 -9.22e-02 -0.13567 1.311 9.86e-03 0.1238
13 -0.0035  3.72e-03  0.00491 1.390 1.31e-05 0.1560
14 -0.4406  4.64e-01  0.57152 1.179 1.59e-01 0.1952
15 -1.3653  1.43e+00  1.67669 0.514 8.78e-01 0.2417   *
```

영향력이 큰 관찰값을 발견하는데 사용되는 통계량들은 각기 일반적으로 통용되는 임계값이 있다. 만일 어느 하나의 통계량이라도 이러한 임계값을 초과하게 되는 관찰값에 대해서는 오른쪽 끝에 있는 'inf'에 별표(*)가 붙게 된다. 따라서 회귀모형 fit_w에서 사용된 관찰값 중에는 첫 번째와 15번째 관찰값이 영향력이 큰 관찰값으로 분류가 되었음을 알 수 있다.

함수 influence.measures()는 모든 관찰값에 대하여 여러 통계량의 값을 계산하여 출력하기 때문에 지나치게 많은 숫자가 말 그대로 쏟아져 나오게 된다. 또한 그 중 어느 하나의 통계량이라도 임계값을 초과하게 되면 별표가 붙기 때문에 이러한 관찰값들을 바로 영향력이 큰 관찰값으로 단순 분류하는 것도 문제가 있다.

많은 숫자에 포함되어 있는 정보를 효과적으로 나타내는 방법으로는 그래프가 최선이며, 패키지 car의 함수 infIndexPlot()과 함수 dfbetasPlots()가 회귀진단에 유용하게 사용할 수 있는 그래프를 작성한다. 회귀모형 fit_s에 대한 그래프를 작성해 보자.

```
> # 그림 15.14
> infIndexPlot(fit_s)
```

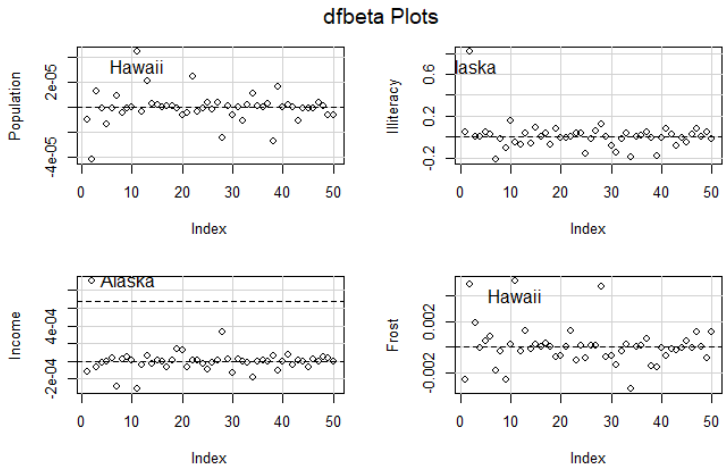



<그림 15.14> 함수 `infIndexPlot()`에 의한 회귀진단

4개의 패널에 각 관찰값에 대한 진단 통계량 값이 index plot의 형태로 작성되며, 각 패널마다 가장 극단적인 값을 갖는 두 관찰값의 라벨이 점 옆에 표시된다. 위에서 첫 번째 패널은 Cook's distance에 대한 것이다. 이것은 각 관찰값이 회귀계수의 추정에 미치는 영향력을 나타낸 통계량이다. 두 번째 패널은 스튜던트화 잔차에 대한 것이고, 세 번째 패널은 스튜던트화 잔차를 이용한 Bonferroni outlier 검정의 p-값이다. 마지막 패널은 leverage에 대한 것이다. 50개 주 중에서 Alaska의 Cook's distance와 leverage가 매우 큰 값을 갖고 있음을 알 수 있다. 또한 Nevada는 스튜던트화 잔차와 Cook's distance가 매우 큰 값을 갖고 있음을 알 수 있다. 주의해서 살펴보아야 할 관찰값이라 하겠다.

각 관찰값이 개별 회귀계수 추정에 미치는 영향력을 보여주는 통계량인 DFBETAS를 나타내는 그래프는 함수 `dfbetaPlots()`와 `dfbetasPlots()`로 작성 할 수 있다. 두 함수의 차이점은 첫 번째 함수가 각 관찰값의 제외로 인한 개별 회귀계수 추정값의 단순 차이를 보여주는 것인 반면에 두 번째 함수는 그 차이를 표준오차로 나누어서 표준화시킨 차이를 보여준다는 점이다. 또한 작성된 그래프에 특이한 관찰값이 있는 경우, 그 관찰값의 라벨을 그래프에 나타내기 위해서는 옵션 `id.method="identify"`를 추가해야 한다.

```
> # 그림 15.15
> dfbetaPlots(fit_s, id.method="identify")
```




<그림 15.15> DFBETAS 그래프

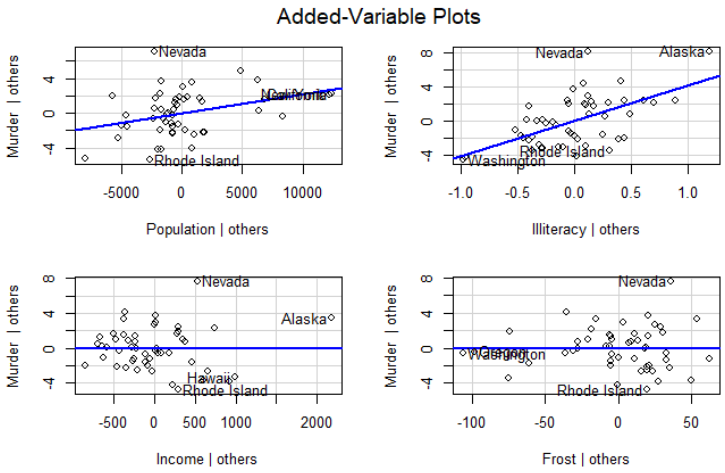
옵션 `id.method="identify"`를 추가된 상태로 함수 `dfbetaPlots()`이 실행되면, 왼쪽 위 패널의 그래프만 작성된다. 이어서 마우스를 Plots 창으로 옮기면 마우스 포인터가 십자로 바뀌게 되는데 이것을 특정 점 위로 이동하고 마우스의 왼쪽 단추를 누르면 그 점의 라벨이 점 주위에 표시된다. 첫 번째 그래프에서 더 이상 확인할 점이 없어서 두 번째 그래프를 나타나게 하려면 Esc 키를 누르면 된다. 같은 방식으로 점의 라벨을 확인하고 나머지 그래프를 작성하면 된다. Hawaii와 Alaska가 특이한 관찰값으로 선택되었다.

영향력이 있는 관찰값을 나타내는 유용한 그래프로 added variable plot 혹은 partial regression plot으로 알려진 그래프가 있다. 원래 이 그래프는 하나 또는 그 이상의 설명변수로 구성되어 있는 회귀모형에 설명변수 한 개를 더 추가함으로써 얻는 효과를 나타내려는 그래프이다.

반응변수 Y와 설명변수 $X_1, \ldots, X_k$로 구성된 회귀모형에서 $X_1$에 대한 added variable plot의 작성 절차를 살펴보면 다음과 같다. 우선 Y와 $X_2, \ldots, X_k$(즉, $X_1$을 제외한 모든 설명변수)로 회귀모형을 설정하고 잔차를 구한다. 이 잔차를 $e_{Y|2,\cdots,k}$라고 하자. 이어서 변수 $X_1$을 종속변수로 하고 나머지 모든 설명변수가 포함된 회귀모형을 설정하고 잔차를 구한다. 이 잔차를 $e_{1|2,\cdots,k}$ 라고 하자. 그러면 이제 두 잔차 $e_{Y|2,\cdots,k}$와 $e_{1|2,\cdots,k}$의 산점도를 작성할 수 있는데, 이 산점도가 변수 $X_1$의 added variable plot이 된다. 이렇게 작성된 산점도는 다른 변수의 영향력이 모두 제거되고 순수하게 변수 Y와 $X_1$만의 관계를 살펴볼 수 있는 것으로 이상값과 영향력이 큰 관찰값 등을 발견하는데 매우 효과적으로 사용되는 그래프가 된다.

R에서 added variable plots의 작성은 패키지 `car`에는 있는 함수 `avPlots()`로 할 수 있다. 데이터 프레임 `states`에 대한 회귀모형인 `fit_s`의 added variable plot을 작성해 보자.

```
> # 그림 15.16
> avPlots(fit_s)
```

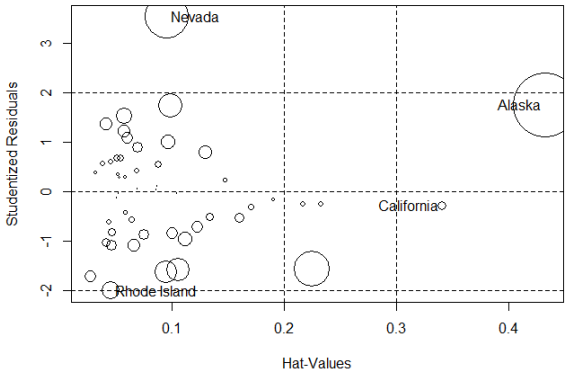



<그림 15.16> 회귀모형인 fit_s의 added variable plot

각 그래프마다 X축과 Y축상에서 각각 가장 극단적인 두 점에 대한 라벨이 점 옆에 추가되었다. X축상에서 극단적인 점은 partial leverage가 큰 값을 의미하고, Y축상에서의 극단적인 점은 잔차가 큰 값을 의미하는 것이다. Nevada와 Alaska가 특이한 관찰값으로 선택되었음을 볼 수 있다.

이상값이란 추정된 회귀모형으로는 설명이 잘 안 되는 관찰값이라고 할 수 있는데, 대부분의 경우 이상값은 큰 잔차를 갖게 된다. 그러나 만일 이상값에 해당되는 관찰값이 영향력도 크다면 회귀계수의 추정을 왜곡시켜 그렇게 크지 않은 잔차를 갖게 될 수도 있다. 따라서 이상값을 판단하고자 한다면 일반적인 잔차보다는 앞 절에서 정의된 스튜던트화 잔차를 이용하는 것이 더 효과적이라고 하겠다. 스튜던트화 잔차와 leverage, 그리고 Cook's distance를 하나의 그래프에서 한꺼번에 보여줄 수 있다면 특이한 관찰값을 분류하는데 큰 도움이 될 수 있다. 패키지 car에 있는 함수 influencePlot()으로 그러한 그래프를 작성할 수 있다.

```
> # 그림 15.17
> influencePlot(fit_s)
                StudRes         Hat       CookD
Alaska        1.7536917 0.43247319 0.448050997
California   -0.2761492 0.34087628 0.008052956
Nevada        3.5429286 0.09508977 0.209915743
Rhode Island -2.0001631 0.04562377 0.035858963
```




<그림 15.17> 회귀모형 fit_s에 대한 influence plot

함수 influencePlot()으로 작성된 그래프는 각 관찰값의 leverage를 X축 좌표로, 스튜던트화 잔차를 Y축 좌표로 하는 산점도이며, 점의 크기는 Cook's distance에 비례하여 결정된다. 또한 사용된 세 통계량의 값이 특이하게 큰 관찰값은 점 옆에 라벨이 표시된다.

그래프에서 확인할 수 있는 것은 우선 Alaska의 경우에는 가장 큰 값의 leverage와 Cook's distance 값을 갖고 있으며 스튜던트화 잔차도 작지 않다는 것도 알 수 있다. 따라서 Alaska는 영향력이 큰 관찰값으로 분류할 수 있다. 또한 Nevada의 경우 비록 leverage는 작으나 Cook's distance의 값은 상당히 크며, 크기가 매우 큰 스튜던트화 잔차를 갖고 있어 이상값으로 분류될 수 있다는 점이다.

이상값을 탐지하는데 가장 일반적으로 사용되는 통계량은 스튜던트화 잔차라고 할 수 있는데, 앞에서 언급한 바와 스튜던트화 잔차는 자유도가 $n-k-2$ 인 t-분포를 하고 있기 때문에 스튜던트화 잔차의 크기가 큰 관찰값을 이상값으로 분류할지 여부에 대한 p-값을 계산할 수 있다. 예를 들어 회귀모형 fit_s에서 스튜던트화 잔차가 가장 큰 Nevada에 대한 p-값을 계산해보자. 이 경우 $n=50$, $k=4$가 되므로 스튜던트화 잔차는 자유도가 44인 t-분포를 하게 된다. 따라서 p-값은 자유도가 44인 t-분포에서 Nevada의 스튜던트화 잔차보다 더 극단적인 값을 갖게 될 확률이 된다.

```
> 2*pt(max(rstudent(fit_s)),df=44,lower.tail=FALSE)
[1] 0.0009508836
```

실제로 이러한 검정은 스튜던트화 잔차의 크기가 상당히 큰 몇 개의 관찰값에 대해서만 이루어지게 되지만, 사실상 모든 관찰값이 검정 대상이 된다고 할 수 있다. 즉, 다중 검정이 실시되는 것이고 따라서 일종 오류를 적절하게 유지하기 위해서는 p-값에 대한 조절이 필요하다. 이 경우 Bonferroni 수정을 사용할 수 있는데, 패키지 car에 있는 함수 outlierTest()로 그러한 수정을 할 수 있다. 회귀모형 fit_s에 대하여 함수 outlierTest()를 적용시켜

이상값을 탐지해 보자.

```
> outlierTest(fit_s)
        rstudent unadjusted p-value  Bonferonni p
Nevada 3.542929          0.00095088     0.047544
```

디폴트 상태에서 함수 outlierTest()는 Bonferroni p-값이 0.05 이하가 되는 관찰값에 대해서만 검정결과를 출력시키며, 만일 모든 관찰값의 Bonferroni p-값이 0.05를 초과하는 경우에는 스튜던트화 잔차의 절대값이 가장 큰 관찰값에 대해서만 검정결과를 출력한다. 검정결과 Nevada의 경우 Bonferroni p-값이 0.047이 되어 이상값으로 분류할 수 있다는 결론을 얻게 되었다.

영향력이 큰 관찰값 혹은 이상값으로 분류된 관찰값에 대한 가장 단순한 처리방법은 분석에서 제거하는 것이지만 모든 경우에 사용할 수 있는 방식이라 할 수는 없다. 특이한 관찰값에 대한 처리 방법에 대해서는 Faraway(2005) 및 회귀분석과 관련된 다양한 문헌을 참고하기 바란다.

### 15.3.3 대안탐색

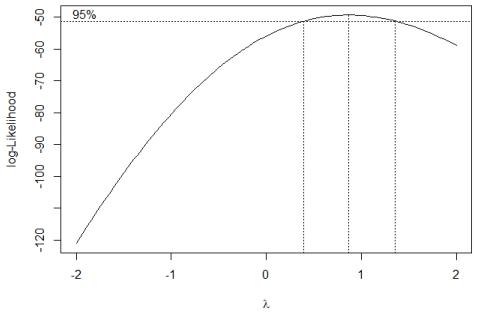
회귀모형의 가정이 만족되지 않는 경우 가장 먼저 고려할 수 있는 대안은 변수 변환이라고 할 수 있다. 예를 들어 동일분산 혹은 정규성에 대한 가정에 문제가 있는 경우 반응변수의 적절한 변환이 대안이 될 수 있으며, 반응변수와 설명변수의 관계가 선형이 아닌 경우에는 설명변수의 변환을 고려할 수 있다. 선형성에 문제가 있는 경우에는 다항회귀모형을 고려하는 것도 적절한 대안이 될 수 있다. 본 절에서는 반응변수 혹은 설명변수의 변환에 관련된 R 함수들을 살펴보고, 다항회귀에 대해서도 예제를 통하여 살펴보고자 한다.

- 반응변수의 변환

동일분산 또는 정규성의 가정이 만족되지 않아 반응변수의 변환이 필요한 경우에 가장 많이 사용되는 방법은 Box-Cox 변환이다. 이것은 양의 값을 갖는 반응변수에 대해서만 적용할 수 있는 것으로 변환 방식은 $Y$를 $Y^{\lambda}(\lambda \neq 0$의 경우)로 하는 지수변환(power transformation) 혹은 $\log Y$ ($\lambda = 0$의 경우)로 하는 로그변환이며, 변환모수 $\lambda$는 우도함수를 최대화시키는 조건으로 계산된다. 패키지 MASS에 있는 함수 boxcox() 또는 패키지 car에 있는 함수 powerTransform()으로 변환모수 $\lambda$값을 구할 수 있다. 음의 값도 갖는 반응변수에 적용되는 two parameter Box-Cox 변환의 경우에는 함수 powerTransform()에 옵션 family="bcnPower"를 지정하면 두 모수를 추정할 수 있다.

데이터 프레임 states에 대한 회귀모형 fit_s의 경우, 반응변수의 변환이 필요한지 여부를 확인해 보자. 함수 boxcox()에 회귀모형 fit_s를 입력하고 실행시키면 $-2 \leq \lambda \leq 2$에 대한 로그 우도함수의 값이 그래프로 그려진다.

```
> library(MASS)

> # 그림 15.18
> bc <- boxcox(fit_s)
> names(bc)
[1] "x" "y"
> bc$x[which.max(bc$y)]
[1] 0.8686869
```




<그림 15.18> 회귀모형 fit_s에 대한 Box-Cox 변환 그래프

그래프에서 로그 우도함수를 최대로 하는 대략적인 λ값을 찾을 수는 있지만, 정확한 값을 찾기 위해서는 함수 boxcox()의 결과를 객체에 할당해야 한다. 생성된 객체 bc는 리스트이고 bc$x는 λ값이, bc$y에는 로그 우도함수 값이 입력되어 있다. 함수 which.max()는 입력된 벡터의 최대값이 있는 인덱스를 구하는 것으로, 로그 우도함수 값이 최대가 되는 λ값을 구하게 된다. 그 결과 $\hat{\lambda} = 0.868$로 계산되었다. 즉 반응변수 Murder를 $Murder^{0.868}$로 변환할 것을 제안한 것이다.

반응변수의 변환에서 주의해야 점은 변환된 새로운 반응변수가 쉽게 해석할 수 있어야 한다는 것인데, $Murder^{0.868}$에 대한 적절한 해석은 찾기 어렵다고 할 것이다. 따라서 이러한 경우에는 계산된 λ를 그대로 사용하는 것보다는 계산된 값에 가장 가까우면서도 나름 해석이 가능한 값으로 수정하는 것이 필요하다. 그림 15.18에서 $\lambda = 1$ 이 최적 λ에 대한 95% 신뢰구간에 포함되어 있는 것을 알 수 있으므로 굳이 반응변수의 변환을 고려할 필요가 없다고 할 것이다.

이번에는 패키지 car에 있는 함수 powerTransform()에 적용시켜 회귀모형 fit_s의 최적변환을 탐색해 보자.

```
> library(car)

> summary(powerTransform(fit_s))
bcPower Transformation to Normality
   Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
Y1    0.8653           1       0.3853       1.3453

Likelihood ratio test that transformation parameter is equal to 0
 (log transformation)
                           LRT df     pval
```

```
LR test, lambda = (0) 13.14361  1 0.0002885

Likelihood ratio test that no transformation is needed
                            LRT df    pval
LR test, lambda = (1) 0.2990438  1 0.58448
```

함수 powerTransform()의 실행결과는 함수 summary()로 더 자세하게 출력된다. λ 의 추정값과 반올림된 값이 제공되는데, 실질적으로 의미가 있는 λ값은 반올림된 값이 된다. 또한 $\lambda = 0$ 과 $\lambda = 1$이라는 가설에 대한 검정을 실시하고 p-값을 출력시킨다. $\lambda = 1$에 대한 p-값이 0.58로 계산되어서 변환이 꼭 필요하다는 어떠한 근거도 제시할 수 없게 되었다.

- 설명변수의 변환

반응변수와 설명변수 사이의 선형관계 가정이 어긋나는 경우에는 설명변수의 변환이나 다항회귀모형을 고려해 볼 수 있다. 설명변수의 변환으로는 Box-Tidwell 변환이 있는데, 이 변환은 패키지 car에 있는 함수 boxTidwell( )로 실시할 수 있다. 일반적인 사용법은 다음과 같다.

```
boxTidwell(formula ,other.x= ,data= )
```

본 절에서 지금까지 살펴본 여러 함수들과는 다르게 함수 boxTidwell()에는 함수 lm()으로 생성된 객체대신 회귀모형을 설정하는 R 공식(formula)를 입력해야 한다. 옵션 other.x에는 변환에 포함시키지 않을 설명변수를 '~x1+x2'와 같이 물결표 오른쪽에 나열할 수 있다. 옵션 data는 데이터 프레임을 지정한다.

데이터 프레임 women에 대한 회귀모형 fit_w에서 반응변수 weight와 설명변수 height의 관계는 선형이 아니다. 함수 boxTidwell()로 설명변수의 적절한 변환을 시도해 보자.

```
> boxTidwell(weight~height, data=women)
 MLE of lambda Score Statistic (z)   Pr(>|z|)
        4.2008              13.067  < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

iterations =  2
```

설명변수 $X$를 $X^{\lambda}$로 변환하여 반응변수와 선형 관계를 유지하는 것이 변환의 목적이라 하겠다. 최적 변환은 $\lambda = 4.2$로 제안되었고, score 검정 결과도 유의한 것으로 나와 변환이 필요한 경우로 판단된다. 그러나 $\text{height}^{4.2}$보다는 $\text{height}^4$이 더 현실적인 변환이기 때문에 변수 height 대신 $\text{height}^4$를 설명변수로 사용하는 회귀모형을 설정할 수 있을 것이다.

변수 height 대신 $\text{height}^4$를 설명변수로 설정한 모형은 lm(wieght~I(height^4), women)으로 적합시킬 수 있다. 그러나 최고차항인 $\text{height}^4$만이 설명변수로 있는 모형은 바람직한 형태라고 할 없다. 그 이유는 $\text{height} \mapsto \text{height} + a$로 위치이동을 시키게 되면 다음과 같이 모든 차수의 항이 모형에 다시 나타나게 되는 모형의 중대한 변화를 유발할 수 있기

때문이다.

$$Y = \beta_0 + \beta_1(height + a)^4 + \varepsilon$$
$$= \beta_0 + \beta_1 a^4 + 4\beta_1 a^3 height + 6\beta_1 a^2 height^2 + 4\beta_1 a\ height^3 + \beta_1 heigth^4 + \varepsilon$$

따라서 데이터 프레임 women에 대해서는 다음과 같은 다항회귀모형을 고려하는 것이 바람직하다고 하겠다.

```
> mod1.fit_w <- lm(weight~poly(height, degree=4, raw=TRUE), women)
```

다항회귀모형은 함수 poly()를 이용하여 설정할 수 있다. 옵션 raw의 디폴트는 FALSE로 직교다항회귀모형이 사용된다. 위에 설정된 일반적인 4차 다항회귀모형의 추정결과를 함수 summary()로 확인해보면 알 수 있는데, 모든 회귀계수가 0이라는 가설은 p-값이 2.2e-16보다 작게 되어 기각되지만 개별 회귀계수들은 $\alpha = 0.05$ 에서 모두 비유의적으로 나타난다. 이것은 다중공선성이 존재할 때 나타나는 전형적인 현상으로 다항회귀모형의 차수가 너무 고차인 경우에 발생할 수 있는 문제가 된다. 따라서 차수를 하나 낮추어 3차 다항회귀모형을 설정해보도록 하자.

```
> mod2.fit_w <- lm(weight~poly(height,degree=3,raw=TRUE),women)
> summary(mod2.fit_w)

Call:
lm(formula = weight ~ poly(height, degree = 3, raw = TRUE), data = women)

Residuals:
     Min       1Q   Median       3Q      Max
-0.40677 -0.17391  0.03091  0.12051  0.42191

Coefficients:
                                      Estimate Std. Error t value Pr(>|t|)
(Intercept)                         -8.967e+02  2.946e+02  -3.044  0.01116
poly(height, degree = 3, raw = TRUE)1  4.641e+01  1.366e+01   3.399  0.00594
poly(height, degree = 3, raw = TRUE)2 -7.462e-01  2.105e-01  -3.544  0.00460
poly(height, degree = 3, raw = TRUE)3  4.253e-03  1.079e-03   3.940  0.00231
---

Residual standard error: 0.2583 on 11 degrees of freedom
Multiple R-squared:  0.9998,   Adjusted R-squared:  0.9997
F-statistic: 1.679e+04 on 3 and 11 DF,  p-value: < 2.2e-16
```

개별회귀계수가 모두 유의적으로 나타났다. 또한 회귀모형 mod3.fit_w에 대한 회귀진단에서도 별다른 문제는 발견되지 않았다.

## 15.4 변수 선택

변수 선택이란 반응변수에 영향을 줄 수 있다고 생각되는 많은 설명변수들 중에서 '최적'의

변수를 선택하여 모형에 포함시키는 절차를 의미한다. 변수 선택 방법은 크게 두 가지로 구분되는데, 검정에 의하여 단계적으로 모형을 찾아가는 방법이 하나이고, 다른 하나는 다양한 평가측도를 최적화시키는 모형을 찾아가는 방법이다. 어떤 방법을 이용하여 변수 선택을 할 것인지는 회귀모형을 수립하는 목적에 가장 잘 부합되는 방법을 이용하면 될 것이다. 즉, 어떤 모형을 '최적' 모형으로 정의할 것인지를 먼저 결정해야 한다는 것이다.

변수 선택 과정은 회귀진단 또는 변수 변환 등과 분리된 분석 과정이 아닌, 서로 연관된 분석 과정이 된다. 따라서 만일 이상값이 발견되어 분석에서 제외가 되었거나 혹은 변수 변환이 이루어진 경우에는 반드시 변수 선택 과정을 다시 밟아야 할 것이다.

### 15.4.1 검정에 의한 방법

검정에 의한 방법이란 SAS나 SPSS 등에서 일반적으로 이루어지는 변수 선택 방법으로써 후진소거법, 전진선택법과 단계별 선택법으로 불리는 방법을 의미한다. 변수 선택 과정에서 소요되는 계산이 방대하지 않기 때문에 대규모의 설명변수가 있는 경우 손쉽게 중요 변수를 선택할 수 있다는 이점이 있는 방법이지만 변수의 선택과 제거가 '한 번에 하나씩' 이루어지기 때문에 이른바 '최적' 모형을 놓치는 경우가 발생할 수도 있으며, 각 단계마다 여러 번의 검정이 동시에 이루어지기 때문에 일종 오류가 증가하는 검정의 정당성 문제가 발생할 수도 있다. 또한 모형의 수립목적이 예측인 경우 변수선택 과정이 목적과 잘 어울리지 않는다는 문제도 지니고 있다.

SAS나 SPSS의 경우와는 다르게 R에서는 한 번의 실행으로 검정에 의한 최종모형을 얻는 방법은 없다. 대신 패키지 MASS에 있는 함수 addterm() 혹은 dropterm()을 반복적으로 실행시키며 사용자가 추가할 변수나 제거할 변수를 직접 선택해야 한다.

예제 데이터로 행렬 state.x77을 이용해 보자. 우선 분석의 편의를 위해 데이터 프레임 state_df로 변환시키고, 변수의 이름을 확인하자.

```
> state_df <- as.data.frame(state.x77)
> names(state_df)
[1] "Population" "Income"     "Illiteracy" "Life Exp"   "Murder"
[6] "HS Grad"    "Frost"      "Area"
```

변수 'Life Exp'와 'HS Grad'는 이름 중간에 빈 칸이 있는데, 이것은 함수 lm()을 사용할 때 문제가 될 수 있다. 따라서 'Life Exp'를 Life.Exp로, 'HS Grad'를 HS.Grad로 이름을 패키지 dplyr의 함수 rename()으로 수정하도록 하자.

```
> library(dplyr)

> state_df <- rename(state_df, Life.Exp='Life Exp', Hs.Grad='HS Grad')
> names(state_df)
```

```
[1] "Population" "Income"     "Illiteracy" "Life.Exp"   "Murder"
[6] "Hs.Grad"    "Frost"      "Area"
```

변수 Murder를 반응변수로 하고, 나머지 변수를 설명변수로 설정하여 검정에 의한 변수선택법을 실시해 보자.

**1) 전진선택법**

전진선택법은 절편만이 있는 모형에서 시작하여 영향력이 큰 변수를 각 단계마다 한 개씩 추가해가는 방법을 의미한다. 영향력이 큰 변수란 해당 변수가 모형에 추가됨으로써 증가되는 회귀제곱합의 증가분이 가장 큰 변수가 되며, 만일 그 증가분이 $\alpha_{ENTRY}$수준에서 유의하면 모형에 포함시키게 된다. 추가할 변수의 선택은 패키지 MASS의 함수 addterm()으로 할 수 있는데, 기본적인 사용법은 addterm(object, scope, test="F")이다.

첫 번째 요소 object는 함수 lm()으로 생성된 객체로서 현 단계의 회귀모형이며, 두 번째 요소 scope에는 고려 대상이 되고 있는 모든 설명변수가 포함된 완전회귀모형을 지정하는 것이다. 세 번째 요소 test="F"를 포함시키면 회귀제곱합의 증가분에 대한 F-검정 결과가 산출된다.

데이터 프레임 state_df에 대하여 $\alpha_{ENTRY} = 0.1$을 기준으로 하여 전진선택법을 실시해 보자. 먼저 함수 addterm()의 요소 scope를 위한 완전회귀모형을 설정하자.

```
> full.fit <- lm(Murder ~ ., state_df)
```

또한 절편만이 있는 모형도 설정하자.

```
> fit <- lm(Murder ~ 1, state_df)
```

이제 모형에 포함시킬 첫 번째 변수를 선택해 보자.

```
> library(MASS)

> addterm(fit, scope=full.fit, test="F")
Single term additions

Model:
Murder ~ 1
           Df Sum of Sq    RSS     AIC   F Value    Pr(F)
<none>                  667.75 131.594
Population  1    78.85  588.89 127.311    6.427 0.0145504 *
Income      1    35.35  632.40 130.875    2.683 0.1079683
Illiteracy  1   329.98  337.76  99.516   46.894 1.258e-08 ***
Life.Exp    1   407.14  260.61  86.550   74.989 2.260e-11 ***
Hs.Grad     1   159.00  508.75 119.996   15.002 0.0003248 ***
Frost       1   193.91  473.84 116.442   19.643 5.405e-05 ***
Area        1    34.83   632.91 130.916    2.642 0.1106495
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

결과에서 'Sum of Sq'는 개별 변수가 모형에 추가됨으로써 증가되는 회귀제곱합의 증가분을 나타내고 있으며, RSS와 AIC는 각각의 개별 변수가 포함된 모형의 잔차제곱합과 AIC가 된다. 또한 계산된 p-값은 각 개별 변수의 회귀제곱합 증가분에 대한 F-검정 결과가 된다. 위 결과에서는 회귀제곱합의 증가분이 가장 큰 변수 Life.Exp가 영향력이 가장 큰 변수가 되며, 또한 p-값이 매우 작게 계산되었으므로 모형에 포함시킬 첫 번째 변수가 된다.

두 번째 단계로 넘어가기 위해서는 함수 addterm()의 첫 번째 요소인 fit을 변수 Life.Exp가 포함된 회귀모형으로 수정해야 한다. 회귀모형의 수정은 함수 lm()을 이용하여 다시 모형을 설정하는 것으로도 가능하지만, 매 단계마다 함수 lm()으로 다시 모형을 설정하는 것은 약간 번거로운 작업이 될 수 있다. 이러한 경우 다음과 같이 함수 update()를 사용할 수 있다.

```
> fit <- update(fit, . ~ . + Life.Exp, state_df)
```

함수 update()안의 첫 번째 요소인 객체 fit은 현재 상태의 회귀모형이다. 두 번째 요소로 사용된 기호 '. ~ . + Life.Exp'의 의미는 기존의 반응변수와 설명변수를 그대로 유지한 모형을 나타내는 기호 '. ~ .'에 변수 Life.Exp를 설명변수에 포함시킨 회귀모형을 나타내는 것이다. 함수 update()는 수정된 회귀모형을 함수 lm()을 이용하여 다시 적합시키고 그 결과를 객체 fit에 할당한다.

두 번째 변수선택을 위한 결과물은 다음과 같다.

```
> addterm(fit, scope=full.fit, test="F")
Single term additions

Model:
Murder ~ Life.Exp
           Df Sum of Sq    RSS    AIC  F Value    Pr(F)
<none>                  260.61 86.550
Population  1    56.615  203.99 76.303 13.0442 0.0007374 ***
Income      1     0.958  259.65 88.366  0.1733 0.6790605
Illiteracy  1    60.549  200.06 75.329 14.2249 0.0004533 ***
Hs.Grad     1     1.124  259.48 88.334  0.2035 0.6539823
Frost       1    80.104  180.50 70.187 20.8575 3.576e-05 ***
Area        1    14.121  246.49 85.764  2.6926 0.1074933
```

변수 Frost의 회귀제곱 증가분이 가장 크고 p-값도 $\alpha_{ENTRY}$보다 작기 때문에 모형에 포함될 두 번째 변수로 선택되었다. 세 번째 단계로 선택할 변수를 살펴보자.

```
> fit <- update(fit, . ~ . + Frost, state_df)
> addterm(fit, scope=full.fit, test="F")
Single term additions
```

```
Model:
Murder ~ Life.Exp + Frost
           Df Sum of Sq    RSS    AIC  F Value   Pr(F)
<none>                   180.50 70.187
Population  1   23.7098  156.79 65.146  6.9559 0.01136 *
Income      1    5.5598  174.94 70.622  1.4619 0.23281
Illiteracy  1    6.0663  174.44 70.477  1.5997 0.21231
Hs.Grad     1    2.0679  178.44 71.610  0.5331 0.46901
Area        1   21.0840  159.42 65.976  6.0837 0.01743 *
```

변수 Population이 모형에 포함될 세 번째 변수로 선택되었다. 네 번째 단계로 선택할 변수를 살펴보자.

```
> fit <- update(fit, . ~ . + Population, state_df)
> addterm(fit, scope=full.fit, test="F")
Single term additions

Model:
Murder ~ Life.Exp + Frost + Population
           Df Sum of Sq    RSS    AIC F Value   Pr(F)
<none>                  156.79 65.146
Income      1    0.7393 156.06 66.909  0.2132 0.64650
Illiteracy  1   11.8262 144.97 63.225  3.6710 0.06173 .
Hs.Grad     1    1.8215 154.97 66.561  0.5289 0.47083
Area        1   19.0402 137.75 60.672  6.2198 0.01637 *
```

변수 Area가 모형에 포함될 네 번째 변수로 선택되었다. 다음으로 선택할 변수가 있는지 살펴보자.

```
> fit <- update(fit, . ~ . + Area, state_df)
> addterm(fit, scope=full.fit, test="F")
Single term additions

Model:
Murder ~ Life.Exp + Frost + Population + Area
           Df Sum of Sq    RSS    AIC F Value   Pr(F)
<none>                  137.75 60.672
Income      1    1.2408 136.51 62.220 0.39993 0.53040
Illiteracy  1    8.7227 129.03 59.402 2.97446 0.09161 .
Hs.Grad     1    0.7708 136.98 62.392 0.24758 0.62126
```

$\alpha_{ENTRY} = 0.1$으로 설정했기 때문에 변수 Illiteracy도 모형에 포함될 수 있다.

```
> fit <- update(fit, . ~ . + Illiteracy, state_df)
> addterm(fit, scope=full.fit, test="F")
Single term additions

Model:
Murder ~ Life.Exp + Frost + Population + Area + Illiteracy
          Df Sum of Sq    RSS    AIC   F Value   Pr(F)
<none>                  129.03 59.402
Income    1   0.02595 129.01 61.392 0.008651 0.9263
Hs.Grad   1   0.76279 128.27 61.105 0.255713 0.6157
```

나머지 두 변수의 p-값이 매우 크게 계산되어 더 이상 추가할 변수가 없게 되었다. 따라서 전진선택법은 종료되며, 선택된 변수는 Life.Exp와 Frost, Population, Area, 그리고 Illiteracy이다.

### 2) 후진소거법

후진소거법은 고려대상이 되는 모든 설명변수가 포함된 완전회귀모형에서 시작하여 영향력이 미약한 변수를 하나씩 제거해가는 방법이다. 영향력이 미약한 변수의 선택은 전진선택법과 동일하게 회귀제곱합의 증가분을 기준으로 하고 있으나 여기에서는 증가분이 가장 작은 변수가 선택되며 F-검정의 p-값이 $\alpha_{STAY}$에서 비유의적인 경우 모형에서 제거된다. 제거할 변수의 선택은 패키지 MASS의 함수 dropterm()으로 할 수 있으며, 기본적인 사용법은 dropterm(object, test="F")이다.

요소 object는 함수 lm()으로 생성된 객체로서 현재 단계의 회귀모형이다. 옵션 test="F"를 포함시키면 회귀제곱의 증가분에 대한 F-검정 결과가 산출된다.

데이터 프레임 state_df에 대하여 $\alpha_{STAY} = 0.1$을 기준으로 후진소거법을 실시해 보자.

```
> fit <- lm(Murder ~ ., state_df)
> dropterm(fit, test="F")
Single term deletions

Model:
Murder ~ Population + Income + Illiteracy + Life.Exp + Hs.Grad +
    Frost + Area
           Df Sum of Sq    RSS    AIC  F Value    Pr(F)
<none>                  128.03 63.013
Population  1    25.719 153.75 70.166   8.437 0.005842 **
Income      1     0.236 128.27 61.105   0.077 0.782318
Illiteracy  1     8.299 136.33 64.154   2.722 0.106409
Life.Exp    1   127.175 255.21 95.503  41.719 8.68e-08 ***
Hs.Grad     1     0.973 129.01 61.392   0.319 0.575191
Frost       1     9.260 137.29 64.505   3.038 0.088673 .
Area        1     7.514 135.55 63.865   2.465 0.123914
```

변수 Income의 'Sum of Sq'의 값이 가장 작게 계산되었고, 따라서 가장 큰 값의 p-값이 산출되었다. 계산된 p-값이 지나치게 큰 값이므로 제거에 문제가 없어 보인다.

기존의 모형에서 변수 Income를 제거하는 방법도 다음과 같이 함수 update()를 사용할 수 있다.

```
> fit <- update(fit, . ~ . - Income, state_df)
```

기존 모형에서의 변수 제거는 기존 모형을 의미하는 '. ~ .' 기호에 제거할 변수 income 앞에 마이너스(-) 기호를 붙여 설명변수의 자리에 놓는 것으로써 이루어진다. 두 번째로 제거할 변수를 선택하기 위하여 함수 update()로 수정한 객체 fit을 다시 함수 dropterm()에 입력해 보자.

```
> dropterm(fit, test="F")
Single term deletions

Model:
Murder ~ Population + Illiteracy + Life.Exp + Hs.Grad + Frost +
    Area
           Df Sum of Sq    RSS    AIC  F Value    Pr(F)
<none>                  128.27 61.105
Population  1    27.142 155.41 68.702   9.099  0.004283 **
Illiteracy  1     8.715 136.98 62.392   2.921  0.094617 .
Life.Exp    1   127.500 255.77 93.613  42.742 6.032e-08 ***
Hs.Grad     1     0.763 129.03 59.402   0.256  0.615664
Frost       1     9.345 137.61 62.621   3.133  0.083831 .
Area        1     7.310 135.58 61.877   2.451  0.124802
```

변수 HS.Grad가 두 번째로 제거할 변수로 선택되었다. 세 번째로 제거할 변수를 살펴보자.

```
> fit <- update(fit, . ~ . - HS.Grad, state_df)
> dropterm(fit, test="F")
Single term deletions

Model:
Murder ~ Population + Illiteracy + Life.Exp + Hs.Grad + Frost +
    Area
           Df Sum of Sq    RSS    AIC  F Value    Pr(F)
<none>                  128.27 61.105
Population  1    27.142 155.41 68.702   9.099  0.004283 **
Illiteracy  1     8.715 136.98 62.392   2.921  0.094617 .
Life.Exp    1   127.500 255.77 93.613  42.742 6.032e-08 ***
Hs.Grad     1     0.763 129.03 59.402   0.256  0.615664
Frost       1     9.345 137.61 62.621   3.133  0.083831 .
Area        1     7.310 135.58 61.877   2.451  0.124802
```

모형에 있는 나머지 변수들의 p-값이 $\alpha_{STAY} = 0.1$ 보다 작게 계산되어 더 이상 제거할 변수가 없는 것으로 나타났다. 따라서 후진소거법은 종료되며, 선택된 변수는 전진선택법과 동일하게 되었다.

### 3) 단계별 선택법

전진선택법에서는 일단 모형에 포함된 변수에 대해서는 추가적인 검정이 이루어지지 않는다. 따라서 일단 모형에 포함된 변수는 당연히 최종모형에도 포함이 된다. 그러나 모형에 포함될 단계에서는 유의적이었던 변수도 다른 변수가 모형에 포함되면 비유의적으로 바뀔 가능성도 있는 것이기 때문에 모형에 있는 변수에 대해서도 추가적인 검정이 이루어질 필요가 있다.

단계별 선택법은 전진선택법과 동일하게 진행된다. 차이점은 모형에 변수가 추가되고 나면 모형에 있는 모든 변수들을 대상으로 후진소거법을 실시하여 제거할 변수를 찾는다는 점이다. 따라서 함수 addterm()으로 추가할 변수를 찾아, 함수 update()로 모형을 수정한 후 함수 dropterm()으로 제거할 변수를 찾는 과정을 추가하면 된다.

### 15.4.2 모형선택 기준에 의한 방법

모형을 수립할 때에는 모형의 사용목적이 있기 마련이다. 따라서 주어진 모형이 이러한 목적을 얼마나 잘 만족시키는지를 측정할 수 있는 통계량을 변수선택의 기준으로 삼는다면 사용목적에 가장 부합되는 모형을 선택할 수 있을 것이다. 모형선택 기준으로 사용할 수 있는 통계량으로는 결정계수, 수정결정계수, 잔차제곱평균(MSE), $C_p$통계량, AIC와 BIC 등이 있다. 이 통계량들이 어떤 의미에서 모형선택 기준으로 사용할 수 있는지는 회귀분석과 관련된 문헌을 참고하기 바란다.

모형선택 기준에 의한 변수선택에는 크게 두 가지 방법이 있다. 하나는 모든 가능한 회귀(All possible regression)이고 다른 하나는 단계별 선택법이다. 모든 가능한 회귀는 설명변수의 모든 가능한 조합에 대하여 모형선택 기준으로 사용되는 통계량 값을 계산하여 최적모형을 선택하는 방법이다. 이 방법의 문제는 설명변수의 수가 증가하면 비교해야 할 모형의 수가 기하급수적으로 증가한다는 것이다. 즉, 고려대상이 되는 설명변수가 $k$ 개 있다면 비교해야 할 모형의 수는 $2^k$개 된다. 컴퓨터 속도가 느렸던 과거에는 비현실적인 방법이었으나 컴퓨터 성능의 향상으로 점차 사용 가능한 방법으로 인식되고 있다. 그러나 data mining 등의 분야에서와 같은 대규모의 데이터에 대해서는 여전히 지나치게 시간이 많이 걸리는 방법이라고 하겠다. 이와 같은 경우 탐색의 범위를 제한하는 방법을 생각할 수 있는데, AIC 혹은 BIC를 기준으로 하는 단계별 선택법이 대안이 될 수 있다.

예제 데이터 프레임 state_df를 대상으로 두 방법을 각각 적용시켜보자.

#### 1) 모든 가능한 회귀

모든 가능한 회귀는 패키지 leaps에 있는 함수 regsubsets()으로 실시할 수 있다. 기본적인 사용법은 regsubsets(formula, data, ...)이다

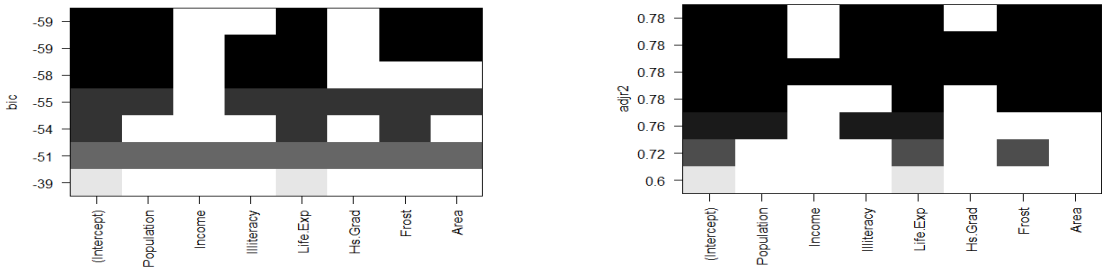
```
> library(leaps)
> fits <- regsubsets(Murder ~ ., state_df)
```

분석 결과는 설명변수의 수가 $k$ 인 모형 중 특정 통계량을 기준으로 가장 최적인 모형이 무엇인지를 보여주는 방식이며, 함수 plot()이나 summary()를 사용해야 볼 수 있다. 설명변수의 수가 $k$인 모형 중 가장 최적인 한 개의 모형이 아니라 최상의 몇 개 모형을 보고

싶다면 옵션 nbest에 원하는 모형의 개수를 지정하면 된다.

함수 plot()을 사용하는 경우, 옵션 scale에 최적 모형의 기준으로 사용할 통계량을 지정하면 된다. 디폴트는 scale="bic"이다.

```
> # 그림 15.19 (a)
> plot(fits)

> # 그림 15.19 (b)
> plot(fits, scale="adjr2")
```




(a) (b)

<그림 15.19> 함수 regsubsets()에 의한 모든 가능 회귀의 결과

그래프의 각 행은 하나의 모형을 의미한 것으로 색이 채워진 직사각형은 모형에 포함된 변수를 나타내고 있다. 예를 들어 BIC를 기준으로 변수 Life.Exp만 포함된 모형의 BIC가 -39 로 가장 크고, 변수 Population, Life.Exp, Frost, Area로 이루어진 모형의 BIC가 가장 작다는 것을 의미한다. Y축의 눈금은 반올림된 숫자를 보여주고 있다. 수정결정계수를 기준으로 하는 경우에는 변수 변수 Population, Illiteracy, Life.Exp, Frost, Area로 이루어진 모형이 최적이라는 것도 보여주고 있다.

분석 결과를 나타내는 다른 방법으로써 함수 summary()를 사용하면 많은 분석 결과가 리스트로 출력된다.

```
> best_fits <- summary(fits)
> names(best_fits)
[1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat"
[8] "obj"
```

이 중 outmat의 결과를 먼저 확인해 보자.

```
> best_fits$outmat
         Population Income Illiteracy Life.Exp Hs.Grad Frost Area
1  ( 1 ) " "        " "    " "        "*"      " "     " "   " "
2  ( 1 ) " "        " "    " "        "*"      " "     "*"   " "
3  ( 1 ) "*"        " "    "*"        "*"      " "     " "   " "
4  ( 1 ) "*"        " "    " "        "*"      " "     "*"   "*"
```

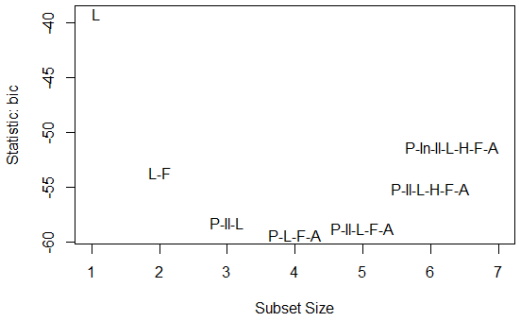
```
5  ( 1 ) "*"        " "    "*"        "*"      " "     "*"    "*"
6  ( 1 ) "*"        " "    "*"        "*"      "*"     "*"    "*"
7  ( 1 ) "*"        "*"    "*"        "*"      "*"     "*"    "*"
```

설명변수의 수가 $k$인 모형 중 결정계수가 가장 높은 모형을 구성하고 있는 변수에 별표가 표시되어 있다. 즉, 변수의 개수가 하나인 모형 중에는 변수 Life.Exp가 있는 모형이 최적이고, $k=2$인 모형 중에는 변수 Life.Exp와 Frost로 이루어진 모형이, $k=3$인 모형 중에는 변수 Population, Illiteracy, Life.Exp로 이루어진 모형이 결정계수가 가장 높다는 의미가 된다. 설명변수의 개수가 $k$인 모형 중 결정계수가 가장 높은 2개의 모형을 표시하고자 한다면 함수 regsubsets()에 옵션 nbest=2를 추가하면 된다.

객체 best_fits의 다른 요소인 rsq에는 선정된 7개 모형의 결정계수가 입력되어 있고, rss에는 잔차제곱합이, adjr2에는 수정결정계수가, cp에는 $C_p$통계량의 값이, 그리고 bic에는 각 모형의 BIC 통계량의 값이 각각 입력되어 있다.

선정된 7개 모형 중 각 통계량을 기준으로 최적 모형을 보여주는 그래프를 작성하는 것이 필요한데, 패키지 car의 함수 subsets()를 이용하면 비교적 편하게 필요한 그래프를 작성할 수 있다. 기본적인 사용법은 subsets(object, statistics=c("bic", "cp", "adjr2", "rsq", "rss"), legend="interactive", ...)이다. object는 함수 regsubsets()으로 생성된 객체이고, 옵션 statistics는 기준으로 삼는 통계량을 지정하는 것으로 디폴트는 "bic"이다. 작성되는 그래프는 가능한 짧게 줄인 변수의 이름을 산점도 기호로 사용한다. 따라서 줄여서 쓴 문자가 어떤 변수에 해당되는지 범례를 제공하는데, 옵션 legend의 디폴트인 "interactive"는 범례의 위치를 마우스로 지정하겠다는 것을 의미한다. 만일 옵션 legend에 FALSE를 지정하면 범례가 Console 창에 출력된다.

```
> library(car)

> # 그림 15.20
> subsets(fits, legend=FALSE)
           Abbreviation
Population           P
Income              In
Illiteracy          Il
Life.Exp             L
Hs.Grad              H
Frost                F
Area                 A
```

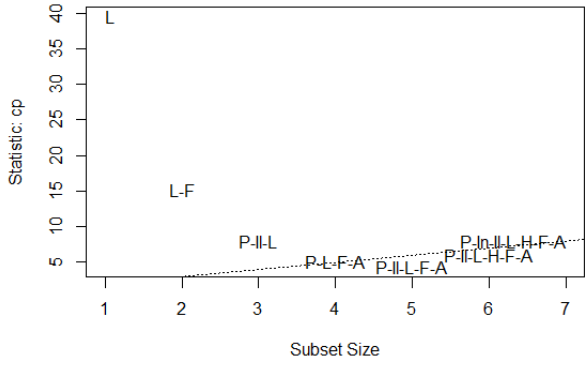



<그림 15.20> BIC 통계량 그래프

옵션 statistics의 디폴트인 BIC를 기준으로 하는 최적 모형은 변수의 개수가 4인 P-L-F-A가 되는데, 이것은 변수 Population, Life.Exp, Frost, Area으로 이루어진 모형을 지칭하는 것이다.

이제 $C_p$ 통계량을 이용하여 최종모형을 선택해 보자.

```
> # 그림 15.21
> subsets(fits, statistic="cp", legend=FALSE)
           Abbreviation
Population           P
Income              In
Illiteracy          Il
Life.Exp             L
Hs.Grad              H
Frost                F
Area                 A
> abline(a=1, b=1, lty=3)
```




<그림 15.20> $C_p$통계량 그래프

$C_p$ 통계량에서 $p$는 모형에 포함된 모수의 개수이다. 따라서 절편이 포함된 모형에서는 모형에 포함된 변수의 개수는 $p-1$이 된다. 그래프를 살펴보면 변수의 개수가 4개인 P-L-F-A 모형의 $C_p$ 통계량의 값이 $C_p = p$ 의 점선 바로 위에 있고 변수의 개수가 5개인 P-II-L-F-A 모형의 $C_p$ 통계량의 값이 $C_p = p$의 선 아래에 있는 것을 알 수 있다. 따라서 변수의 개수가 5개인 모형이

최소의 $C_p$ 통계량 값을 갖고 있으나, 변수의 개수가 4개인 모형의 $C_p$값과 큰 차이가 없음을 알 수 있다.

BIC와 $C_p$ 기준에서 P-L-F-A 모형과 P-II-L-F-A 모형에는 큰 차이가 없음을 알 수 있다. 가능한 단순한 모형이 선호된다는 관점에서 보면 변수가 4개인 모형을 최종 모형으로 선택할 수 있지만, 두 모형을 함께 고려하는 것이 필요하다고 보겠다.

함수 regsubsets()에서는 선정된 모형의 회귀계수를 미리 계산하지 않는다. 선정된 모형의 회귀계수 계산은 regsubsets 객체를 함수 coef()에 입력하고 변수 id에 몇 번째 모형인지를 지정하면 된다.

```
> coef(fits, id=4)
  (Intercept)    Population       Life.Exp          Frost
 1.387215e+02  1.581235e-04 -1.837437e+00 -2.204187e-02
         Area
 7.387061e-06
```

**2) 단계별 선택법**

AIC를 기준으로 하는 단계별 변수선택은 함수 step() 혹은 패키지 MASS의 함수 stepAIC()로 할 수 있다. 함수 step()은 함수 stepAIC()의 일부 기능을 단순화시킨 함수이지만, 함수 lm()으로 생성된 모형에 대해서는 제한사항 없이 동일하게 사용할 수 있다. 데이터 프레임 state_df에 대하여 단계별 선택법으로 최적 모형을 적용시켜보자. 함수 stepAIC()에 lm 객체를 입력하고 탐색 범위를 지정하는 옵션 scope가 생략되면 입력된 lm 모형을 대상으로 후진소거법이 디폴트로 적용된다. 모든 설명변수가 다 포함된 회귀모형인 fit.full을 함수 stepAIC()에 입력해 보자.

```
> fit.full <- lm(Murder ~ ., state_df)

> fit_1 <- stepAIC(fit.full)
Start:  AIC=63.01
Murder ~ Population + Income + Illiteracy + Life.Exp + Hs.Grad +
    Frost + Area

             Df Sum of Sq     RSS     AIC
- Income      1     0.236 128.27 61.105
- Hs.Grad     1     0.973 129.01 61.392
<none>                   128.03 63.013
- Area        1     7.514 135.55 63.865
- Illiteracy  1     8.299 136.33 64.154
- Frost       1     9.260 137.29 64.505
- Population  1    25.719 153.75 70.166
- Life.Exp    1   127.175 255.21 95.503

Step:  AIC=61.11
Murder ~ Population + Illiteracy + Life.Exp + Hs.Grad + Frost +
    Area

             Df Sum of Sq     RSS     AIC
```

```
- Hs.Grad     1     0.763 129.03 59.402
<none>                   128.27 61.105
- Area        1     7.310 135.58 61.877
- Illiteracy  1     8.715 136.98 62.392
- Frost       1     9.345 137.61 62.621
- Population  1    27.142 155.41 68.702
- Life.Exp    1   127.500 255.77 93.613

Step:  AIC=59.4
Murder ~ Population + Illiteracy + Life.Exp + Frost + Area

             Df Sum of Sq    RSS    AIC
<none>                   129.03 59.402
- Illiteracy  1     8.723 137.75 60.672
- Frost       1    11.030 140.06 61.503
- Area        1    15.937 144.97 63.225
- Population  1    26.415 155.45 66.714
- Life.Exp    1   140.391 269.42 94.213
```

7개 설명변수가 다 포함된 모형의 AIC가 63.01임을 알 수 있다. 함수 `stepAIC()`에 의한 단계별 선택법에서 변수를 제거시키는 기준은 AIC를 최소화시키는 모형을 찾는 것이다. 첫 번째 단계에서 변수 `Income`을 제거시킨 모형의 AIC가 61.105로 최소가 됨을 알 수 있다. 따라서 변수 `Income`이 제거되고 두 번째 단계로 넘어가게 된다. 두 번째 단계에서는 변수 `HS.Grad`를 제거한 모형의 AIC가 59.402로 최소가 됨을 알 수 있고, 변수 `HS.Grad`가 제거된 모형으로 세 번째 단계로 넘어간다. 세 번째 단계에서는 어떤 변수를 제거하든지 현재의 모형보다 AIC가 증가되는 것을 알 수 있고, 따라서 변수제거는 종료되고, 최종 모형이 `fit_1`에 할당된다.

```
> fit_1

Call:
lm(formula = Murder ~ Population + Illiteracy + Life.Exp + Frost +
    Area, data = state_df)

Coefficients:
(Intercept)   Population   Illiteracy    Life.Exp        Frost
  1.202e+02    1.780e-04    1.173e+00   -1.608e+00   -1.373e-02
       Area
  6.804e-06
```

AIC를 기준으로 모형을 선택했기 때문에 최종 모형에는 비유의적인 변수가 포함될 수 있다.

```
> confint(fit_1)
                    2.5 %        97.5 %
(Intercept)  8.553677e+01  1.547913e+02
Population   5.846453e-05  2.974973e-04
Illiteracy  -1.977147e-01  2.543676e+00
Life.Exp    -2.076162e+00 -1.139511e+00
Frost       -2.799858e-02  5.379601e-04
Area         9.217859e-07   1.268632e-05
```

변수 Illiteracy와 Frost의 회귀계수에 대한 95% 신뢰구간에 0이 포함된 것을 알 수 있다.

함수 stepAIC()에서는 디폴트인 후진소거법이 적용되면 일단 제거된 변수는 다음 단계에서 고려대상이 되지 않는다. 만일 제거된 변수도 최소 AIC 모형을 찾는 고려대상에 여전히 포함시키고자 한다면 옵션 direction="both"를 포함시키면 된다.

```
> stepAIC(fit.full, direction="both")
Start:  AIC=63.01
Murder ~ Population + Income + Illiteracy + Life.Exp + Hs.Grad +
    Frost + Area

             Df Sum of Sq    RSS    AIC
- Income      1     0.236 128.27 61.105
- Hs.Grad     1     0.973 129.01 61.392
<none>                    128.03 63.013
- Area        1     7.514 135.55 63.865
- Illiteracy  1     8.299 136.33 64.154
- Frost       1     9.260 137.29 64.505
- Population  1    25.719 153.75 70.166
- Life.Exp    1   127.175 255.21 95.503

Step:  AIC=61.11
Murder ~ Population + Illiteracy + Life.Exp + Hs.Grad + Frost +
    Area

             Df Sum of Sq    RSS    AIC
- Hs.Grad     1     0.763  129.03 59.402
<none>                    128.27 61.105
- Area        1     7.310 135.58 61.877
- Illiteracy  1     8.715 136.98 62.392
- Frost       1     9.345 137.61 62.621
+ Income      1     0.236 128.03 63.013
- Population  1    27.142 155.41 68.702
- Life.Exp    1   127.500 255.77 93.613

Step:  AIC=59.4
Murder ~ Population + Illiteracy + Life.Exp + Frost + Area

             Df Sum of Sq    RSS    AIC
<none>                    129.03 59.402
- Illiteracy  1     8.723 137.75 60.672
+ Hs.Grad     1     0.763 128.27 61.105
+ Income      1     0.026 129.01 61.392
- Frost       1    11.030 140.06 61.503
- Area        1    15.937 144.97 63.225
- Population  1    26.415 155.45 66.714
- Life.Exp    1   140.391 269.42 94.213

Call:
lm(formula = Murder ~ Population + Illiteracy + Life.Exp + Frost +
    Area, data = state_df)

Coefficients:
(Intercept)   Population   Illiteracy    Life.Exp        Frost
  1.202e+02    1.780e-04    1.173e+00   -1.608e+00   -1.373e-02
       Area
  6.804e-06
```

두 방법에서 모두 동일한 결과가 산출되었다. 따라서 AIC 통계량에 의한 단계별 선택법으로 선택된 변수는 Population, Illiteracy, Life.Exp, Frost, Are 가 된다.

AIC가 아닌 BIC를 기준으로 단계별 모형 선택을 하고자 한다면 함수 stepAIC()의 변수 k값을 수정해 주어야 한다. 변수 k값은 AIC를 계산하는 과정에서 모형에 포함된 변수의 개수만큼 불이익을 주기 위한 상수로써 디폴트 값은 k=2가 된다. 이 값을 k=log(n)으로 수정하면 BIC에 의한 단계별 선택법이 적용된다. 단, n은 표본의 크기이다.

```
> stepAIC(fit.full, k=log(nrow(state_df)), trace=FALSE)

Call:
lm(formula = Murder ~ Population + Life.Exp + Frost + Area, data =
state_df)

Coefficients:
(Intercept)   Population     Life.Exp        Frost         Area
  1.387e+02    1.581e-04   -1.837e+00   -2.204e-02    7.387e-06
```

모형 탐색 과정을 출력하지 않고 최종 모형만을 바로 보려고 한다면, 옵션 trace=FALSE를 지정하면 된다. AIC를 기준으로 하는 최종 모형과 다른 결과가 나왔다.

지금까지 변수선택을 위한 방법으로 검정에 의한 방법과 모형선택 기준에 의한 방법이 R에서 어떻게 구현되는지를 살펴보았다. 변수선택에 대해서는 많은 문헌에서 지적하고 있는 것과 같이 목적을 이루는 수단일 뿐 목적자체가 아님을 명심해야 한다. 선택된 모형에 대하여 회귀진단을 포함한 다각적인 검증이 필요하며 모형수립 목적에 부합하고 있는지도 함께 평가해야 할 것이다.

## 15.5 purrr과 broom을 이용한 모형 적합

## 15.6 15장을 마치며

회귀분석은 가장 적용범위가 넓은 통계분석 방법론이라고 할 수 있다. 따라서 여러 분야에서 중요한 분석도구로 빈번하게 사용되고 있는 실정이다. 그러나 통계분석 목적에 부합하는 모형을 찾는 작업 및 추정된 모형의 적합성 진단 등의 과정은 실제로 많은 경험이 필요한 고급기법이라고 할 수 있다. 본 장에서 살펴본 많은 R 함수들은 최적모형을 찾아가는 지루한 과정에서 매우 유용하게 사용될 수 있을 것이다.

## 15.7 연습문제

1. 패키지 MASS에 있는 데이터 프레임 Cars93는 1993년 미국에서 판매된 93대 자동차에 대한 27개의 변수로 이루어진 데이터이다. 시내주행 시 연비를 나타내는 변수 MPG.city를 반응변수로 하는 회귀모형을 수립하고자 한다. 모형수립 목적은 연비의 예측이며 설명변수로는 Price, EngineSize, Horsepower, RPM, Passengers, Length, Wheelbase, Width, Weight를 고려하고 있다.

   1) 모든 변수들이 포함된 산점도 행렬을 작성하여 변수들 사이의 관계를 그래프로 나타내라.

   2) 고려대상이 되는 모든 설명변수를 포함한 다중회귀모형을 적합시키고 각 회귀계수에 대한 검정을 실시하라.

   3) 2)번에서 추정된 모형에 대한 회귀진단을 실시하라. 추정된 모형이 오차항에 대한 가정사항을 모두 만족시키고 있는가? 다중공선성의 문제는 존재하는가? 또한 특이한 관찰값은 존재하는가? 만일 특이한 관찰값이 존재한다면 그 관찰값에 대한 변수 Manufacturer와 Model, Type의 값을 구하라.

   4) 고려대상이 되고 있는 설명변수들을 대상으로 변수선택을 실시하라. 검정에 의한 방법 및 모형선택기준에 의한 방법을 각각 사용하여 모형을 선택하라.

   5) 4)에서 선택된 모형에 대하여 3)번과 동일한 방법으로 회귀진단을 실시하라. 문제가 발견되었다면 그 문제를 해결하고 최종모형을 선택하라.

   6) 변수 Type의 값이 “Sporty”와 “Van”인 경우를 제외하고 회귀모형을 다시 적합시키고자 한다. 최종모형을 선택하라.

2. 패키지 UsingR에 있는 데이터 프레임 fat에는 체지방률 및 여러 신체 부위를 체중계와 줄자를 이용해서 측정한 결과가 들어있다. 분석목적은 체지방률을 효과적으로 예측하는 모형을 수립하는 것이다. 반응변수는 체지방률을 나타내는 body.fat이고 설명변수로는 age, weight, height, BMI, neck, chest, abdomen, hip, thigh, knee, ankle, bicep, forearm, wrist를 고려하고자 한다.

1) 최종모형을 선택하라. 선택된 최종모형은 회귀진단에서 어떤 문제도 발견되지 않아야 한다.

2) 피실험자들을 나이에 따라 20-30대(age < 40), 40-50대(40 ≤ age < 60), 60대 이상(age ≥ 60)으로 구분하여 각각의 경우에 체지방률을 예측하는 최종모형을 각각 수립하라. 추정된 각 모형도 회귀진단에서 아무런 문제도 발견되지 않아야 한다.