

모두를 위한 R 데이터 분석 입문

2판



Chapter 11

회귀분석



목차

1. 단순선형 회귀분석
2. 다중선형 회귀분석
3. 로지스틱 회귀분석

Section 01

단순선형 회귀분석

1. 회귀분석 관련 용어

- 증권회사에서는 미래의 주식 시세를 예측하기 위해 많은 연구
- 주식 시세는 기업의 매출액, 원유가격, 국제정세, 정부정책 발표 등 매우 많은 요인들에 의해 영향 받음
- **독립변수(independent variable)**:주식시세에 영향을 미치는 요인들(기업의 매출액, 원유가격, 국제정세, 정부정책 발표)
- **종속변수(dependent variable)**: 독립변수의 영향에 따라 값이 결정되는 주식시세
- 독립변수와 종속변수를 다른 용어로 각각 설명변수(explanatory variable)와 반응변수(response variable)라고도 함

1. 단순선형 회귀분석

1. 회귀분석 관련 용어

- **예측모델(prediction model) 또는 예측모형**: 독립변수에 해당하는 자료와 종속변수에 해당하는 자료를 모아 관계를 분석하고 이를 예측에 사용할 수 있는 통계적 방법으로 정리한 것
- **회귀분석(regression analysis)**: 회귀 이론을 기초로 독립변수(설명변수)가 종속변수(반응변수)에 미치는 영향을 파악하여 예측 모델을 도출하는 통계적 방법
- 회귀분석은 여러 가지 종류가 있는데, 회귀분석에서 독립변수의 수가 하나인 경우를 **단순 회귀(simple regression)**라고 하고, 독립변수의 수가 두 개 이상인 경우를 **다중 회귀(multiple regression)**라고 함

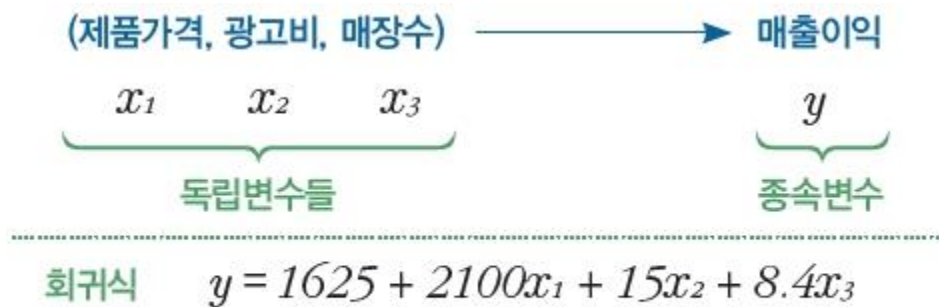


그림 11-1 독립변수, 종속변수, 회귀식

1. 단순선형 회귀분석

2. 단순선형 회귀분석의 목표

- **단순선형 회귀:** 독립변수(x)와 종속변수(y) 사이의 선형관계를 파악하고 이를 예측에 활용하는 통계적 방법
ex) 기온(x) 자료를 가지고 아이스크림 판매량(y)을 예측하는 문제
- 단순선형 회귀모델 또는 단순선형 회귀식은 다음과 같이 1차식의 형태를 가짐

$$y = Wx + b \text{ (} W, b \text{ 는 상수)}$$

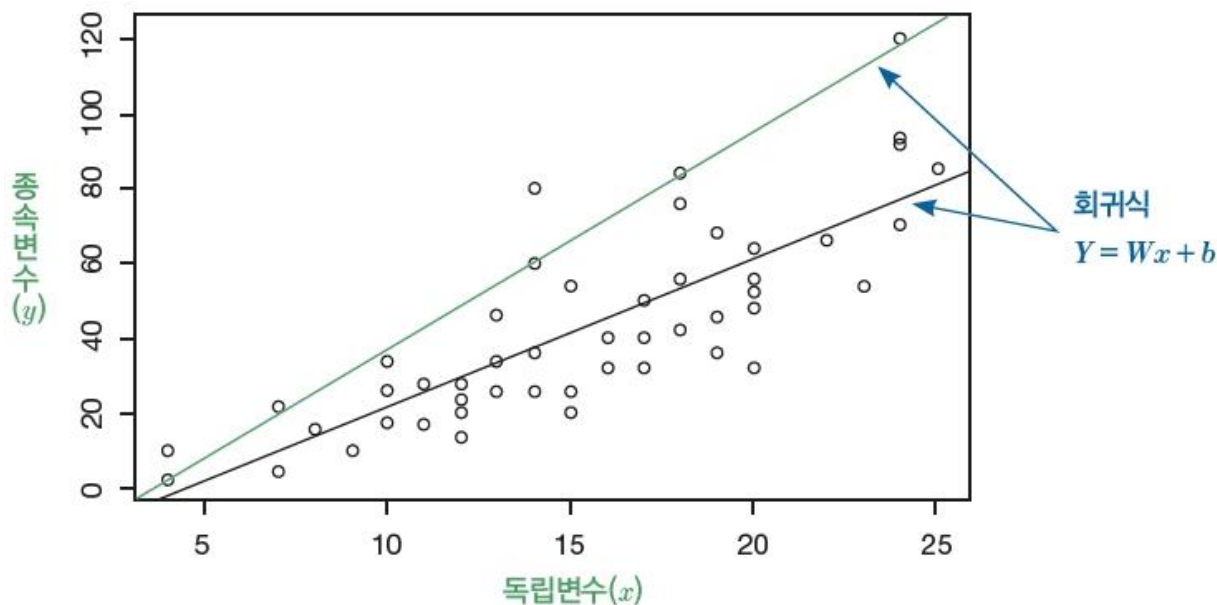


그림 11-2 산점도와 단순선형 회귀식

1. 단순선형 회귀분석

3. R을 이용한 단순선형 회귀분석

3.1 주행속도와 제동거리 사이의 회귀모델 구하기

- 단순선형 회귀식을 구하기 위해서는 이론적인 이해가 필요하지만, R에서 제공하는 `lm()` 함수를 이용하여 쉽게 회귀식을 구할 수 있음

코드 11-1

```
head(cars)
plot(dist~speed, data=cars)           # 산점도를 통해 선형 관계 확인

model <- lm(dist~speed, cars)         # 회귀모델 구하기
model

abline(model)                         # 회귀선을 산점도 위에 표시
coef(model)[1]                       # b 값 출력
coef(model)[2]                       # W 값 출력
```


1. 단순선형 회귀분석

```
> head(cars)
```

```
  speed dist
```

```
1     4     2
```

```
2     4    10
```

```
3     7     4
```

```
4     7    22
```

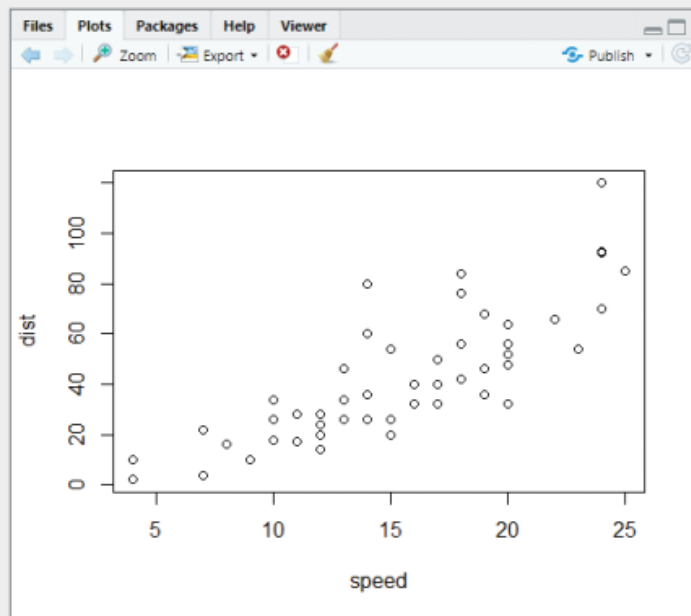
```
5     8    16
```

```
6     9    10
```

```
> plot(dist~speed, data=cars)
```

```
>
```

산점도를 통해 선형 관계 확인



1. 단순선형 회귀분석

```
> model <- lm(dist~speed, cars)           # 회귀모델 구하기
> model
Call:
lm(formula = dist ~ speed, data = cars)

Coefficients:
(Intercept)      speed 
   -17.579       3.932
```

- **dist~speed**

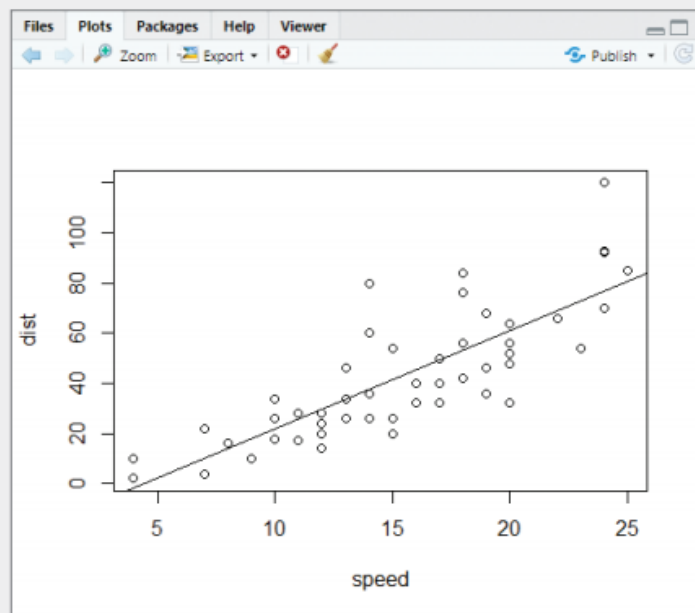
회귀모델에서 독립변수와 종속변수를 지정하는 것으로, ~를 기준으로 '종속변수~독립변수'의 순서로 지정해야 한다. 여기서 순서가 바뀌면 안 된다.

- **cars**

회귀모델을 만드는 데 사용할 데이터셋이다. 여기에서는 dist와 speed가 cars의 열이어야 한다.

1. 단순선형 회귀분석

```
> abline(model)
```



```
# 회귀선을 산점도 위에 표시
```

```
> coef(model)[1]
```

```
(Intercept)
```

```
-17.57909
```

```
> coef(model)[2]
```

```
speed
```

```
3.932409
```

```
# b 값 출력
```

```
# W 값 출력
```

1. 단순선형 회귀분석

3.2 주행속도에 따른 제동거리 구하기

코드 11-2

```
b <- coef(model)[1]
W <- coef(model)[2]

speed <- 30                # 주행속도
dist <- W*speed + b        # 제동거리
dist

speed <- 35                # 주행속도
dist <- W*speed + b        # 제동거리
dist

speed <- 40                # 주행속도
dist <- W*speed + b        # 제동거리
dist
```

1. 단순선형 회귀분석

```
> b <- coef(model)[1]
> W <- coef(model)[2]
> speed <- 30                                # 주행속도
> dist <- W*speed + b
> dist                                        # 제동거리
    speed
100.3932
> speed <- 35                                # 주행속도
> dist <- W*speed + b
> dist                                        # 제동거리
    speed
120.0552
> speed <- 40                                # 주행속도
> dist <- W*speed + b
> dist                                        # 제동거리
    speed
139.7173
```

1. 단순선형 회귀분석

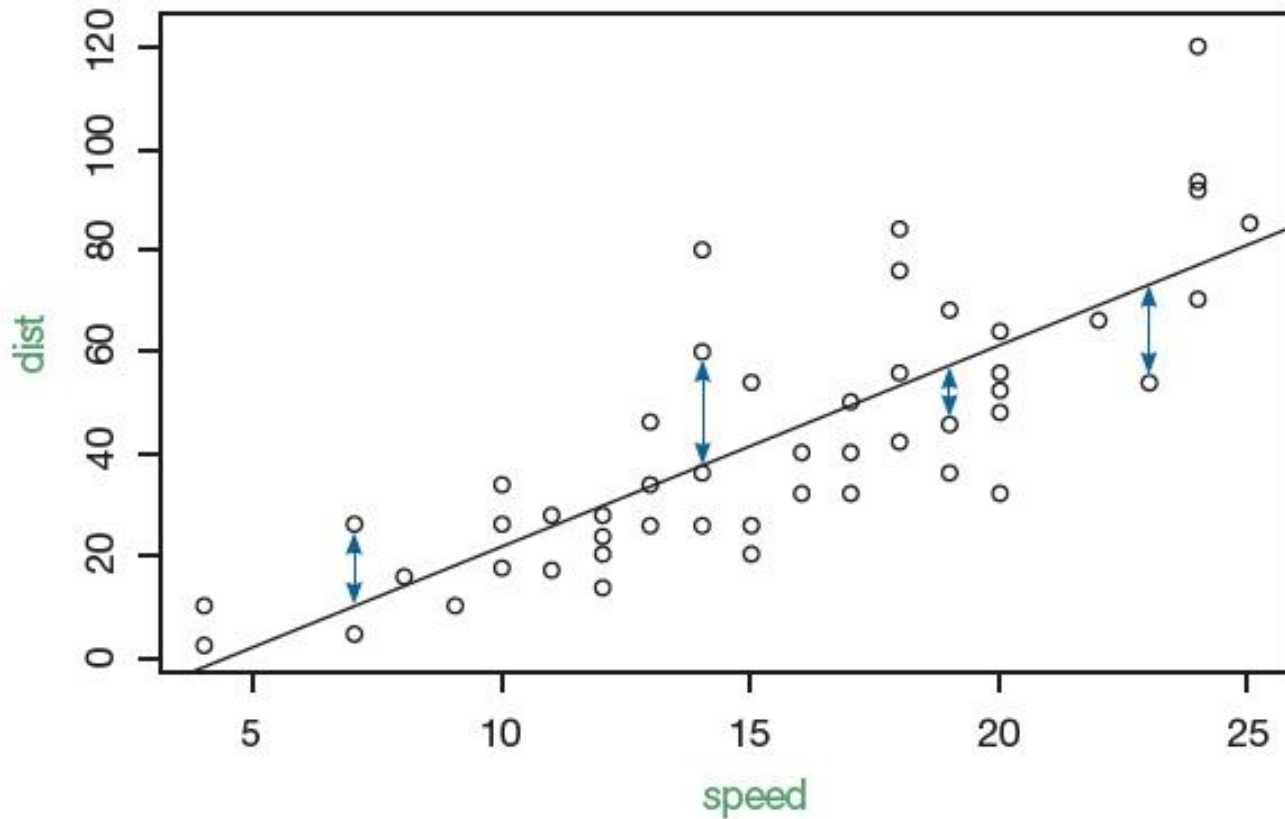


그림 11-3 회귀모델에 의한 예측값과 실제값의 차이

1. 단순선형 회귀분석

3.3 예상 제동거리, 실제 제동거리, 오차 구하기

- cars 데이터셋의 주행속도(speed) 데이터를 앞에서 구한 회귀식에 대입

코드 11-3

```
speed <- cars[,1]                # 주행속도
pred <- W * speed + b
pred                             # 예상 제동거리

compare <- data.frame(pred, cars[,2], pred-cars[,2])
colnames(compare) <- c('예상','실제','오차')
head(compare)
```

```
> speed <- cars[,1]                # 주행속도
> pred <- W * speed + b
```

1. 단순선형 회귀분석

```
> pred                                # 예상 제동거리
[1] -1.849460 -1.849460  9.947766  9.947766 13.880175
[6] 17.812584 21.744993 21.744993 21.744993 25.677401
[11] 25.677401 29.609810 29.609810 29.609810 29.609810
[16] 33.542219 33.542219 33.542219 33.542219 37.474628
[21] 37.474628 37.474628 37.474628 41.407036 41.407036
[26] 41.407036 45.339445 45.339445 49.271854 49.271854
[31] 49.271854 53.204263 53.204263 53.204263 53.204263
[36] 57.136672 57.136672 57.136672 61.069080 61.069080
[41] 61.069080 61.069080 61.069080 68.933898 72.866307
[46] 76.798715 76.798715 76.798715 76.798715 80.731124
>
```

```
> compare <- data.frame(pred, cars[,2], pred-cars[,2])
> colnames(compare) <- c('예상', '실제', '오차')
> head(compare)
```

	예상	실제	오차
1	-1.849460	2	-3.849460
2	-1.849460	10	-11.849460
3	9.947766	4	5.947766
4	9.947766	22	-12.052234
5	13.880175	16	-2.119825
6	17.812584	10	7.812584

Section 02

다중선형 회귀분석

2. 다중선형 회귀분석

1. 다중선형 회귀모델 만들기

- 단순선형 회귀가 하나의 독립변수를 다룬다면 다중선형 회귀는 여러 개의 독립변수를 다룸 ex)키와 몸무게를 가지고 혈당 수치를 예측하는 문제

키(x1), 몸무게(x2): 독립변수

혈당수치(y): 종속변수

- 다중 회귀모델 (다중 회귀식)의 일반적인 형태

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \dots + \beta_n x_n$$

- R에서는 다중 회귀모델도 lm() 함수로 구함

2. 다중선형 회귀분석

코드 11-4

```
library(car)
head(Prestige)
newdata <- Prestige[,c(1:4)]           # 회귀식 작성을 위한 데이터 준비
plot(newdata, pch=16, col="blue",     # 산점도를 통해 변수 간 관계 확인
      main="Matrix Scatterplot")
mod1 <- lm(income ~ education + prestige + # 회귀식 도출
            women, data=newdata)
summary(mod1)
```

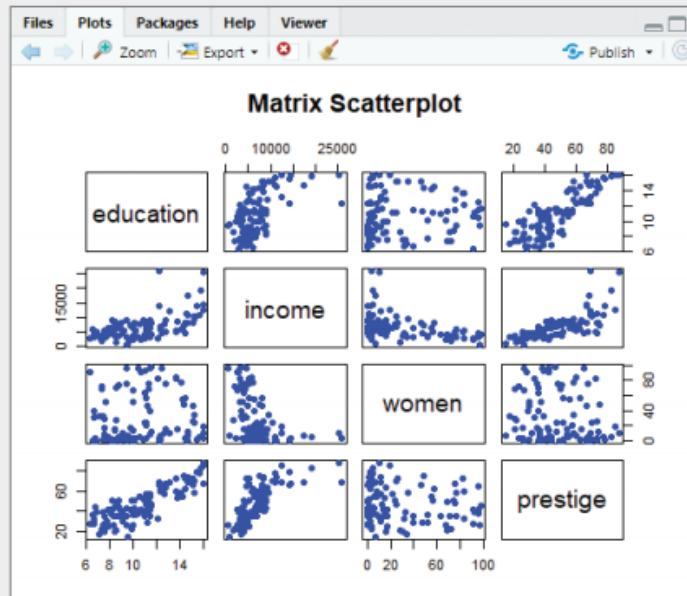
```
> library(car)
> head(Prestige)
```

	education	income	women	prestige	census	type
gov.administrators	13.11	12351	11.16	68.8	1113	prof
general.managers	12.26	25879	4.02	69.1	1130	prof
accountants	12.77	9271	15.70	63.4	1171	prof
purchasing.officers	11.42	8865	9.11	56.8	1175	prof
chemists	14.62	8403	11.68	73.5	2111	prof
physicists	15.64	11030	5.13	77.6	2113	prof

2. 다중선형 회귀분석

```
> newdata <- Prestige[,c(1:4)]  
> plot(newdata, pch=16, col="blue",  
+       main="Matrix Scatterplot")
```

회귀식 작성을 위한 데이터 준비
산점도를 통해 변수 간 관계 확인



2. 다중선형 회귀분석

```
> mod1 <- lm(income ~ education + prestige +      # 회귀식 도출
+             women, data=newdata)
> summary(mod1)
```

Call:

```
lm(formula = income ~ education + prestige + women, data = newdata)
```

Residuals:

Min	1Q	Median	3Q	Max
-7715.3	-929.7	-231.2	689.7	14391.8

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	-253.850	1086.157	-0.234	0.816	
education	177.199	187.632	0.944	0.347	
prestige	141.435	29.910	4.729	7.58e-06	***
women	-50.896	8.556	-5.948	4.19e-08	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

2. 다중선형 회귀분석

Residual standard error: 2575 on 98 degrees of freedom

Multiple R-squared: 0.6432, Adjusted R-squared: 0.6323

F-statistic: 58.89 on 3 and 98 DF, p-value: < 2.2e-16

- **income ~ education + prestige + women**

회귀모델에서 무엇이 독립변수이고 무엇이 종속변수인지 지정하는 것으로, ~ 앞에 있는 것이 종속변수, ~ 뒤 쪽에 있는 것이 독립변수이다. 독립변수가 여러 개이면 +로 연결한다.

- **data=newdata**

회귀모델 도출에 사용할 데이터셋을 지정한다. 변수명 income, education, prestige, women 은 newdata에 속한 열의 이름이다.

2. 다중선형 회귀분석

```
> summary(mod1)

call:
lm(formula = income ~ education + prestige + women, data = newdata)

Residuals:
    Min       1Q   Median       3Q      Max
-7715.3  -929.7  -231.2    689.7 14391.8

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -253.850   1086.157  -0.234    0.816
education    177.199    187.632    0.944    0.347
prestige     141.435     29.910    4.729 7.58e-06 ***
women        -50.896      8.556   -5.948 4.19e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2575 on 98 degrees of freedom
Multiple R-squared:  0.6432,    Adjusted R-squared:  0.6323 
F-statistic: 58.89 on 3 and 98 DF,  p-value: < 2.2e-16
```

① income을 설명하는 데 얼마나 중요한 변수인지를 나타냄

② 구한 모델이 의미 있는 모델인지를 나타냄

③ 모델이 income을 얼마나 잘 설명할 수 있는지를 나타냄

그림 11-4 회귀모델에 대한 설명 내용

- ①에 있는 *는 해당 변수가 종속변수를 설명하는 데 얼마나 중요한 변수인가를 나타냄. *가 많을수록 통계적으로 중요하다는 의미
- ②에 있는 p-value(유의수준) 값은 구한 회귀모델이 의미 있는 모델인지(신뢰할 수 있는 모델인지)를 나타내는 것으로, 이 값이 작을수록 의미 있는 모델인 것을 나타냄
- ③에 있는 Adjusted R-squared 값은 모델의 설명력을 나타내며 0~1 사이의 값을 갖음

2. 다중선형 회귀모델에서 변수의 선택

- 다중선형 회귀모델에서는 종속변수를 설명하는 데 도움되는 독립변수가 다수 존재
- 그런데 모든 독립변수가 종속변수를 설명하는 데 동일하게 기여하는 것은 아님
- 어떤 변수는 기여도가 높고, 어떤 변수는 기여도가 낮음
- 예를 들어 '수면시간', '학습시간'은 '성적'을 예측하는 데 중요한 기여를 할 수 있지만, '점심식사 여부'는 '성적'을 예측하는 데 별로 도움이 되지 않는 변수
- 기여도가 낮거나 거의 없는 변수들은 모델에서 제외하는 것이 좋음(적은 변수를 가지고 현실을 잘 설명할 수 있는 것이 좋은 모델이기 때문)
- R에서는 모델에 기여하는 변수들을 선별할 수 있는 `stepAIC()` 함수를 제공
 - AIC (Akaike Information Criterion)
- **stepAIC 함수의 목적**
 - 각 단계에서 AIC 값이 가장 크게 감소하는 변수를 제거하면서 최적의 모형을 찾는 것
 - 단계가 올라갈수록 단계의 AIC는 작아짐

2. 다중선형 회귀분석

코드 11-5

```
library(MASS)                # stepAIC( ) 함수 제공
newdata2 <- Prestige[,c(1:5)] # 모델 구축에 사용할 데이터셋 생성
head(newdata2)
mod2 <- lm(income ~ education + prestige +
            women + census, data= newdata2)
mod3 <- stepAIC(mod2)        # 변수 선택 진행
mod3                                # 변수 선택 후 결과 확인
summary(mod3)                # 회귀모델 상세 내용 확인
```

```
> library(MASS)                # stepAIC( ) 함수 제공
> newdata2 <- Prestige[,c(1:5)] # 모델 구축에 사용할 데이터셋 생성
> head(newdata2)
```

	education	income	women	prestige	census
gov.administrators	13.11	12351	11.16	68.8	1113
general.managers	12.26	25879	4.02	69.1	1130
accountants	12.77	9271	15.70	63.4	1171
purchasing.officers	11.42	8865	9.11	56.8	1175
chemists	14.62	8403	11.68	73.5	2111
physicists	15.64	11030	5.13	77.6	2113

```
> mod2 <- lm(income ~ education + prestige +
+            women + census, data= newdata2)>
```

2. 다중선형 회귀분석

```
> mod3 <- stepAIC(mod2)
```

변수 선택 진행

Start: AIC=1607.93

income ~ education + prestige + women + census

동일

	Df	Sum of Sq	RSS	AIC
- census	1	639658	649654265	1606.0
- education	1	5558323	654572930	1606.8
<none>			649014607	1607.9
- prestige	1	143207106	792221712	1626.3
- women	1	212639294	861653901	1634.8

1 단계에서 제거 대상:
AIC가 가장 작은 열

Step: AIC=1606.03

income ~ education + prestige + women

	Df	Sum of Sq	RSS	AIC
- education	1	5912400	655566665	1605.0
<none>			649654265	1606.0
+ census	1	639658	649014607	1607.9
- prestige	1	148234959	797889223	1625.0
- women	1	234562232	884216497	1635.5

2 단계에서 제거 대상:
AIC가 가장 작은 열

Step: AIC=1604.96

2. 다중선형 회귀분석

```
income ~ prestige + women
```

	Df	Sum of Sq	RSS	AIC
<none>			655566665	1605.0
+ education	1	5912400	649654265	1606.0
+ census	1	993735	654572930	1606.8
- women	1	234647032	890213697	1634.2
- prestige	1	811037947	1466604612	1685.1

```
> mod3
```

```
# 변수 선택 후 결과 확인
```

```
Call:
```

```
lm(formula = income ~ prestige + women, data = newdata2)
```

```
Coefficients:
```

(Intercept)	prestige	women
431.57	165.87	-48.38

```
> summary(mod3)
```

```
# 회귀모델 상세 내용 확인
```

```
Call:
```

2. 다중선형 회귀분석

```
lm(formula = income ~ prestige + women, data = newdata2)
```

Residuals:

Min	1Q	Median	3Q	Max
-7620.9	-1008.7	-240.4	873.1	14180.0

Coefficients:

	Estimate	Std.Error	t value	Pr(> t)	
(Intercept)	431.574	807.630	0.534	0.594	
prestige	165.875	14.988	11.067	< 2e-16	***
women	-48.385	8.128	-5.953	4.02e-08	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2573 on 99 degrees of freedom

Multiple R-squared: 0.64, Adjusted R-squared: 0.6327

F-statistic: 87.98 on 2 and 99 DF, p-value: < 2.2e-16

Section 03

로지스틱 회귀분석

3. 로지스틱 회귀분석

1. 로지스틱 회귀분석의 개념

- **로지스틱 회귀(logistic regression)** : 회귀모델에서 종속변수의 값의 형태가 연속형 숫자가 아닌 범주형 값인 경우를 다루기 위해서 만들어진 통계적 방법
ex) iris 데이터셋에서 4개의 측정값을 가지고 품종을 예측. 품종이 범주형 값
- R에서 로지스틱 회귀 모델은 glm() 함수 이용
 - 일반화 선형 모델(Generalized Linear Model)

2. 로지스틱 회귀모델 만들기

코드 11-6

```
iris.new <- iris
iris.new$Species <- as.integer(iris.new$Species) # 범주형 자료를 정수로 변환
head(iris.new)
mod.iris <- glm(Species ~., data= iris.new)      # 로지스틱 회귀모델 도출
summary(mod.iris)                               # 회귀모델의 상세 내용 확인
```

3. 로지스틱 회귀분석

```
> iris.new <- iris
> iris.new$Species <- as.integer(iris.new$Species) # 범주형 자료를 정수로 변환
> head(iris.new)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1          3.5          1.4          0.2        1
2          4.9          3.0          1.4          0.2        1
3          4.7          3.2          1.3          0.2        1
4          4.6          3.1          1.5          0.2        1
5          5.0          3.6          1.4          0.2        1
6          5.4          3.9          1.7          0.4        1

> mod.iris <- glm(Species ~., data= iris.new) # 로지스틱 회귀모델 도출
```

- **Species ~.**

회귀모델에서 종속변수가 Species이고, 나머지 변수들은 모두 독립변수이다.

- **data=iris.new**

회귀모델 도출에 사용할 데이터셋이 iris.new이다.

3. 로지스틱 회귀분석

```
> summary(mod.iris)
```

회귀모델의 상세 내용 확인

Call:

```
glm(formula = Species ~ ., data = iris.new)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-0.59215	-0.15368	0.01268	0.11089	0.55077

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.18650	0.20484	5.792	4.15e-08 ***
Sepal.Length	-0.11191	0.05765	-1.941	0.0542 .
Sepal.Width	-0.04008	0.05969	-0.671	0.5030
Petal.Length	0.22865	0.05685	4.022	9.26e-05 ***
Petal.Width	0.60925	0.09446	6.450	1.56e-09 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

3. 로지스틱 회귀분석

(Dispersion parameter for gaussian family taken to be 0.04800419)

Null deviance: 100.0000 on 149 degrees of freedom

Residual deviance: 6.9606 on 145 degrees of freedom

AIC: -22.874

Number of Fisher Scoring iterations: 2

3. 로지스틱 회귀분석

3. 로지스틱 회귀모델을 이용한 예측

- 수작업으로 계산하여 품종을 예측하는 방법 대신, 구해놓은 회귀모델을 이용하여 보다 편리한 방법으로 품종을 예측

코드 11-7

```
# 예측 대상 데이터 생성(데이터프레임)
unknown <- data.frame(rbind(c(5.1, 3.5, 1.4, 0.2)))
names(unknown) <- names(iris)[1:4]
unknown                                     # 예측 대상 데이터

pred <- predict(mod.iris, unknown)          # 품종 예측
pred                                       # 예측 결과 출력
round(pred,0)                            # 예측 결과 출력(소수 첫째 자리에서 반올림)

# 실제 품종명 알아보기
pred <- round(pred,0)
pred
levels(iris$Species)
levels(iris$Species)[pred]
```

3. 로지스틱 회귀분석

```
> # 예측 대상 데이터 생성(데이터프레임)
> unknown <- data.frame(rbind(c(5.1, 3.5, 1.4, 0.2)))
> names(unknown) <- names(iris)[1:4]
> unknown                                     # 예측 대상 데이터
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1          5.1          3.5          1.4          0.2
pred <- predict(mod.iris, unknown)           # 품종 예측
```

- **mod.iris**
로지스틱 회귀모델을 의미한다.
- **unknown**
예측 대상 데이터를 의미한다. 데이터 1건을 입력할 수도 있고 여러 개를 묶어서 입력할 수도 있다.

3. 로지스틱 회귀분석

```
> pred                                     # 예측 결과 출력
      1
0.9174506
> round(pred,0)                           # 예측 결과 출력(소수 첫째 자리에서 반올림)
      1
      1
> # 실제 품종명 알아보기
> pred <- round(pred,0)
> pred
      1
      1
> levels(iris$Species)
[1] "setosa"    "versicolor" "virginica"
> levels(iris$Species)[pred]
[1] "setosa"
```

3. 로지스틱 회귀분석

4. 다수의 데이터에 대한 예측

- 예측 대상 데이터가 여러 개인 경우에도 유사한 방법으로 예측

코드 11-8

```
test <- iris[,1:4]           # 예측 대상 데이터 준비
pred <- predict(mod.iris, test) # 모델을 이용한 예측
pred <- round(pred,0)
pred                         # 예측 결과 출력
answer <- as.integer(iris$Species) # 실제 품종 정보
pred == answer               # 예측 품종과 실제 품종이 같은지 비교
acc <- mean(pred == answer)  # 예측 정확도 계산
acc                         # 예측 정확도 출력
```

```
> test <- iris[,1:4]           # 예측 대상 데이터 준비
> pred <- predict(mod.iris, test) # 모델을 이용한 예측
```

3. 로지스틱 회귀분석

```
> pred <- round(pred,0)
```

```
> pred
```

예측 결과 출력

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
... (중간 생략)																
103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136
2	3	3	3	3	3	3	3	3	3	3	3	3	3	2	3	3
137	138	139	140	141	142	143	144	145	146	147	148	149	150			
3	3	3	3	3	3	3	3	3	3	3	3	3	3			

3. 로지스틱 회귀분석

```
> answer <- as.integer(iris$Species)      # 실제 품종 정보
> pred == answer                          # 예측 품종과 실제 품종이 같은지 비교
```

1	2	3	4	5	6	7	8	9	10	11
TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
12	13	14	15	16	17	18	19	20	21	22
TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
23	24	25	26	27	28	29	30	31	32	33
TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE

... (중간 생략)

122	123	124	125	126	127	128	129	130	131	132
TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
133	134	135	136	137	138	139	140	141	142	143
TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
144	145	146	147	148	149	150				
TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE				

```
> acc <- mean(pred == answer)      # 예측 정확도 계산
> acc                              # 예측 정확도 출력
[1] 0.9733333
```

Thank you!