빅데이터 통계적 자료 분석의 기초 : R 을 중심으로

- 필요 패키지 : `data.table`, `MASS`, `ggplot2`, `KernSmooth`, `car`, `faraway`, `ROCR`, `nnet`, `tree`
- 필요 데이터 : `titanic.csv`, `germancredit.csv`, `protein.csv`

A. 데이터 수집단계 가이드

데이터 안에 숨겨진 정보를 파악하여 의미있는 스토리를 제공하기 위해서는 올바른 데이터의 수집이 필수적이다.
특정한 분석을 위해 스스로 데이터를 수집할 수도 있지만 이에는 한계가 있고, 이미 많은 양의 데이터가 오픈 API 를 통해 제공되고 있기 때문에 이를 이용하는 것도 한가지 방편이 될 것이다.
다만, 공개되어 있는 데이터를 바로 이용하기에는 어려움이 있고, 자신의 분석 목적에 맞게 테이블을 변경하고 데이터를 가공해야 데이터의 새로운 가치를 부여할 수 있다.
특히, 의미있는 스토리를 만들어 내기 위해서는 다른 데이터의 정보도 접목시켜 이용해야 하는 경우가 많이 있기 때문에 올바른 데이터의 구성은 중요하다고 할 것이다.

빅데이터를 수집하는 기술로는 SNS, 뉴스 등의 웹정보를 인터넷에서 수집하는 크롤링(crawling), 각종 센서를 이용해 수집하는 센싱(sensing), 분산 시스템에서 데이터베이스 관리 시스템인 카산드라(Cassandra), 운영체제와 응요프로그램 간의 통신에 사용되는 메시지 형식의 개방된 오픈 API 등이 있다.
이들 기술에 대해 여기서 깊이 다루기는 한계가 있으므로 생략하기로 하고, 회귀분석, 기계학습 등과 같이 특정한 분석 목적에 부합하는 맞는 형태의 데이터를 구성하는 것은 이후 자료분석의 각 절에서 다루어질 것이다.

B. 데이터 전처리 및 저장 단계 가이드

- R-package 인 **data.table** 의 주요 사용법을 정리 및 숙지를 목표로 함
  - 다양한 기능들 중 데이터탐색, 연산, 병합에 초점
  - `data.table()`의 함수 옵션을 알아보고 example 을 통해 적용방법을 알아볼 것임
- Data : **titanic data**
  - titanic 호에 탔던 승객들의 정보에 대한 데이터

B.1. Data load

- 데이터를 R 로 불러온 후, 데이터의 class 를 data.table 로 변환

```
library(data.table)
titanic <- read.csv("./Data/titanic.csv", fileEncoding="UTF-8")
class(titanic)
```

```
## [1] "data.frame"
```

```
titanic.dt <- data.table(titanic)
class(titanic.dt)
```

```
## [1] "data.table" "data.frame"
```

```
str(titanic.dt)
```

```
## Classes 'data.table' and 'data.frame':   1309 obs. of  15 variables:
##  $ X        : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ pclass   : Factor w/ 3 levels "1st","2nd","3rd": 1 1 1 1 1 1 1 1 1 1 ...
##  $ survived : int  1 1 0 0 0 1 1 0 1 0 ...
##  $ name     : Factor w/ 1307 levels "Abbing, Mr. Anthony",..: 22 24 25 26
27 31 46 47 51 55 ...
##  $ sex      : Factor w/ 2 levels "female","male": 1 2 1 2 1 2 1 2 1 2 ...
##  $ age      : num  29 0.917 2 30 25 ...
##  $ sibsp    : int  0 1 1 1 1 0 1 0 2 0 ...
##  $ parch    : int  0 2 2 2 2 0 0 0 0 0 ...
##  $ ticket   : Factor w/ 929 levels "110152","110413",..: 188 50 50 50 50 1
25 93 16 77 826 ...
##  $ fare     : num  211 152 152 152 152 ...
##  $ cabin    : Factor w/ 187 levels "","A10","A11",..: 45 81 81 81 81 151 1
47 17 63 1 ...
##  $ embarked : Factor w/ 3 levels "Cherbourg","Queenstown",..: 3 3 3 3 3 3
3 3 3 1 ...
```

```
##  $ boat     : Factor w/ 28 levels "","1","10","11",..: 13 4 1 1 1 14 3 1 2
8 1 ...
##  $ body     : int  NA NA NA 135 NA NA NA NA NA 22 ...
##  $ home.dest: Factor w/ 369 levels "","?Havana, Cuba",..: 309 231 231 231
231 237 163 25 23 229 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
head(titanic.dt)
```

```
##    X pclass survived                             name    sex     age sibsp
## 1: 1    1st        1   Allen, Miss. Elisabeth Walton female 29.0000     0
## 2: 2    1st        1  Allison, Master. Hudson Trevor   male  0.9167     1
## 3: 3    1st        0    Allison, Miss. Helen Loraine female  2.0000     1
## 4: 4    1st        0 Allison, Mr. Hudson Joshua Crei   male 30.0000     1
## 5: 5    1st        0 Allison, Mrs. Hudson J C (Bessi female 25.0000     1
## 6: 6    1st        1             Anderson, Mr. Harry   male 48.0000     0
##    parch ticket     fare   cabin    embarked boat body
## 1:     0  24160 211.3375      B5 Southampton    2   NA
## 2:     2 113781 151.5500 C22 C26 Southampton   11   NA
## 3:     2 113781 151.5500 C22 C26 Southampton        NA
## 4:     2 113781 151.5500 C22 C26 Southampton       135
## 5:     2 113781 151.5500 C22 C26 Southampton        NA
## 6:     0  19952  26.5500     E12 Southampton    3   NA
##                         home.dest
## 1:                     St Louis, MO
## 2: Montreal, PQ / Chesterville, ON
## 3: Montreal, PQ / Chesterville, ON
## 4: Montreal, PQ / Chesterville, ON
## 5: Montreal, PQ / Chesterville, ON
## 6:                     New York, NY
```

B.2. data.table 과 data.frame

B.2.1 처리 속도 차이

- `data.table` 을 사용하는 가장 큰 이유중 하나는 빠른 처리 속도임 (`data.frame` 과 최소 20 배가량 차이)
  - `data.table` 은 `data.frame` 을 계승한 것임
  - 속도의 차이가 존재하는 이유는 `data.table` 은 데이터를 처리할 때 특정 column 을 key 값으로 색인을 지정하기 때문임

- 데이터가 크고 복잡할수록 속도의 차이는 더욱 확연하게 드러남

- `system.time()` : 처리 속도를 보여줌

- `setkey()` : data.table 을 sort 시켜줌

```
DF <- data.frame(x = runif(2.6e+07), y = rep(LETTERS, each = 10000))
df <- data.frame(x = runif(2.6e+07), y = rep(letters, each = 10000))
system.time(x <- DF[DF$y == "C", ])

##    user  system elapsed
##    2.05    0.48    2.53

DT <- as.data.table(DF)
setkey(DT,y)
system.time(x <- DT[J("C"), ])

##    user  system elapsed
##    0.02    0.00    0.02
```

B.2.2 데이터 선택

- `data.table` 의 형식 : **[행, 표현식, 옵션]**
  - "표현식"에는 선택하고자 하는 열의 번호, 변수명, J 표현식등이 올수 있음
    - 변수명으로 지정할 때, 문자는 따옴표를 씌워야함

- 1 열을 가져오기 위해 `data.frame` 에서는 [, 1]을 사용하지만 `data.table` 에서는 다른 결과를 얻음

- **with=F** : data.frame 의 형태로 출력하기 위한 option

```
head(titanic[,1])

## [1] 1 2 3 4 5 6

titanic.dt[,1]

## [1] 1

titanic.dt[,1, with=F]

##          X
##    1:    1
##    2:    2
```

```
##    3:    3
##    4:    4
##    5:    5
##   ---
## 1305: 1305
## 1306: 1306
## 1307: 1307
## 1308: 1308
## 1309: 1309
```

B.3. 조건을 이용한 데이터 선택

- `data.table` 의 **J 표현식**
  - **DT[J('제약조건')]** 형식으로 사용가능
  - **J** 를 사용하기 위해서는 key 를 반드시 설정해줘야함
    - key : 행을 정렬해주기 위한 기준변수

B.3.1 특정 등급 승객들만 선택하는 방법

- `data.frame` 에서 사용하는 방법과 동일함
- 예시 : 아래 예는 같은 결과를 가짐

```
titanic.dt[pclass=="1st",]
```

```
##        X pclass survived                                name    sex     age
##   1:   1    1st        1   Allen, Miss. Elisabeth Walton female 29.0000
##   2:   2    1st        1  Allison, Master. Hudson Trevor   male  0.9167
##   3:   3    1st        0    Allison, Miss. Helen Loraine female  2.0000
##   4:   4    1st        0 Allison, Mr. Hudson Joshua Crei   male 30.0000
##   5:   5    1st        0 Allison, Mrs. Hudson J C (Bessi female 25.0000
##  ---
## 319: 319    1st        0  Williams-Lambert, Mr. Fletcher   male      NA
## 320: 320    1st        1       Wilson, Miss. Helen Alice female 31.0000
## 321: 321    1st        1               Woolner, Mr. Hugh   male      NA
## 322: 322    1st        0              Wright, Mr. George   male 62.0000
## 323: 323    1st        1        Young, Miss. Marie Grice female 36.0000
##      sibsp parch   ticket     fare   cabin    embarked boat body
##   1:     0     0    24160 211.3375      B5 Southampton    2   NA
##   2:     1     2   113781 151.5500 C22 C26 Southampton   11   NA
##   3:     1     2   113781 151.5500 C22 C26 Southampton        NA
##   4:     1     2   113781 151.5500 C22 C26 Southampton       135
```

```
##   5:     1     2   113781 151.5500 C22 C26 Southampton        NA
##  ---
## 319:     0     0   113510  35.0000    C128 Southampton        NA
## 320:     0     0    16966 134.5000 E39 E41   Cherbourg    3   NA
## 321:     0     0    19947  35.5000     C52 Southampton    D   NA
## 322:     0     0   113807  26.5500         Southampton        NA
## 323:     0     0 PC 17760 135.6333     C32   Cherbourg    8   NA
##                          home.dest
##   1:                    St Louis, MO
##   2: Montreal, PQ / Chesterville, ON
##   3: Montreal, PQ / Chesterville, ON
##   4: Montreal, PQ / Chesterville, ON
##   5: Montreal, PQ / Chesterville, ON
##  ---
## 319:                 London, England
## 320:
## 321:                 London, England
## 322:                     Halifax, NS
## 323:   New York, NY / Washington, DC
```

```
titanic.dt[pclass=="1st"]
```

```
##        X pclass survived                             name    sex     age
##   1:   1    1st        1   Allen, Miss. Elisabeth Walton female 29.0000
##   2:   2    1st        1  Allison, Master. Hudson Trevor   male  0.9167
##   3:   3    1st        0    Allison, Miss. Helen Loraine female  2.0000
##   4:   4    1st        0 Allison, Mr. Hudson Joshua Crei   male 30.0000
##   5:   5    1st        0 Allison, Mrs. Hudson J C (Bessi female 25.0000
##  ---
## 319: 319    1st        0  Williams-Lambert, Mr. Fletcher   male      NA
## 320: 320    1st        1       Wilson, Miss. Helen Alice female 31.0000
## 321: 321    1st        1               Woolner, Mr. Hugh   male      NA
## 322: 322    1st        0              Wright, Mr. George   male 62.0000
## 323: 323    1st        1        Young, Miss. Marie Grice female 36.0000
##      sibsp parch   ticket     fare   cabin    embarked boat body
##   1:     0     0    24160 211.3375      B5 Southampton    2   NA
##   2:     1     2   113781 151.5500 C22 C26 Southampton   11   NA
##   3:     1     2   113781 151.5500 C22 C26 Southampton        NA
##   4:     1     2   113781 151.5500 C22 C26 Southampton       135
##   5:     1     2   113781 151.5500 C22 C26 Southampton        NA
##  ---
## 319:     0     0   113510  35.0000    C128 Southampton        NA
## 320:     0     0    16966 134.5000 E39 E41   Cherbourg    3   NA
## 321:     0     0    19947  35.5000     C52 Southampton    D   NA
## 322:     0     0   113807  26.5500         Southampton        NA
## 323:     0     0 PC 17760 135.6333     C32   Cherbourg    8   NA
##                          home.dest
##   1:                    St Louis, MO
##   2: Montreal, PQ / Chesterville, ON
```

```
##   3: Montreal, PQ / Chesterville, ON
##   4: Montreal, PQ / Chesterville, ON
##   5: Montreal, PQ / Chesterville, ON
##  ---
## 319:                 London, England
## 320:
## 321:                 London, England
## 322:                     Halifax, NS
## 323:   New York, NY / Washington, DC
```

B.3.2 key 를 이용한 선택

- `data.table` 의 `setkey()`을 이용해 key 설정
  - `setkeyv()`의 경우엔 key 를 두 개 이상 설정 가능

- `tables()`를 통해 key 지정 확인 가능

- 예시
  1. "sex"와 "pclass"를 key 로 지정
  2. "pclass"를 key 로 지정
  3. 다음 3 가지는 모두 같은 결과를 얻음

```
#1
setkeyv(titanic.dt, c("sex","pclass"))
tables()
```

```
##      NAME             NROW NCOL  MB
## [1,] DT         26,000,000    2 298
## [2,] titanic.dt      1,309   15   1
## [3,] x           1,000,000    2  12
##      COLS

## [1,] x,y

## [2,] X,pclass,survived,name,sex,age,sibsp,parch,ticket,fare,cabin,embarked,
boat,body,
## [3,] x,y

##      KEY
## [1,] y
```

```
## [2,] sex,pclass
## [3,] y
## Total: 311MB

#2
setkey(titanic.dt,pclass)
tables()

##      NAME             NROW NCOL  MB
## [1,] DT         26,000,000    2 298
## [2,] titanic.dt      1,309   15   1
## [3,] x           1,000,000    2  12
##      COLS

## [1,] x,y

## [2,] X,pclass,survived,name,sex,age,sibsp,parch,ticket,fare,cabin,embarked,
boat,body,
## [3,] x,y

##      KEY
## [1,] y
## [2,] pclass
## [3,] y
## Total: 311MB

#3
v <-  "pclass"
setkeyv(titanic.dt,v)
setkey(titanic.dt, pclass)
setkey(titanic.dt, "pclass")
tables()

##      NAME             NROW NCOL  MB
## [1,] DT         26,000,000    2 298
## [2,] titanic.dt      1,309   15   1
## [3,] x           1,000,000    2  12
##      COLS

## [1,] x,y

## [2,] X,pclass,survived,name,sex,age,sibsp,parch,ticket,fare,cabin,embarked,
boat,body,
## [3,] x,y

##      KEY
## [1,] y
## [2,] pclass
```

```
## [3,] y
## Total: 311MB
```

B.3.3 key 지정 후

- key 를 지정했으므로 **J 표현식** 을 이용하여 데이터를 추출할 수 있음

- 다음 2 가지는 같은 결과를 얻음

```
#1
titanic.dt["1st"]
```

```
##        X pclass survived                            name    sex age sibsp
##   1:   1    1st        1   Allen, Miss. Elisabeth Walton female  29     0
##   2:   3    1st        0    Allison, Miss. Helen Loraine female   2     1
##   3:   5    1st        0 Allison, Mrs. Hudson J C (Bessi female  25     1
##   4:   7    1st        1 Andrews, Miss. Kornelia Theodos female  63     1
##   5:   9    1st        1 Appleton, Mrs. Edward Dale (Cha female  53     2
##  ---
## 319: 317    1st        0     Williams, Mr. Charles Duane   male  51     0
## 320: 318    1st        1 Williams, Mr. Richard Norris II   male  21     0
## 321: 319    1st        0  Williams-Lambert, Mr. Fletcher   male  NA     0
## 322: 321    1st        1               Woolner, Mr. Hugh   male  NA     0
## 323: 322    1st        0              Wright, Mr. George   male  62     0
##      parch   ticket     fare   cabin    embarked boat body
##   1:     0    24160 211.3375      B5 Southampton    2   NA
##   2:     2   113781 151.5500 C22 C26 Southampton        NA
##   3:     2   113781 151.5500 C22 C26 Southampton        NA
##   4:     0    13502  77.9583      D7 Southampton   10   NA
##   5:     0    11769  51.4792    C101 Southampton    D   NA
##  ---
## 319:     1 PC 17597  61.3792           Cherbourg        NA
## 320:     1 PC 17597  61.3792           Cherbourg    A   NA
## 321:     0   113510  35.0000    C128 Southampton        NA
## 322:     0    19947  35.5000     C52 Southampton    D   NA
## 323:     0   113807  26.5500         Southampton        NA
##                           home.dest
##   1:                     St Louis, MO
##   2: Montreal, PQ / Chesterville, ON
##   3: Montreal, PQ / Chesterville, ON
##   4:                       Hudson, NY
##   5:             Bayside, Queens, NY
##  ---
## 319: Geneva, Switzerland / Radnor, P
## 320: Geneva, Switzerland / Radnor, P
## 321:                 London, England
```

```
## 322:                 London, England
## 323:                     Halifax, NS
```

```
#2
titanic.dt[J("1st")]
```

```
##        X pclass survived                             name    sex age sibsp
##   1:   1    1st        1   Allen, Miss. Elisabeth Walton female  29     0
##   2:   3    1st        0    Allison, Miss. Helen Loraine female   2     1
##   3:   5    1st        0 Allison, Mrs. Hudson J C (Bessi female  25     1
##   4:   7    1st        1 Andrews, Miss. Kornelia Theodos female  63     1
##   5:   9    1st        1 Appleton, Mrs. Edward Dale (Cha female  53     2
##  ---
## 319: 317    1st        0     Williams, Mr. Charles Duane   male  51     0
## 320: 318    1st        1 Williams, Mr. Richard Norris II   male  21     0
## 321: 319    1st        0  Williams-Lambert, Mr. Fletcher   male  NA     0
## 322: 321    1st        1               Woolner, Mr. Hugh   male  NA     0
## 323: 322    1st        0              Wright, Mr. George   male  62     0
##      parch   ticket     fare   cabin    embarked boat body
##   1:     0    24160 211.3375      B5 Southampton    2   NA
##   2:     2   113781 151.5500 C22 C26 Southampton        NA
##   3:     2   113781 151.5500 C22 C26 Southampton        NA
##   4:     0    13502  77.9583      D7 Southampton   10   NA
##   5:     0    11769  51.4792    C101 Southampton    D   NA
##  ---
## 319:     1 PC 17597  61.3792           Cherbourg        NA
## 320:     1 PC 17597  61.3792           Cherbourg    A   NA
## 321:     0   113510  35.0000    C128 Southampton        NA
## 322:     0    19947  35.5000     C52 Southampton    D   NA
## 323:     0   113807  26.5500         Southampton        NA
##                           home.dest
##   1:                     St Louis, MO
##   2: Montreal, PQ / Chesterville, ON
##   3: Montreal, PQ / Chesterville, ON
##   4:                       Hudson, NY
##   5:             Bayside, Queens, NY
##  ---
## 319: Geneva, Switzerland / Radnor, P
## 320: Geneva, Switzerland / Radnor, P
## 321:                 London, England
## 322:                 London, England
## 323:                     Halifax, NS
```

- key 를 활용하여 집계를 할 수 있음

```
titanic.dt[J("1st"),mean(survived)]
```

```
## [1] 0.619195
```

B.4. Grouping 연산

- grouping 하여 연산하는 `data.table` 의 형식 : **DT[, 연산식, by='variable']**

- reserved keyword
  - **.N**
    - by 에 의해 grouping 될 때 매칭된 변수의 행의 개수를 나타냄
    - 매칭이 되지 않으면 NA 나 0 으로 출력
  - **.SD**
    - Subset of x's Data for each group, excluding any columns used in by(or keyby)
    - **.SDcols** : .SD 에 특정한 변수 이름을 지정하여 해당 변수로만 이루어진 새 .SD 를 구성하는 것임
    - **nrow(.SD)** : **J** 와 by 로 분류된 변수를 포함하는 행의 개수
    - 단, 만약 by 변수가 sex 라면 "male"과 "female"은 분리되어 계산됨
- 예시

1. 좌석등급별 생존률

```
titanic.dt[, mean(survived), by="pclass"]
```

```
##    pclass        V1
## 1:    1st 0.6191950
## 2:    2nd 0.4296029
## 3:    3rd 0.2552891
```

2. 남녀 생존률

```
titanic.dt[,lapply(.SD,mean), by=sex, .SDcols=c("survived")]
```

```
##       sex  survived
## 1: female 0.7274678
## 2:   male 0.1909846
```

```
titanic.dt[,mean(survived),by=sex]
```

```
##       sex        V1
## 1: female 0.7274678
## 2:   male 0.1909846
```

3. 1 등급 승객 중 성별 생존률

```
titanic.dt[J("1st"), mean(survived), by="sex"]
```

```
##       sex        V1
## 1: female 0.9652778
## 2:   male 0.3407821
```

4. 다수 group key 지정(sex, boat)

```
titanic.dt[J("2nd"), mean(survived), by="sex,boat"]
```

```
##        sex boat  V1
##  1: female   10 1.0
##  2: female   11 1.0
##  3: female   13 1.0
##  4: female   12 1.0
##  5: female   14 1.0
##  6: female    9 1.0
##  7: female      0.4
##  8: female   16 1.0
##  9: female    4 1.0
## 10:   male      0.0
## 11:   male   13 1.0
## 12:   male   11 1.0
## 13:   male    9 1.0
## 14:   male   14 1.0
## 15:   male   10 1.0
## 16:   male    4 1.0
## 17:   male   15 1.0
## 18:   male    B 1.0
## 19:   male    D 1.0
## 20:   male    7 1.0
## 21:   male   12 0.0
##        sex boat  V1
```

5. 1 등급 승객의 counting

```
titanic.dt[,length(which(pclass=="1st"))]
```

```
## [1] 323
```

```
titanic.dt[pclass=="1st",.N]
```

```
## [1] 323
```

6. 등급별, 성별 counting
    1. 1 등급 중 성별 counting
    2. 등급별 counting

```
#1
titanic.dt[pclass=="1st", .N, by="sex"]
```

```
##       sex   N
## 1: female 144
## 2:   male 179
```

```
#2
titanic.dt[, .N ,by="pclass"]
```

```
##    pclass   N
## 1:    1st 323
## 2:    2nd 277
## 3:    3rd 709
```

7. 1 등급 승객 중 성별 20 세 이상 성인 비율

```
titanic.dt[J("1st"), length(which(age>20))/.N, by="sex"]
```

```
##       sex        V1
## 1: female 0.8125000
## 2:   male 0.7877095
```

```
titanic.dt[J("1st"), length(which(age>20))/nrow(.SD), by="sex"]
```

```
##       sex        V1
## 1: female 0.8125000
## 2:   male 0.7877095
```

8. 성별, 등급, 나이별 생존률

```
titanic.dt[,":="(isminor,"adult")]
titanic.dt[age<15,":="(isminor,"child")]
```

```
survived_pclass_sex_isminor <- titanic.dt[,list(cntsurv=length(which(survived
 == 1)), cntdie=length(which(survived == 0))), by=list(pclass, sex, isminor)]
[,list(psurvived=cntsurv/(cntsurv + cntdie)),by=list(pclass, sex, isminor)]

survived_pclass_sex_isminor

##     pclass    sex isminor  psurvived
##  1:    1st female   adult 0.97183099
##  2:    1st female   child 0.50000000
##  3:    1st   male   child 1.00000000
##  4:    1st   male   adult 0.32183908
##  5:    2nd female   adult 0.86813187
##  6:    2nd female   child 1.00000000
##  7:    2nd   male   adult 0.08805031
##  8:    2nd   male   child 0.91666667
##  9:    3rd female   adult 0.49450549
## 10:    3rd female   child 0.47058824
## 11:    3rd   male   adult 0.13716814
## 12:    3rd   male   child 0.31707317

survived_pclass_sex_isminor$sex_age <- apply(survived_pclass_sex_isminor[,lis
t(sex,isminor)], 1, paste, collapse="_")

survived_pclass_sex_isminor

##     pclass    sex isminor  psurvived      sex_age
##  1:    1st female   adult 0.97183099 female_adult
##  2:    1st female   child 0.50000000 female_child
##  3:    1st   male   child 1.00000000   male_child
##  4:    1st   male   adult 0.32183908   male_adult
##  5:    2nd female   adult 0.86813187 female_adult
##  6:    2nd female   child 1.00000000 female_child
##  7:    2nd   male   adult 0.08805031   male_adult
##  8:    2nd   male   child 0.91666667   male_child
##  9:    3rd female   adult 0.49450549 female_adult
## 10:    3rd female   child 0.47058824 female_child
## 11:    3rd   male   adult 0.13716814   male_adult
## 12:    3rd   male   child 0.31707317   male_child
```

C. 데이터 시각화 단계 가이드

C.1. 기본 도표 및 그래프

수집된 자료를 효과적으로 정리, 요약하기 위해서 적절한 도표나 그래프를 사용하여 시각화할 필요가 있다. 이러한 작업은 많은 경우 수치를 이용하는 것보다 자료에 내포된 정보를 보다 쉽고 빠르게 파악할 수 있게 하며, 탐색적 자료 분석 단계에서는 이러한 것들이 필수적이라 할 수 있다.
이 절에서는 그래프를 이용하여 자료를 시각화 및 요약하는 방법에 대해서 학습한다.

자료의 시각화는 대칭 혹은 비대칭의 정도, 대부분의 자료로부터 동떨어진 이상점(outliers)의 유무, 그리고 상대적으로 많은 자료가 분포되어 있는 봉우리의 위치 등 자료의 대략적인 분포형태 및 특성 등을 파악하게 한다.

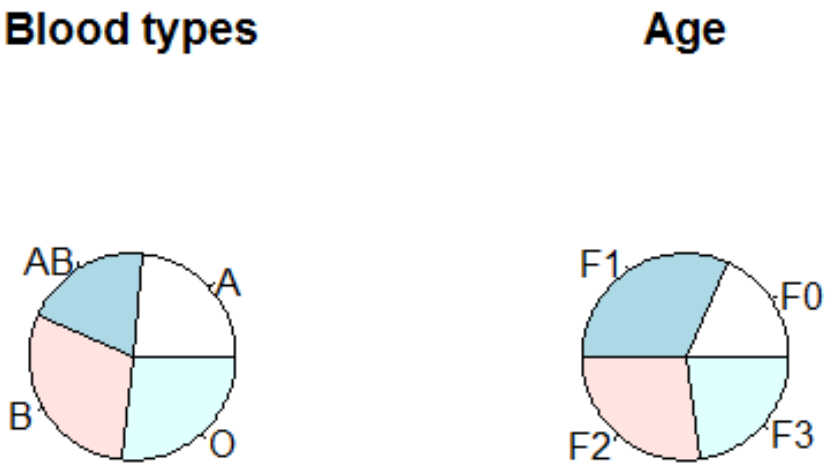
- 실습
  - 크기가 30 명인 집단의 혈액형(A, B, AB, O)을 조사한 자료를 임의로 생성하여 이 자료를 Blood.Type 이라는 이름으로 저장하고, 도수분포표와 상대도수분포표를 작성해보자.

```
set.seed(1)
Blood.Type <- sample(x=1:4, size=30, replace=TRUE)
Blood.Type <- c("A", "B", "AB", "O")[Blood.Type]
Blood.Type <- factor(Blood.Type)
Blood.Type

##  [1] B  B  AB O  A  O  O  AB AB A  A  A  AB B  O  B  AB O  B  O  O  A
 AB
## [24] A  B  B  A  B  O  B
## Levels: A AB B O

table(Blood.Type)                          # frequency table

## Blood.Type
##  A AB  B  O
##  7  6  9  8

table(Blood.Type)/length(Blood.Type)       # relative frequency table

## Blood.Type
##         A        AB         B         O
## 0.2333333 0.2000000 0.3000000 0.2666667
```

- MASS 패키지에 있는 quine 데이터는 호주 NSW 의 학생들에 관련된 조사 자료이다.
  이 자료와 Blood.Type 데이터로 도수분포표와 몇 가지 기초 그래프를 그려보자.

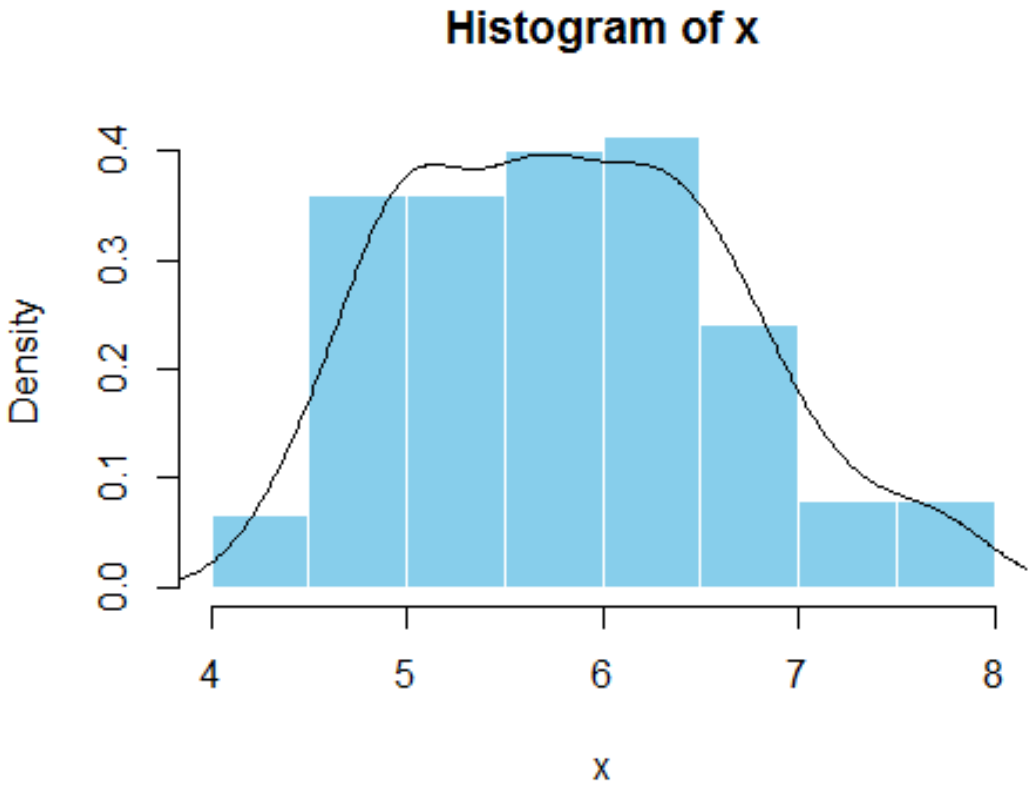
```
library(MASS)
attach(quine)
table(Age)

## Age
## F0 F1 F2 F3
## 27 46 40 33

par(mfrow=c(2,2))
barplot(table(Blood.Type), col=3, xlab="Blood types", ylab="Frequency")
barplot(table(Blood.Type)/length(Blood.Type), col=3, xlab="Blood types",
ylab="Relative frequency")
barplot(table(Age), col=7, xlab="Age", ylab="Frequency")
barplot(table(Age)/length(Age), col=7, xlab="Age", ylab="Relative frequen
cy")
```

```r
par(mfrow=c(1,2))
pie(table(Blood.Type))
title("Blood types")
pie(table(Age))
title("Age")
```

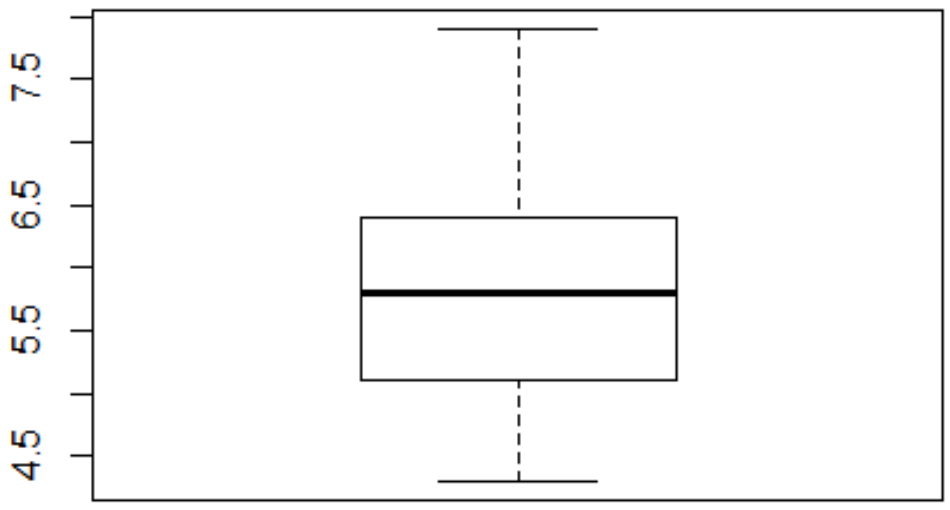


- 다음은 R 의 built-in 데이터셋인 `iris` 에 포함된 Sepal.Length 의 분포의 형태를 상자그림과 히스토그램을 이용해 개략적으로 알아보는 예이다.

```r
data(iris)
head(iris)

##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa

x <- iris$Sepal.Length
hist(x, prob=TRUE, col="skyblue", border="white")
lines(density(x))
```




```
boxplot(x)          # boxplot
```



C.2. 자료 분포의 요약

주어진 자료 속에 내재해 있는 구조를 파악하여 필요한 정보를 얻기 위해서는 자료의 분포를 파악하는 것이 분석의 기본이다.
이 절에서는 자료의 특성을 수치화하여 분포에 대해 요약하는 방법을 익힌다.

자료 분포의 특성은 1) 분포의 중심 위치, 2) 자료값의 분포 내 상대적 위치, 3) 자료 분포의 산포 정도 등을 통해 대부분 설명된다.

- 자료값: $x_1, x_2, \cdots, x_n$
- 자료 분포의 중심 위치 (central location)
  - 평균(mean): 자료값의 합을 자료의 개수로 나눈 값

$$\bar{x} = \frac{x_1 + x_2 + \cdots + x_n}{n} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

```
x <- c(1, 2, 4, 5, 7)
mean(x)
```

```
## [1] 3.8
```

```
+ 계산이 간편하고 직관적으로 이해하기 쉬움
+ 자료에 이상점(outlier)이 포함된 경우 문제 발생. 특히 자료의 개수가 많지 않은 경우 유의할 필요가 있음
```

```
x <- c(1, 2, 4, 5, 70)   # 70: 오타에 의한 이상점
mean(x)
```

```
## [1] 16.4
```

  - 중앙값(median): 자료를 크기 순으로 늘어놓았을 때 중앙에 위치해 있는 값. 이상점이 포함되어 있더라도 거의 영향을 받지 않음

```
x <- c(1, 2, 4, 5, 70)   # 70: 오타에 의한 이상점
median(x)
```

```
## [1] 4
```

  - 절사평균(trimmed mean): 자료값 중 양쪽 꼬리 부분의 극단값을 일정한 비율만큼 제거하고 남은 값들로 구한 평균. 평균이 가지고 있는 장점을 일부

공유하기 때문에 중앙값에 비해 좋은 이론적 성질을 가지고 있음

- 다음은 앞에서 다룬 데이터셋인 iris 에 포함된 변수 Sepal.Length 의 다양한 중심위치를 구하는 예이다.

```
x <- iris$Sepal.Length
mean(x)             # 평균

## [1] 5.843333

median(x)           # 중앙값

## [1] 5.8

mean(x, trim=0.1)   # 절사평균: 크기순으로 배열하여 양쪽 극단의 10%(= 0.1) 값들을 버린 후 평균

## [1] 5.808333
```

- 자료 분포 내 상대적 위치 (relative location)
  - 분위수(quantile): 자료값을 크기 순으로 늘어놓았을 때 작은 값으로부터 특정한 비율의 위치에 해당하는 값
  - 사분위수(quartile): 25%, 50%, 75%에 해당하는 분위수들. 25%에 해당하는 값을 제 1 사분위수, 75%에 해당하는 값을 제 3 사분위수라 함
  - 아래 코드는 quantile()함수를 이용해 자료값의 0.1, 0.25, 0.5, 0.75, 0.9 의 비율의 위치에 해당하는 값을 구하는 예

```
quantile(x, probs=c(0.1, 0.25, 0.5, 0.75, 0.9))

## 10% 25% 50% 75% 90%
## 4.8 5.1 5.8 6.4 6.9
```

- 자료 분포의 산포 정도 (scale)
  - 분산(variance): 자료값에서 평균값을 뺀 편차(deviation)의 제곱값의 평균에 해당하는 값

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

-

- 표준편차(standard deviation): 분산의 제곱근 $s = \sqrt{s^2}$
- 범위(range): 자료의 최댓값과 최솟값의 차이
- 사분위수 범위(interquartile range, IQR): 자료의 제 3 사분위수와 제 1 사분위수의 차이. 자료에 이상점이 포함된 경우 등에 유용하게 활용 가능. 자료값의 분포가 정규분포인 경우 IQR 값은 표준편차의 1.35 배 정도임

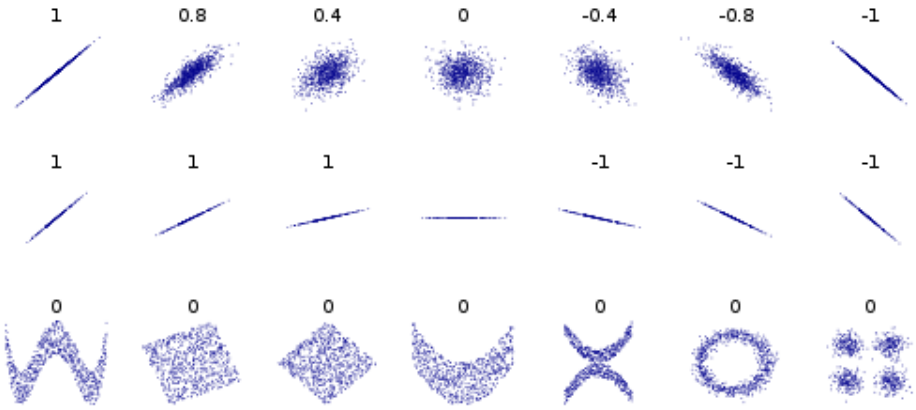
```
sd(x)             # 표준편차
## [1] 0.8280661
var(x)            # 분산
## [1] 0.6856935
diff(range(x))    # 범위
## [1] 3.6
IQR(x)            # 사분위수 범위 = quantile(x, 0.75) - quantile(x, 0.25)
## [1] 1.3
IQR(rnorm(10000)) # 1.35 정도
## [1] 1.350945
```

자료에 포함된 변수가 두 개 이상인 경우 서로 다른 변수 간의 종속 관계를 분석해 주로 예측적 분석에 활용하게 된다.
다변수 자료의 분석에서 가장 기본이 되는 두 변수 간의 연관성을 알아보는 것으로 시작하자.

- 자료값: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- 변수 간 종속성
  - Pearson 의 상관계수(correlation coefficient)

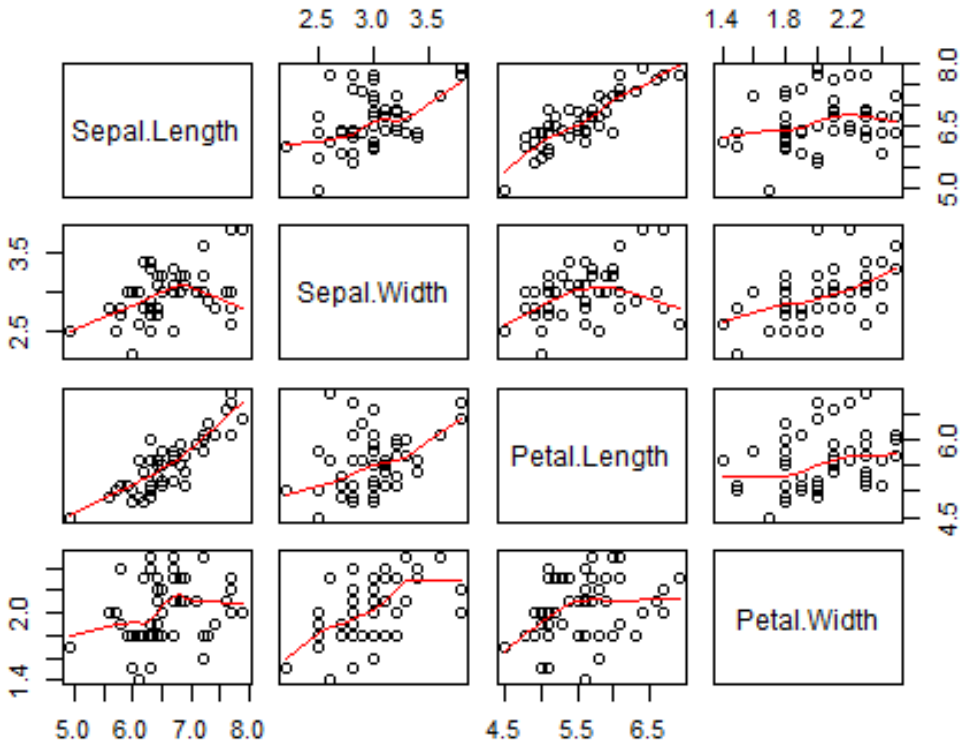
- 두 변수 간의 선형(linear) 종속성을 재는 측도로서 -1 ~ +1 사이에서 값을 가짐

- 절대값이 1 에 가까울수록 두 변수 간의 선형 종속성이 강함을 의미. 변수 간에 명백한 종속관계가 있더라도 선형 종속성을 갖지 않으면 상관계수 값은 0 에 가까운 값이 됨



- 변수의 크기 간 연관성을 분석하면 선형 종속성을 넘어 단조적 관계성(monotonicity)에 대한 분석 가능.
  - Kendall τ(tau): concordance-discordance 개념 이용
  - Spearman 의 순위상관계수 ρ(rho): 순위 자료의 상관계수
  - 아래 코드는 `pairs()`함수를 이용해 `iris` 데이터에 포함된 변수들에 대한 산점도 행렬을 작성해 변수 간 연관성을 시각적으로 확인하고, 변수 간 상관계수 값으로 구성된 상관계수 행렬을 구하는 예임. 품종(Species)이 virginica 인 경우만 추출해 분석에 사용했음에 유의

```
pairs(iris[iris$Species=="virginica",-5], panel = panel.smooth)
```




```r
cor(iris[iris$Species=="virginica",-5]) # Pearson 상관계수 (default)
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    1.0000000   0.4572278    0.8642247   0.2811077
## Sepal.Width     0.4572278   1.0000000    0.4010446   0.5377280
## Petal.Length    0.8642247   0.4010446    1.0000000   0.3221082
## Petal.Width     0.2811077   0.5377280    0.3221082   1.0000000
```

```r
cor(iris[iris$Species=="virginica",-5], method="kendall")
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    1.0000000   0.3070869    0.6698154   0.2189892
## Sepal.Width     0.3070869   1.0000000    0.2912823   0.4186479
## Petal.Length    0.6698154   0.2912823    1.0000000   0.2714149
## Petal.Width     0.2189892   0.4186479    0.2714149   1.0000000
```

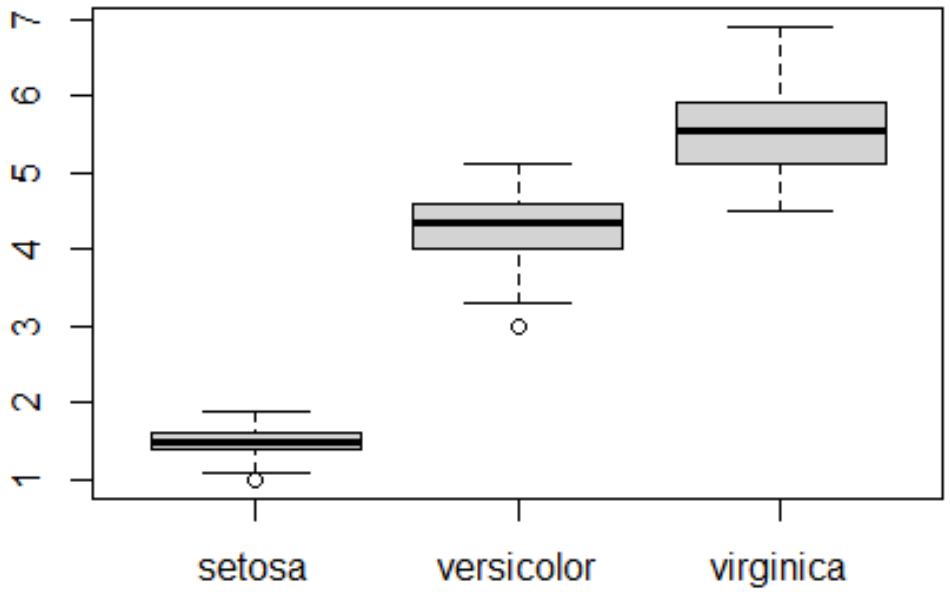
```r
cor(iris[iris$Species=="virginica",-5], method="spearman")
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    1.0000000   0.4265165    0.8243234   0.3157721
## Sepal.Width     0.4265165   1.0000000    0.3873587   0.5443098
## Petal.Length    0.8243234   0.3873587    1.0000000   0.3629133
## Petal.Width     0.3157721   0.5443098    0.3629133   1.0000000
```

- 연속형 변수와 범주형 변수 간의 종속성(연관성)
  - 상자그림(boxplot)을 이용하면 간단하면서도 효과적으로 시각적 정보를 얻을 수 있음

  - 아래 코드는 iris 데이터의 품종(Species)별 꽃잎 길이(Petal.Length)의 분포를 나란히 상자그림으로 그려 품종 특성을 비교하는 예임

```
boxplot(Petal.Length ~ Species, data = iris, col = "lightgray")
```



C.3. 모집단, 표본, 표본분포

통계적 추론, 특히 예측 관점에서 분석을 실시하려면 표본분포에 대한 이해가 필수적이다.
이를 위한 기본 개념들을 익혀보자.

- 모집단, 모수
  - 관심의 대상이 되는 집단 전체 혹은 개체를 전부 다 모은 것

- 모집단을 대상으로 측정한 모든 값의 분포(distribution)가 궁극적 관심의 대상

- 모집단 분포의 특성을 요약한 수치를 모수(parameter)라 함

- 모수의 값은 알려지지 않은 경우가 대부분이며, 따라서 추론의 대상임

- 표본, 통계량
  - 모집단의 일부, 자료

  - 모수의 표본 버전을 통계량(statistic)이라 함

  - 통계량의 예측 분포를 표본분포(sampling distribution)라 함
- 가장 자주 접하게 되는 대표적 통계량인 표본평균의 표본분포에 대해 알아보자.
  - **중심극한정리 (central limit theorem)** : 평균이 μ이고 분산이 σ^2인 모집단에서 추출한 표본 $X_1, X_2, \dots, X_n$을 추출해 계산한 표본평균 $\overline{X} = n^{-1}\sum_{i=1}^{n} X_i$의 표본분포는 n이 충분히 클 때 정규분포 $N\left(\mu, \frac{\sigma^2}{n}\right)$와 비슷해진다.
    - 모집단의 분포가 어떤 형태이든 평균과 분산이 존재하는 분포이기만 하면 항상 성립

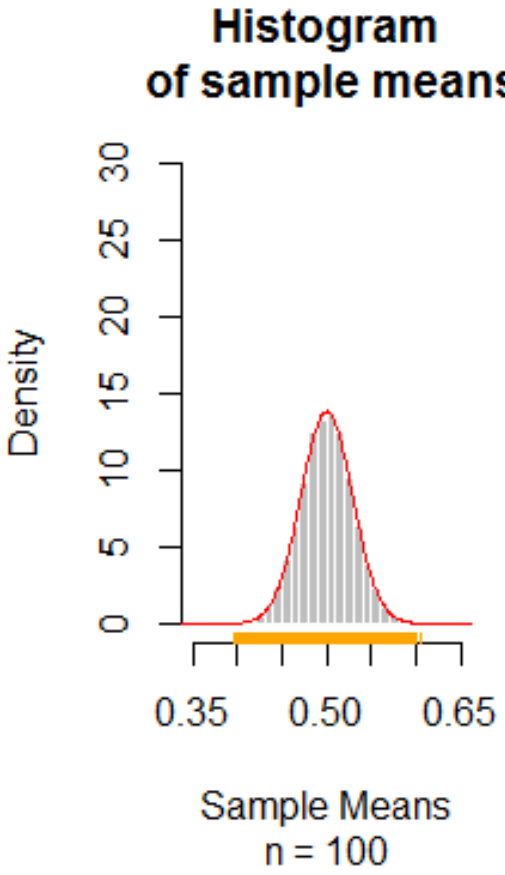
  - 예측적 의미: 평균 μ에서 봉우리를 가지고 분산에 의해 산포의 정도가 결정되는 정규분포 곡선의 특성을 고려하면, 표본의 크기가 클 수록 표본평균 $\overline{X}$의 값이 참값인 μ 근처에서 값을 가지는 경향성이 강해짐을 예측할 수 있음

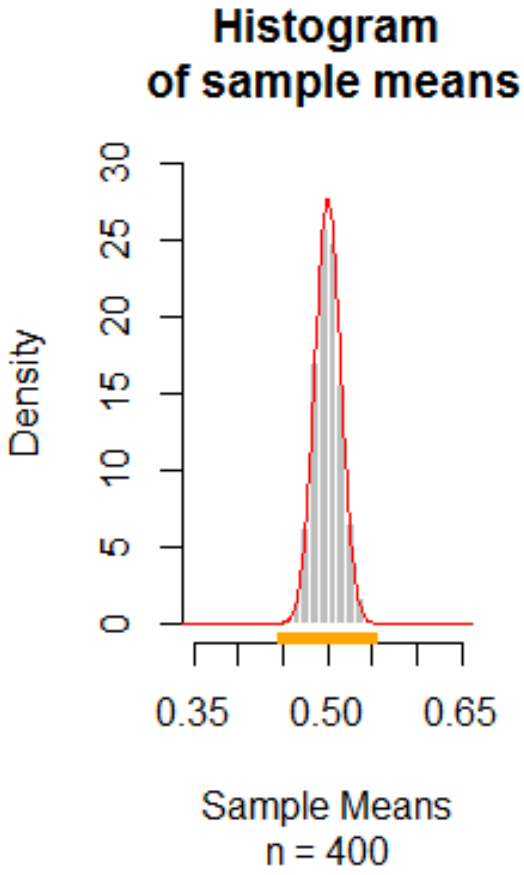
  - 아래 코드는 모의실험을 통해 중심극한정리를 실증한 것임
    - [1] 0 과 1 사이에서 정의된 균일분포 Unif(0,1) 에서 난수 n개를 발생시킨 후 표본평균의 값을 계산

    - [2] [1]를 10,000 번 반복해 얻은 10,000 개의 표본평균값들로 히스토그램을 그림

    - [3] 히스토그램과 중심극한정리에 따른 이론적인 표본분포(즉, 정규분포)와 비교

```r
mu <- 1/2; sig.sq <- 1/12    # Unif(0,1)의 평균, 분산

par(mfrow=c(1,2))

n <- 100
X.bar <- NULL
for ( k in 1:10000) {
  X <- runif(n)
  X.bar <- c(X.bar, mean(X))
}
hist(X.bar, prob=T, col="gray", border="white",
     xlim=c(0.35, 0.65), ylim=c(0, 30),
     xlab="Sample Means", main="Histogram\n of sample means", sub=paste("
n =",n))
rug(X.bar, col="orange")
z <- seq(from=0, to=1, by=0.001)
lines(z, dnorm(z, mean=mu, sd=sqrt(sig.sq/n)), col=2)

n <- 400
X.bar <- NULL
for ( k in 1:10000) {
  X <- runif(n)
  X.bar <- c(X.bar, mean(X))
}
hist(X.bar, prob=T, col="gray", border="white",
     xlim=c(0.35, 0.65), ylim=c(0, 30),
     xlab="Sample Means", main="Histogram\n of sample means", sub=paste("
n =",n))
rug(X.bar, col="orange")
z <- seq(from=0, to=1, by=0.001)
lines(z, dnorm(z, mean=mu, sd=sqrt(sig.sq/n)), col=2)
```





- 위 모의실험 결과에 대한 해석
  - 히스토그램과 이론적 정규분포 곡선(빨간색 실선)이 놀랍도록 비슷. 중심극한정리가 현실에 의미가 있는 이론임을 확인.
  - 실현된 표본평균 10,000 개의 값이 참값 1/2 에서 그리 멀지 않은 구간 내에 집중되어 있음
  - 표본평균을 평균 μ에 대한 추정치로 삼는다면 매우 정확한 값을 얻게 될 가능성이 높음을 예상할 수 있게 하는 결과임
  - 또한, 집중된 정도가 표본 크기가 100 일 때 보다 400 일 때 더 강한 것으로 보아 표본 크기가 더 큰 경우에 참값에 가까운 값을 얻을 가능성이 높아짐을 예상할 수 있음

C.4. ggplot2 를 이용한 고급 시각화

- **ggplot2** 는 데이터 시각화를 위한 그래픽 라이브러리로는 현재까지 가장 많이 사용되고 있음

- 명령어 형식의 ggplot2 의 특징
  - ggplot2(Grammar of Graphic)의 약자와 같이 grammar 기반의 명령어를 제공
  - 문장 형성을 통해 사용자는 데이터 시각화를 가능하게 함

- **ggplot2** 의 활용 예를 통해 데이터 시각화의 컴포넌트와 실제 데이터와의 연결을 이해함

C.4.1 Example 을 통한 기초 이해

1. iris 데이터 사용 예제

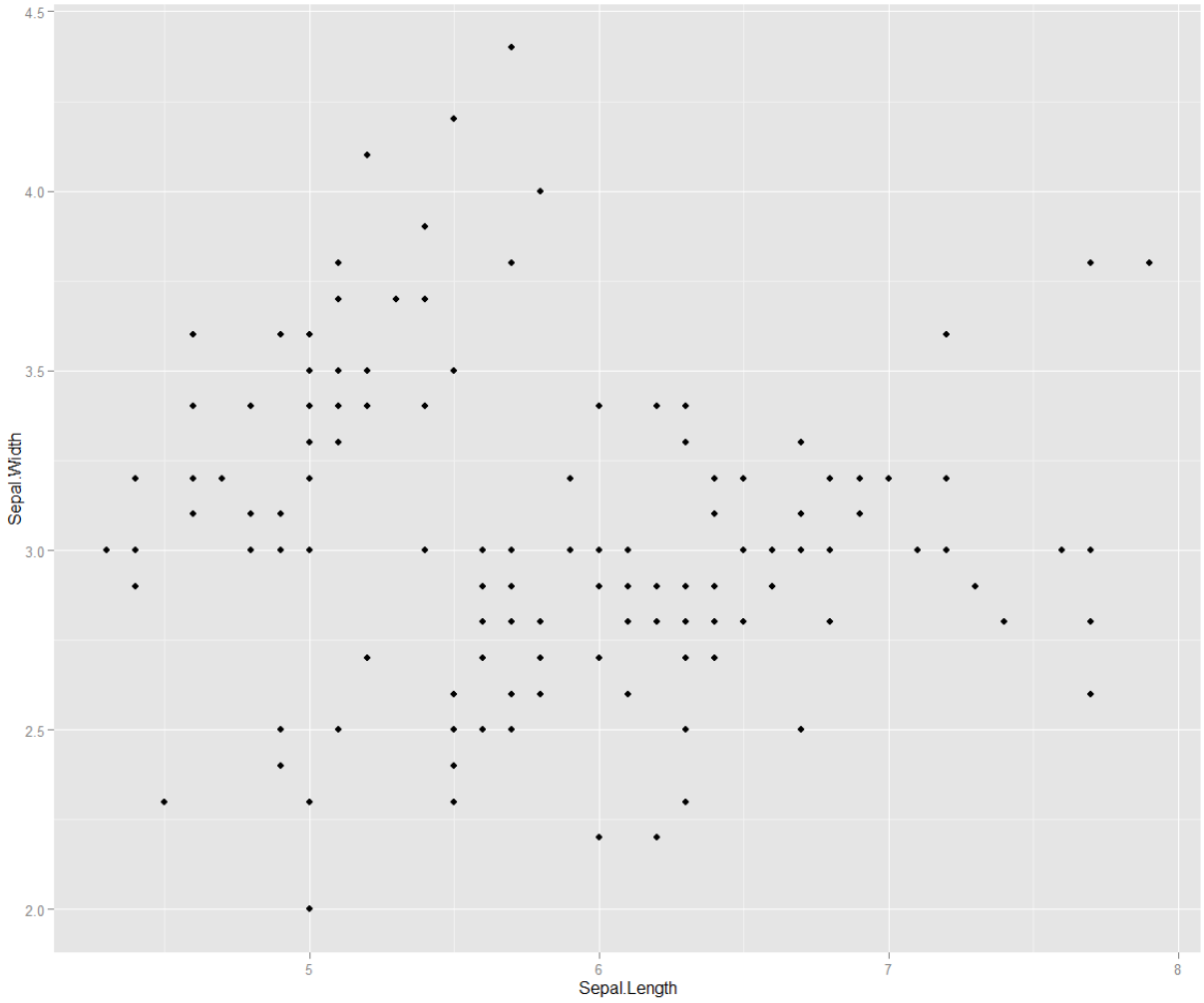
```
str(iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1
1 1 1 1 ...
```

```
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
library(ggplot2)
ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width)) + geom_point()
```



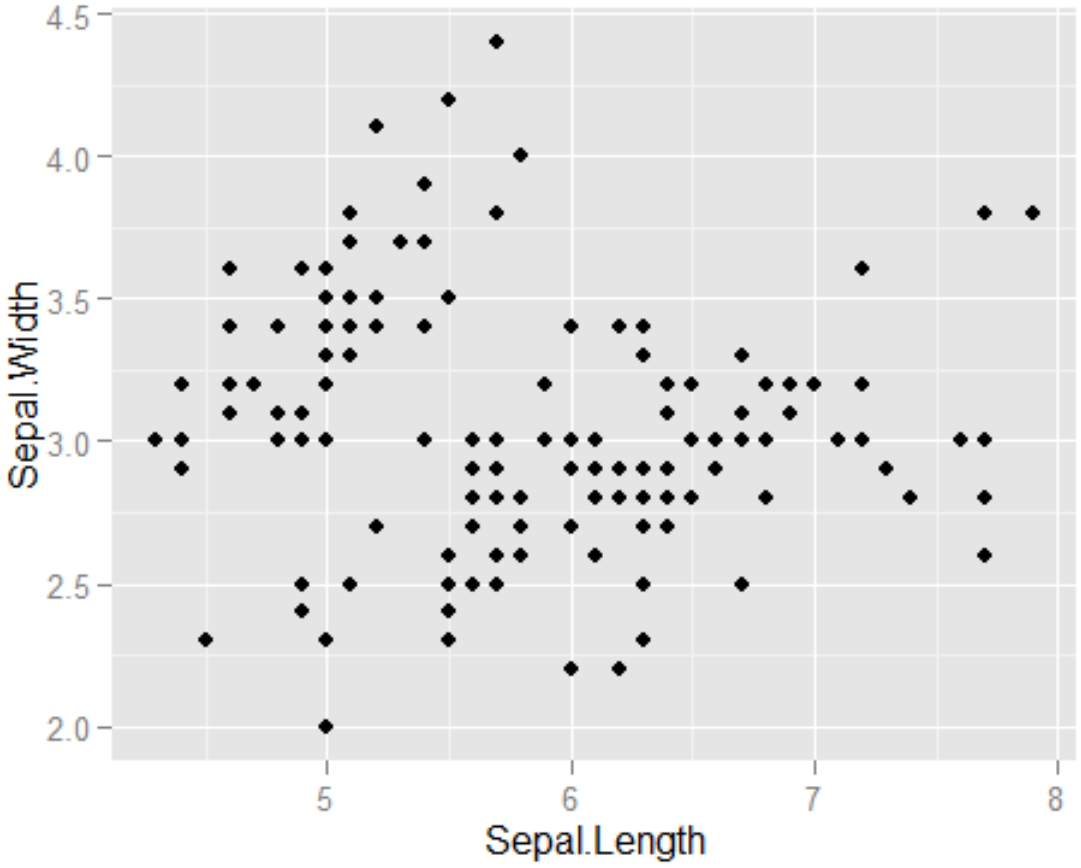
2. 기초 구조의 이해

- 함수 `ggplot()` 안에서 데이터와 변수를 정의함

- 일종의 백지 위에 그림을 그릴 대상을 정의하는 것, 즉 데이터 시각화를 위한 셋팅 부분임

- 그 이후 어떤 시각화를 할지의 구체적인 데이터 시각화 컴포넌트 대상, 통계모형, 판넬(panels) 등을 백지 상에 레이어 형태로 그림을 그리는 것임

```
myplot <- ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width))
myplot + geom_point()
```

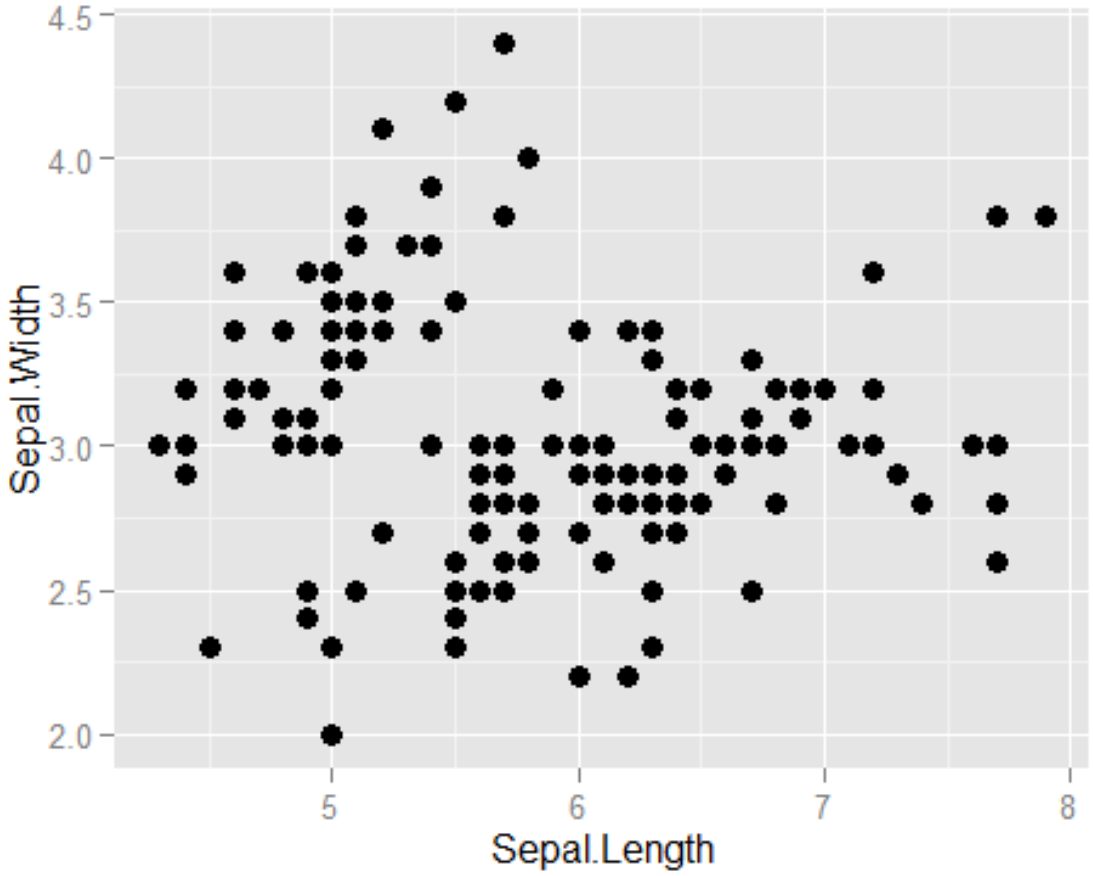


3. Further examples

(1) Increase the size of points

- 위에서 보다 점(point)의 크기가 커진 것을 확인 할 수 있음

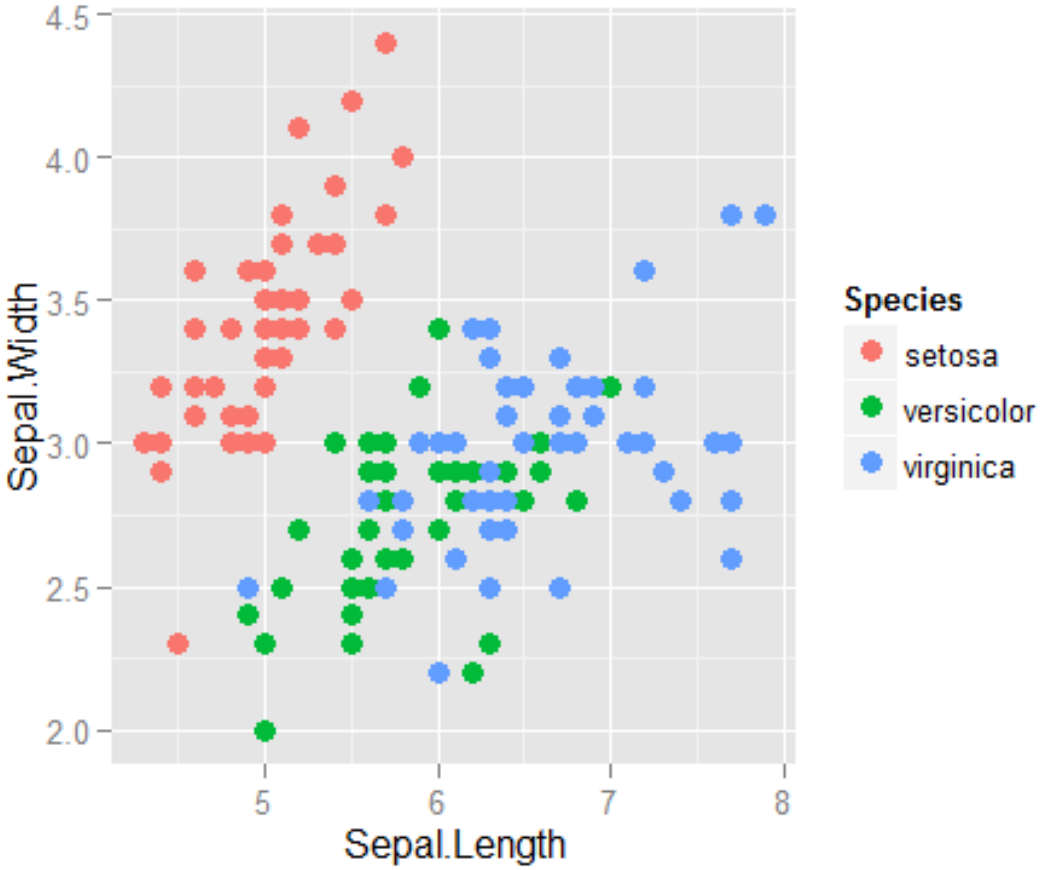
```
ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width)) + geom_point(size
 = 3)
```



(2) Add some color

- 종류(Species)에 따라 point 에 색이 각각 부여됨

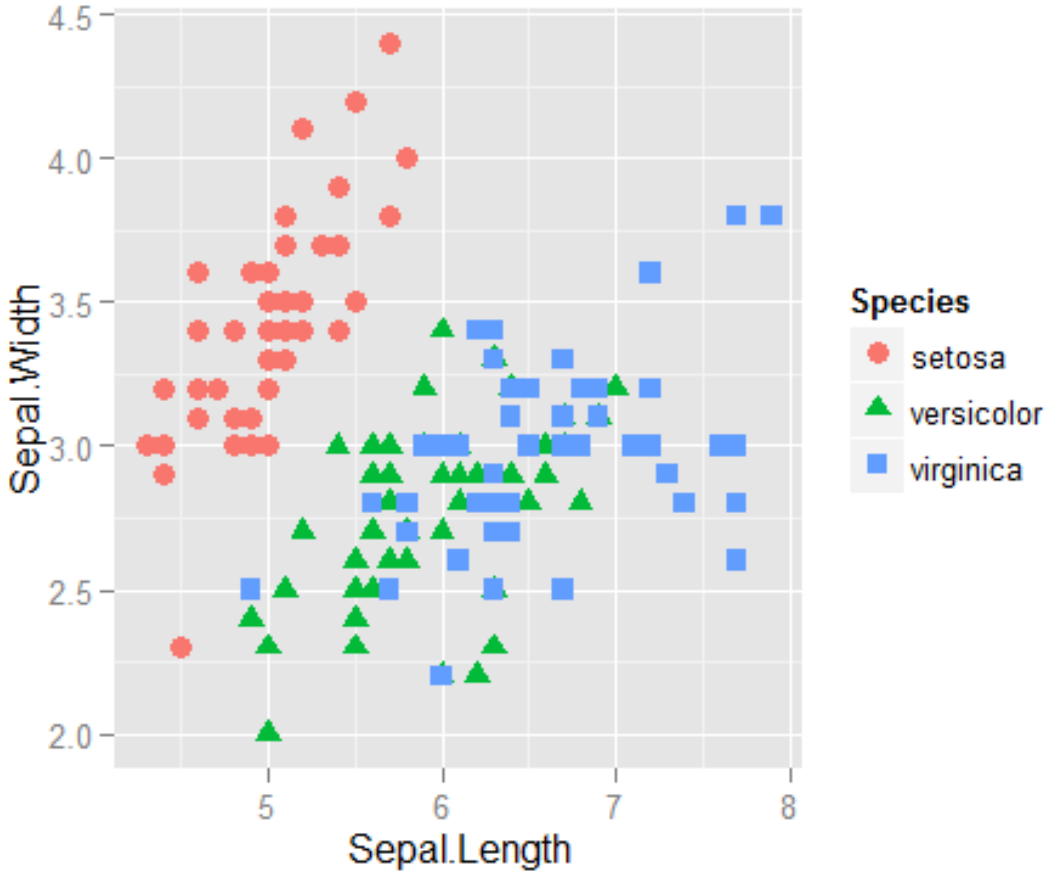
```
ggplot(iris, aes(Sepal.Length, Sepal.Width, color = Species)) +
    geom_point(size = 3)
```



(3) Differentiate points by shape

- 종류(Species)에 따라 point 에 모양이 각각 부여됨

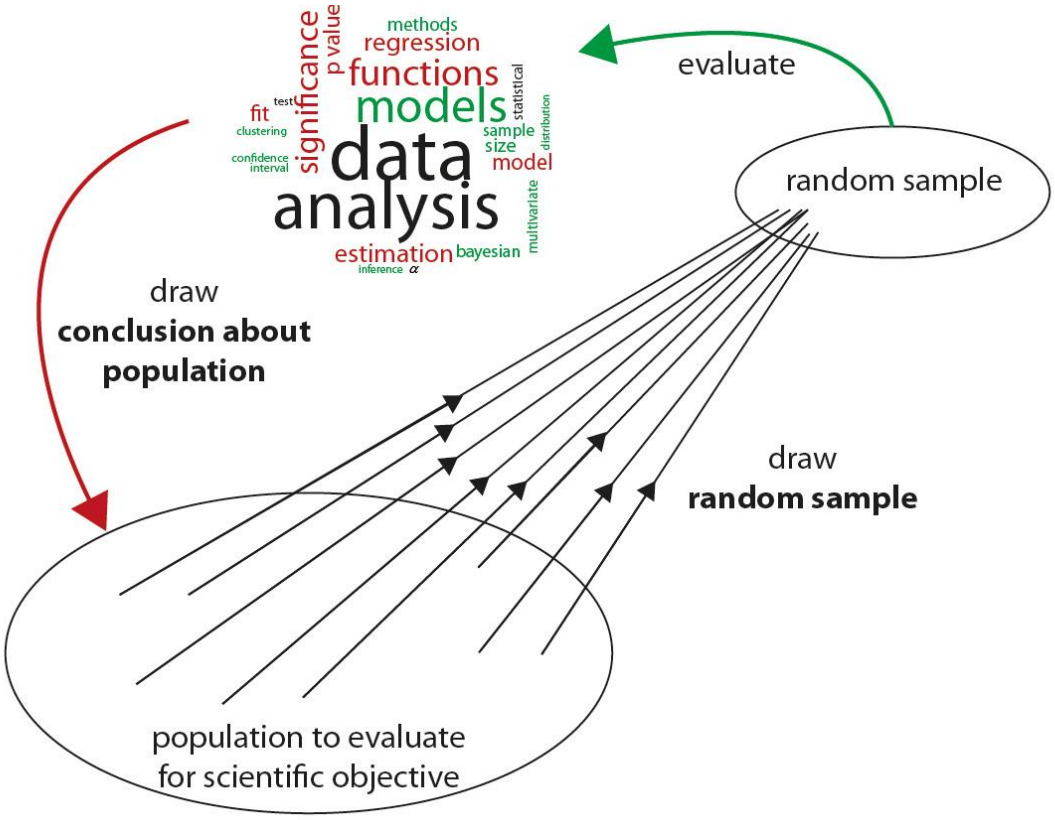
```
ggplot(iris, aes(Sepal.Length, Sepal.Width, color = Species)) +
    geom_point(aes(shape = Species), size = 3)
```


4.5
4.0
3.5
3.0
2.5
2.0
Sepal.Width
5
6
7
8
Sepal.Length
Species
setosa
versicolor
virginica

D. 프로세스별 데이터셋 구성 및 분석단계 가이드

D.1. 예측 모형에 기반한 분석

통계적 추론은 표본에 의한 부분적 정보를 이용해 모집단에 대한 의사결정을 할 수 있도록 해 준다.
아래 그림은 일련의 통계적 분석의 과정을 도식화한 것이다.



(출처 http://www.dkfz.de/de/biostatistics/)

이 절에서는 통계적 추론 결과를 예측(prediction)에 이용하는 분석 방법 중 가장 기본이 되는 회귀분석, 로지스틱 회귀분석 등을 익힌다.

D.1.1 회귀분석

서로 다른 변수 간의 함수 관계를 규명하는 통계적 분석 절차를 **회귀분석(regression analysis)**이라 한다.
즉, $y = f(x)$ 형태의 변수 간의 관계식을 얻어 변수 간 관계를 규명하고, 더 나아가 보조정보(x)를 이용해 y의 예측에 활용하는 것을 주요 목적으로 하는 분석법으로 예측 방법론의 핵심이 되는 기법이다.

- 회귀모형: $y = f(x) + e$
  - y: 반응변수(response variable), 종속변수(dependent v.)
  - x: 설명변수(explanatory v.), 독립변수(independent v.), 공변량(covariates), 예측변수(predictor)
  - f(x): 회귀함수(regression function), x의 값이 주어졌을 때 y의 조건부평균[기댓값]
  - $e \sim N(0, \sigma^2)$: 오차(error)항으로 정규분포를 가정함
- 회귀분석의 주요 목적
  - 오차 e의 방해를 극복하고 관심 대상 변수인 x와 y 사이의 관계를 규명함
  - 주어진 x값을 이용해 y값을 예측
- 선형회귀모형(linear regression model)
  - 회귀함수 f가 선형식임을 가정: $f(x) = \beta_0 + \beta_1 x$, 독립변수가 하나인 경우
  - 회귀계수 β_1의 의미: x의 값이 한 단위 증가함에 따라 y의 값에 생기는 평균 변화량 (partial regression coefficients)
  - 상관분석 결과에 의해 정당성을 확보할 수 있고, 해석이 용이하며 추론이 쉬움
  - 다만 회귀함수가 실제로 선형이 아닌 경우 예측력이 상당히 떨어질 수 밖에 없음
  - 다항회귀, 비선형회귀 혹은 비모수적함수추정 기법 등을 이용해 해결 가능

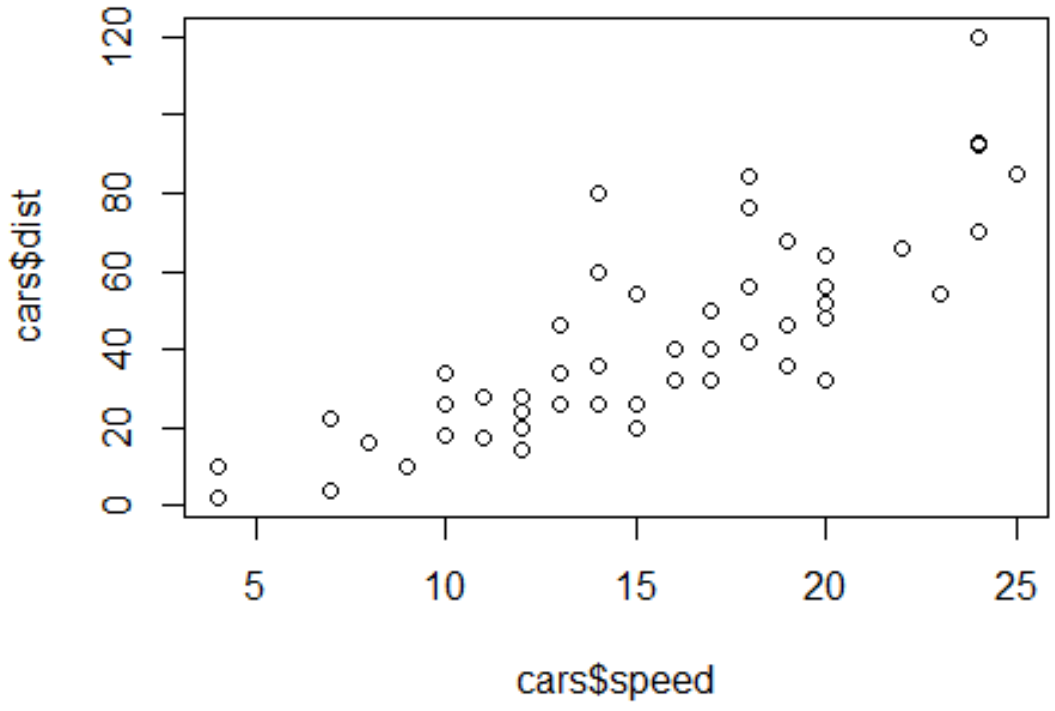
회귀분석의 일반적 분석 절차를 예제를 통해 알아보자.

1. 산점도 작성
   - 분석의 대상이 되는 변수 간의 연관성을 시각적으로 확인

- 회귀분석을 실시하기 전에 실행되어야 할 가장 기본적인 분석 절차

- 아래 코드는 R 의 built-in dataset 인 cars 데이터에 포함된 두 변수 speed 와 dist 간의 산점도를 그린 예임. 자동차의 속력(speed)과 제동 거리(dist) 사이의 관계를 시각적으로 확인 가능함

```
data(cars)
summary(cars)

##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00

plot(cars$speed, cars$dist)
```



2. 모형 적합(model fitting)

- 최소제곱법(least squares method)을 이용
  - 오차 제곱합을 최소로 하는 회귀계수 β_0와 β_1을 구함
  - 즉,

$$\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2$$

  - 을 최소로 하는 β_0와 β_1을 구함

- 선형회귀모형 $\text{dist} = \beta_0 + \beta_1 \text{speed} + \text{error}$ 적합하기 위해 `lm()`를 이용

```
res.lm <- lm(dist ~ speed, data=cars)
summary(res.lm)

##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.069  -9.525  -2.272   9.215  43.201
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.5791     6.7584  -2.601   0.0123 *
## speed         3.9324     0.4155   9.464 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.38 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.6438
## F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

- 분석 결과 해석
  - 출력 결과의 Coefficients 에서 회귀계수 β_0와 β_1의 추정치 및 통계적 유의성(significance)을 확인함.
    - (Intercept)에 표시된 Estimate 이 β_0에 대한 추정치이고 speed 에 표시된 값이 β_1의 추정치임.

- 즉, 제동거리 dist 와 주행속력 speed 에 대해 $\text{dist} = -17.5791 + 3.9324\text{speed}$의 회귀식을 얻음.

- 오른쪽 끝에 표시된 Pr(>|t|)는 추정치의 통계적 유의성을 확인하기 위해 실시한 t-test 결과 도출된 유의확률(p-value). 미리 정한 유의수준 (통상적으로 0.05)보다 작으면 유의하다고 판단

- 이 예의 경우 두 회귀계수 추정치 모두 유의함

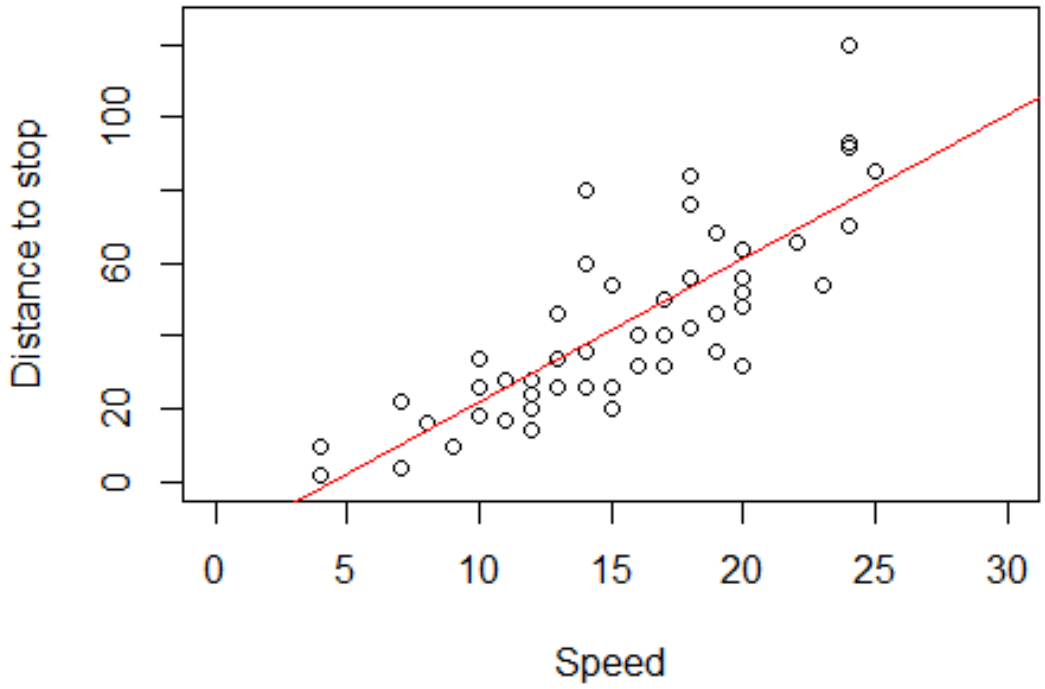
- Multiple R-squared 는 적합된 회귀모형의 설명력을 0 부터 1 사이의 수로 나타냄.
  - 1 이면 추정된 회귀모형이 데이터에 대해 100 퍼센트의 설명력을 가지는 완벽한 모형임을 의미하고, 따라서 클수록 설명력이 좋은 것으로 판단함

  - 이 예의 경우 0.6511(65.11 퍼센트)로서 양호한 편임

- F-statistic 은 적합된 회귀모형의 유의성을 검정하는 결과임
  - 해당 p-value 가 미리 정한 유의수준(통상적으로 0.05)보다 작으면 적합된 회귀모형이 유의하다고 판단

  - 이 예의 경우 1.49e-12 로 매우 작으므로 유의하다고 판단

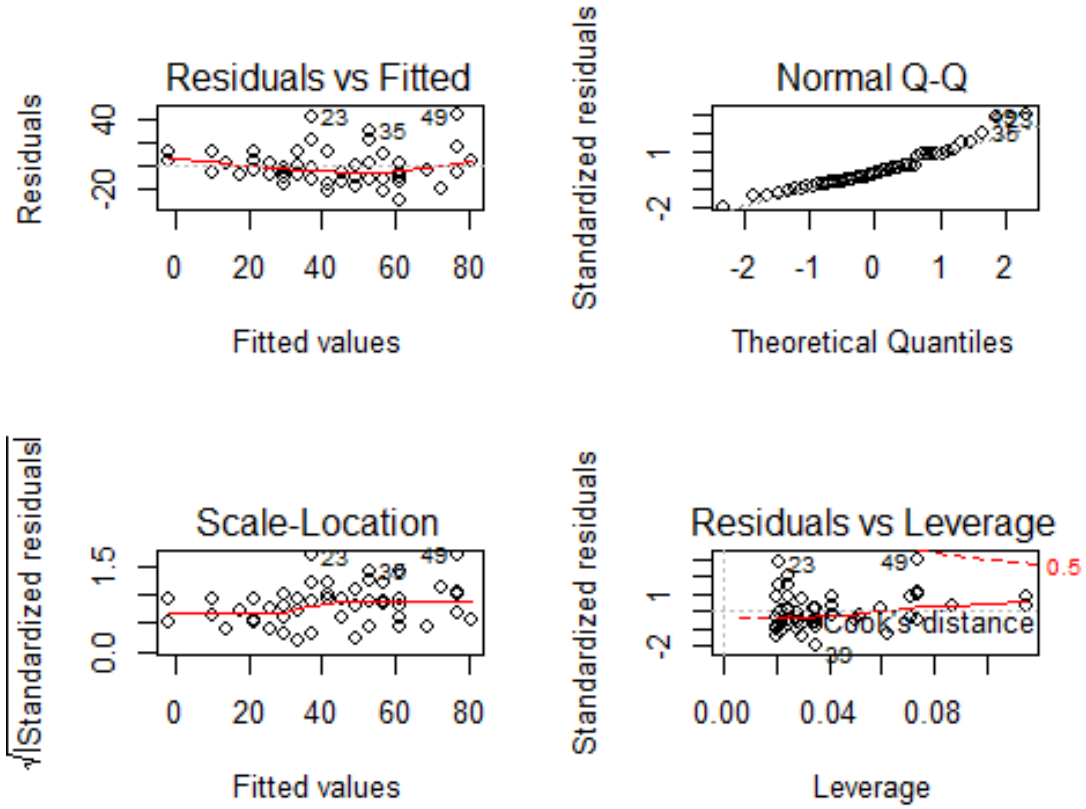
```
# 축 제목을 수정하고 적합된 회귀함수 그래프를 겹쳐 그림
plot(cars$speed, cars$dist, xlab="Speed", ylab="Distance to stop",
     xlim=c(0, 30), ylim=c(0, 125))
abline(coef(res.lm), col=2)
```



3. 회귀 진단 - 잔차 분석
   - 회귀식에 의한 예측치와 실제 관측치 사이의 차이를 **잔차** (residual)라 함

   - 잔차(`res.lm$residuals`)와 예측치(`res.lm$fitted.values`) 간의 산점도를 그려보면 적합된 회귀모형에 문제는 없는 지 진단을 해볼 수 있음
     - 적합된 회귀모형에 문제가 없다면 잔차 그림에서 특정 패턴을 발견할 수 없어야 함

```
par(mfrow=c(2,2))
plot(res.lm)      # Diagnostic plots
```



- Residual vs Fitted 그림을 보면 이차식 적합의 필요성이 보임

- 정규성 가정은 대체로 만족하는 것으로 판담함

- 또한 β_0의 추정치가 -17.5791 인데, 이는 주행속도가 0 일 때 평균 제동거리를 -17.5791 로 예측하게 하는 결과로서 물리적으로 불합리한 결과임. β_0는 0 으로 고정할 필요가 있음

4. 모형 수정 및 확정
   - 회귀진단과정에서 발견된 문제점을 고려해 모형을 수정하는 단계

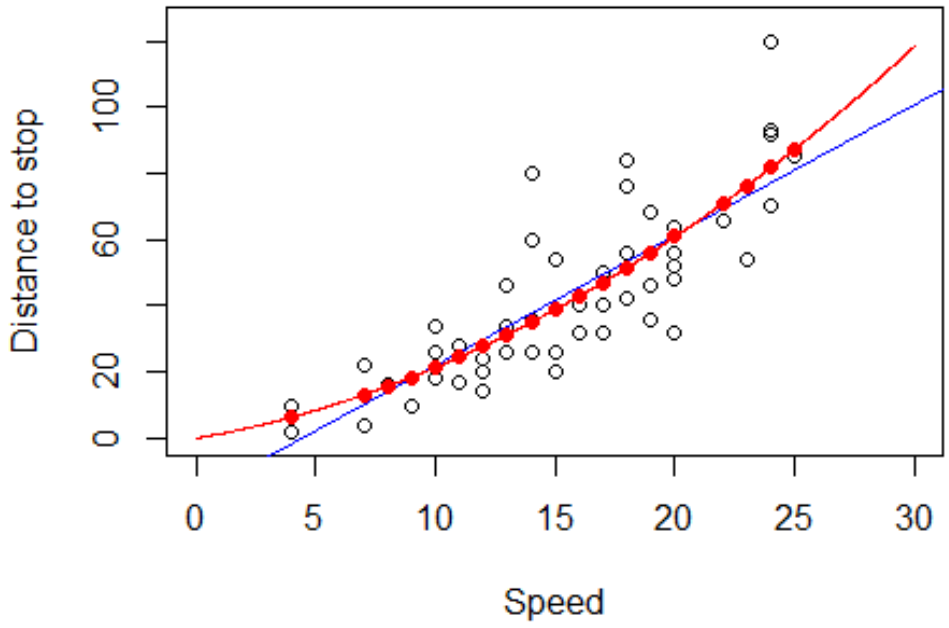
   - 이 예의 경우 이차식

$$\text{dist} = \beta_1 \text{speed} + \beta_2 \text{speed}^2$$

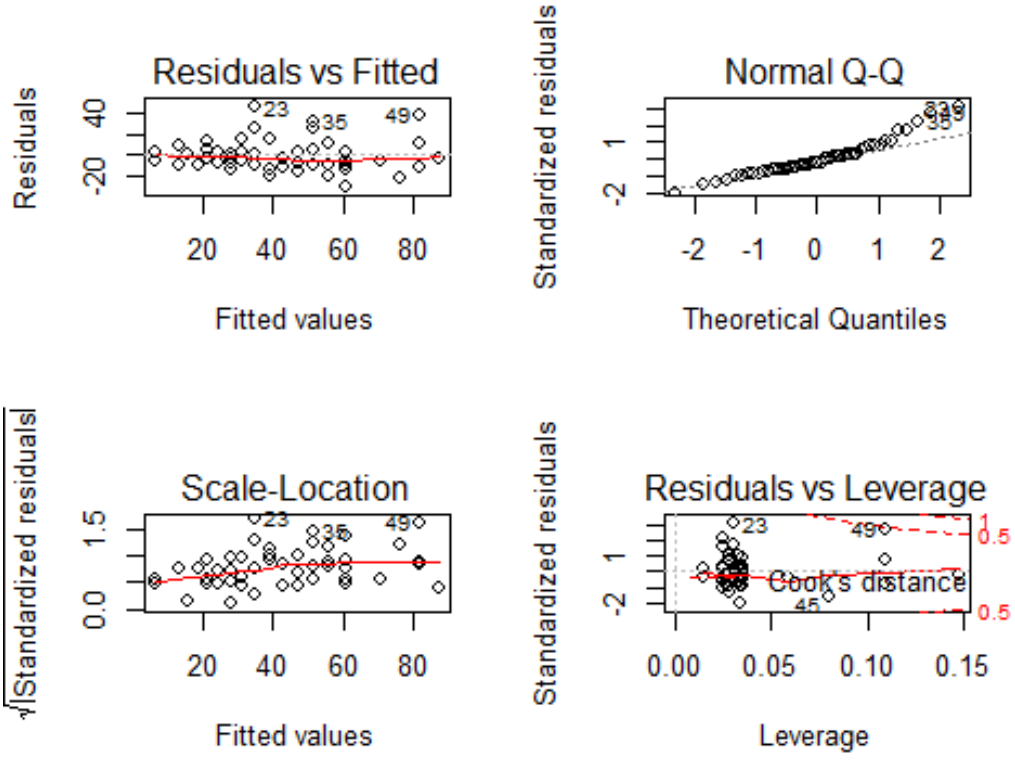
   - 을 회귀식으로 하는 것이 좋음
     - β_0를 0 으로 고정하려면 `lm()` 의 모형식에 `-1` 를 사용

- speed^{2}을 모형식에 넣으려면 As-Is 함수인 `I()`를 이용

```
res.lm2 <- lm(dist ~ speed + I(speed^2) - 1, data=cars)
summary(res.lm2)

##
## Call:
## lm(formula = dist ~ speed + I(speed^2) - 1, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.836  -9.071  -3.152   4.570  44.986
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## speed       1.23903    0.55997   2.213  0.03171 *
## I(speed^2)  0.09014    0.02939   3.067  0.00355 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.02 on 48 degrees of freedom
## Multiple R-squared:  0.9133, Adjusted R-squared:  0.9097
## F-statistic: 252.8 on 2 and 48 DF,  p-value: < 2.2e-16

plot(cars$speed, cars$dist, xlab="Speed", ylab="Distance to stop",
     xlim=c(0, 30), ylim=c(0, 125))
abline(coef(res.lm), col=4)
x <- seq(from=0, to=30, by=0.01)
lines(x, res.lm2$coefficients[1]*x+res.lm2$coefficients[2]*x*x, col=2)
points(cars$speed, predict(res.lm2), col=2, pch=19)
```




```r
par(mfrow=c(2,2))
plot(res.lm2)      # New diagnostic plots
```



- 일차식으로 적합했을 때(파란 실선)보다 이차식 적합(빨간 실선)이 훨씬 더 자료를 잘 설명

- 잔차 그림에서도 일차식 적합 때 발견되었던 특이점을 찾을 수 없음

- Scale-Location 에서 고르게 분포되는 것으로 보아 등분산성 가정을 만족한다고 봄

- Leverage plot 을 보면 23 번째와 49 번째 자료가 이상치 혹은 영향치로 짐작됨

- 최종 회귀식:

$$\text{dist} = 1.23903\text{speed} + 0.09014\text{speed}^2$$

5. 예측에 활용
   - 적합된 회귀식을 이용하면 설명변수의 특정값에 대한 반응변수의 값을 예측할 수 있음

   - 아래 코드는 `predict()` 함수를 이용해 주행속력이 15 인 경우와 20 인 경우에 대한 제동거리 예측치를 구하는 예임

```
newdata <- data.frame(speed=c(15, 20))
predict(res.lm2, newdata)   # 최종 회귀식인 이차식에 의한 예측

##        1        2
## 38.86667 60.83611
```

   - 참고로 앞에서 적합했던 일차식 적합 결과에 의한 예측 결과와 비교해보라

```
predict(res.lm, newdata)    # 일차식에 의한 예측

##        1        2
## 41.40704 61.06908
```

D.1.2 다중회귀분석

설명변수가 여러 개인 경우를 **다중회귀**(multiple regression)라 한다.
현실적으로 회귀분석의 거의 대부분은 다중회귀분석에 해당한다.
설명변수가 하나인 경우는 **단순회귀**(simple regression)이라 한다.

많은 경우 설명변수에 범주형 변수를 포함하는데, 이 때 해당 변수의 타입은 factor 로 처리된다.
해당 변수에 대한 가변수(dummy variable)을 만들어 사용하게 되며, 특정 수준(보통은 알파벳 순으로 맨 처음 값에 해당하는 수준) 대비 추정치를 계산한다.

아래 코드는 **MASS** 패키지의 built-in dataset 인 `Cars93` 를 이용해 차량 가격을 예측하는 모형을 적합하는 예이다.
분석의 흐름은 단순회귀분석의 경우와 동일하다.

- **MASS** 패키지의 `Cars93` 데이터셋을 불러와 자료 구조 파악

```
library(MASS)
data(Cars93)
str(Cars93)

## 'data.frame':    93 obs. of  27 variables:
##  $ Manufacturer      : Factor w/ 32 levels "Acura","Audi",..: 1 1 2 2
3 4 4 4 4 5 ...
##  $ Model             : Factor w/ 93 levels "100","190E","240",..: 49 5
6 9 1 6 24 54 74 73 35 ...
##  $ Type              : Factor w/ 6 levels "Compact","Large",..: 4 3 1
3 3 3 2 2 3 2 ...
##  $ Min.Price         : num  12.9 29.2 25.9 30.8 23.7 14.2 19.9 22.6 26.
3 33 ...
##  $ Price             : num  15.9 33.9 29.1 37.7 30 15.7 20.8 23.7 26.3
 34.7 ...
##  $ Max.Price         : num  18.8 38.7 32.3 44.6 36.2 17.3 21.7 24.9 26.
3 36.3 ...
##  $ MPG.city          : int  25 18 20 19 22 22 19 16 19 16 ...
##  $ MPG.highway       : int  31 25 26 26 30 31 28 25 27 25 ...
##  $ AirBags           : Factor w/ 3 levels "Driver & Passenger",..: 3 1
 2 1 2 2 2 2 2 ...
##  $ DriveTrain        : Factor w/ 3 levels "4WD","Front",..: 2 2 2 2 3
2 2 3 2 2 ...
##  $ Cylinders         : Factor w/ 6 levels "3","4","5","6",..: 2 4 4 4
2 2 4 4 4 5 ...
##  $ EngineSize        : num  1.8 3.2 2.8 2.8 3.5 2.2 3.8 5.7 3.8 4.9 ...
##  $ Horsepower        : int  140 200 172 172 208 110 170 180 170 200 ...
##  $ RPM               : int  6300 5500 5500 5500 5700 5200 4800 4000 48
00 4100 ...
##  $ Rev.per.mile      : int  2890 2335 2280 2535 2545 2565 1570 1320 16
90 1510 ...
##  $ Man.trans.avail   : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 1 1
1 1 ...
##  $ Fuel.tank.capacity: num  13.2 18 16.9 21.1 21.1 16.4 18 23 18.8 18
 ...
##  $ Passengers        : int  5 5 5 6 4 6 6 6 5 6 ...
##  $ Length            : int  177 195 180 193 186 189 200 216 198 206 ...
```

```
##  $ Wheelbase         : int  102 115 102 106 109 105 111 116 108 114 ...
##  $ Width             : int  68 71 67 70 69 69 74 78 73 73 ...
##  $ Turn.circle       : int  37 38 37 37 39 41 42 45 41 43 ...
##  $ Rear.seat.room    : num  26.5 30 28 31 27 28 30.5 30.5 26.5 35 ...
##  $ Luggage.room      : int  11 15 14 17 13 16 17 21 14 18 ...
##  $ Weight            : int  2705 3560 3375 3405 3640 2880 3470 4105 34
95 3620 ...
##  $ Origin            : Factor w/ 2 levels "USA","non-USA": 2 2 2 2 2 1
 1 1 1 1 ...
##  $ Make              : Factor w/ 93 levels "Acura Integra",..: 1 2 4 3
 5 6 7 9 8 10 ...
```

- lm()를 이용해 다중회귀모형 적합

```
res.lm <- lm(Price ~ Type + AirBags + Cylinders + Man.trans.avail, data=C
ars93)
summary(res.lm)

##
## Call:
## lm(formula = Price ~ Type + AirBags + Cylinders + Man.trans.avail,
##     data = Cars93)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.3474  -3.1590  -0.2354   2.2646  30.5086
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          19.4499     4.7898   4.061 0.000114 ***
## TypeLarge            -5.2185     3.4470  -1.514 0.134030
## TypeMidsize           3.2177     2.4771   1.299 0.197721
## TypeSmall            -5.4357     2.1709  -2.504 0.014342 *
## TypeSporty           -4.2684     2.3349  -1.828 0.071310 .
## TypeVan              -1.8335     3.2904  -0.557 0.578949
## AirBagsDriver only   -4.0773     1.9038  -2.142 0.035299 *
## AirBagsNone          -8.0897     2.2203  -3.644 0.000479 ***
## Cylinders4            0.8688     3.7392   0.232 0.816872
## Cylinders5            4.5607     6.1692   0.739 0.461940
## Cylinders6            8.7238     4.2596   2.048 0.043877 *
## Cylinders8           19.5575     4.8409   4.040 0.000123 ***
## Cylindersrotary      18.8536     7.3074   2.580 0.011731 *
## Man.trans.availYes    2.5422     2.1141   1.202 0.232762
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.953 on 79 degrees of freedom
## Multiple R-squared:  0.6738, Adjusted R-squared:  0.6202
## F-statistic: 12.56 on 13 and 79 DF,  p-value: 2.375e-14
```

- 회귀분석 결과에 따른 가격 예측치 산출

```
newdata <- data.frame(Type='Large', AirBags='Driver only', Cylinders='6',
 Man.trans.avail='Yes')
predict(res.lm, newdata)

##        1
## 21.42005
```

변수선택(variable selection)은 불필요한 설명변수를 제거해 보다 설득적이고 간명한 회귀모형을 구축하는 문제로서 다중회귀분석에서 매우 중요한 주제이다.

- Stepwise
  - 변수들을 추가하고 제거하는 과정을 반복해 최적의 모형을 찾아감
  - 아래 코드는 AIC 를 기준으로 stepwise 를 적용하는 예임
  - 최종 선택변수는 Type, AirBags, Cylinders 이상 세 변수임. Man.trans.avail 는 제거.

```
step <- stepAIC(res.lm, direction="both")

## Start:  AIC=344.63
## Price ~ Type + AirBags + Cylinders + Man.trans.avail
##
##                   Df Sum of Sq    RSS    AIC
## - Man.trans.avail  1     51.25 2850.9 344.32
## <none>                         2799.7 344.63
## - AirBags          2    483.64 3283.3 355.45
## - Type             5    927.78 3727.5 361.25
## - Cylinders        5   1556.16 4355.9 375.74
##
## Step:  AIC=344.32
## Price ~ Type + AirBags + Cylinders
##
##                   Df Sum of Sq    RSS    AIC
## <none>                         2850.9 344.32
## + Man.trans.avail  1     51.25 2799.7 344.63
## - AirBags          2    515.77 3366.7 355.79
## - Type             5    887.72 3738.7 359.53
## - Cylinders        5   1505.46 4356.4 373.75

anova(step)

## Analysis of Variance Table
##
```

```
## Response: Price
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Type       5 3421.4  684.29 19.2017 1.660e-12 ***
## AirBags    2  806.2  403.09 11.3110 4.720e-05 ***
## Cylinders  5 1505.5  301.09  8.4489 1.868e-06 ***
## Residuals 80 2850.9   35.64
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- (c.f.) LASSO (least absolute shrinkage and selection operator)
  - 변수의 개수가 많아지면서 모형의 적합도는 좋아지지만 복잡성이 증가하는 것에 대해 penalty 를 부여해 변수의 개수를 줄여 최적의 모형을 찾아감
  - 예측 관점에서 변수 선택 문제에 접근하는 방식
  - 고차원 자료 분석에 효과적

다중 회귀 (multiple regression)를 실시하는 경우 **다중공선성** 문제에 유의해야 한다.

- 다중 공선성(multiple collinearity)
  - x 변수 간 선형종속관계를 심한 경우 분석이 어려움
  - 탐지 방법: 추정된 회귀계수 중 얼토당토 않은 값이 포함된 경우, 또는 분산팽창계수(variance inflation factor, VIF)가 10 보다 큰 변수가 있으면 다중공선성이 있다고 봄
  - 해결 방안: 문제가 되는 변수를 제거하거나, 주성분회귀(principal component regression), 능형회귀(ridge regression) 등의 방법 사용
  - 아래 코드는 `car` 패키지의 내장 데이터셋인 `mtcars` 의 mpg(연비)를 반응변수로 하는 회귀모형의 다중공선성에 대한 예임

```
library(car)
str(mtcars)
```

```
## 'data.frame':    32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
```

```
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

```
fit <- lm(mpg~., data=mtcars)
summary(fit)
```

```
##
## Call:
## lm(formula = mpg ~ ., data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4506 -1.6044 -0.1196  1.2193  4.6271
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.30337   18.71788   0.657   0.5181
## cyl         -0.11144    1.04502  -0.107   0.9161
## disp         0.01334    0.01786   0.747   0.4635
## hp          -0.02148    0.02177  -0.987   0.3350
## drat         0.78711    1.63537   0.481   0.6353
## wt          -3.71530    1.89441  -1.961   0.0633 .
## qsec         0.82104    0.73084   1.123   0.2739
## vs           0.31776    2.10451   0.151   0.8814
## am           2.52023    2.05665   1.225   0.2340
## gear         0.65541    1.49326   0.439   0.6652
## carb        -0.19942    0.82875  -0.241   0.8122
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.65 on 21 degrees of freedom
## Multiple R-squared:  0.869,  Adjusted R-squared:  0.8066
## F-statistic: 13.93 on 10 and 21 DF,  p-value: 3.793e-07
```

- 연비에 결정적 영향을 줄 것으로 예상되는 주요 변수들(cyl: 실린더수, disp: 배기량, hp: 마력, am: 오토/수동, gear: 기어단수)이 유의하지 않은 회귀계수 추정치를 갖고 부호도 거꾸로인 경우(disp: 배기량)도 보이는 등 납득하기 어려운 결과임

- 따라서 다중공선성을 의심하고 `vif()` 함수를 이용해 다중공선성 문제가 있는지 알아봄

```
vif(fit)

##       cyl      disp        hp      drat        wt      qsec        vs
## 15.373833 21.620241  9.832037  3.374620 15.164887  7.527958  4.965873
##        am      gear      carb
##  4.648487  5.357452  7.908747
```

- cyl(실린더 수), disp(배기량), wt(무게) 등의 변수의 vif 값이 10 을 넘음
- 이들 변수의 특징을 살펴보면 서로 간에 매우 높은 수준의 상관계수값을 가질 것을 예상할 수 있음
- 그러나 이들 변수 모두 연비를 예측할 때 매우 중요한 변수들임을 고려하면 단순히 변수를 제거하는 방식은 바람직하지 않음 → 주성분회귀, 능형회귀 등을 고려할 필요가 있음

D.1.3 비모수적 회귀분석

비모수적 회귀분석은 두 변수 간의 함수관계에 특정 형태를 가정하지 않고 평활법을 이용해 함수 관계를 구하는 최신 기법으로 매우 유연한 모형을 통해 정확한 예측이 가능하다.
다양한 방법이 있지만 그중 가장 널리 쓰이는 방법인 커널 방법을 소개한다.

커널 평활법: 국소선형회귀 (kernel smoothing: local linear regression)

- 관측데이터: $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$
- $f(x) = E(y|x)$에 대해 특별한 함수적 형태를 가정하지 않고 직접 함수 $f(x)$의 값을 추정하는 방법
  - 회귀함수의 특수한 형태를 가정할 만한 정보가 없을 때, 자료의 구조를 탐색적으로 살펴보고 싶을 때 유용
- 국소선형회귀
  - x_0에서의 회귀함수값 $f(x_0)$을 추정하는 문제
  - 회귀함수 $f(x)$에 대해 다음의 테일러급수전개에 의한 선형근사(linear approximation)을 고려: $x \approx x_0$일 때,

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) = \beta_0 + \beta_1(x - x_0)$$

-

- 즉, $x \approx x_0$인 영역에서 국소적으로 회귀함수 $f(x)$의 형태를 선형식으로 가정하는 것이 큰 문제가 없음

- 아래 그림은 회귀함수가 $f(x) = \sin(x)$(검은색 점선)인 경우 $x = 2$에서의 국소선형근사(파란색 실선)를 나타낸 것인데, 매우 비슷한 값을 가짐을 확인할 수 있음
    - 비슷한 정도는 더 좁은 $x = 2$ 근방을 고려할 수록 좋아짐

- 그림의 작은 동그라미들은 모의실험에 의해 생성된 100 개의 데이터 포인트

- 빨간색 작은 동그라미는 생성된 데이터 중 $x = 2$ 근방에 놓인 점들

- 빨간색으로 표시된 점들을 이용해 선형회귀를 실시하면, 위에서 보듯 $f(x_0) = \beta_0$ 관계식을 이용해 $f(x_0)$의 값을 추정할 수 있음

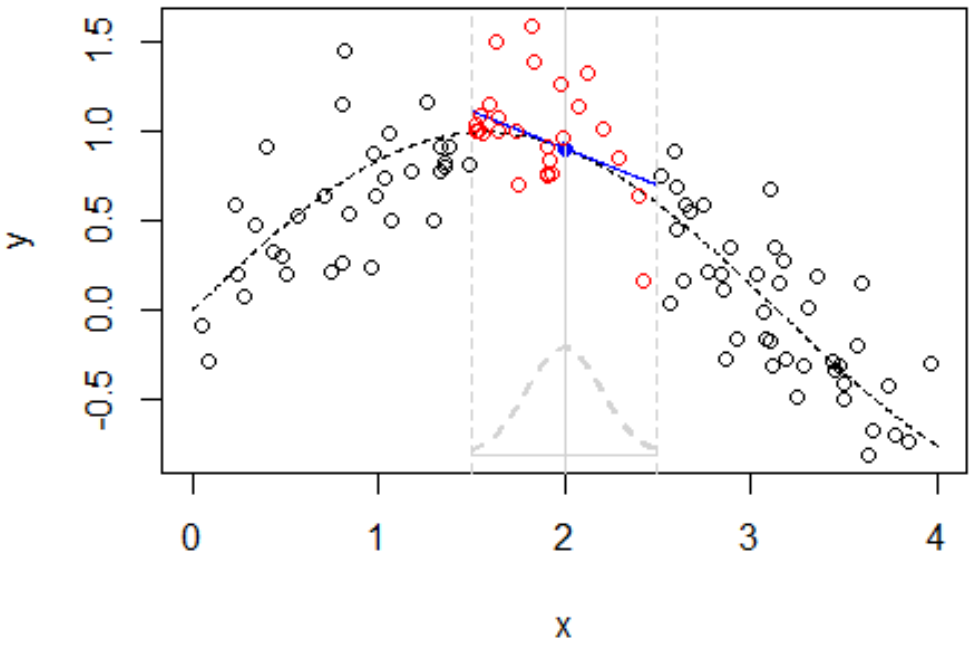
- 다만 $x_0 = 2$에 가까운 점일 수록 높은 가중치를 부여하면 보다 정확한 추정 가능
    - 이 가중치를 결정하는 함수 $K(x)$를 커널(kernel)이라 부름 (그림의 아래 회색 점선)

    - 회귀함수를 선형근사시킬 영역의 너비, 소위 bandwidth 를 정하는 문제가 매우 중요

    - x_0 근방에서 국소오차제곱에 대한 커널가중합
    $$\sum_{i=1}^{n} \{ y_i - \beta_0 - \beta_1 (x_i - x_0) \}^2 K((x_i - x_0)/h)$$
    - 을 최소로 하는 (β_0, β_1)을 찾은 후 $f(x_0) = \beta_0$로 $f(x_0)$의 추정치를 삼음. 단, $h > 0$는 bandwidth.

- 이상의 절차를 다양한 x_0의 값에서 반복

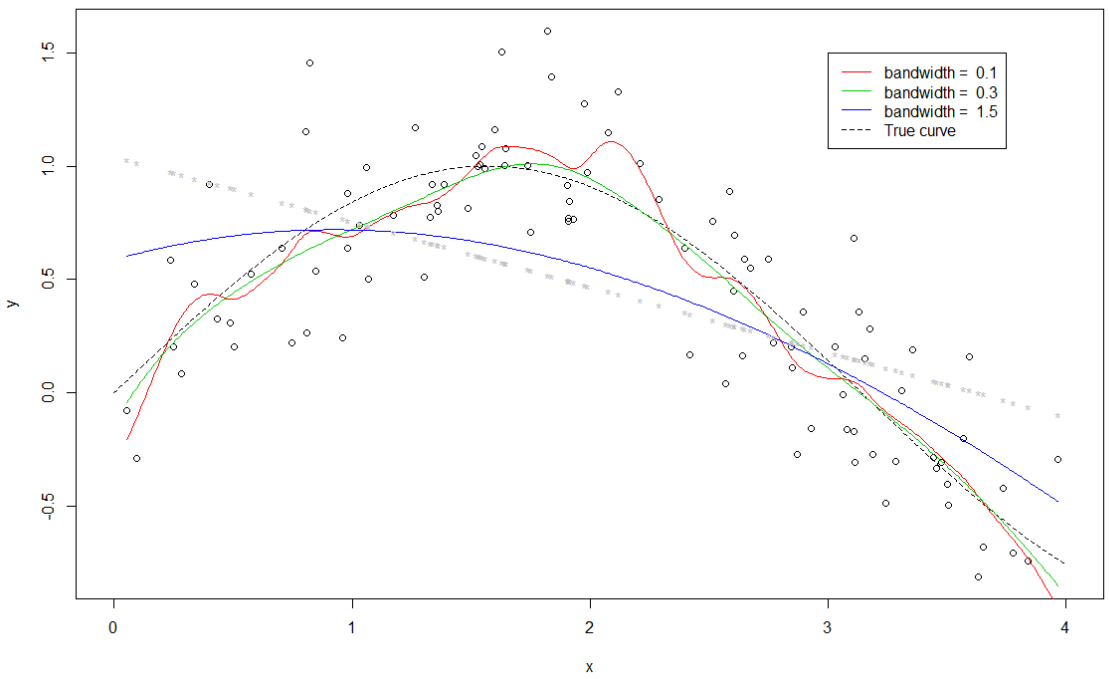
- 보통 의미있는 영역에서 grid 를 잡아 각 grid 점을 x_0로 놓고 모든 grid 점에서 반복 시행



- 커널을 이용한 국소선형회귀분석은 **KernSmooth** 패키지의 `locpoly()`함수를 이용하면 편리
  - 우선 f(x) = sin(x)로 하여 n = 100인 데이터셋을 생성하고 그림으로 나타냄
  - `loess()` 함수를 이용해 커널 국소선형회귀를 실시
    - 커널함수는 표준정규분포곡선(default)
    - bandwidth 는 0.1, 0.3, 1.5 등의 값을 적용해 비교

```
set.seed(1)
n <- 100
X <- runif(n)*4
Y <- sin(X) + rnorm(n, sd=.3)

x <- seq(from=0, to=4, by=0.005)
```

```r
y <- sin(x)
plot(x, y, type='l', ylim=c(min(Y), max(Y)), lty=2)
points(X, Y)

library(KernSmooth)

## KernSmooth 2.23 loaded
## Copyright M. P. Wand 1997-2009

h <- c(0.1, 0.3, 1.5)

for ( k in 1:length(h) ) {
  res.lp <- locpoly(X, Y, bandwidth=h[k])
  lines(res.lp, col=k+1)
}
legend(3, 1.5, c(paste("bandwidth = ",h[1]), paste("bandwidth = ",h[2]),
                 paste("bandwidth = ",h[3]), "True curve"),
       col=c(2:4, 1), lty=c(rep(1, 3), 2))

res.lm <- lm(Y~X)
points(X, predict(res.lm), col=8, pch="*")
```



- 각 색깔이 있는 실선은 국소선형회귀에 의해 추정된 회귀함수의 그래프임
  - bandwidth 가 0.3 인 경우(녹색 실선) 실제 회귀함수와 매우 유사

        - bandwidth 가 작으면(이 예에서는 0.1 인 경우, 빨간색 실선) 평활 정도가 너무 작아 추정 결과가 불안정

        - bandwidth 가 너무 크면(이 예에서는 1.5 인 경우, 파란색 실선) 평활량이 너무 커서 추정 결과가 자료의 trend 을 잘 표현하지 못함

    - 참고로 회색 별표는 선형회귀에 의한 예측 결과임 (bandwidth 가 무한대인 경우에 해당)

    - 즉, 국소선형회귀방법을 사용할 때 적절한 bandwidth 를 선택하면 매우 정확한 예측이 가능함을 시사함

- 실제 자료 분석: **MASS** 패키지의 built-in 데이터셋인 `mcycle` 분석
    - 자동차 충돌시험 결과 데이터
        - times: 충돌 후 시간(1000 분의 1 초 단위)

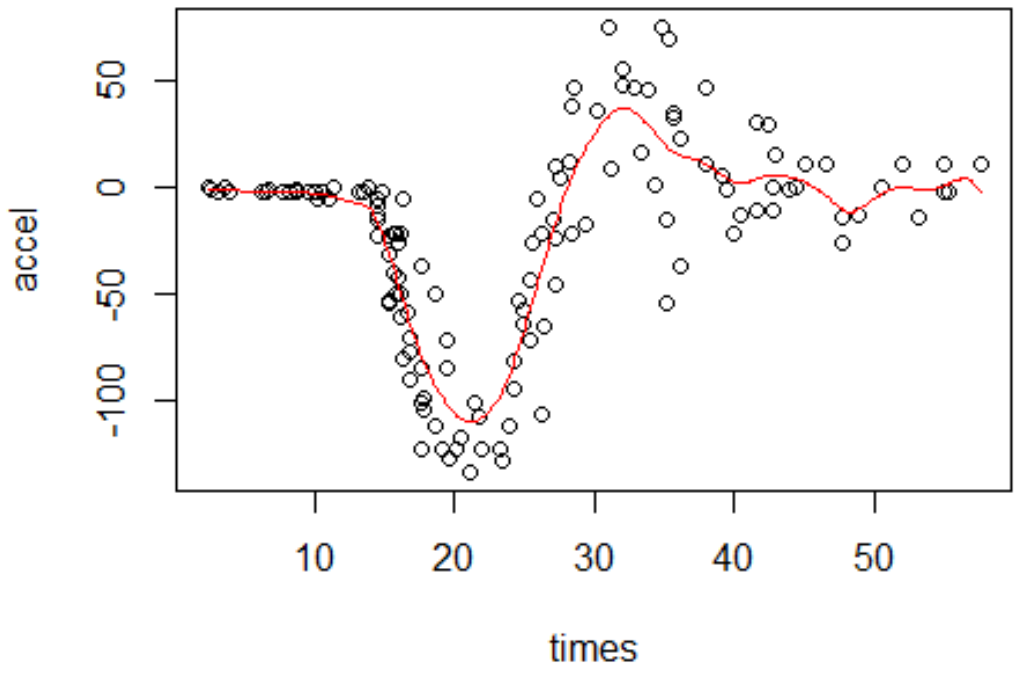
        - accel: 충돌 후 가속도 (in g)

    - 시간에 따른 가속도의 변화를 모델링

```
library(MASS)
data(mcycle)
str(mcycle)
```

```
## 'data.frame':    133 obs. of  2 variables:
##  $ times: num  2.4 2.6 3.2 3.6 4 6.2 6.6 6.8 7.8 8.2 ...
##  $ accel: num  0 -1.3 -2.7 0 -2.7 -2.7 -2.7 -1.3 -2.7 -2.7 ...
```

```
plot(mcycle)

res.lp <- with(mcycle, locpoly(times, accel, bandwidth=1.5))
lines(res.lp, col=2)
```



D.1.4 로지스틱회귀분석

위 절에서는 반응변수가 연속형 변수로서 반응변수의 분포에 대해 정규분포를 가정할 수 있는 경우를 배웠다.
이 절에서는 반응변수에 대해 정규분포 가정을 하는 것이 불가능한 경우, 특히 반응변수가 이항변수인 경우에 대한 분석 방법을 익힌다.

- 반응변수 y가 성공(1), 실패(0) 두 가지 값을 갖는 이항(binary) 변수로 생각할 수 있는 경우
  - (예) 환자 사망, 전염병 발병, 신용 부도, 고객 claim, 교통사고 등
- 반응변수가 연속형 변수인 경우 $E(y|x)$가 임의의 실수값을 취할 수 있는 것에 반해, 이항 변수인 경우 $E(y|x) = Prob(y = 1|x)$가 되어 0 과 1 사이의 범위로 취할 수 있는 값의 범위가 제한됨

- 이를 무시하고 통상적인 회귀분석을 적용하면 예측값이 허용 범위인 0 과 1 사이를 벗어나는 경우가 발생해 상당한 문제가 발생하게 됨

- 로짓 변환:
  - $\text{logit}(p) = \log\frac{p}{1-p}, 0 < p < 1$
    0 에서 1 사이의 범위를 갖는 숫자를 실수 전체 영역의 값을 갖도록 변환
- 로지스틱(logistic) 회귀모형:
  - $f(x) = \text{logit}p(x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$
    로짓변환 후 predictor 에 대해 선형모형을 유지

  - 로짓 대신 표준정규분포의 누적분포함수의 역함수를 이용한 프로빗(probit) 모형도 많이 쓰임
- **faraway** 패키지의 built-in 데이터셋인 `orings` 분석
  - 1986 년 우주왕복선 챌린저호 폭발 사건이 로켓엔진의 O-ring 불량과 관련이 있다고 알려짐

  - 이전 23 회의 우주 왕복 임무 과정에서 수집된 데이터임
    - temp: 발사 때 기온 (화씨)

    - damage: 6 회 시험 중 O-ring 손상 회수

  - 챌린저호 발사 당시 기온은 화씨 31 도였음. O-ring 손상확률은 얼마인 지 예측해보자.

```
  library(faraway)
```

```
##
## Attaching package: 'faraway'
##
## The following objects are masked from 'package:car':
##
##     logit, vif
```

```
  data(orings)
  str(orings)
```

```
## 'data.frame':    23 obs. of  2 variables:
##  $ temp  : num  53 57 58 63 66 67 67 67 68 69 ...
##  $ damage: num  5 1 1 1 0 0 0 0 0 0 ...
```

```
  res.glm <- glm(cbind(damage, 6-damage) ~ temp, family=binomial, data=or
ings)
  summary(res.glm)
```

```
##
## Call:
## glm(formula = cbind(damage, 6 - damage) ~ temp, family = binomial,
##     data = orings)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9529  -0.7345  -0.4393  -0.2079   1.9565
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 11.66299    3.29626   3.538 0.000403 ***
## temp        -0.21623    0.05318  -4.066 4.78e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 38.898  on 22  degrees of freedom
## Residual deviance: 16.912  on 21  degrees of freedom
## AIC: 33.675
##
## Number of Fisher Scoring iterations: 6
```

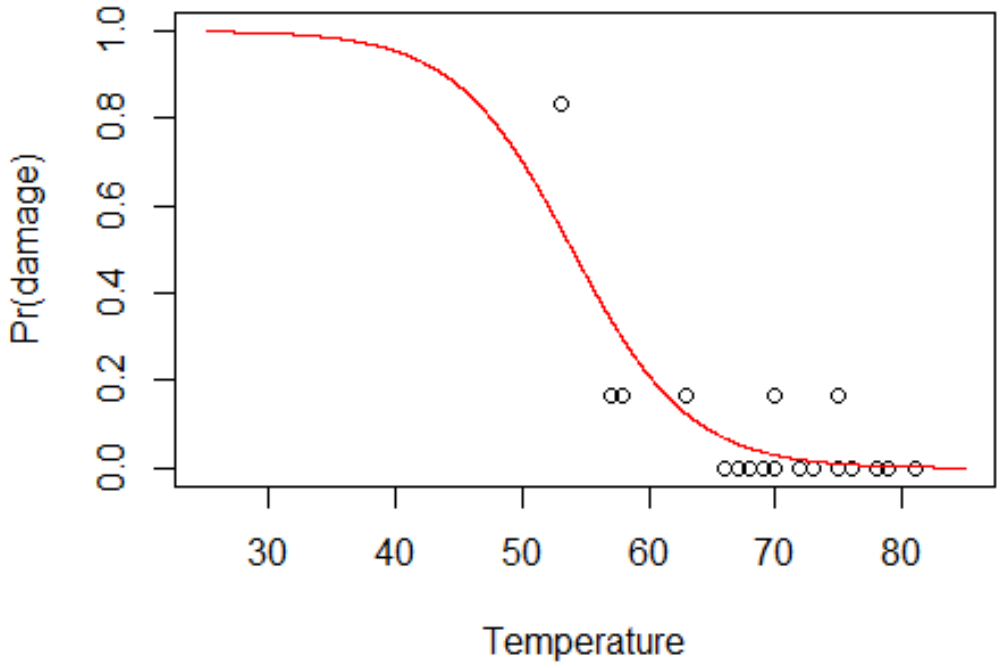
- 기온(temp)에 대한 회귀계수가 음수(-0.21623)이며 통계적으로 유의함(p-value = 4.78e-05). 즉, 추운 날씨에서 O-ring 의 손상 가능성이 높아짐을 의미

- 예측식:
  $\text{logit}p(damage|temp) = 11.66299 - 0.21623 \times temp$

-

- 다음은 분석 결과를 그래프로 나타내는 코드임

```
  plot(damage/6 ~ temp, xlim=c(25, 85), ylim=c(0, 1), xlab="Temperature",
 ylab="Pr(damage)", data=orings)
  x <- seq(from=25, to=85, by=0.01)
  beta <- res.glm$coefficients
  lines(x, ilogit(beta[1]+beta[2]*x), col=2)
```



- 챌린저호 발사 당시 기온이 화씨 31 도로서 매우 추운 날씨였음. 손상확률이 높을 것으로 예상됨.

- 아래는 챌린저호 발사 당시 기온인 화씨 31 도에서의 손상확률을 예측하는 코드임

```
predict(res.glm, newdata=data.frame(temp=31), type="response")
```

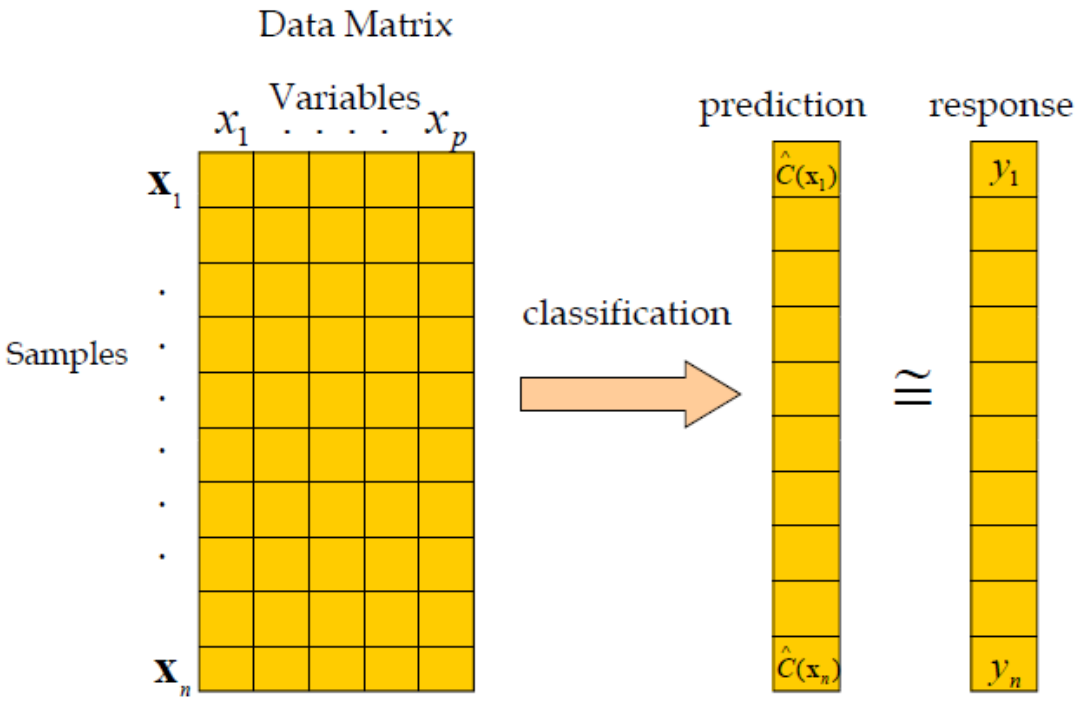
```
##         1
## 0.9930342
```

- 손상확률이 99.3%로 예측됨. OMG!

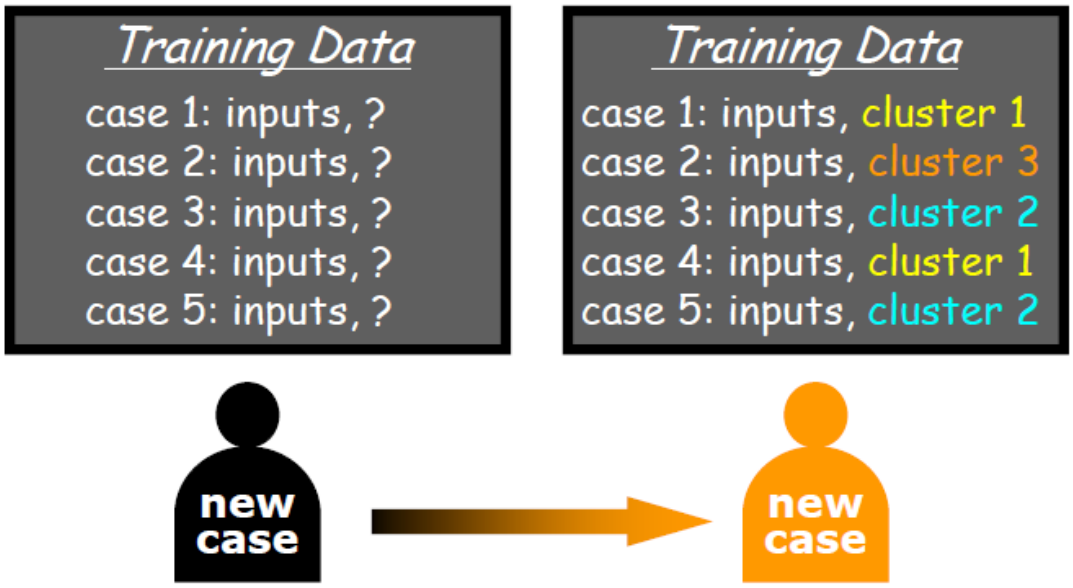
D.2. 통계적 기계 학습

통계적 학습(statistical learning)이란 데이터 내 구조를 이해하기 위한 다양한 도구를 통칭한다.
크게 지도학습과 자율학습으로 분류할 수 있다.

- 지도학습 (supervised learning) : 자료에 outcome (or response) measurement 가 포함된 경우
  - Training dataset $(x_1, y_1), \ldots, (x_n, y_n)$ 을 이용해 예측모델을 구축해 new case 에 대한 예측 실행
  - x 변수를 input 으로 y 변수를 output 으로 하는 예측 모델을 이용해 새로이 주어진 x 변수에 대응되는 y 변수의 값을 예측
  - (예) Logistic regression 을 이용한 classification



- 자율학습 (unsupervised learning): 자료에 outcome (or response) measurement 가 포함되지 않은 경우
  - 분석 목적이 다소 불분명한 문제가 있을 수 있음
  - (예) Clustering



D.2.1 지도학습에 의한 분류

로지스틱 회귀 이용법

앞의 2 절에서 학습한 로지스틱 회귀방법, 즉 주어진 입력변수 x에 대해 출력변수 Y가 1 이 될 확률 $p(x) = P(Y = 1|x)$를 추정하는 방법에 기초한 분류법을 익혀보자. 클래스가 두 개인 경우 특히 신용평점모형(credit scoring model)에 많이 활용되는 방법론이다.

- Classification rule: 0 과 1 사이의 적당한 값 c를 기준으로
  - $p(x) \geq c$ 이면 자료를 $Y = 1$인 클래스로 분류하고
  - $p(x) < c$ 이면 자료를 $Y = 0$인 클래스로 분류
- 절단값(cutoff value) c의 선택
  - 사전정보 활용
  - 손실함수 활용: 오분류에 의한 손실 정도를 그룹별로 비교 분석

- 전문가 의견

- 불균형 자료(unbalanced data)의 분석
  - $Y = 1$인 그룹과 $Y = 0$인 그룹의 크기가 많이 차이 나는 경우
    - (예) 사기 방지 분류 문제: 정상($Y = 1$)인 자료가 절대 다수

  - 원자료를 그대로 사용하지 않고 정상($Y = 1$) 자료에서 적절한 수의 표본을 추출해 균형을 맞추는 방법이 널리 사용됨

- ROC(receiver operating characteristic) 곡선
  - $Y = 1$인 경우에 대해 대부분 $Y = 1$인 것으로 예측을 하는 분류규칙은 sensitive 하다고 함

  - $Y = 0$인 경우에 대해 대부분 $Y = 0$인 것으로 예측을 하면 specific 하다고 함

  - Sensitivity 와 specificity 는 절단값 c에 따라 달라지기 때문에 예측 알고리즘의 평가를 위해 다양한 절단값에 대한 'sensitivity' vs '1-specificity' 도표를 작성하면 곡선이 나타나는데 이를 ROC 곡선이라 함

- 아래는 German credit data 에 대해 로지스틱 회귀에 의한 분류법을 적용한 예임
  - p: 부도 확률(default probability)

  - 절단값 c를 결정: 손실함수 활용법을 이용
    - 부도가 발생하는 경우 우수고객에게 대출을 하지 않는 경우 감내해야 하는 비용에 비해 평균적으로 5 배 정도의 비용이 발생하는 것을 가정함

    - 이 경우 $5p = 1 - p$가 성립하므로 절단값을 $c = 1/6$로 결정함.

    - 즉, 예측확률이 1/6 미만인 경우에만 대출을 승인하는 시나리오를 가정

  - 데이터 불러오기

```
credit <- read.csv("./Data/germancredit.csv", fileEncoding="UTF-8")
dim(credit)

## [1] 1000   21
```

- 불러온 자료에 포함된 몇몇 변수에 대해 알기 쉬운 표현으로 수준(level) 부여

```
## re-level the credit history and a few other variables
credit$history <- factor(credit$history,
                         levels=c("A30","A31","A32","A33","A34"))
levels(credit$history) <- c("good","good","poor","poor","terrible")
credit$foreign <- factor(credit$foreign,
                         levels=c("A201","A202"),
                         labels=c("foreign","german"))
credit$rent <- factor(credit$housing=="A151")
credit$purpose <- factor(credit$purpose,
                         levels=c("A40","A41","A42","A43","A44","A45",
                                  "A46","A47","A48","A49","A410"))
levels(credit$purpose) <- c("newcar","usedcar",rep("goods/repair",4),
                            "edu",NA,"edu","biz","biz")
```

- 분석에 사용할 변수만 잘라내어 새로운 데이터셋 생성

```
## for demonstration, cut the dataset to these variables
credit <- credit[,c("Default","duration","amount","installment","age",
                    "history", "purpose","foreign","rent")]
summary(credit) # check out the data

##     Default       duration        amount       installment
##  Min.   :0.0   Min.   : 4.0   Min.   :  250   Min.   :1.000
##  1st Qu.:0.0   1st Qu.:12.0   1st Qu.: 1366   1st Qu.:2.000
##  Median :0.0   Median :18.0   Median : 2320   Median :3.000
##  Mean   :0.3   Mean   :20.9   Mean   : 3271   Mean   :2.973
##  3rd Qu.:1.0   3rd Qu.:24.0   3rd Qu.: 3972   3rd Qu.:4.000
##  Max.   :1.0   Max.   :72.0   Max.   :18424   Max.   :4.000
##       age            history            purpose       foreign
##  Min.   :19.00   good    : 89   newcar      :234   foreign:963
##  1st Qu.:27.00   poor    :618   usedcar     :103   german : 37
##  Median :33.00   terrible:293   goods/repair:495
##  Mean   :35.55                  edu         : 59
##  3rd Qu.:42.00                  biz         :109
##  Max.   :75.00
##     rent
##  FALSE:821
##  TRUE :179
##
##
##
##
```

- 디자인 행렬 생성

```
## create a design matrix
## factor variables are turned into indicator variables
## the first column of ones is omitted
Xcred <- model.matrix(Default~.,data=credit)[,-1]
```

- 분류모형의 예측력 평가를 위해 데이터셋을 training set 과 test set 으로 구별: 1,000 개의 관측치 중 900 개를 임의로 선택해 training data 로, 나머지 100 개는 test set 으로 지정

```
## creating training and prediction datasets
## select 900 rows for estimation and 100 for testing
set.seed(1)
train <- sample(1:1000,900)
xtrain <- Xcred[train,]
xnew <- Xcred[-train,]
ytrain <- credit$Default[train]
ynew <- credit$Default[-train]
```

- training set 에 대해 로지스틱 회귀를 적용해 예측 모형 구축

```
cred.glm <- glm(Default~.,family=binomial, data=data.frame(Default=ytrain,
 xtrain))
summary(cred.glm)

##
## Call:
## glm(formula = Default ~ ., family = binomial, data = data.frame(Defaul
t = ytrain,
##     xtrain))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2912  -0.7951  -0.5553   0.9922   2.2601
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -2.705e-01  4.833e-01  -0.560 0.575693
## duration             2.721e-02  8.464e-03   3.215 0.001303 **
## amount               9.040e-05  3.854e-05   2.346 0.018987 *
## installment          2.228e-01  8.064e-02   2.763 0.005722 **
## age                 -1.327e-02  7.704e-03  -1.723 0.084961 .
## historypoor         -1.102e+00  2.641e-01  -4.173 3.01e-05 ***
## historyterrible     -1.860e+00  3.007e-01  -6.184 6.25e-10 ***
## purposeusedcar      -1.793e+00  3.555e-01  -5.043 4.58e-07 ***
## purposegoods.repair -7.447e-01  1.976e-01  -3.769 0.000164 ***
```

```
## purposeedu          -6.809e-02  3.401e-01  -0.200 0.841325
## purposebiz          -7.342e-01  2.916e-01  -2.518 0.011812 *
## foreigngerman       -1.363e+00  6.638e-01  -2.053 0.040054 *
## rentTRUE             7.011e-01  2.075e-01   3.378 0.000730 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1102.92  on 899  degrees of freedom
## Residual deviance:  955.21  on 887  degrees of freedom
## AIC: 981.21
##
## Number of Fisher Scoring iterations: 5
```

- test set 으로 예측력 평가

```
ptest <- predict(cred.glm, newdata=data.frame(xnew), type="response")
dat <- data.frame(labels=ynew, preds=ptest)
## What are our misclassification rates on that training set?
## We use probability cutoff 1/6
## coding as 1 (predicting default) if probability 1/6 or larger
cut <- 1/6
gg1 <- as.numeric(ptest >= cut)
ttt <- table(ynew, gg1)
print(ttt)

##     gg1
## ynew  0  1
##    0 30 42
##    1  5 23
```

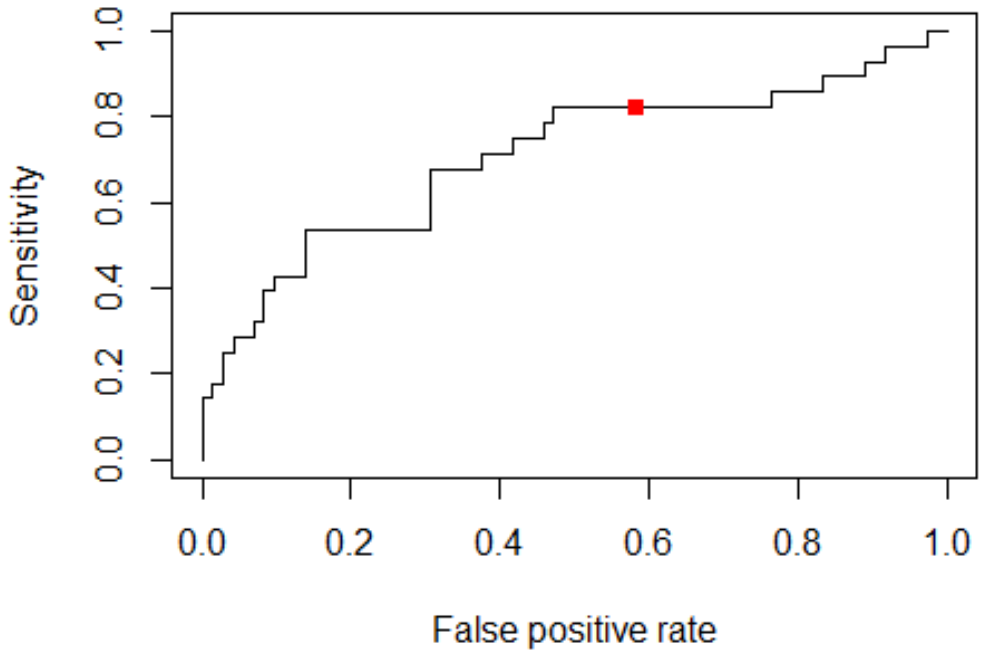
- 예측 성능을 sensitivity, specificity 측면에서 확인

```
truepos <- ynew==1 & ptest>=cut
trueneg <- ynew==0 & ptest<cut
# Sensitivity (predict default when it does happen)
sensitivity <- sum(truepos)/sum(ynew==1)
# Specificity (predict no default when it does not happen)
specificity <- sum(trueneg)/sum(ynew==0)
print(c(sensitivity, specificity))

## [1] 0.8214286 0.4166667
```

- ROC curve

```
library(ROCR)

## Loading required package: gplots
##
```

```
## Attaching package: 'gplots'
##
## The following object is masked from 'package:stats':
##
##     lowess

pred <- prediction(dat$preds, dat$labels)
perf <- performance(pred, "sens", "fpr")
plot(perf)
points(1-specificity, sensitivity, col=2, pch=15)
```



선형판별분석

선형판별분석(linear discriminant analysis, LDA)은 주어진 다변수 자료가 뿌려져 있는 다차원 공간에 자료를 잘 분류할 수 있는 초평면(hyperplane)을 찾아 예측에 활용하는 기법이다.
로지스틱 회귀를 이용할 때와 달리 class 의 개수가 두 개보다 많은 경우에도 활용 가능하다.

- 일반적으로 분류 기법은 다음의 두 가지를 목표로 함
  - Discrimination: 주어진 자료를 이용해 classifier 를 구축하는 것

  - Classification: 구축된 classifier 를 이용해 unlabeled observation 의 class 를 예측하는 것

- LDA: Classifier 가 input variable 들의 linear combination 형태인 경우를 통칭함
- Welch (1939)의 방법
  - Bayes rule

    $$P(Y = C_l|X) = P(Y = C_l)P(X|Y = C_l) / \sum_l P(Y = C_l)P(X|Y = C_l)$$

  - 에 기초한 방법

  - 사후 확률을 기준으로 하는 classifier:

    $$l^* = \underset{l}{\operatorname{argmax}} P(Y = C_l|X) = \underset{l}{\operatorname{argmax}} P(Y = C_l)P(X|Y = C_l)$$

  -

  - 정규분포 가정 하의 classifier: $X|Y = C_l \sim N(\mu_l, \Sigma)$일 때,

    $$l^* = \underset{l}{\operatorname{argmax}}\{X^T\Sigma - 1\mu_l - 0.5\mu_l^T\Sigma \quad -1\mu_l + \log P(Y = C_l)\}$$

  - (linear in X)

  - c.f. $X|Y = C_l \sim N(\mu_l, \Sigma_l)$일 때,

    $$l^* = \underset{l}{\operatorname{argmax}}\{-0.5X^T\Sigma_l^{-1}X + X^T\Sigma_l^{-1}\mu_l - 0.5\mu_l^T\Sigma_l^{-1}\mu_l + \log P(Y = C_l)\}$$

  - (quadratic in X: QDA)
- Fisher (1936)의 방법
  - 그룹 내 분산(W) 대비 그룹 간 분산(B)을 최대로 하는 classifier 를 찾는 방법

    $$b^* = \underset{b}{\operatorname{argmax}} \frac{b^TBb}{b^TWb} = \text{the principal eigenvectors of} W^{-1}B$$

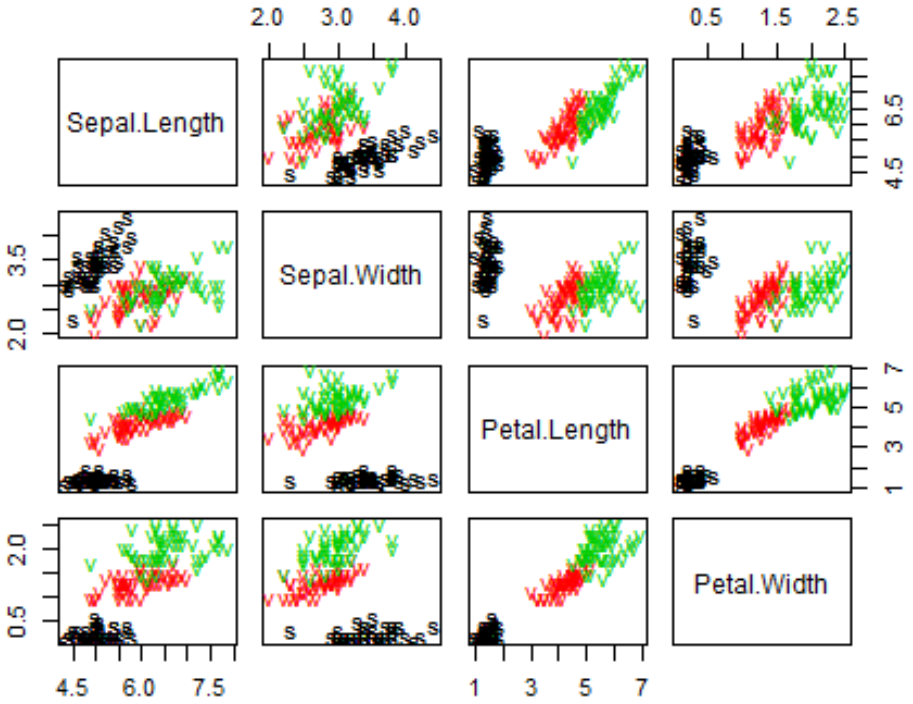
  -

  - Discriminant score = $b^{*T}X$ 에 기초해 분류 규칙을 결정
- 아래는 R 내장 데이터셋인 `iris` 에 대해 LDA 를 적용해 분석한 예임
  - 데이터 불러오기

```
data(iris)
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
plot(iris[,1:4], col=as.integer(iris$Species),
                 pch=substring((iris$Species), 1, 1))
```



- 위 플롯을 보면 데이터에 포함된 4 개 변수를 이용해 iris 의 종(Species)를 분류하는 것이 비교적 용이할 것으로 예상됨. 다만 versicolor 와 virginica 의 분류에서 약간의 오류 가능성이 있을 것으로 보임

- 각 종(Species)별로 30 개씩 비복원추출해 training set 결정

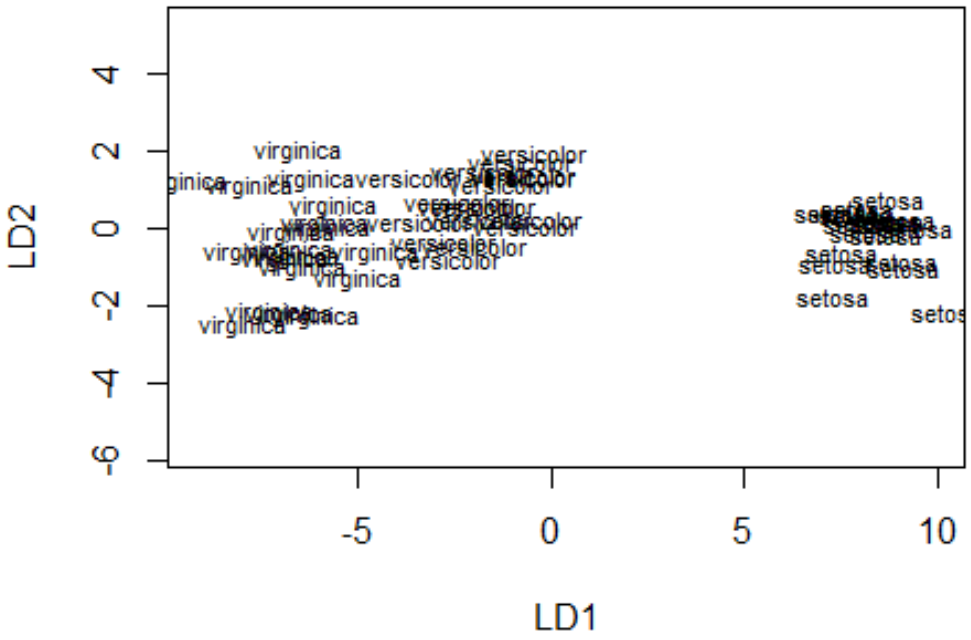
```
n <- 30
tr <- c(sample(1:50, n), sample(51:100, n), sample(101:150, n))
```

- MASS 패키지의 내장 함수인 lda()를 iris 데이터에 적용해 선형판별분석 실시한 후 분석 결과를 res 라는 이름의 객체로 저장

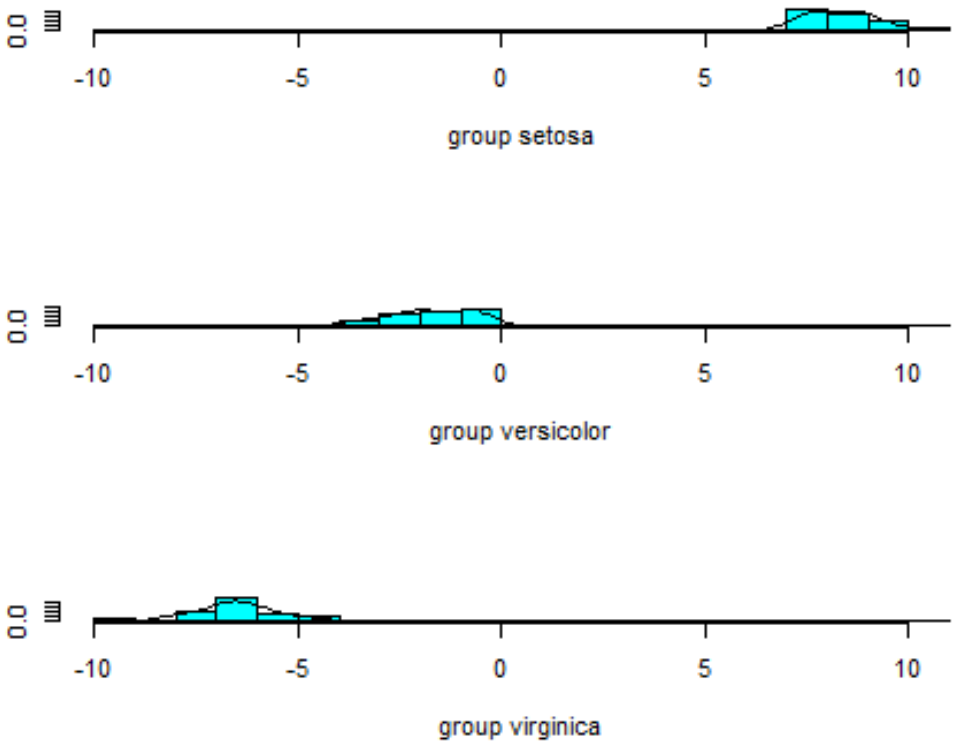
```
library(MASS)  # for lda()
res <- lda(Species~ ., data=iris[tr,], prior = c(1,1,1)/3, subset = tr)
```

- 분석 결과 시각화

```
plot(res)
```




```
plot(res, dimen=1, type="both")
```

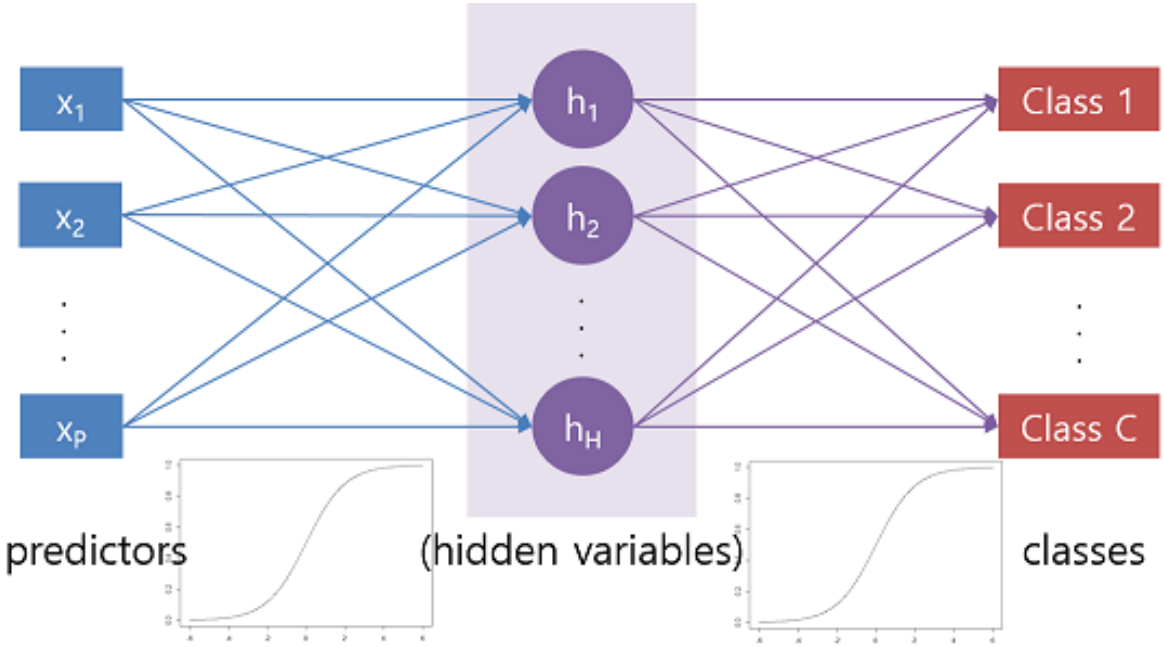


- test set 에 대해 예측 시행, 예측 성능 평가

```
pred <- predict(res, iris[-tr,-5])$class
actual <- iris[-tr,]$Species
table(pred, actual)

##             actual
## pred         setosa versicolor virginica
##   setosa         20          0         0
##   versicolor      0         19         1
##   virginica       0          1        19
```

신경망 모형을 이용한 분류

신경망 모형(Neural networks)은 인간 두뇌의 신경망 구조를 모방한 기계학습 알고리즘으로서, 입력값과 출력값 간에 복잡한 비선형 관계를 고려한 방법이다.
일반적으로 예측력이 우수한 방법론이나 해석이 어렵다는 문제 때문에 다소 활용 분야에 제약이 있다.



- 아래는 위에서 살펴본 `iris` 데이터의 분류 문제를 신경망 모형을 이용해 분석한 예임
  - 신경망 모형을 위한 R 패키지로 **nnet** 을 사용

```
library(nnet)
```

  - training set 에 대해 신경망 모형을 적합해 예측 모형 구축

```
ir1 <- nnet(Species~., data=iris[tr,], size=5, decay=.1)

## # weights:  43
## initial  value 93.713353
## iter  10 value 34.361598
## iter  20 value 21.233955
## iter  30 value 18.784817
## iter  40 value 17.022358
## iter  50 value 16.557264
## iter  60 value 16.210185
## iter  70 value 16.191405
## iter  80 value 16.188497
## final  value 16.188480
## converged

summary(ir1)

## a 4-5-3 network with 43 weights
## options were - softmax modelling  decay=0.1
```

```
##  b->h1 i1->h1 i2->h1 i3->h1 i4->h1
##  -1.56  -1.01  -1.03   1.34   2.64
##  b->h2 i1->h2 i2->h2 i3->h2 i4->h2
##   1.46   0.97   0.98  -1.29  -2.51
##  b->h3 i1->h3 i2->h3 i3->h3 i4->h3
##  -0.19  -0.26  -0.92   1.50   0.63
##  b->h4 i1->h4 i2->h4 i3->h4 i4->h4
##   0.18   0.25   0.92  -1.50  -0.62
##  b->h5 i1->h5 i2->h5 i3->h5 i4->h5
##   0.18   0.25   0.92  -1.50  -0.62
##  b->o1 h1->o1 h2->o1 h3->o1 h4->o1 h5->o1
##  -0.20  -0.85   0.66  -2.18   1.97   1.97
##  b->o2 h1->o2 h2->o2 h3->o2 h4->o2 h5->o2
##  -0.03  -2.44   2.28   1.61  -1.64  -1.64
##  b->o3 h1->o3 h2->o3 h3->o3 h4->o3 h5->o3
##   0.23   3.29  -2.94   0.57  -0.33  -0.33
```

- test set 에 대해 예측 실시 및 예측력 확인

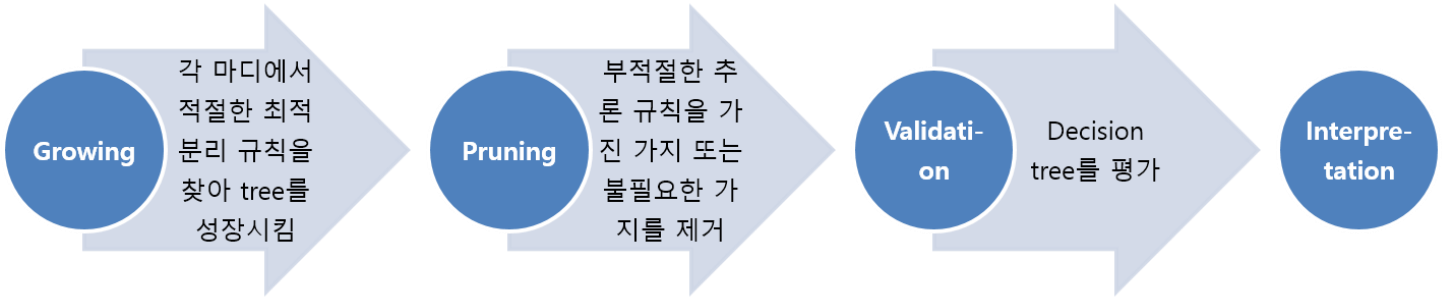
```
pred <- predict(ir1, iris[-tr,-5], type="class")
table(pred, actual)

##             actual
## pred         setosa versicolor virginica
##   setosa         20          0         0
##   versicolor      0         19         1
##   virginica       0          1        19
```

의사결정나무를 이용한 분류

의사결정나무(decision tree)를 이용한 분류는 각 변수의 영역을 반복적으로 분할해 전체 영역에서의 규칙을 생성하는 기법이다.
예측력은 여타 지도학습 방법에 비해 다소 떨어지지만, 해석력이 매우 좋기 때문에 선호되는 상황이 있다.
즉, If-then 으로 구성된 규칙이 도출되므로 이해가 쉽고 SQL 과 같은 데이터베이스 언어로 쉽게 구현 가능하다.

예측력과 해석력이 모두 중요하지만, 상황에 따라 다르다 할 것이다.
예를 들어 잠재고객이 가장 많은 반응을 보일 만한 고객유치방안을 예측하고자 하는 경우는 예측력이 중요하지만, 신용평가에서는 심사 결과 부적격 판정이 나와 고객에게 부적격 이유를 설명할 필요가 있는 경우는 해석력이 중요하다.

의사결정나무를 이용한 분류는 일반적으로 다음과 같은 프로세스를 따른다.

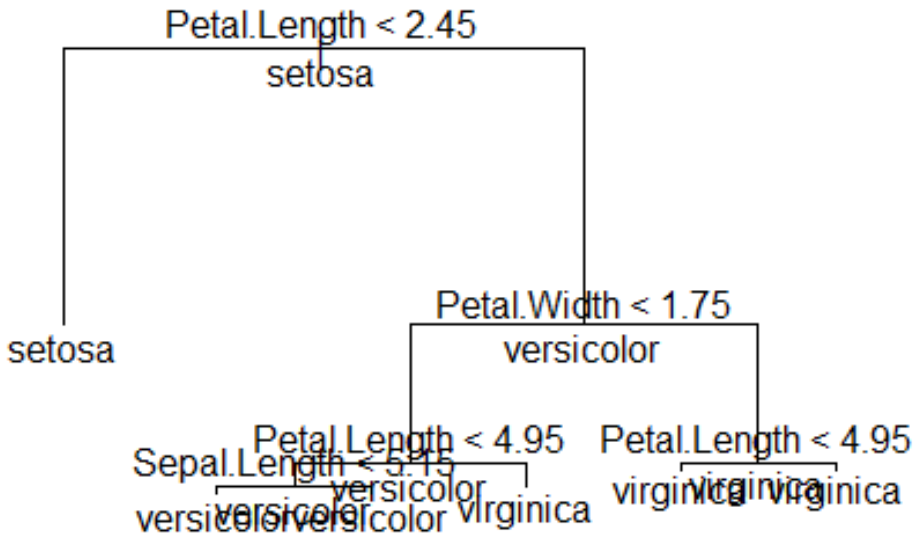


- 아래는 iris 데이터의 분류 문제를 의사결정나무를 이용해 분석한 예임
  - 의사결정나무 기법을 위해 tree 패키지의 tree() 함수 사용

```
library(tree)
iris.tr <- tree(Species~., iris)
```

  - 분류 결과 확인 및 시각화

```
summary(iris.tr)

##
## Classification tree:
## tree(formula = Species ~ ., data = iris)
## Variables actually used in tree construction:
## [1] "Petal.Length" "Petal.Width"  "Sepal.Length"
## Number of terminal nodes:  6
## Residual mean deviance:  0.1253 = 18.05 / 144
## Misclassification error rate: 0.02667 = 4 / 150

plot(iris.tr)
text(iris.tr, all=T)
```

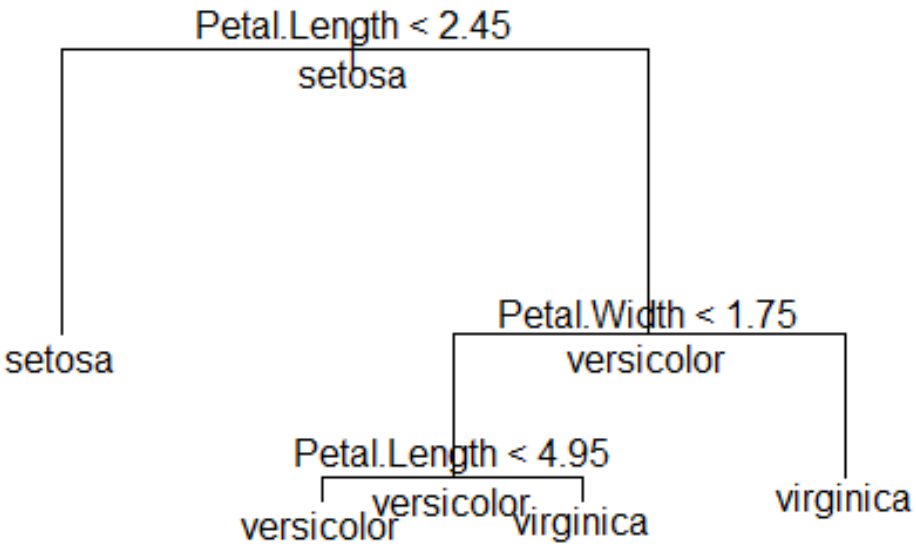


- 다소 의사결정규칙이 엉켜있으므로 가지치기(pruning) 실시

```
iris.tr2 <- prune.misclass(iris.tr, best=4)
```

- 새로운 분류 결과 확인 및 시각화

```
summary(iris.tr2)

##
## Classification tree:
## snip.tree(tree = iris.tr, nodes = c(7L, 12L))
## Variables actually used in tree construction:
## [1] "Petal.Length" "Petal.Width"
## Number of terminal nodes:  4
## Residual mean deviance:  0.1849 = 26.99 / 146
## Misclassification error rate: 0.02667 = 4 / 150

plot(iris.tr2)
text(iris.tr2, all=T)
```



D.2.2 자율학습에 의한 군집분석

군집분석(clustering)은 대표적인 자율학습(혹은 비지도학습)법으로서, 관측값을 적절한 기준에 따라 그룹화하는 규칙을 찾는 방법이다.
대표적 방법은 K-평균 군집법(K-means clustering), 계층적 군집법(hierarchical clustering) 등이 있다.

관측치 간의 유사성에 의해 군집을 형성하므로 유사성/비유사성을 측정하기 위한 적절한 거리 측도가 필요하다.

* Euclidean distance (L2 distance): 양적 데이터(quantitative data)에 사용

$$\sqrt{\sum_{i=1}^{d}(x_i - y_i)^2}$$

* Hamming distance (# of mismatches): 범주형 자료(categorical data)에 사용

$$\sum_{i=1}^{d} I(x_i \neq y_i)$$

* Manhattan (city block) distance (L1 distance): 양적 데이터(quantitative data)에 사용

$$\sum_{i=1}^{d} |x_i - y_i|$$

- 아래는 유럽 25 개 국가의 단백질 섭취에 대한 데이터에 대해 군집분석을 실시한 결과임
  - 단백질 관련 9 개 음식 군에 대한 섭생 정보를 이용해 25 개 국가를 그룹화하는 것을 목표로 함

  - 데이터 불러오기

```
protein <- read.table("./Data/protein.txt", sep="\t", header=T, fileEncod
ing="UTF-8")
summary(protein)
```

```
##           Country      RedMeat         WhiteMeat           Eggs
##  Albania       : 1   Min.   : 4.400   Min.   : 1.400   Min.   :0.500
##  Austria       : 1   1st Qu.: 7.800   1st Qu.: 4.900   1st Qu.:2.700
##  Belgium       : 1   Median : 9.500   Median : 7.800   Median :2.900
##  Bulgaria      : 1   Mean   : 9.828   Mean   : 7.896   Mean   :2.936
##  Czechoslovakia: 1   3rd Qu.:10.600   3rd Qu.:10.800   3rd Qu.:3.700
##  Denmark       : 1   Max.   :18.000   Max.   :14.000   Max.   :4.700
##  (Other)       :19
##       Milk            Fish           Cereals          Starch
##  Min.   : 4.90   Min.   : 0.200   Min.   :18.60   Min.   :0.600
##  1st Qu.:11.10   1st Qu.: 2.100   1st Qu.:24.30   1st Qu.:3.100
##  Median :17.60   Median : 3.400   Median :28.00   Median :4.700
##  Mean   :17.11   Mean   : 4.284   Mean   :32.25   Mean   :4.276
##  3rd Qu.:23.30   3rd Qu.: 5.800   3rd Qu.:40.10   3rd Qu.:5.700
##  Max.   :33.70   Max.   :14.200   Max.   :56.70   Max.   :6.500
##
##       Nuts           Fr.Veg
##  Min.   :0.700   Min.   :1.400
##  1st Qu.:1.500   1st Qu.:2.900
##  Median :2.400   Median :3.800
##  Mean   :3.072   Mean   :4.136
##  3rd Qu.:4.700   3rd Qu.:4.900
##  Max.   :7.800   Max.   :7.900
##
```

  - 분석에 사용할 변수 지정 및 scaling

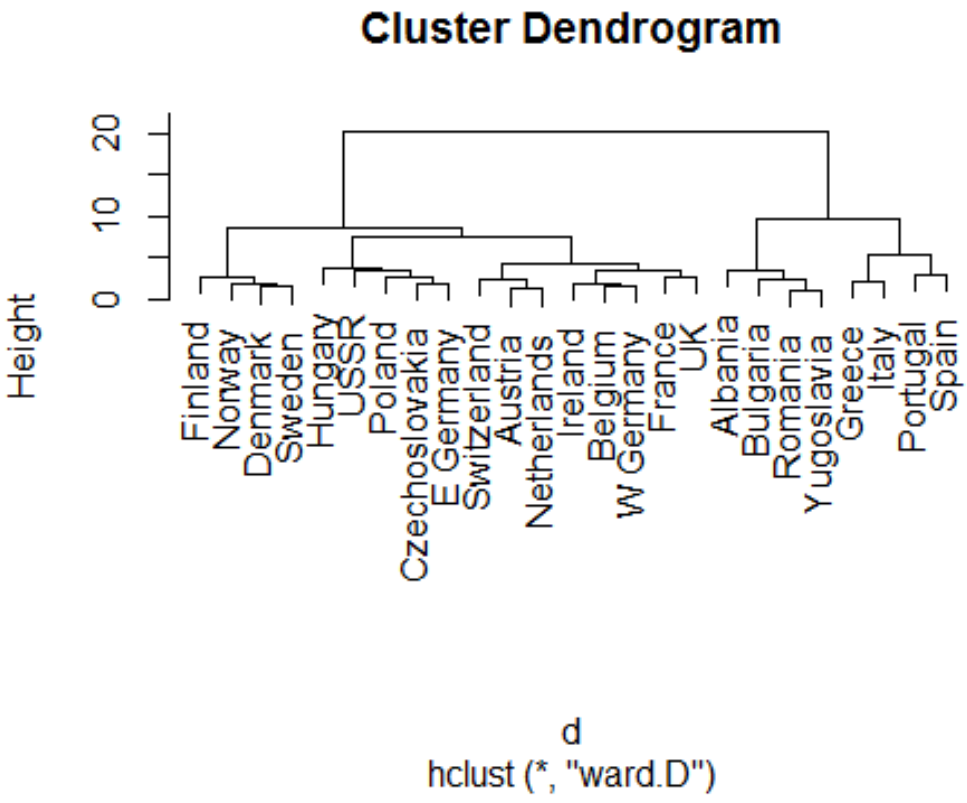
```
vars.to.use <- colnames(protein)[-1]
pmat <- scale(protein[,vars.to.use])
pcenter <- attr(pmat, "scaled:center")
pscale <- attr(pmat, "scaled:scale")
```

- Euclidean 거리에 기반(dist() 이용)한 계층적 군집분석 실시(hclust() 함수 이용)

```
d <- dist(pmat, method="euclidean") # Euclidean distances
pfit <- hclust(d, method="ward")  # hierarchical clustering

## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
```

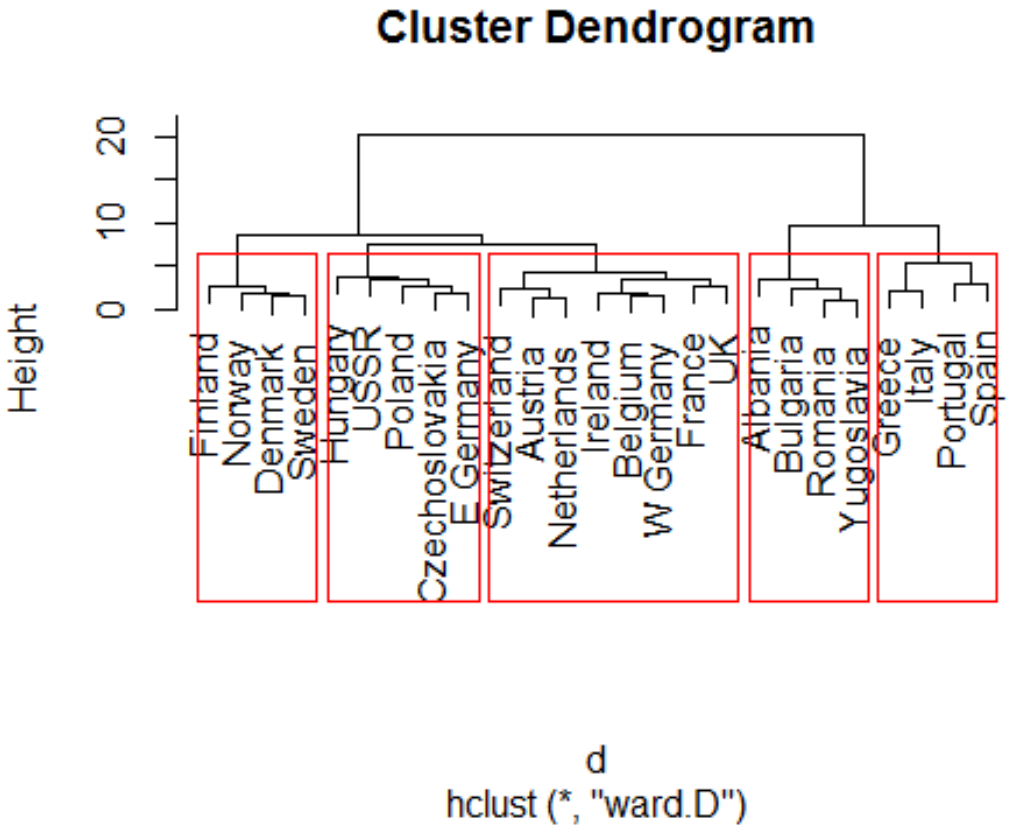
- 분석 결과 시각화를 위한 dendrogram 을 작성

```
plot(pfit, labels=protein$Country)  # draw dendrogram
```



- 5 개 군집을 형성함을 확인. dendrogram 위에 그룹별로 직사각형 표시

```
plot(pfit, labels=protein$Country)  # draw dendrogram
rect.hclust(pfit, k=5) # draw rentangles
```



- 25 개 국가를 5 개 그룹으로 할당한 그룹 인덱스

```
groups <- cutree(pfit, k=5)
```

- 각 그룹별 특성을 알아보기 위해 그룹별로 Country, RedMeat, Fish, Fr.Veg 등 4 개 변수 값 출력해보기

```
print.clusters <- function(labels, k) {
  for(i in 1:k) {
  print(paste("cluster", i))
  print(protein[labels==i,c("Country","RedMeat","Fish","Fr.Veg")])
  }
}

print.clusters(groups, 5)

## [1] "cluster 1"
##       Country RedMeat Fish Fr.Veg
## 1     Albania    10.1  0.2    1.7
## 4    Bulgaria     7.8  1.2    4.2
## 18    Romania     6.2  1.0    2.8
```

```
## 25 Yugoslavia     4.4  0.6    3.2
## [1] "cluster 2"
##        Country RedMeat Fish Fr.Veg
## 2      Austria     8.9  2.1    4.3
## 3      Belgium    13.5  4.5    4.0
## 9       France    18.0  5.7    6.5
## 12     Ireland    13.9  2.2    2.9
## 14 Netherlands     9.5  2.5    3.7
## 21 Switzerland    13.1  2.3    4.9
## 22          UK    17.4  4.3    3.3
## 24   W Germany    11.4  3.4    3.8
## [1] "cluster 3"
##           Country RedMeat Fish Fr.Veg
## 5  Czechoslovakia     9.7  2.0    4.0
## 7       E Germany     8.4  5.4    3.6
## 11        Hungary     5.3  0.3    4.2
## 16         Poland     6.9  3.0    6.6
## 23           USSR     9.3  3.0    2.9
## [1] "cluster 4"
##    Country RedMeat Fish Fr.Veg
## 6  Denmark    10.6  9.9    2.4
## 8  Finland     9.5  5.8    1.4
## 15  Norway     9.4  9.7    2.7
## 20  Sweden     9.9  7.5    2.0
## [1] "cluster 5"
##     Country RedMeat Fish Fr.Veg
## 10   Greece    10.2  5.9    6.5
## 13    Italy     9.0  3.4    6.7
## 17 Portugal     6.2 14.2    7.9
## 19    Spain     7.1  7.0    7.2
```

- K-평균 군집법 적용

```
pKmeans <- kmeans(pmat, 5) # K means clustering
summary(pKmeans)

##              Length Class  Mode
## cluster      25     -none- numeric
## centers      45     -none- numeric
## totss         1     -none- numeric
## withinss      5     -none- numeric
## tot.withinss  1     -none- numeric
## betweenss     1     -none- numeric
## size          5     -none- numeric
## iter          1     -none- numeric
## ifault        1     -none- numeric
```

- K-평균 군집법으로 생성된 그룹별 특징 알아보기

```
pKmeans$centers
```

```
##        RedMeat  WhiteMeat        Eggs       Milk       Fish    Cereals
## 1 -0.068119111 -1.0411250 -0.07694947 -0.2057585  0.1075669  0.6380079
## 2  0.006572897 -0.2290150  0.19147892  1.3458748  1.1582546 -0.8722721
## 3 -0.790141851 -0.5267887 -1.16557572 -0.9047559 -0.9504683  1.4383272
## 4  0.613615213  0.7738178  0.71613441  0.3066547 -0.2598064 -0.5146342
## 5 -0.949484801 -1.1764767 -0.74802044 -1.4583242  1.8562639 -0.3779572
##       Starch       Nuts     Fr.Veg
## 1 -1.3010340  1.4997366  1.3659270
## 2  0.1676780 -0.9553392 -1.1148048
## 3 -0.7604664  0.8870168 -0.5373533
## 4  0.4208082 -0.6131165  0.1060327
## 5  0.9326321  1.1220326  1.8925628
```

```
pKmeans$size
```

```
## [1]  2  4  6 11  2
```

```
print.clusters(pKmeans$cluster, 5)
```

```
## [1] "cluster 1"
##    Country RedMeat Fish Fr.Veg
## 10  Greece    10.2  5.9    6.5
## 13   Italy     9.0  3.4    6.7
## [1] "cluster 2"
##    Country RedMeat Fish Fr.Veg
## 6  Denmark    10.6  9.9    2.4
## 8  Finland     9.5  5.8    1.4
## 15  Norway     9.4  9.7    2.7
## 20  Sweden     9.9  7.5    2.0
## [1] "cluster 3"
##       Country RedMeat Fish Fr.Veg
## 1     Albania    10.1  0.2    1.7
## 4    Bulgaria     7.8  1.2    4.2
## 11    Hungary     5.3  0.3    4.2
## 18    Romania     6.2  1.0    2.8
## 23       USSR     9.3  3.0    2.9
## 25 Yugoslavia     4.4  0.6    3.2
## [1] "cluster 4"
##           Country RedMeat Fish Fr.Veg
## 2         Austria     8.9  2.1    4.3
## 3         Belgium    13.5  4.5    4.0
## 5  Czechoslovakia     9.7  2.0    4.0
## 7       E Germany     8.4  5.4    3.6
## 9          France    18.0  5.7    6.5
## 12        Ireland    13.9  2.2    2.9
## 14    Netherlands     9.5  2.5    3.7
## 16         Poland     6.9  3.0    6.6
## 21    Switzerland    13.1  2.3    4.9
```

```
## 22             UK    17.4  4.3    3.3
## 24      W Germany    11.4  3.4    3.8
## [1] "cluster 5"
##     Country RedMeat Fish Fr.Veg
## 17 Portugal     6.2 14.2    7.9
## 19    Spain     7.1  7.0    7.2
```