Part2 : R 통계패키지 & 알고리즘 기반

머신 러닝
Machine Learning

목차

iris Flowers

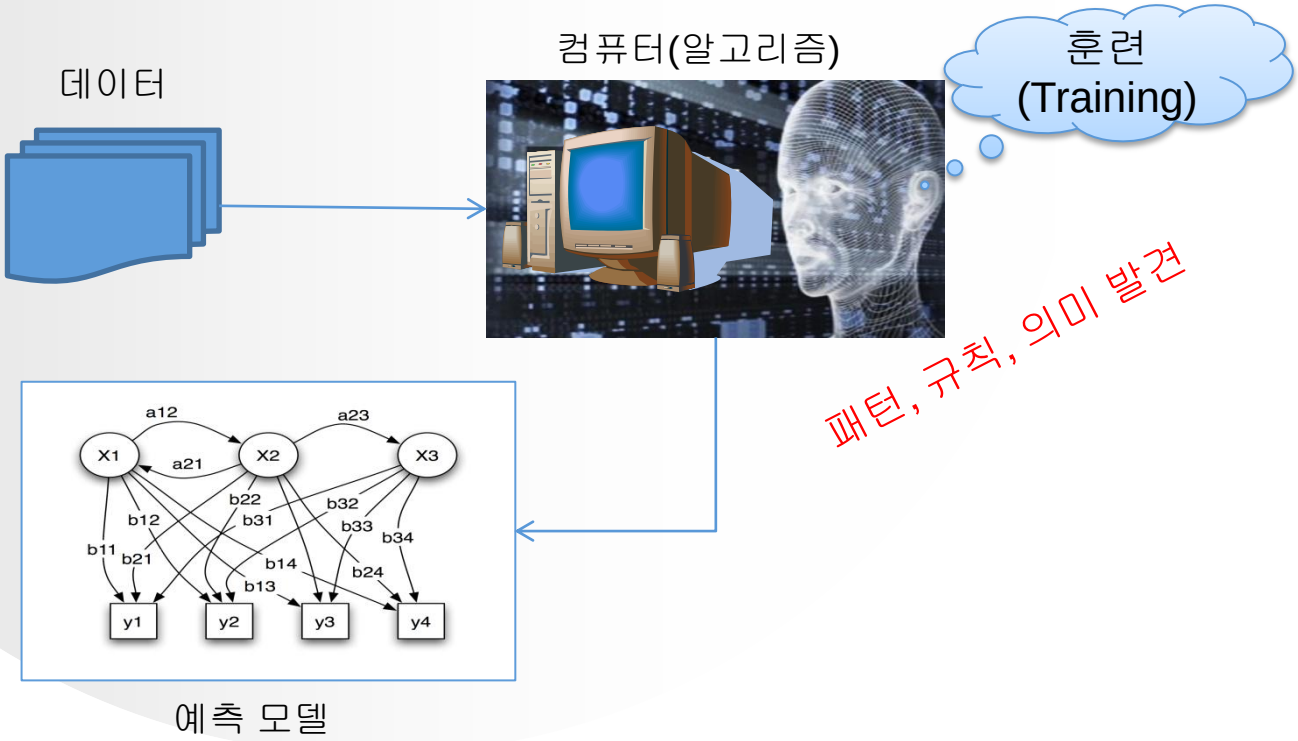
1. ML 데이터 전처리

chap01_ML_DataPreprocessing 수업내용

1. 머신 러닝(Machine Learning) 개요
2. 데이터 모양변경(Data Reshape)
   - ✓ 수집한 원형 데이터를 데이터 분석의 목적에 맞게 데이터를 가공.처리하는 과정
   - ✓ 데이터 리모델링 관련 패키지 학습
   - ✓ 관련패키지 : plyr, dplyr, reshape, reshape2
3. 텍스트 전처리
   - ✓ 텍스트 마이닝 패키지 이용 텍스트 데이터 전처리
   - ✓ 관련패키지 : tm

1) Machine Learning 개요

- **기계학습(Machine Learning)**
  - 인공지능(Artificial Intelligence)분야, 1950년대 연구 시작
  - 빅 데이터 분석과 Deep Learning 이슈로 최근 새롭게 조명
  - 기계학습(머신 러닝)은 데이터에 내재된 패턴, 규칙, 의미 등을 알고리즘을 기반으로 컴퓨터가 스스로 학습 하도록 하여 새롭게 입력되는 데이터에 대한 결과를 예측하게 하는 기술



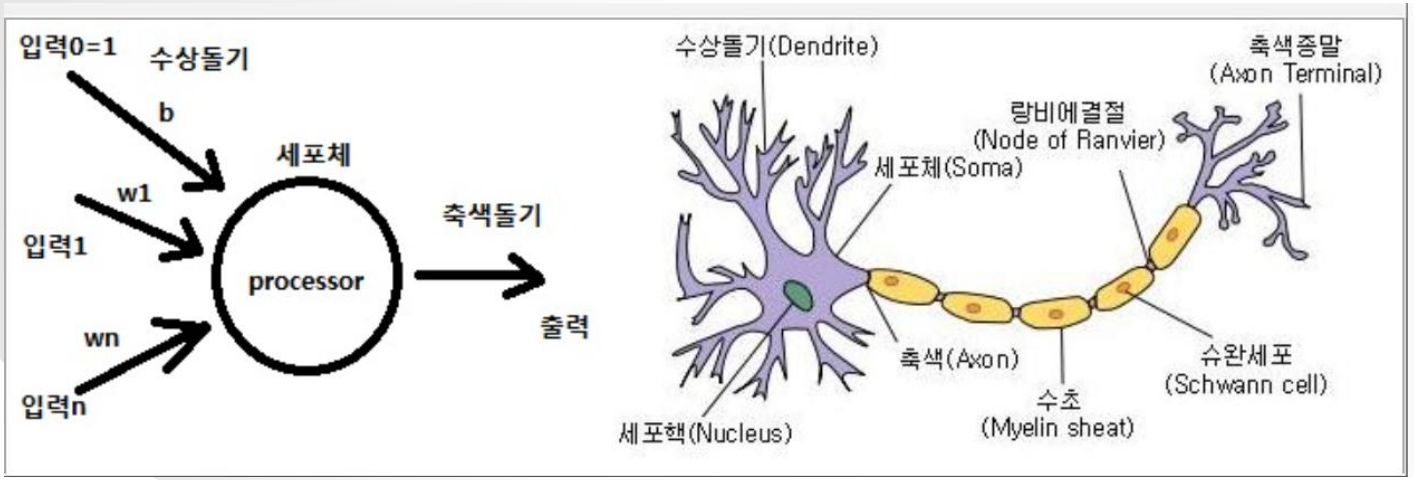
1) Machine Learning 개요

- **딥 러닝(Deep Learning)**

  컴퓨터가 여러 데이터를 이용해 사람처럼 스스로 학습할 수 있도록 인공 신경망(ANN: artificial neural network)을 기반으로 한 기계 학습

  인간의 두뇌가 수많은 데이터 속에서 패턴을 발견한 뒤 사물을 구분하는 정보처리 방식을 모방해 컴퓨터가 사물을 분별하는 기계 학습
  인간 개입 없이 컴퓨터가 스스로 인지·추론·판단

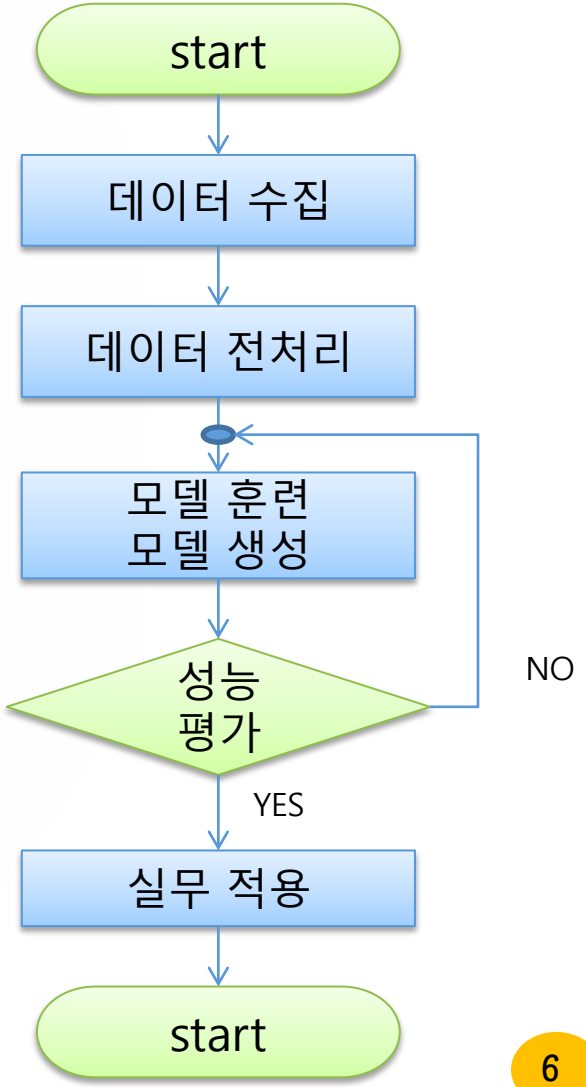
  음성, 이미지 인식, 사진 분석 등 광범위하게 활용, 구글 알파고



1) Machine Learning 개요

- **기계학습 적용 단계**

1. 데이터 수집 단계
   - ✓ File, DB, Web, SNS 등
2. 데이터 전처리(준비/탐구)
   - ✓ 인간이 해석(80% 작업)
3. 모델 훈련/ 모델생성
   - ✓ 데이터 학습(알고리즘) -> 모델 생성
4. 모델 성능 평가
   - ✓ 모델 평가(테스트 데이터 이용 정확도)
5. 모델 성능 향상/업무 적용
   - ✓ 모델 변경, 추가 데이터 실험, 실무 적용



1) Machine Learning 개요

● 기계학습 알고리즘

분류		알고리즘	적용분야
지도	회귀모델	선형회귀	수치예측
		모델 트리	
	분류모델	최근접 이웃(kNN)	분류예측
		나이브 베이즈(NB)	
		결정트리(DT)/랜덤 포레스트(RF)	
	블랙박스모델	서포트 벡터 머신(SVM)	다중 용도
		신경망(ANN)	
비지도	군집모델	K평균 군집화	군집 예측
	연관모델	연관규칙	패턴 예측

1) Machine Learning 개요

● 혼돈(confusion) 매트릭스 : 예측 값과 실제 값을 비교하는 표

	NO	YES
NO	참 부정(TN)	거짓 긍정(FP)
YES	거짓 긍정(FN)	참 긍정(TP)

위험

$$\text{예측 정확도} = \frac{TN + TP}{TN + FP + FN + TP}$$

$$\text{오차비율} = \frac{FP + FN}{TN + FP + FN + TP}$$

2) 데이터 모양변경

1. Plyr 패키지 활용

1) join() : 데이터 병합 하기

```
install.packages('plyr')
library(plyr) # 패키지 로딩

# 병합할 데이터프레임 셋 만들기
x = data.frame(ID = c(1,2,3,4,5), height = c(160,171,173,162,165))
y = data.frame(ID = c(5,4,1,3,2), weight = c(55,73,60,57,80))

# join() : plyr패키지 제공 함수
z <- join(x,y,by='ID') # ID컬럼으로 조인
z
```

	ID	height	weight
1	1	160	60
2	2	171	80
3	3	173	57
4	4	162	73
5	5	165	55

2) 데이터 모양변경

- **Left 조인**

```
x = data.frame(ID = c(1,2,3,4,6), height = c(160,171,173,162,180))
y = data.frame(ID = c(5,4,1,3,2), weight = c(55,73,60,57,80))

# left 조인
z <- join(x,y,by='ID') # ID컬럼으로 left 조인
z
```

```
 ID height weight
1  1    160     60
2  2    171     80
3  3    173     57
4  4    162     73
5  6    180     NA    <- 결측치이면 NA 처리됨
```

2) 데이터 모양변경

- **Inner 조인**

z <- join(x,y,by='ID', type='inner') # type='inner' : 값이 있는 것만 조인
z

```
 ID height weight
1  1    160     60
2  2    171     80
3  3    173     57
4  4    162     73
```

2) 데이터 모양변경

- **Full 조인**

z <- join(x,y,by='ID', type='full') # type='full' : 모든 항목 조인
z

```
#ID height weight
#1  1    160     60
#2  2    171     80
#3  3    173     57
#4  4    162     73
#5  6    180     NA
#6  5     NA     55
```

2) 데이터 모양변경

- key값으로 조인

```
x = data.frame(key1 = c(1,1,2,2,3), key2 = c('a','b','c','d','e'),
               val1 = c(10,20,30,40,50))
y = data.frame(key1 = c(3,2,2,1,1), key2 = c('e','d','c','b','a'),
               val2 = c(500,400,300,200,100))
x;y
join(x,y,by=c('key1', 'key2'))
```

```
#  key1 key2 val1 val2
#1    1    a   10  100
#2    1    b   20  200
#3    2    c   30  300
#4    2    d   40  400
#5    3    e   50  500
```

2) 데이터 모양변경

- **tapply() 이용한 집단별(그룹별) 통계치 구하기(활용도 높음)**

head(iris) # iris 데이터 보기

Sepal.Length(꽃받침 길이), Sepal.Width(꽃받침 너비),

Petal.Length(꽃잎 길이), Petal.Width(꽃잎 너비),

Species(꽃의 종류)

unique(iris$Species) # 꽃의 종류 보기 - 3가지(전체 150개)

예) 꽃의 종류별(Species) 꽃받침 길이(Sepal.Length)의 평균 구하기

tapply 함수 형식) tapply(적용data, 집단변수, 함수)

tapply(iris$Sepal.Length, iris$Species, mean) # 종별 평균치

#setosa versicolor virginica

#5.006 5.936 6.588

2) 데이터 모양변경

- tapply()함수 특징
  - ✓ 집단별로 통계치를 구해주는 기본 함수
  - ✓ 한 번에 하나의 통계치(mean, min, max 등)만 계산
  - ✓ 한 번에 두 이상의 통계치 계산
    - ➢ ddply() 함수 : plyr 패키지 제공

2) 데이터 모양변경

2) ddply() : 집단 변수 대상 통계치 구하기

ddply() : plyr 패키지 제공 함수

형식) ddply(dataframe, .(집단변수), 요약집계, 컬럼명=함수(변수))

설명) dataframe의 집단변수를 기준으로 변수에 함수를 적용하여
컬럼명으로 표현

2) 데이터 모양변경

예제) ddply() 이용 꽃의 종류별로 꽃받침 길이의 평균을 구하여
데이터 프레임 형식으로 저장

```
a <- ddply(iris, .(Species), summarise, avg = mean(Sepal.Length))
# summarise : 새로운 데이터 프레임 형태로 요약
a
# 꽃 종별로 꽃받침 길이의 평균, 표준편차, 최댓값, 최솟값 한번에 계산
b <- ddply(iris, .(Species), summarise,
      avg=mean(Sepal.Length),  std = sd(Sepal.Length),
      max = max(Sepal.Length), min=min(Sepal.Length))
# round() 적용
b <- ddply(iris, .(Species), summarise,
       avg=round(mean(Sepal.Length),2), std=round(sd(Sepal.Length),2),
       max=max(Sepal.Length), min=min(Sepal.Length))
```

	Species	avg	std	max	min
1	setosa	5.01	0.35	5.8	4.3
2	versicolor	5.94	0.52	7.0	4.9
3	virginica	6.59	0.64	7.9	4.9

2) 데이터 모양변경

<연습문제> 차종별 실린더 수와 기어 단수 별로 연비의 평균과 표준편차 통계치를 계산하시오.

1) 데이터 셋 가져오기

```
head(mtcars)

str(mtcars) # data.frame':      32 obs. of  11 variables:
################## mtcars 데이터셋 ####################
# R의 기본 내장 데이터셋
# 32대의 자동차에 대한 특징을 11가지 변수로 제공
# 주요 변수 : 연비(mpg:miles per gallon),실린더수(cyl),
#                 배기량(displacement),기어단수(gear)
#######################################################
```

2) 통계치 계산

2) 데이터 모양변경

2. dplyr 패키지 활용

✓ **데이터를 분석에 필요한 형태로 만드는 데이터 전처리 관련 함수 제공 패키지**

✓ **dplyr 패키지의 주요 함수**

- tbl_df() : 데이터셋 화면창 크기 만큼만 데이터 제공
- filter() : 지정한 조건식에 맞는 데이터 추출- subset()
- select() : 열의 추출 - data[, c("Year", "Month")]
- mutate() : 열 추가 - transform()
- arrange() : 정렬 - order(), sort()
- summarise() : 집계 - aggregate()

Hadley Wickham 개발 패키지 : ggplot2, plyr, reshape2

plyr은 R로 개발, dplyr은 C++ 개발(빠른 처리 속도)

2) 데이터 모양변경

- dplyr 패키지와 데이터 셋 hflight 설치

```
install.packages(c("dplyr", "hflights"))
library(dplyr)
library(hflights)

################### hflights 데이터셋 ################
# 2011년도 미국 휴스턴에서 출발하는 모든 비행기의
# 이착륙기록이 수록된 것으로 227,496건의 이착륙기록에
# 대해 21개 항목을 수집한 데이터
#####################################################
```

2) 데이터 모양변경

- **dim(hflights) # 차원보기**

[1] 227496 21

- **tbl_df() 함수 : 데이터셋 화면창 크기 만큼 데이터 제공**

hflights_df <- tbl_df(hflights)

hflights_df

Source: local data frame [227,496 x 21]

2) 데이터 모양변경

1. filter(dataframe, 조건1, 조건2)함수 이용 데이터 추출

```
# 1월 1일 데이터 추출
filter(hflights_df,  Month == 1, DayofMonth == 1) # and(, or &)
# Source: local data frame [552 x 21]
# 1월 혹은 2월 데이터 추출
filter(hflights_df, Month == 1 | Month == 2) # or(|)
# Source: local data frame [36,038 x 21]
```

2) 데이터 모양변경

2. arrange()함수를 이용한 오름차순/내림차순 정렬

```
# 년도,월,도착시간 오름차순
arrange(hflights_df, Year, Month, ArrTime )

# Month 기준 내림차순 정렬 - desc(변수)
arrange(hflights_df, desc(Month))
```

2) 데이터 모양변경

3. select(), mutate()함수를 이용한 열 조작

Year, Month, DayOfWeek 열을 추출
select(hflights_df, Year, Month, DayOfWeek)

Month기준 내림차순 정렬 -> Year, Month, AirTime, ArrDelay 컬럼 추출
select(arrange(hflights_df, desc(Month)),Year, Month, AirTime, ArrDelay)

Year~DayOfWeek 추출(Year, Month, DayofMonth, DayofWeek)
select(hflights_df, Year:DayOfWeek)

Year부터 DayOfWeek를 제외한 나머지 열 추출
select(hflights_df, -(Year:DayOfWeek))

2) 데이터 모양변경

4. mutate() 함수를 이용하여 열 추가(변형)

```
# gain 변수 추가 -> gain_per_hour 변수 계산에 사용할 수 있음
mutate(hflights_df, gain = ArrDelay - DepDelay,
                    gain_per_hour = gain/(AirTime/60))
# 새로 만든 열을 같은 함수 안에서 바로 사용 가능
```

.. ..

Variables not shown: ActualElapsedTime (int), AirTime (int), ArrDelay (int), DepDelay (int), Origin (chr), Dest (chr), Distance (int), TaxiIn (int), TaxiOut (int), Cancelled (int), CancellationCode (chr), Diverted (int), **gain (int), gain_per_hour(dbl)**

2) 데이터 모양변경

새로 추가된 열을 같은 함수 안에서 select() 함수로 보기

주의 : 생성된 열은 hflights_df에 추가되지 않음 - 추가된 열 결과 보기

```
select(mutate(hflights_df, gain = ArrDelay - DepDelay,
              gain_per_hour = gain/(AirTime/60)),
       Year, Month, ArrDelay, DepDelay,gain, gain_per_hour)
```

```
#Source: local data frame [227,496 x 6]

#Year Month ArrDelay DepDelay gain gain_per_hour
#1  2011     1       -10         0  -10    -15.000000
#2  2011     1        -9         1  -10    -13.333333
```

2) 데이터 모양변경

5. summarise()함수를 이용한 집계

```
#  n(), sum(), mean(), sd(), var(), median() 등의 함수 사용 - 기초 통계량
summarise(hflights_df, n()) # n() : dataframe의 row값 리턴 함수
# 평균 출발지연시간 계산
summarise(hflights_df, cnt = n(), delay=mean(DepDelay, na.rm = TRUE))
# Source: local data frame [1 x 1] -> 결과는 data frame

#      cnt     delay
#1 227496 9.444951
```

2) 데이터 모양변경

<연습문제1> 평균 비행시간(AirTime)을 구하시오
108.1423

<연습문제2)> n(), sum()를 이용하여 평균 비행시간을 구하시오.

```
#      cnt       sum       avg
#1 227496 24210257 106.4206 <- 1.7217 차이발생(NA 원인)
```

<연습문제3> 평균차이를 제거하시오.

```
#     cnt       sum       avg
#1 223874 24210257 108.1423
```

<연습문제4> 도착시간(ArrTime) 표준편차와 분산(표준편차의 제곱근)
1 472.4017
1 223163.4

2) 데이터 모양변경

6. group_by(dataframe, 기준변수)함수를 이용한 그룹화

데이터 프레임을 대상으로 기준변수로 그룹화

예문) 항공기별로 비행편수가 20편 이상, 평균 비행 거리 2,000마일 이내의 평균 연착시간

1) 비행편수를 구하기 위해서 항공기별 그룹화

```
planes <- group_by(hflights_df, TailNum) # TailNum : 항공기 일련번호 그룹
planes
```

2) 항공기별 필요한 변수 요약

```
planesInfo <- summarise(planes, count = n(), dist=mean(Distance, na.rm=T),
                        delay=mean(ArrDelay, na.rm = T))
planesInfo
#Source: local data frame [3,320 x 4]

#    TailNum  count     dist          delay
#    항공기   비행편수  평균비행거리  평균연착시간
#1           795       938.7157         NA
#2   N0EGMQ   40        1095.2500   1.918919
```

2) 데이터 모양변경

str(planesInfo) # Classes 'tbl_df'

planesInfo 객체에 count, dist, delay 변수가 추가됨

3) planesInfo객체를 대상으로 조건 지정

result <- filter(planesInfo, count > 20, dist < 2000)

result

Source: local data frame [1,526 x 4]

```
   TailNum count      dist     delay
1           795  938.7157        NA
2   N0EGMQ   40 1095.2500  1.918919
3   N10156  317  801.7192  8.199357
```

2) 데이터 모양변경

```
#####################################################
# reshape 패키지 활용
#####################################################
result <- read.csv("C:/Rwork/Part-II/reshape.csv", header=FALSE)
head(result) # V1   V2   V3   V4

install.packages("reshape")  # 패키지 설치
library(reshape)
# rename() 함수 이용 컬럼명 지정
result <- rename(result, c(V1="total", V2="num1", V3="num2", V4="num3"))
head(result)
```

```
  total num1 num2 num3
1   5.1    1.4    0.2    3.5
2   4.9    1.4    0.2    3.0
```

2) 데이터 모양변경

3. reshape2 패키지 활용
   - ✓ 긴 형식 <-> 넓은 형식
   - ✓ reshape2 is a reboot of the reshape package.
   - ✓ reboot : 기본골격만을 대상으로 패키지 개발
   - ✓ reshape2 주요 함수
     - ➢ dcast() : 긴 형식 -> 넓은 형식으로 모양 변경
     - ➢ melt() : 넓은 형식 -> 긴 형식으로 모양 변경

2) 데이터 모양변경

```
install.packages('reshape2')
library(reshape2)
# '긴 형식'(Long format)과 '넓은 형식'(wide format)으로 데이터 모양 변경

data <- read.csv("c:/Rwork/Part-II/data.csv")
data
# '넓은 형식'(wide format)으로 변형
wide <- dcast(data, Customer_ID ~ Date, sum)
# 형식) dcast(wide frame 변경 데이터셋, 앞변수~뒤변수, 함수)
# 해설) 데이터셋을 대상으로 앞변수와 뒤변수의 값이 같은 경우 함수 적용)

# 예) data 대상으로 Customer_ID와 Data변수가 같은 Buy변수에 합계 계산)
# Using Buy as value column: use value.var to override. - 정리 대상 변수(Buy)

wide # Buy 변수에 함수 적용 <- 정리대상 변수
```

2) 데이터 모양변경

	Date	Customer_ID	Buy
1	20140101	1	3
2	20140101	2	4
3	20140102	1	2
4	20140102	4	6
5	20140102	5	1
6	20140103	1	5
7	20140103	2	1
8	20140103	4	8
9	20140103	5	5
10	20140104	1	5
11	20140104	2	8
12	20140104	3	5
13	20140105	5	6
14	20140106	2	6
15	20140106	3	6
16	20140107	1	9
17	20140107	5	7

Customer_ID	20140101	20140102	20140103	20140104	20140105	20140106	20140107
1	1	3	2	5	5	0	0	9
2	2	4	0	1	8	0	6	0
3	3	0	0	0	5	0	6	0
4	4	0	6	8	0	0	0	0
5	5	0	1	5	0	6	0	7

2) 데이터 모양변경

- **열 또는 행 단위 통계치 계산 함수**
  - ✓ **colSums(), rowSums(), colMeans(), rowMeans()**

rowSums 와 colSums 이용 날짜별, 고객별 구매횟수의 합계 구하기

rowSums(wide) # Date Customer_ID Buy

[1] 25 21 14 18 24

#	Customer_ID	20140101	20140102	20140103	20140104	20140105	20140106	20140107	
#1	1	3	2	5	5	0	0	9	<- 25
#2	2	4	0	1	8	0	6	0	<- 21
#3	3	0	0	0	5	0	6	0	<- 14
#4	4	0	6	8	0	0	0	0	<- 18
#5	5	0	1	5	0	6	0	7	<- 24

colSums(wide)

# Customer_ID	20140101	20140102	20140103	20140104	20140105	20140106	20140107
#	15	7	9	19	18	6	12	16

2) 데이터 모양변경

- **원래 데이터 셋에 데이터 붙이기**
  - ✓ cbind와 rbind() 이용하여 원래 데이터 셋에 붙임

wide <- cbind(wide, rowSums(wide)) # 컬럼으로 행 합계 붙임

wide <- rbind(wide, colSums(wide)) # 행으로 컬럼 합계 붙임

wide

#Customer_ID	20140101	20140102	20140103	20140104	20140105	20140106	20140107	rowSums(wide)
#1	1	3	2	5	5	0	0	9	25
#2	2	4	0	1	8	0	6	0	21
#3	3	0	0	0	5	0	6	0	14
#4	4	0	6	8	0	0	0	0	18
#5	5	0	1	5	0	6	0	7	24
#6	15	7	9	19	18	6	12	16	102

2) 데이터 모양변경

- **컬럼명 변경 : 'Sum by User'**

```
colnames(wide)
colnames(wide)[9] <- 'Sum by User' # 9번째 컬럼명 변경

# 특정 셀값 변경 : 'Sum by day'
wide[6,1] <- 'Sum by day'
wide
```

# Customer_ID	20140101	20140102	20140103	20140104	20140105	20140106	20140107	Sum by User
#1	1	3	2	5	5	0	0	9	25
#2	2	4	0	1	8	0	6	0	21
#3	3	0	0	0	5	0	6	0	14
#4	4	0	6	8	0	0	0	0	18
#5	5	0	1	5	0	6	0	7	24
#6	Sum by day	7	9	19	18	6	12	16	102

2) 데이터 모양변경

- **필요한 컬럼만 보기**

  wide[,-3:-5] # 3~5열 제외

  wide[-4,-3:-5] # 4행, 3~5열 제외

  wide[-4,c(-3,-5,-7)] # 4행, 3,5,7열 제외

Customer_ID	20140101	20140103	20140105	20140107	Sum by User
1	1	3	5	0	9	25
2	2	4	1	0	0	21
3	3	0	0	0	0	14
5	5	0	5	6	7	24
6	Sum by day	7	19	6	16	102

- **처리결과 파일에 저장**

  setwd("c:/Rwork/Part-II")

  write.csv(wide, 'wide.csv', row.names=FALSE) # 행번호 저장 안함

2) 데이터 모양변경

```
# 긴 형식 변경을 위한 데이터셋 생성
wide <- wide[-6,-9] # 6행과 9컬럼 제거
wide
```

- **melt() 함수 이용 : 넓은 형식 -> 긴 형식 변경**

```
# 형식) melt(long frame 변경 데이터셋, id='열이름 변수')
# 해설) 데이터셋을 대상으로 id로 지정된 변수 기준으로 긴 형식 모양 변경)
x <- melt(wide, id='Customer_ID')
x
# x객체의 컬럼명을 확인한 후 UserID, Date, Buy 이름으로 컬럼명 수정
colnames(x)
name <- c("UserID", "Date", "Buy")
colnames(x) <- name
head(x)
```

3) 텍스트 전처리

- 텍스트 마이닝(Text Mining)
  - 정형화 되지 않은 텍스트를 대상으로 유용한 정보를 찾는 기술
  - 토픽분석, 감성분석 등에서 이용
  - 텍스트 마이닝에서 가장 중요한 것은 형태소 분석
  - 형태소 분석 : 문장을 분해 가능한 최소한의 단위로 분리하는 작업
  - 예) “우리나라는 대한민국 입니다." -> 우리나라(명사) + 는(조사) + 대한민국(명사) + 입니다.(조사)

3) 텍스트 전처리

- **텍스트 전처리 과정**

```
install.packages('tm')
install.packages('SnowballC') # stemDocument() 함수 제공
library(tm)
library(SnowballC)
sms_corpus = Corpus(VectorSource(sms_data$text)) # 1) 말뭉치 생성(vector -> corpus 변환)
sms_corpus = tm_map(sms_corpus, content_transformer(tolower))  # 2) 소문자 변경
sms_corpus = tm_map(sms_corpus, removeNumbers) # 3) 숫자 제거
sms_corpus = tm_map(sms_corpus, removePunctuation) # 4) 문자부호(콤마 등) 제거
sms_corpus = tm_map(sms_corpus, removeWords, stopwords("SMART")) # 5) 불용어 제거 제거
sms_corpus = tm_map(sms_corpus, stripWhitespace) # 6) 여러 공백 제거(공백 제거)
sms_corpus = tm_map(sms_corpus, stemDocument) # 7) 유사 단어 어근 처리
sms_corpus = tm_map(sms_corpus, stripWhitespace) # 8) 여러 공백 제거(공백 제거)
sms_dtm = DocumentTermMatrix(sms_corpus) # 9) 문서와 단어 집계표 작성
```

3) 텍스트 전처리

- **DocumentTermMatrix**
  - ✓ 문서와 단어 이용 희소행렬 생성 함수
  - ✓ 줄 단위로 단어 생성

sms_dtm = DocumentTermMatrix(sms_corpus) # 9) 문서와 단어 집계표 작성

sms_dtm

```
<<DocumentTermMatrix (documents: 5558, terms: 6822)>>
Non-/sparse entries: 33629/37883047
Sparsity           : 100%
Maximal term length: 40
Weighting          : term frequency (tf)
```

<DocumentTermMatrix 예>

	a	aa	bb	cc	dd	ee	ff	gg	kk	….	zz
1	1	0	0	1	0	0	0	1	0	…….	0
2	0	1	0	0	0	0	0	0	0	…….	1

3) 텍스트 전처리

- **TermDocumentMatrix**
  - ✓ DocumentTermMatrix -> TermDocumentMatrix 변경
  - ✓ 행(문서)과 열(단어)를 상호 변경(단어와 문서 구조 변경)
  - ✓ 단어의 출현 빈도수 확인 가능

t(sms_dtm)

```
<<TermDocumentMatrix (terms: 6822, documents: 5558)>>
Non-/sparse entries: 33629/37883047
Sparsity           : 100%
Maximal term length: 40
Weighting          : term frequency (tf)
```

< TermDocumentMatrix 예 >

```
    1  2  3  4  5  6  7  8  9 10  .... n
a   1  0  0  1  0  0  0  1  0  0 ..... 0
aa  0  1  0  0  0  0  0  0  0  0.......1
bb  0  0  0  0  0  0  0  0  0  0..... 0
:                  :
zz
```

2. 시계열 분석

chap02_TimeseriesAnalysis 수업내용

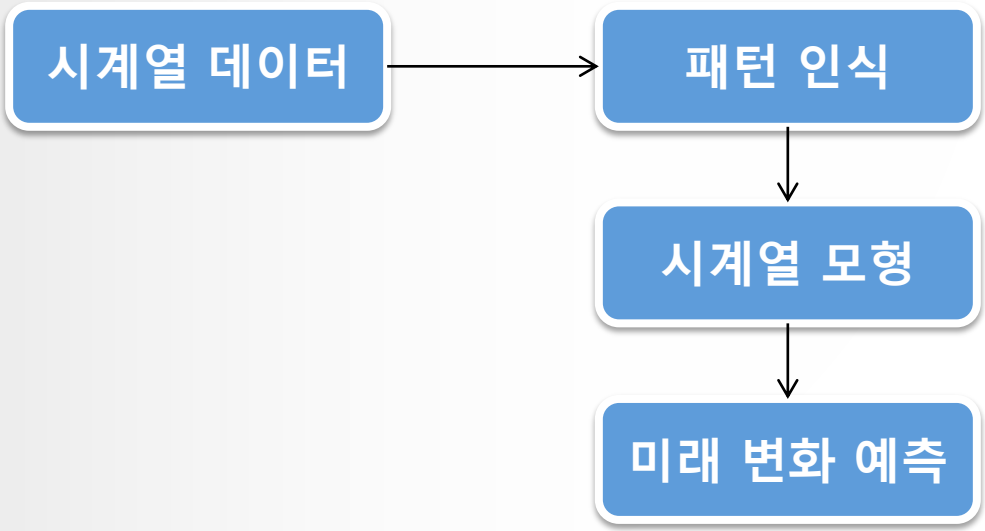
1) 시계열 분석
2) 시계열 데이터
3) 시계열 데이터 추세그래프
4) 시계열 데이터 미래 예측

1) 시계열 분석

● 시계열 분석(Timeseries Analysis)

어떤 현상에 대해서 시간의 변화에 따라 일정한 간격으로 현상의 변화를 기록한 시계열 데이터를 대상으로 미래의 변화에 대한 추세를 분석하는 방법

시간 경과에 따른 관측 값의 변화를 패턴으로 인식하여 시계열 모형을 추정하고, 이 모형을 통해서 미래의 변화에 대한 추세를 예측하는 분석방법



1) 시계열 분석

● 시계열 분석 절차

단계1 : 시계열 자료 수집 및 객체 생성

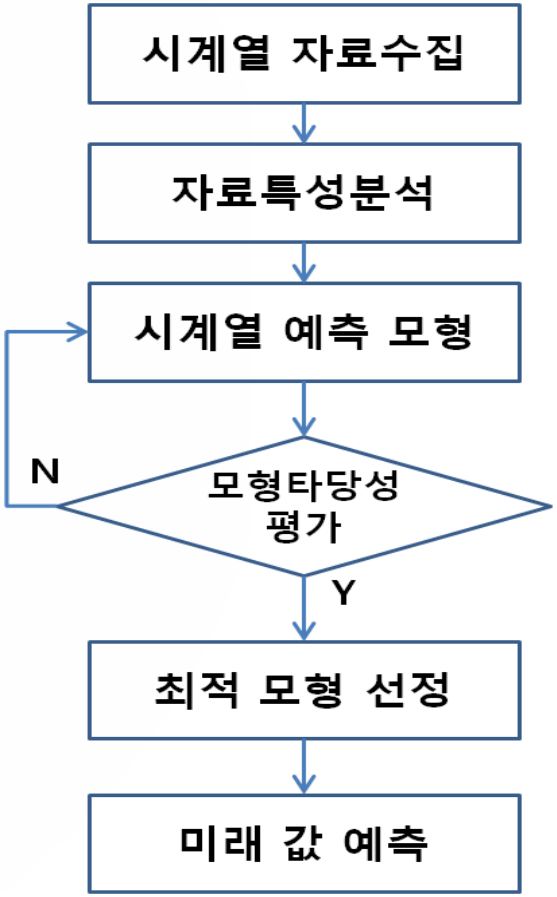
단계2 : 시계열 자료 특성 분석(추세선 시각화)

단계3 : 시계열 예측 모형 추정

단계4 : 모형 타당성 평가(유의성 검정)

단계5 : 최적의 시계열 모형 선정

단계6 : 미래 값 예측(업무 적용)



2) 시계열 데이터

- 시계열 자료 특성

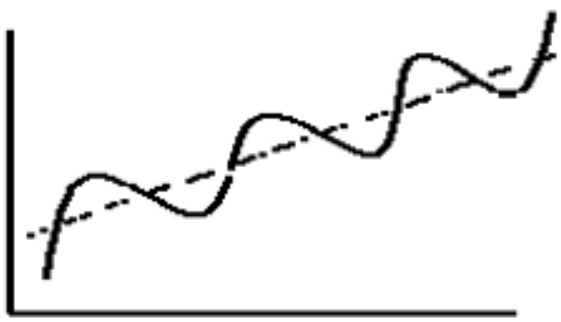
✓ 경향(Trend) : Yt 데이터가 증가(감소)하는 경향이 있는지 혹은 안정적인지 알 수 있다. 직선의 기울기가 있는가?

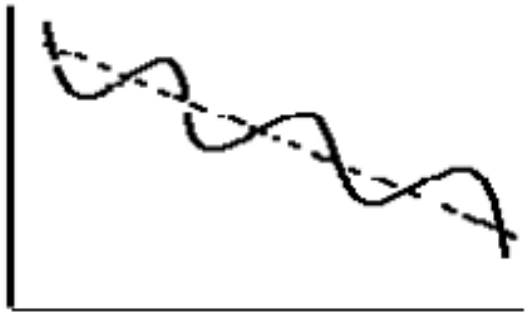
✓ 주기(cycle) : 일정한 주기(진폭)마다 유사한 변동이 반복된다.

✓ 계절성(seasonality) : 주별, 월별, 분기별, 년별 유사 패턴이 반복된다.

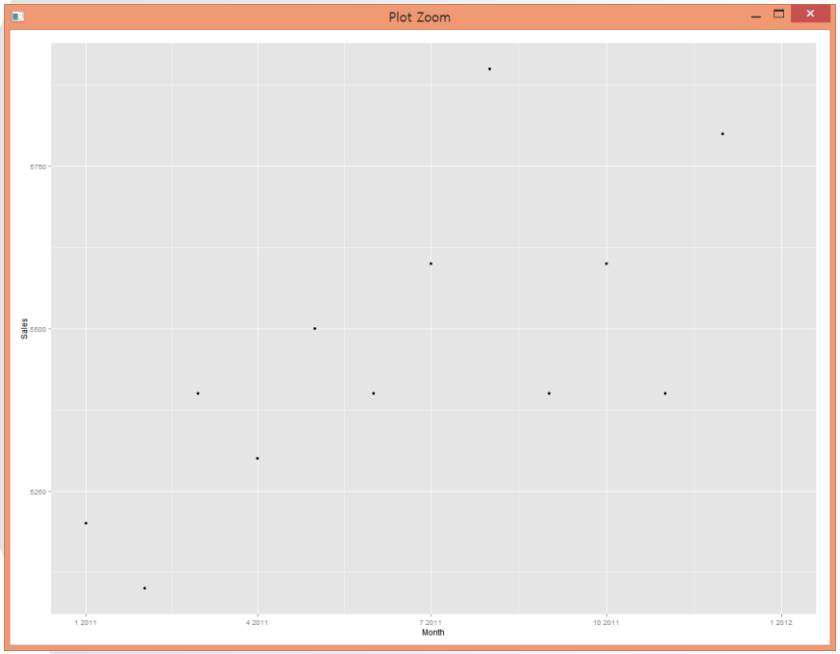
✓ 불규칙성(irregular) : 일정한 패턴을 따르지 않는다.

$$Y_t = Trend + Cycle + Seasonality + Irregular$$

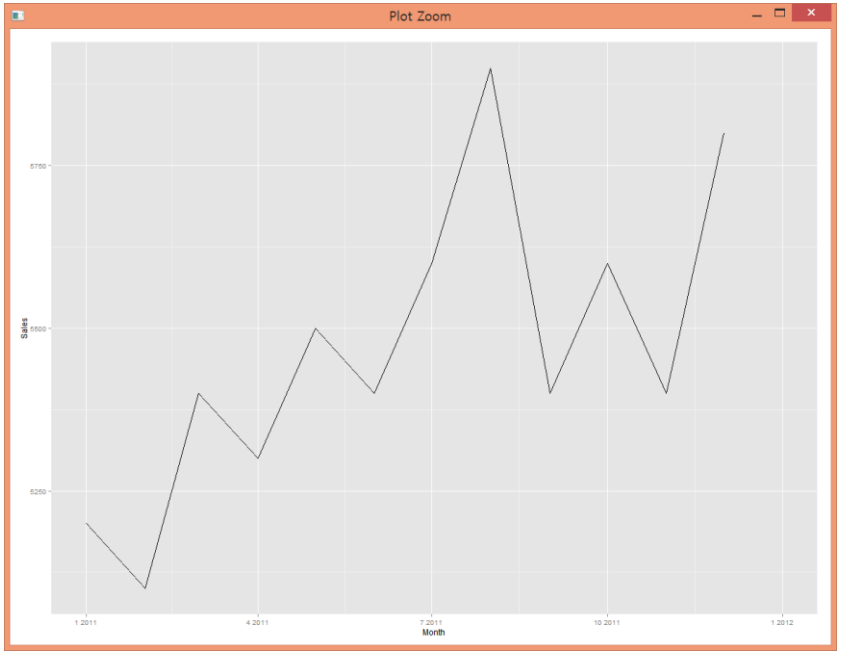




3) 시계열 데이터 추세그래프



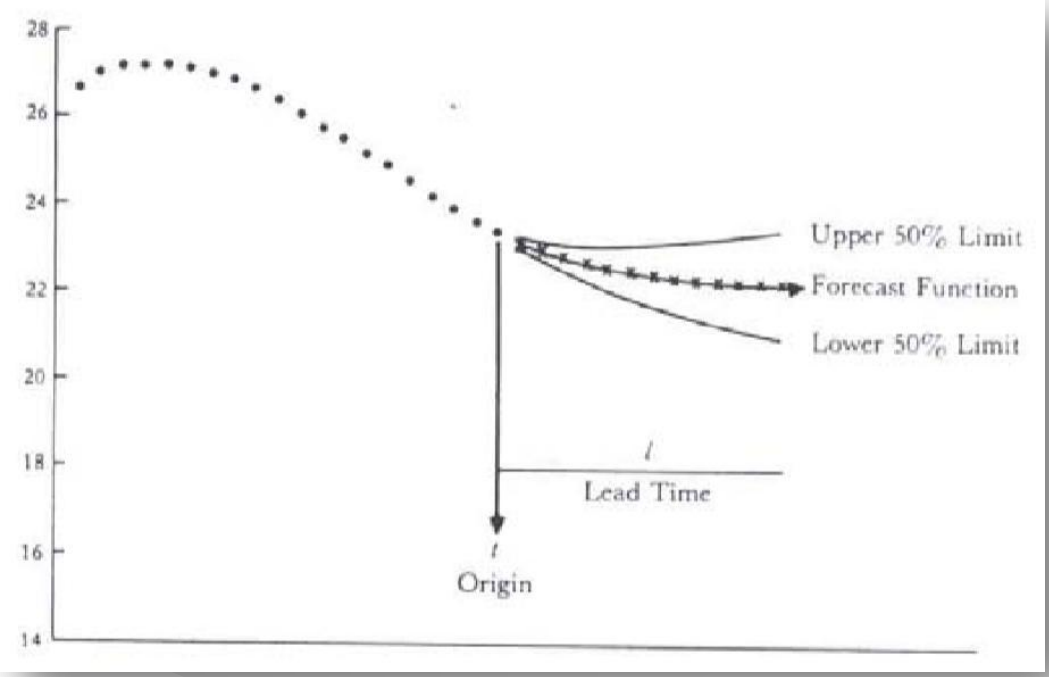
시간도표(Time Plot) 예(산점도)



시간도표(Time Plot) 예(산점도 연결선)

3) 시계열 데이터 추세그래프

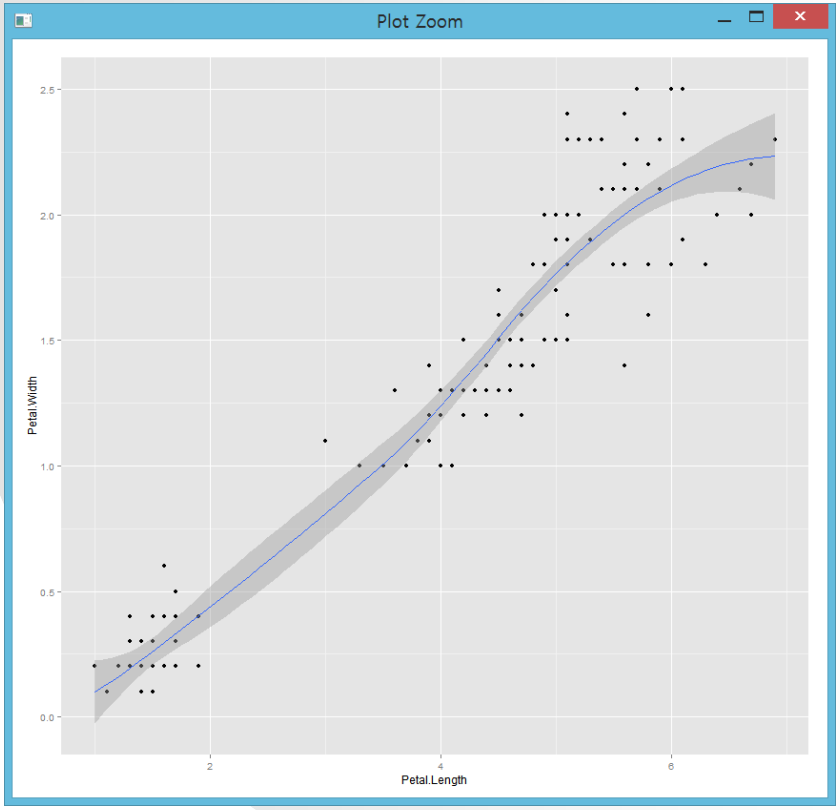
- Autoregressive Integrated Moving-average
  - ✓ 넓은 범위의 시계열 데이터를 표현할 수 있음
  - ✓ 미래 관측치에 대한 Confidence Interval도 구할 수 있음
  - ✓ 1960's Box 와Jenkins가 경제학 관련예측 연구
  - ✓ Box-Jenkins approach 라고 불림



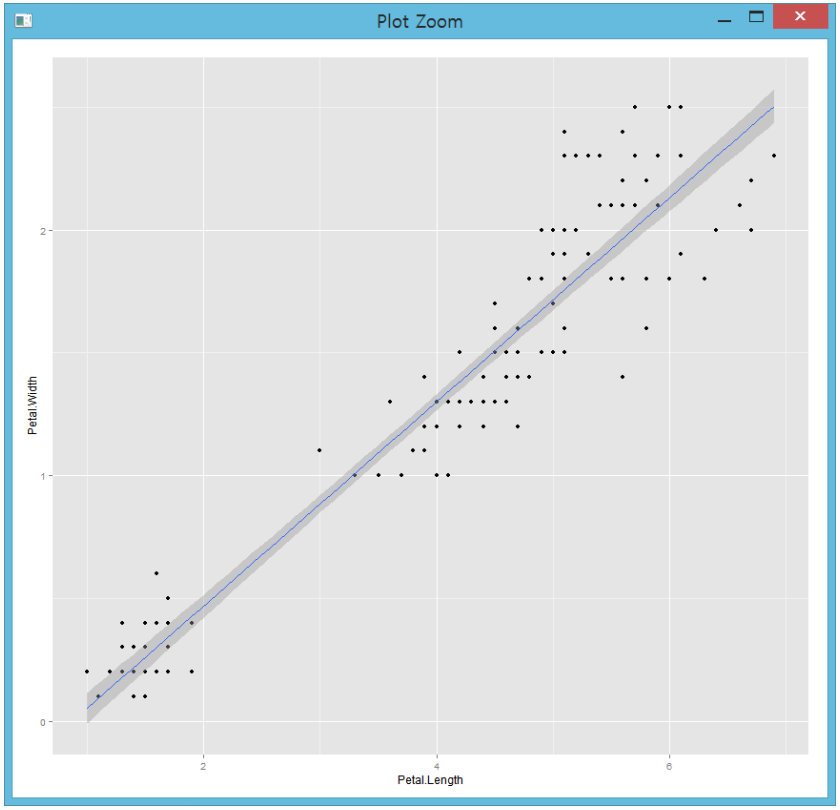
3) 시계열 데이터 추세그래프

【qplot()과 ggplot()- 시계열 분석 결과를 위한 도식화 함수】

smooth : 데이터의 추세선을 나타내고, 표준오차의 정도를 나타내는데 사용



method="auto"

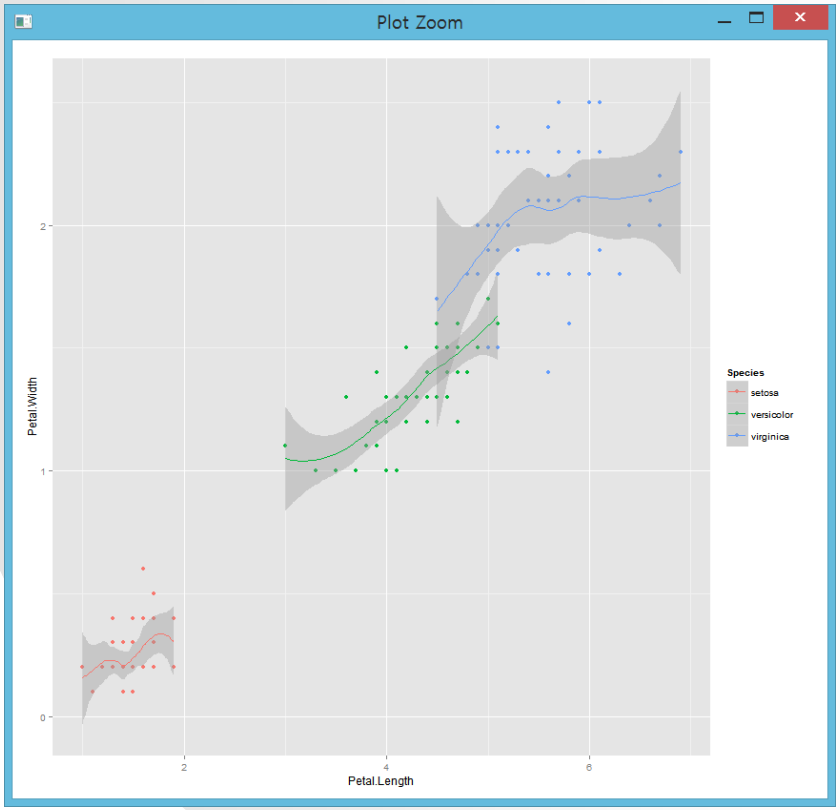


method="lm"

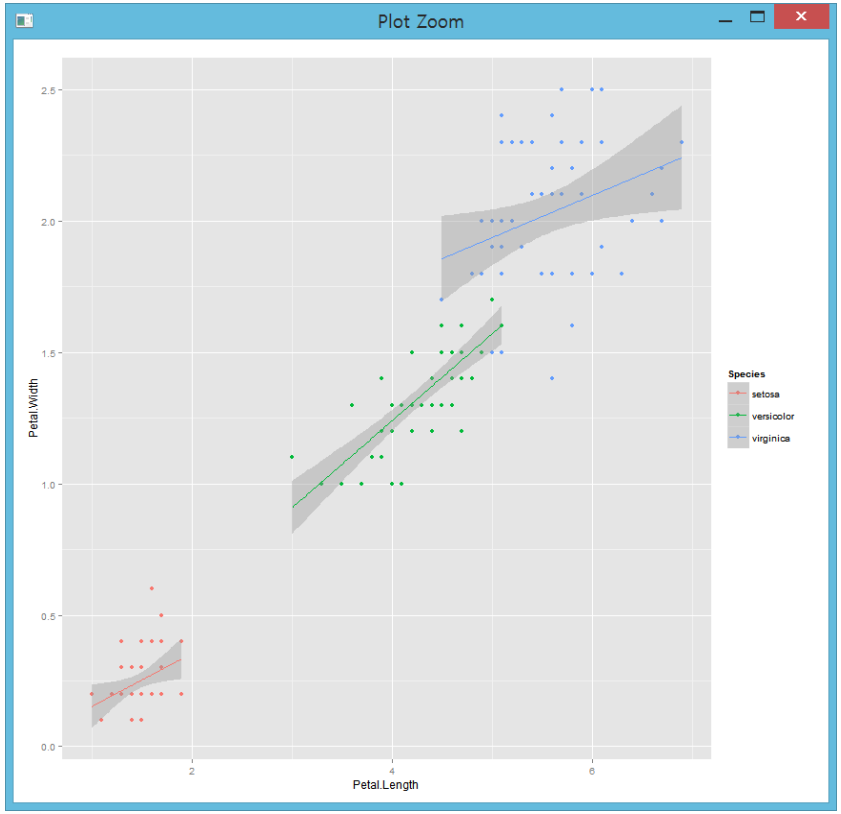
3) 시계열 데이터 추세그래프

smooth : 데이터의 추세선을 나타내고, 표준오차의 정도를 나타내는데 사용

colour=Species



method="auto"

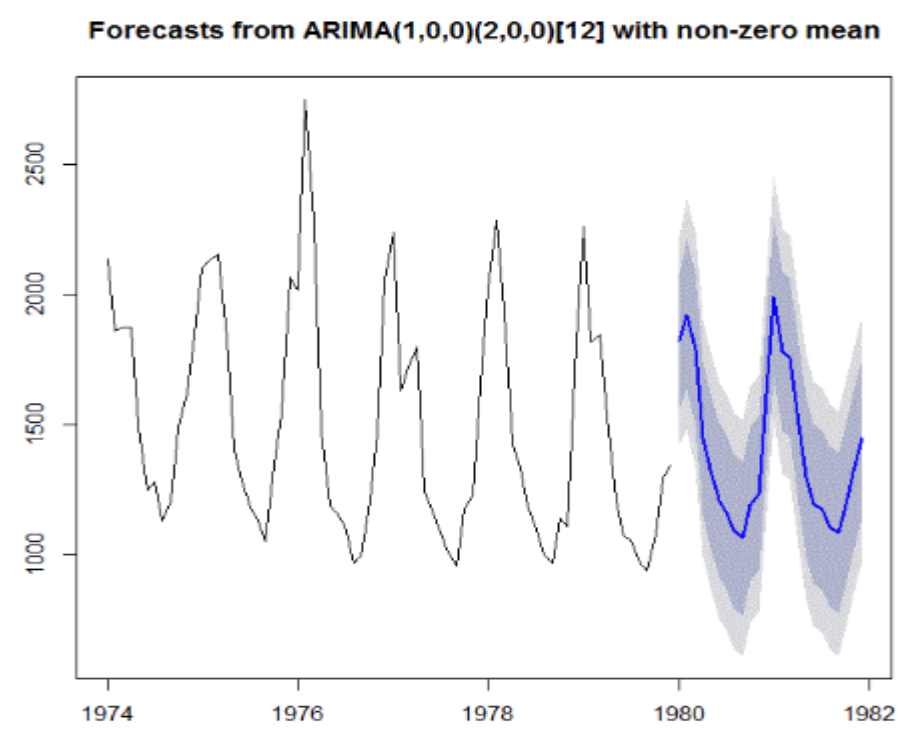


method="lm"

4) 시계열 데이터 미래 예측

mdeaths 샘플 데이터 이용 시계열 데이터 미래 예측

```
mdeaths # 영국인 사망 관련 시계열 데이터
fit <- auto.arima(mdeaths)
```



3. 수치예측 모형

chap03_LinearRegression 수업내용

1) 선형회귀모형

2) 모델 트리

1) 선형 회귀모형

- 의료비 예측

- 의료보험 가입자 1,338명을 대상으로 한 데이터 셋으로 의료비 인상 예측

1. 데이터 셋

```
insurance <- read.csv(file.choose(), header = T) # insurance.csv
str(insurance)
#'data.frame':        1338 obs. of  7 variables:
#$ age      : 수혜자 나이 - int  19 18 28 33 32 31 46 37 37 60 ...
#$ sex      : 성별 Factor w/ 2 levels "female","male": 1 2 2 2 2 1 1 1 2 1 ...
#$ bmi      : 신체 용적 지수(비만 유무)num  27.9 33.8 33 22.7 28.9 ...
#$ children: 보장 받는 자녀수 int  0 1 3 0 0 0 1 3 2 0 ...
#$ smoker  : 흡연 여부 Factor w/ 2 levels "no","yes": 2 1 1 1 1 1 1 1 1 1 ...
#$ region  : 거주지 Factor w/ 4 levels "northeast","northwest",..: 4 3 3 2 2 3 3 2 1 2 ...
#$ charges : 의료비(y) num  16885 1726 4449 21984 3867 ...
```

1) 선형 회귀모형

```
# 5. 회귀모델 성능 평가 - 검정 데이터 이용
    pred <- predict(model_ins, testing_ins)
    pred # 검정데이터의 의료비(charges) 예측

# 1) 요약통계량으로 성능 평가
    summary(pred)
    #Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    #-1620    6576   10450   13340   15410   40610

    summary(testing_ins$charges)
    #Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    #1141    4890    9257   13010   15770   62590
    # [해석] 평균과 중위수는 매우 유사하게 예측

# 2) 상관계수로 성능 평가 - 수치 예측이기 때문에
    cor(pred, testing_ins$charges)
```

2) 모델 트리

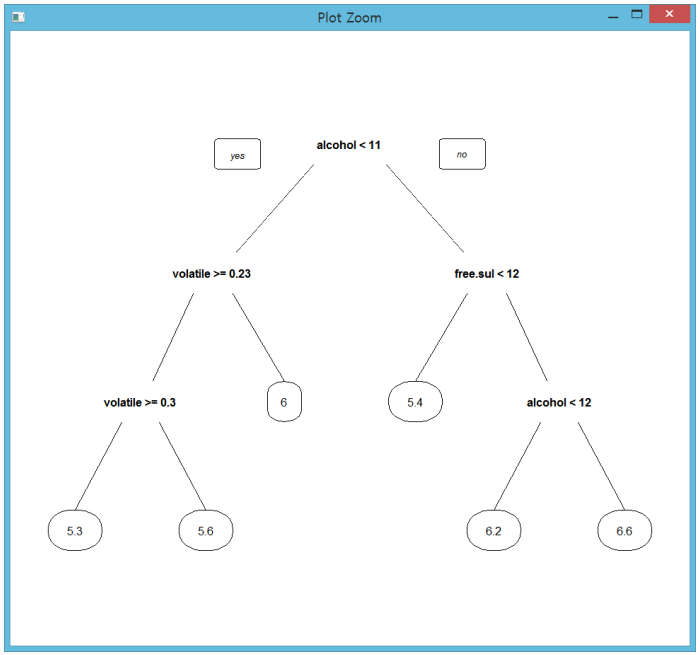
- 포루트갈산의 화이트와 레드 와인 예제

wine <- read.csv(file.choose()) # whitewines.csv 파일 선택

str(wine) # 'data.frame': 4898 obs. of 12 variables:

y 변수 : quality(블라인드 테스트를 통해서 0~10 등급으로 와인의 질 적용)

x변수 : 나머지 11개 변수(화학적 특성-신맛(acidity), 당도(suger), 염화물(chl), 황(sulfur),알코올(alcohol) 등 특성)



4. 분류 예측 모형

분류 예측 알고리즘 수업내용

1)	k-Nearest Neighbors	**Chap04-1_kNN**
2)	Naive Bayes	**Chap04-2_NB**
3)	Decision Tree	**Chap04-3_DT**
4)	Random Forest	**Chap04-4_RF**
5)	Support Vector Machine	**Chap04-5_SVM**

1) kNN 알고리즘

1. 범주가 알려지지 않은 범주를 대상으로 가장 유사한 범주로 분류
2. 기존에 예제 범주가 존재해야 함
   - ✓ 식료품(과일, 채소, 단백질 등)
3. 학습하지 않음 : 게으른 학습
4. 유클리드 거리(Euclidean distance) 계산식 이용
   - ✓ 가장 유사한 범주를 가장 가까운 거리로 선택
5. 적용 분야
   - ✓ 개인별 영화 추천
   - ✓ 이미지/비디오에서 얼굴과 글자 인식
   - ✓ 유전자 데이터 패턴 식별(종양 식별)

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

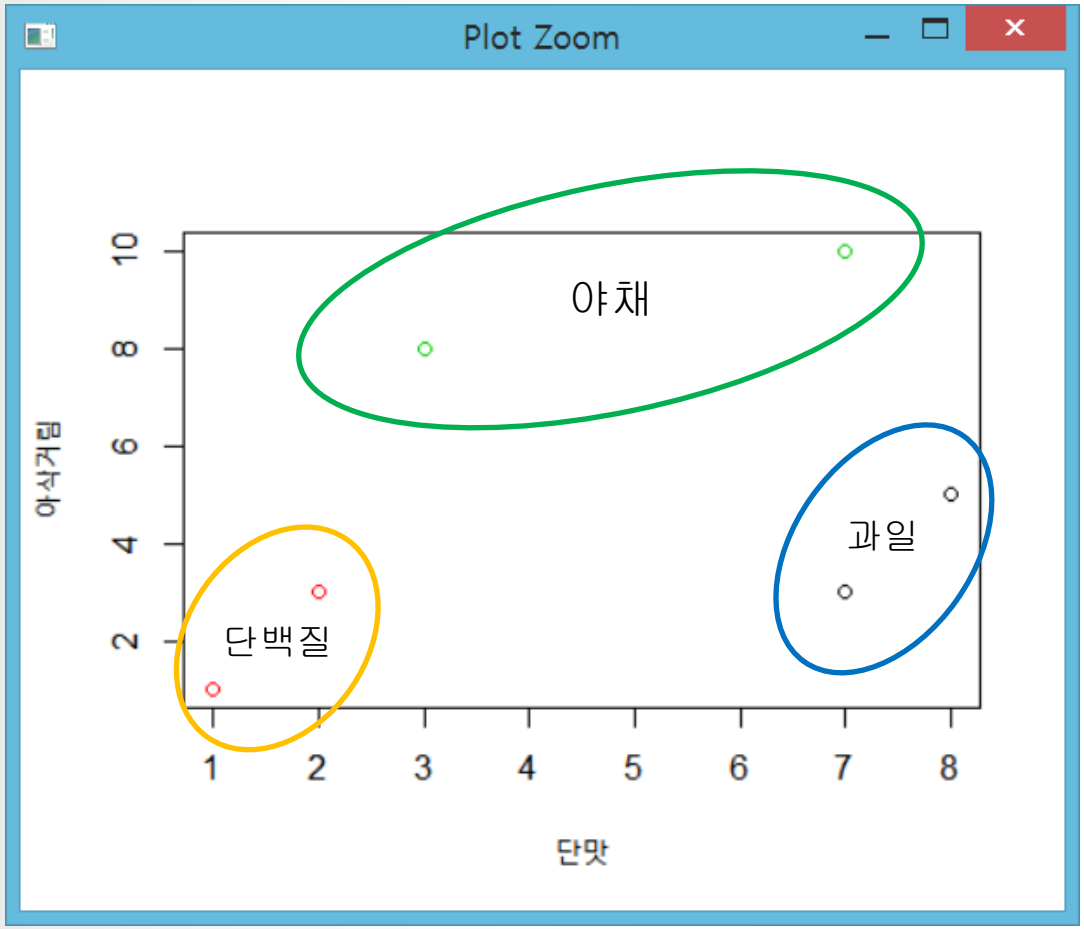
1) kNN 알고리즘

● 식료품 분류 예

식품명	단맛	아삭거림	분류
포도(grape)	8	5	과일
물고기(fish)	2	3	단백질
당근(carrot)	7	10	채소
오랜지(orange)	7	3	과일
셀러리(celery)	3	8	채소
치즈(cheese)	1	1	단백질

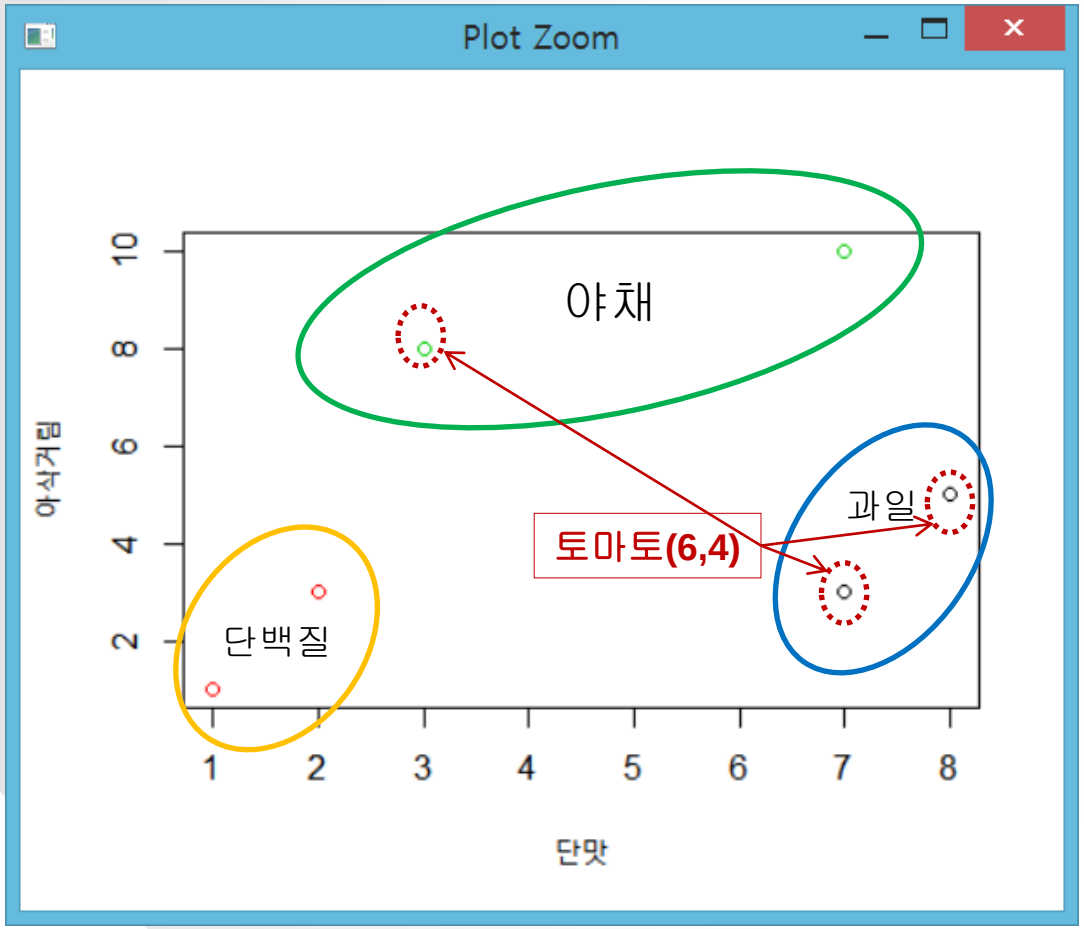
1) kNN 알고리즘

- plot() 함수 적용 결과



1) kNN 알고리즘

- 토마토는 어느 분류에 속하는가?



유클리드 거리

1NN : 오랜지

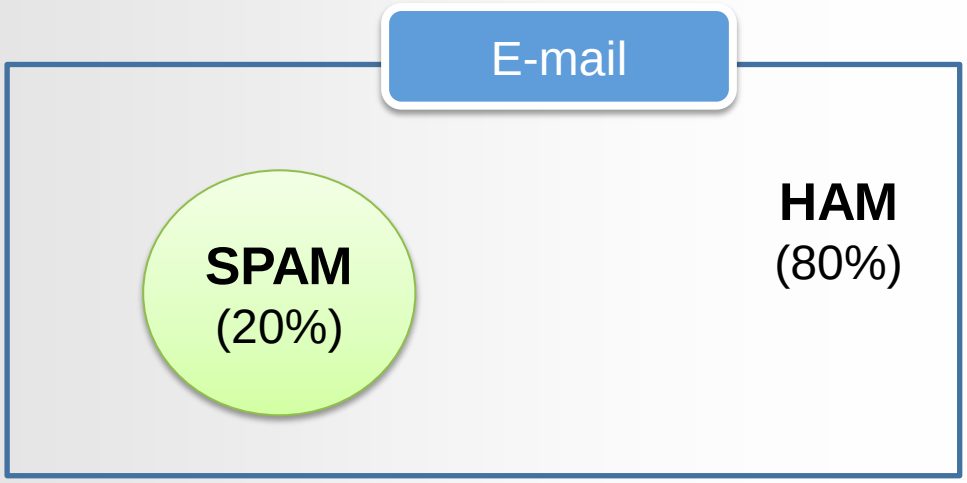
3NN : 오랜지,포도,셀러리

2) Naive Bayes 알고리즘

1. 통계적 분류기
   - ✓ 주어진 데이터가 특정 클래스에 속하는지를 확률을 통해서 예측
   - ✓ 조건부 확률 이용 : P(A|B)
2. 베이즈 확률 이론(Bayes' theorem)을 적용한 기계학습 방법
   - ✓ 두 확률 변수(사전 확률과 사후 확률) 사이의 관계를 나타내는 이론
   - ✓ 사전확률 : 사건이 발생하기 전에 알려진 확률
   - ✓ 사후확률 : 베이즈 이론에 근거한 확률
3. 특정 영역에서는 DT나 NN 분류기 보다 성능이 우수
4. 데이터가 큰 경우 높은 정확도와 속도 제공
5. 적용분야
   - ✓ Spam 메일 분류, 문서(주제) 분류, 비 유무
   - ✓ 컴퓨터 네트워크에서 침입자 분류

2) Naive Bayes 알고리즘

- 상호배타적, 포괄적 사건
  - ✓ Email 메시지에서 이미 알려진 Spam과 Ham 발생 비율 예



<독립사건인 경우 사전확률>

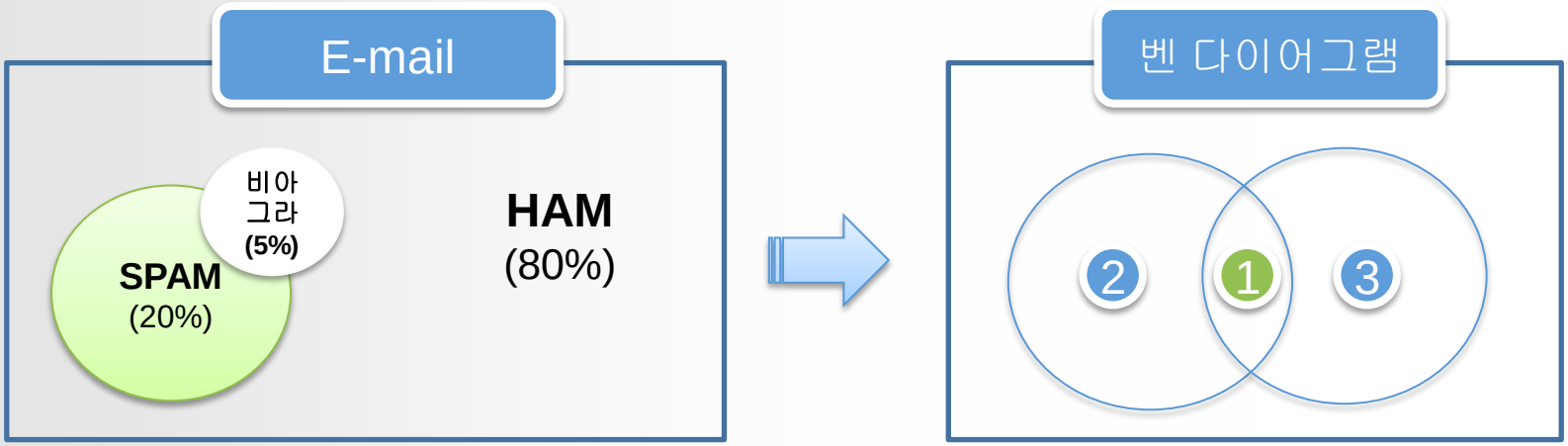
✓ SPAM과 HAM의 확률

- P(spam) = 0.2(20%)
- P(~spam) = 1-0.2=0.8(80%)

1. 상호배타적 : 동시에 두 사건이 일어나지 않음(독립사건)
   - ✓ 예) SPAM이 발생하면 HAM 발생하지 않음
2. 포괄적 : 두 가지 결과만 발생
   - ✓ 예) SPAM 또는 HAM 사건만 발생

2) Naive Bayes 알고리즘

- 비 상호배타적 사건
  - ✓ Email 메시지에서 비아그라 속성 추가 예



1. 비상호배타적 : 동시에 두 사건이 일어남
2. 결합확률 : 두 사건이 동시에 일어날 확률

예) SPAM 메일에 비아그라 단어가 포함될 확률

예) HAM 메일에 비아그라 단어가 포함될 확률

- 벤 다이어그램 : 원소 집합의 중첩 표현

① : SPAM 메일에 비아그라 단어 포함

② + ① : SPAM 메일(20%)

③ : HAM 메일에 비아그라 단어 포함

- 조건부 확률 적용
  - ✓ P(spam ∩ viagra) : 비아그라 단어가 포함된 경우 SPAM일 확률 계산

2) Naive Bayes 알고리즘

- 주변확률 : 조건 없이 사건A가 발생할 확률
  - ✓ SPAM 확률 : 20/100 = 0.2
  - ✓ HAM 확률 : 80/100 = 0.8
  - ✓ VIAGRA 확률 : 5/100 = 0.05
- 결합확률 : 두 사건 A와 B가 동시 일어날 확률
  - ✓ Viagra 단어가 포함된 Spam 메일 확률 : 4/20 = 0.2
  - ✓ Viagra 단어가 포함된 Ham 메일 확률 : 1/80 = 0.0125

구분	VIAGRA		합계
	Yes	No	
SPAM	4	16	20/100
HAM	1	79	80/100
합계	5/100	95/100	100

[주변확률 표]

구분	VIAGRA		합계
	Yes	No	
SPAM	4/20	16/20	20
HAM	1/80	79/80	80
합계	5/100	95/100	100

[결합확률 표]

2) Naive Bayes 알고리즘

- 조건부 확률 : 사건 A가 발생한 상태에서 사건 B가 발생할 확률
  - ✓ P(B|A) = P(A|B) * P(B) / P(A) = P(A∩B) / P(A)
- 비아그라 단어(사건 A)가 포함된 경우 스팸(사건 B)일 확률
  - ✓ P(SPAM|VIAGRA) = P(viagra|spam) * P(spam) / P(viagra)
  - ✓ P(SPAM|VIAGRA) = (4/20) * (20/100) / (5/100) = 0.8
  - ∴ 비아그라 단어가 포함된 메시지는 스팸 확률이 80% 라는 의미

결합 확률

구분	VIAGRA		합계	주변 확률
	Yes	No		
SPAM	4/20	16/20	20	20/100
HAM	1/80	79/80	80	80/100
합계	5/100	95/100	100	

[결합확률 표]

2) Naive Bayes 알고리즘

- 베이즈 이론을 조건부 확률에 적용
  - ✓ P(B|A) = P(A|B) * P(B) / P(A)
  - ✓ P(SPAM|VIAGRA) = P(viagra|spam) * P(spam) / P(viagra)
  - ✓ 사후 확률 = 결합확률(우도) * 사전확률 / 주변확률
  - ✓ P(SPAM|VIAGRA) = (4/20) * (20/100) / (5/100) = 0.8

결합확률

주변확률

구분	VIAGRA		합계	주변 확률
	Yes	No		
SPAM	4/20	16/20	20	20/100
HAM	1/80	79/80	80	80/100
합계	5/100	95/100	100	

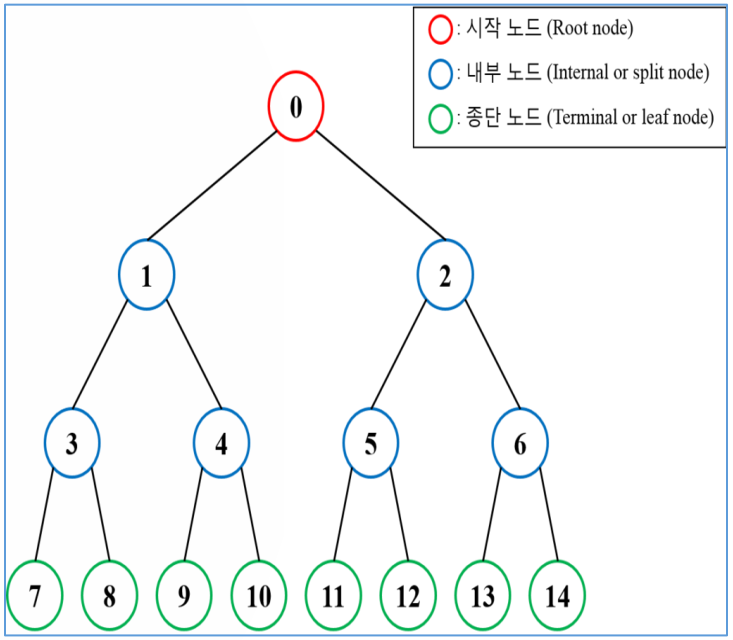
사전확률 :
이미 알려진 확률

[결합확률 표]

∴ 비아그라 단어가 포함된 Email이 스팸 일 확률은 80%

3) Decision Tree 알고리즘

- Tree 구조
  - ✓ 계층 구조로 노드와 에지(edge) 집합
  - ✓ 노드는 내부 노드(internal node)와 종단 노드(leaf node) 분류
  - ✓ 모든 노드의 들어오는 에지는 하나만 제한(그래프와 차이점)
  - ✓ 각 노드의 나가는 에지 개수는 제한 없음 (주로 두 개의 에지가 있는 것으로 가정)
- Decision Tree
  - ✓ 결정을 내리기 위해 사용하는 트리
  - ✓ 복잡한 문제를 간단한 계층 구조형태로 나누기 위한 기술
  - ✓ 간단한 문제에 대해서는 사용자가 직접 매개변수(모든 노드의 매개변수, 종단 노드의 매개변수) 설정 – 파라미터 설정
  - ✓ 복잡한 문제는 학습 데이터로부터 트리 구조와 매개변수를 자동으로 학습

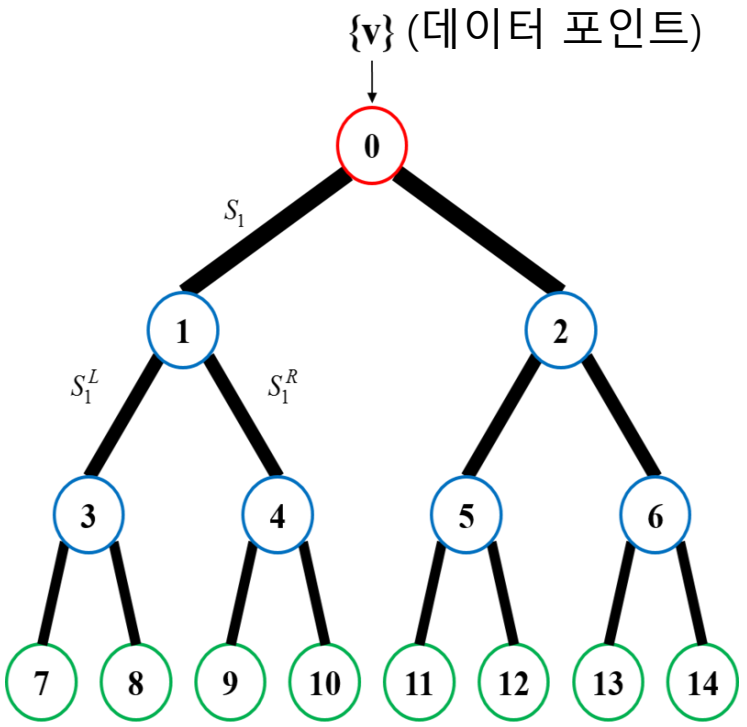


[참고] 위키백과

● 학습(훈련) 단계(Training Step)

- 종단 노드에 대한 매개변수와 내부 노드와 관련된 노드 분할 함수(split function)의 매개변수를 최적화하는 작업 진행
  - ✓ 예) 노드 분할에 최적인 변수 선정
- 트리 노드 관계식
  노드 1(그림에서와 같이 0부터 너비-우선순위로 각 노드에 숫자 부여)에 도달하는 훈련 데이터의 부분집합을 S_1 이라고 하고, 노드 1의 왼쪽과 오른쪽의 자식 노드를 각각 $S_1^{\mathrm{L}}, S_1^{\mathrm{R}}$ 라고 할 때 각 분할 노드는 다음과 같은 관계식을 갖는다.

$$S_j = S_j^{\mathrm{L}} \cup S_j^{\mathrm{R}}, S_j^{\mathrm{L}} \cap S_j^{\mathrm{R}} = \emptyset, S_j^{\mathrm{L}} = S_{2j+1}, S_j^{\mathrm{R}} = S_{2j+2}.$$

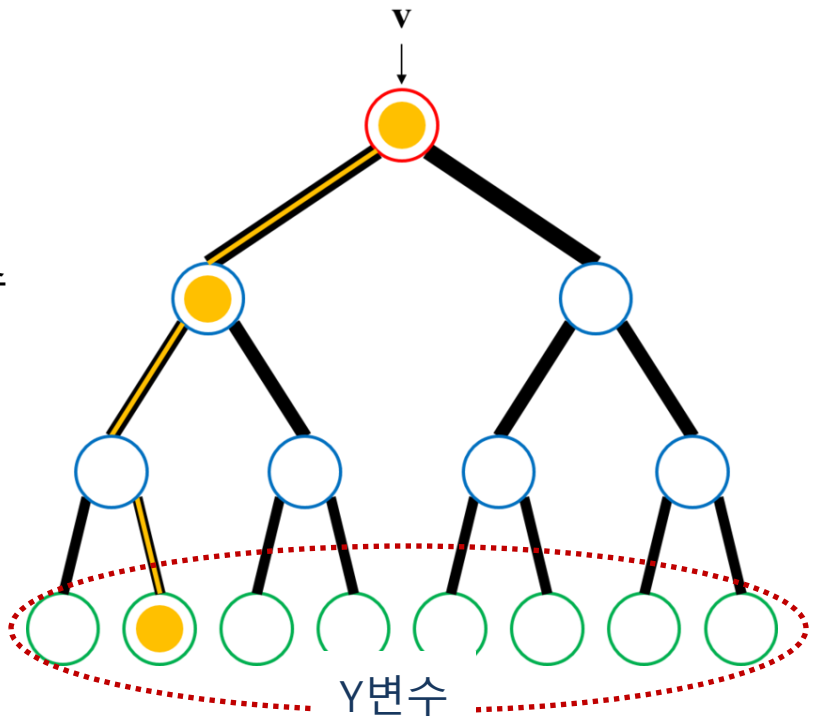


[노드1] 결정 트리 테스트 단계

T개의 트리로 구성된 하나의 포레스트의 경우, 일반적으로 훈련 과정은 각 트리에 대해 독립적으로 T번 반복된다.
랜덤 포레스트(트리)에서 주목할 사실은 임의성(randomness)이 오직 훈련 과정에서만 적용된다. 즉 훈련 단계에서 랜덤 포레스트가 결정되며, 이렇게 결정된 데이터를 이용하여 검정 과정에서 결정적인 특성이 나타난다.

● 검정 단계(Test Step)

- 검정 데이터(v)가 입력으로 주어졌을 때, 각 결정 트리는 사전에 정의된 많은 테스트 값을 계층적으로 적용
- 루트 노드에서 시작해, 각 노드의 분할 함수를 입력 데이터에 적용
- 입력 데이터는 이진 테스트의 결과에 따라 오른쪽 또는 왼쪽의 자식 노드로 보내진다. 이 과정은 입력 데이터 포인트가 단말 노드에 도달할 때까지 반복
- 단말 노드는 대개 입력 에 대한 출력 값을 내는 예측기(분류기(classifier) 또는 회귀기(regressor))
- 포레스트를 구성하는 각 트리 예측기들의 조합으로 단일 예측치를 만든다.



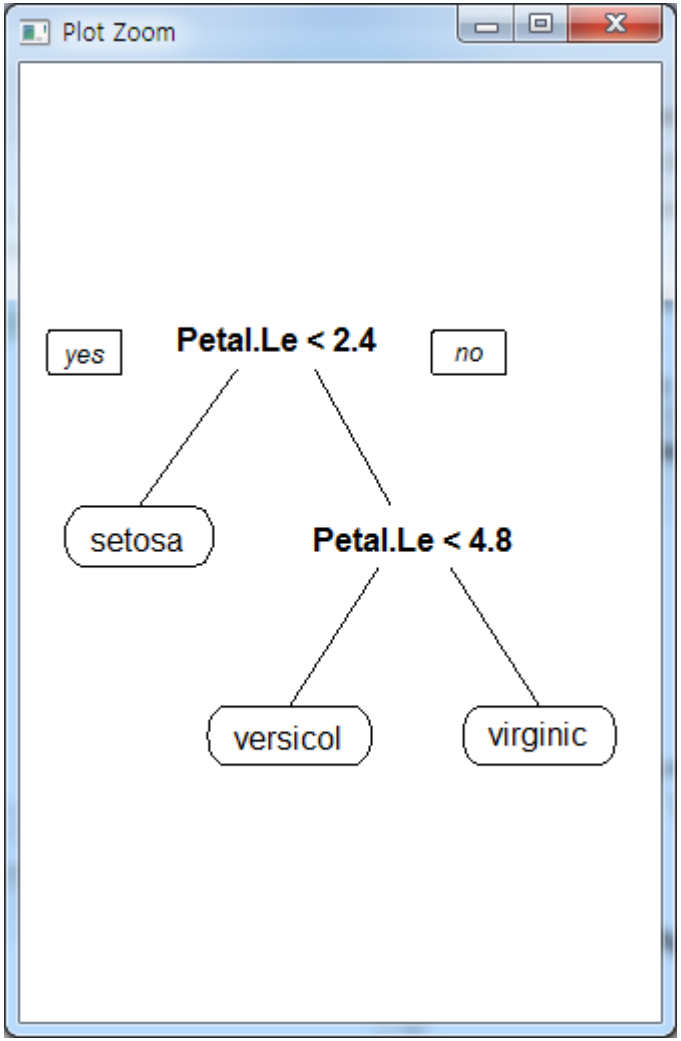
[노드1] 결정 트리 훈련 단계

● 학습 데이터로 모델 생성

```
# 단계1. 패키지 설치 및 실습데이터 생성
library(rpart)
library(rpart.plot) # prp() : rpart 시각화 패키지
data(iris)
set.seed(415)
idx = sample(1:nrow(iris), 0.7*nrow(iris))
training = iris[idx, ]
testing = iris[-idx, ]

# 단계2. 학습데이터로 분류모델 생성
# iris의 4개 칼럼으로 꽃의 종류(Species) 분류
result.dt = rpart(Species~., data=training)
result.dt
# [해설]  Petal.Length  칼럼이 중요변수

prp(result.dt) # 시각화
```



[노드1] 결정 트리 훈련 단계

● 분류모델을 검정데이터에 적용

```
# 단계3. 분류모델을 검정데이터에 적용
predicted.dt = predict(result.dt, testing, type="class")
# 1) 검정 데이터에는 y변수(Species)가 존재해야 함
testing # Sepal.Length Sepal.Width Petal.Length Petal.Width  Species
# 2) 분류모델로 분류된 y변수 보기
predicted.dt
#    8          11         16         20       <- 검정데이터 row
#setosa     setosa     setosa     setosa <- 분류 결과(꽃의 종류)
# 3) 분류결과에 검정데이터 빈도분석
table(testing$Species, predicted.dt)
#            predicted.dt
#            setosa versicolor virginica
#setosa         14          0         0 <- 14개 모두 적중
#versicolor      0         13         4 <- 17개 중 13개 적중
#virginica       0          1        13 <- 14개 중 13개 적중
#[해설] row : 검정 데이터 원형, column : 분류모델에 의한 분류결과
```

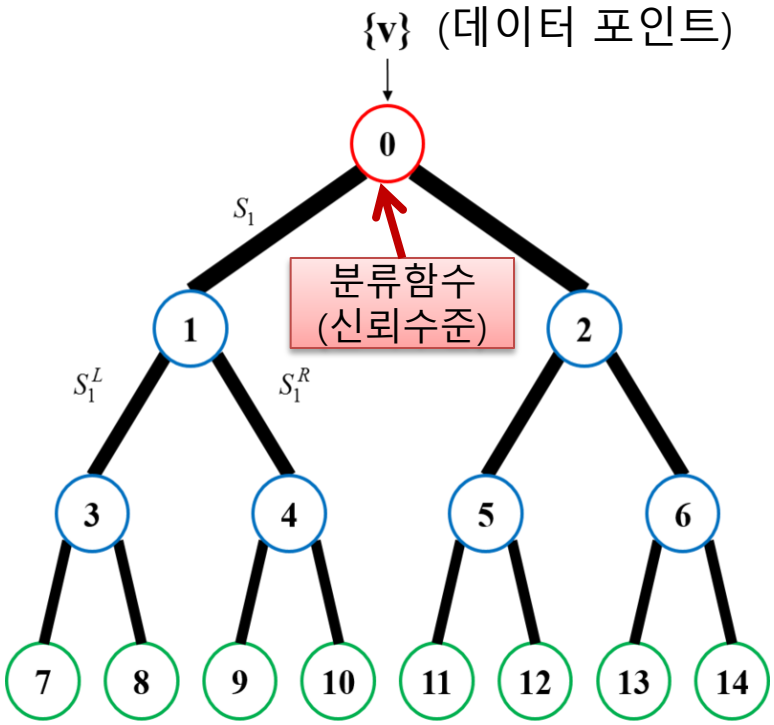
● 분류모델을 검정데이터에 적용

```
# 4) 검정데이터 꽃의 종류 확인
length(testing$Species) # 45 - 전체 data
length(test[testing$Species=='setosa']) # 14개
length(test[testing$Species=='versicolor']) # 17개
length(test[testing$Species=='virginica']) # 14개
```

● 정보 획득량과 섀넌 엔트로피

- 훈련 단계에서 트리의 각 노드는 들어오는 데이터 포인트들을 최적으로 분리하기 위해 정보 획득량(information gain)을 측정 기준으로 사용
- 정보 획득량 측정 기준
  - ✓ 신뢰수준(confidence) 최대화
- 정보 획득량은 다음과 같이 정의된다.

$$I = H(S) - \sum_{i \in \{L,R\}} \frac{|S^i|}{|S|} H(S^i),$$



[노드1] 결정 트리 테스트 단계

섀넌 엔트로피(Shannon entropy)
- 확률 변수의 조합으로 정보원(S)에 대한 불확실성

❖ 정보 획득량을 최대화하기 위해서는 결국 섀넌 엔트로피가 최소화 되어야 한다.

4) Random Forest 알고리즘

- 여러 개의 결정 트리를 임의적으로 학습하는 방식의 앙상블 학습방법
- 다중의사결정트리
- 학습단계 : 다수의 결정 트리 구성
- 검정단계 : 입력 벡터가 들어왔을 때, 분류하거나 예측하는 단계
- 검출, 분류, 회귀 등 다양한 Application 활용

- 학습(훈련) 과정에서 구성한 다수의 결정 트리로 부터 분류 또는 평균 예측치(회귀 분석) 출력

[앙상블 학습방법]
더 좋은 예측 성능을 얻기 위해 다수의 학습 알고리즘을 사용하는 방법이다

[참고] https://www.stat.berkeley.edu/~breiman/RandomForests/

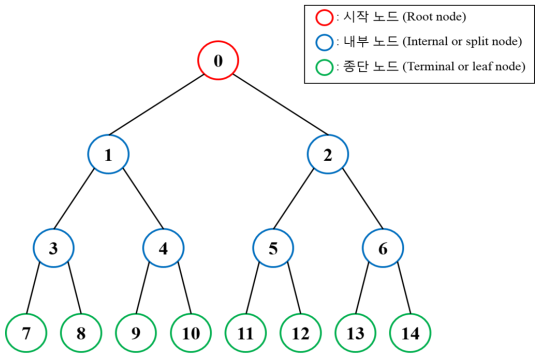
● Decision tree

1. 동일한 하나의 데이터 집합에서 한 개의 훈련용 데이터를 생성
2. 한 번의 학습을 통해서 하나의 분류 트리 생성 및 목표변수 예측
3. 생성된 분류모델을 검정데이터에 적용하여 목표변수 예측

● Random Forest

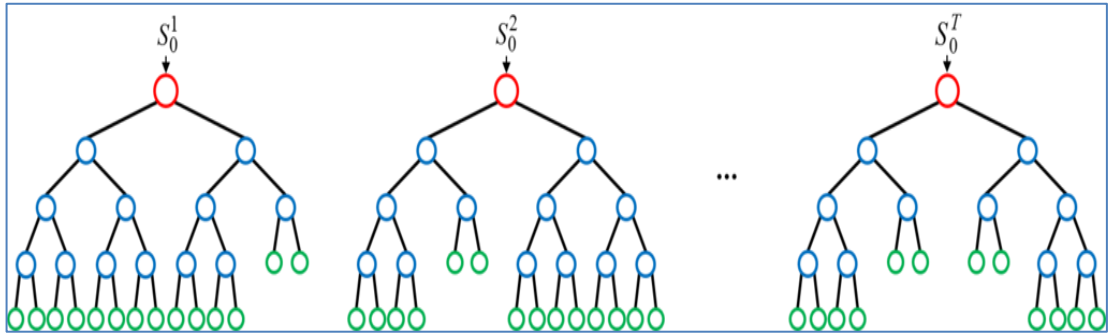
1. 동일한 하나의 데이터 집합에서 임의복원 샘플링을 통해 여러 개의 훈련용 데이터를 생성
2. 여러 번의 학습을 통해 여러 개의 트리 생성하고, 이를 결합하여 최종적으로 목표변수 예측
3. 분류모델을 검정데이터에 적용

학습용 데이터 트리



결정 트리

학습용 트리1 학습용 트리2 학습용 트리3



랜덤 포레스트

● 임의화(randomization)

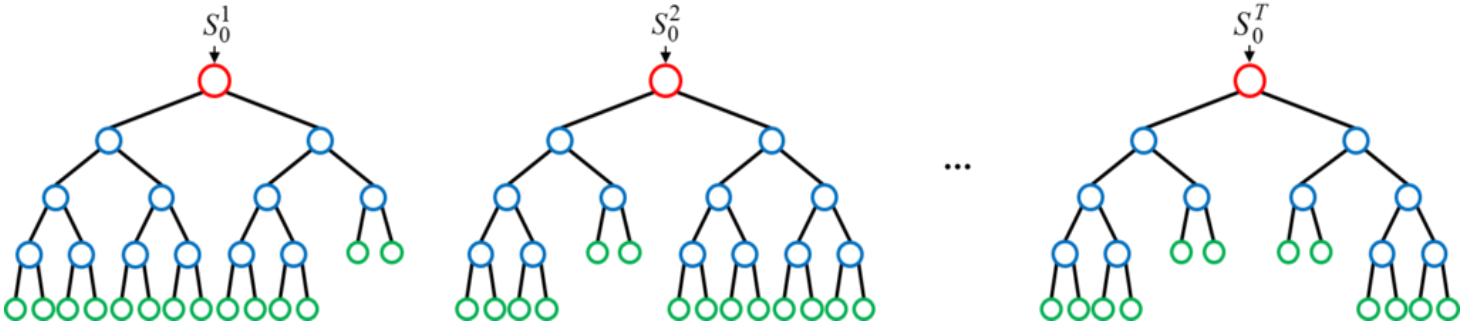
- ➢ 랜덤 포레스트의 가장 핵심적인 특징은 임의성(randomness)에 의해 서로 조금씩 다른 특성을 갖는 트리 들로 구성

- ➢ 임의 학습 데이터 추출 방법
  - ✓ 배깅(bagging)
  - ✓ 임의 노드 최적화(randomized node optimization)

● 배깅(bagging)

- bootstrap aggregating의 약자로, 부트스트랩(bootstrap)을 통해 조금씩 다른 훈련 데이터에 대해 훈련된 기초 분류기(base learner)들을 결합(aggregating)시키는 방법
- 부트스트랩이란, 주어진 훈련 데이터에서 중복을 허용하여 원래 데이터와 같은 크기의 데이터를 만드는 과정
- 배깅을 통해 랜덤 포레스트를 훈련시키는 과정은 다음과 같이 크게 세 단계로 구성되어있다.

1. 부트스트랩 방법을 통해T개의 훈련 데이터 집합을 생성한다.
2. T 개의기초 분류기(트리)들을 훈련시킨다.
3. 기초 분류기(트리)들을 하나의 분류기(랜덤 포레스트)로 결합한다.



배깅을 이용하여 T개의 결정트리들로 구성된 랜덤 포레스트를 학습하는 과정으로
S_0:전체 학습 데이터 집합, S^T_0 : 결정트리를 위해 배깅을 통해 임의로 선택된 학습 데이터

Random Forest 임의화 모형

● 배깅(Bagging)

1. 주어진 데이터에 대해서 여러 개의 부트스트랩(bootstrap)자료를 생성하고, 각 부트스트랩 자료를 모델링한 후 결합하여 최종의 예측 모형을 산출하는 방법
2. 원 데이터로 부터 여러 번의 복원 샘플링을 통해 예측 모형의 분산을 줄여 줌으로써 예측력을 향상시키는 방법

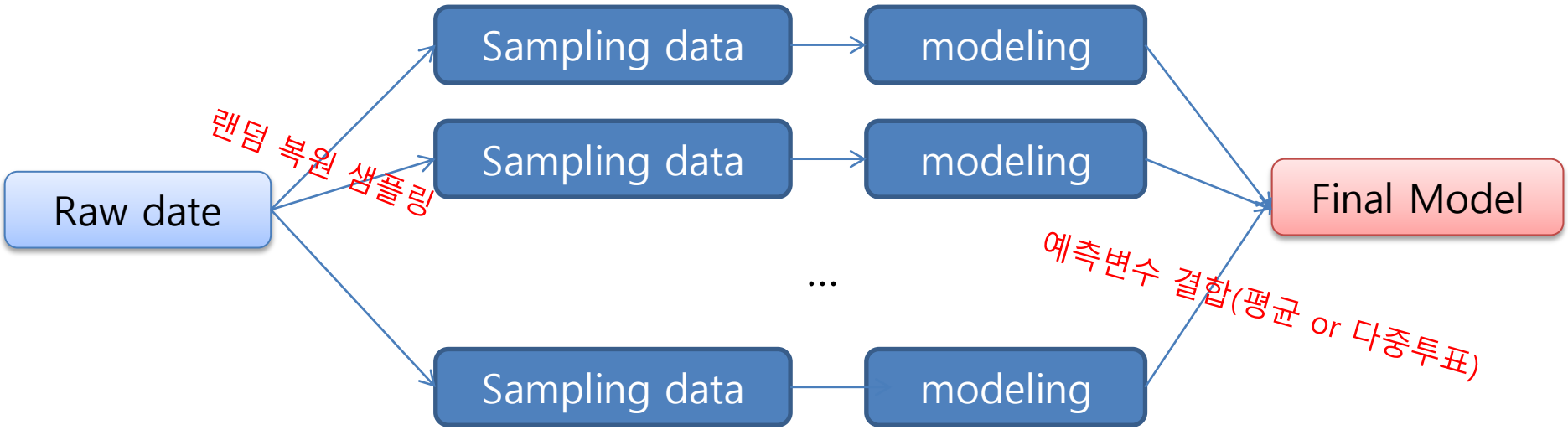
- 부트스트랩 자료란 원 데이터로부터 단순 복원 임의 추출법을 통해 추출된 크기가 동일한 여러 개의 표본 자료

● 임의 노드 최적화

- 분석에 사용될 데이터를 랜덤하게 분석하여 추출하는 방법

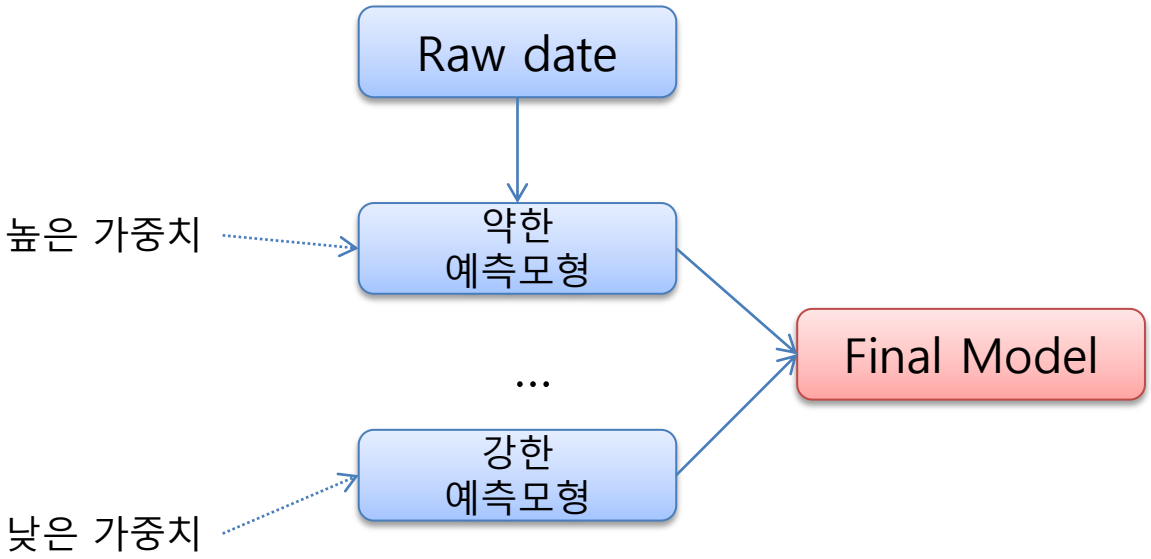
● 배깅 알고리즘

- 원 데이터로 부터 n번 랜덤 복원 샘플링을 하고 각 샘플의 모델링(함수 이용)을 통해서 나온 예측변수들을 결합하여 최종 모형을 생성
- 각 샘플의 예측변수들을 결합하는 방법은 목표변수가 연속형이면 평균(average), 범주형이면 다중 투표(majority vote) 사용



● 부스팅(boosting) 알고리즘

- 잘못 분류된 객체들에 집중하여 새로운 분류규칙을 생성하는 단계를 반복하는 알고리즘
- 약한 예측모형들을 결합하여 강한 예측모형을 만드는 알고리즘
- 오 분류된 개체는 높은 가중치, 정 분류된 개체는 낮은 가중치를 적용하여 예측모형의 정확도를 향상시키기 위한 방법



● Random Forest 사용 이유

1. 단일 의사결정트리는 과대적합(overfitting)의 위험이 큼
2. 배깅을 이용해 각 트리의 평균, 확률, 투표를 통해 목표변수 예측
3. 트리의 편향은 유지되고, 분산은 감소되기 때문에 정확도 높음
4. 빅데이터 시스템에서 분산처리 시스템에 맞는 분류분석 기법
   - 여러 개의 훈련 데이터를 추출하여 트리 생성, 결합을 통해 목표변수 예측의 구조가 분산처리시스템에 적합

- 정리

1. 배깅 알고리즘을 통해 임의 복원 추출되는 훈련용 데이터를 생성하고, 각 트리 생성
2. 예측결과를 투표, 평균, 확률 등으로 결합하여 최종 모형 생성
3. 적용되는 R 패키지 : ramdomForest

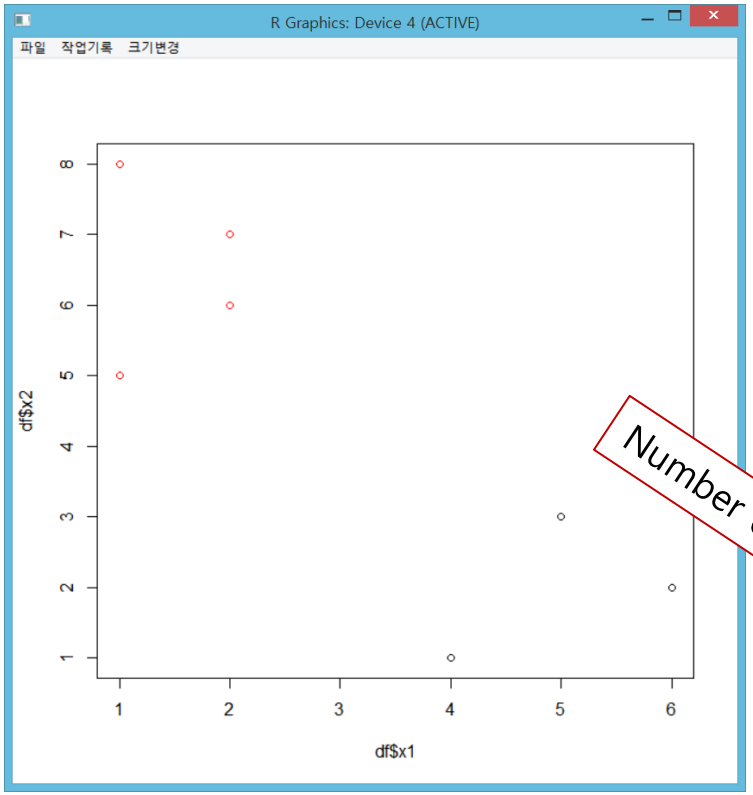
5) Support Vector Machine 알고리즘

- 2000년대 초반에 많이 사용되는 분류 알고리즘
- SVM 알고리즘 - 이진분류 : 두 범주를 직선으로 분류
- 선형분리 - 2개의 집합(초평면:Hyperplane)을 직선으로 분리
- 좋은 직선 – 가상 직선 중심으로 거리 계산하여 직사각형 형태로 영역 넓힘(가장 가까운 점이 만날 때 까지 영역을 최대한 넓혀감)
- 직사각형의 넓이(Margin) : Margin의 최대값을 구하는 것이 관건
- Support Vectors : Margin과 가장 가까운 점들
- kNN과 선형회귀 모델링 기법이 적용 - 분류와 수치 예측 가능
- 비선형 분류를 위해서 데이터를 고차원 특징 공간으로 사상하는 작업 필요(커널 트릭 사용)
- 적용 분야
  - ✓ 바이오인포매틱스의 마이크로 유전자 데이터 분류
  - ✓ 인간의 얼굴, 문자, 숫자 인식 -> 이미지 데이터 패턴 인식에 적합
    예) 스캐너로 스캔된 문서 이미지를 문자로 인식

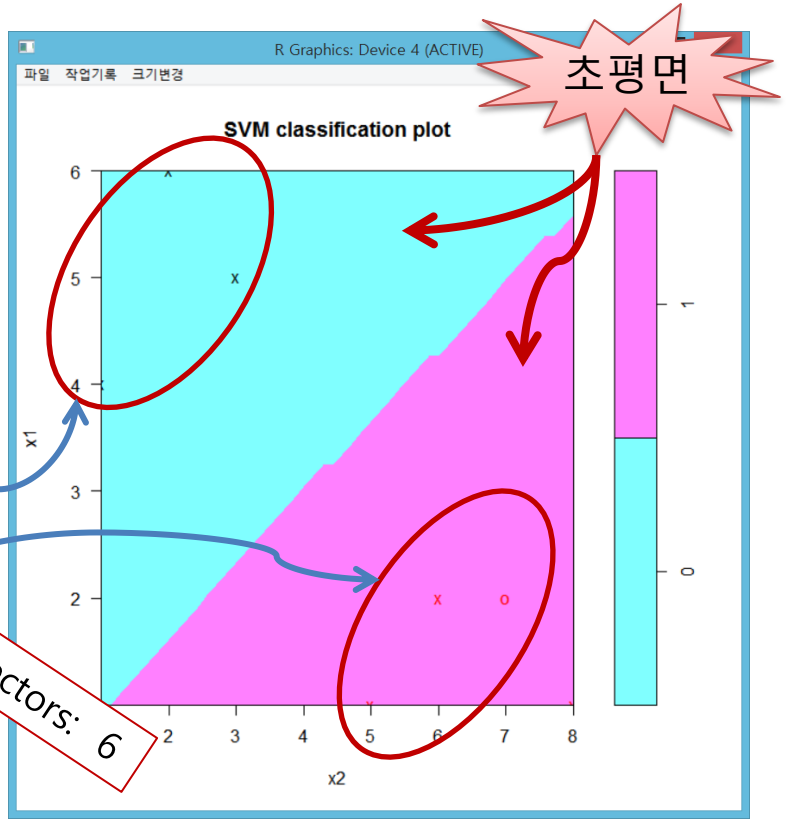
● 초평면(Hyperplane)

서포트 벡터 머신은 분류 또는 회귀 분석에 사용 가능한 초평면(hyperplane) 또는 초평면 들의 집합으로 구성

2개의 집합 직선으로 분리



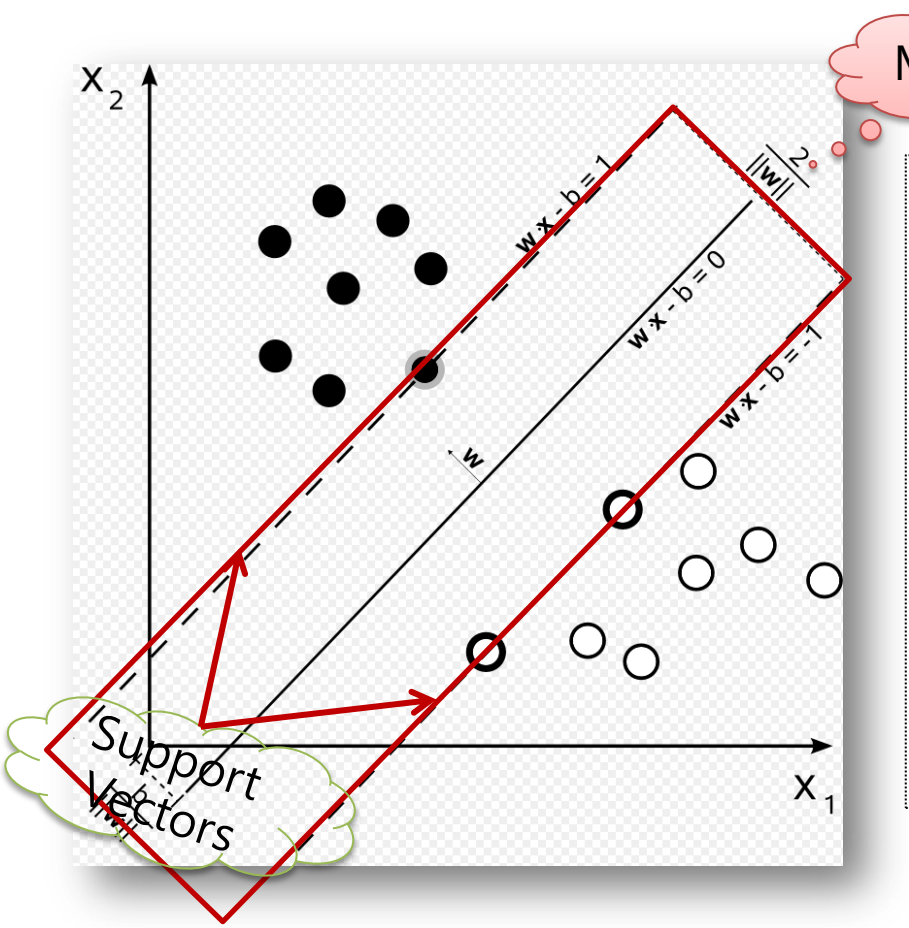
Number of Support Vectors: 6



Margin과 가장 가까운 점 6개

● Linear Classifier Margin

SVM 알고리즘 : 가상 직선 중심으로 거리 계산하여 최대의 직사각형 형태(Margin)로 영역 넓힘



W : 초평면 Normal vector

w.x – b = 0 : 두 집합을 분류하는 가상 직선(1 or -1값)

w.x -b = 1 : X^+ 를 지나는 초평면
X^+ : +1 집합에서 가장 가까운 데이터(점)

w.x-b = -1 : X^- 를 지나는 초평면
X^- : -1 집합에서 가장 가까운 데이터(점)

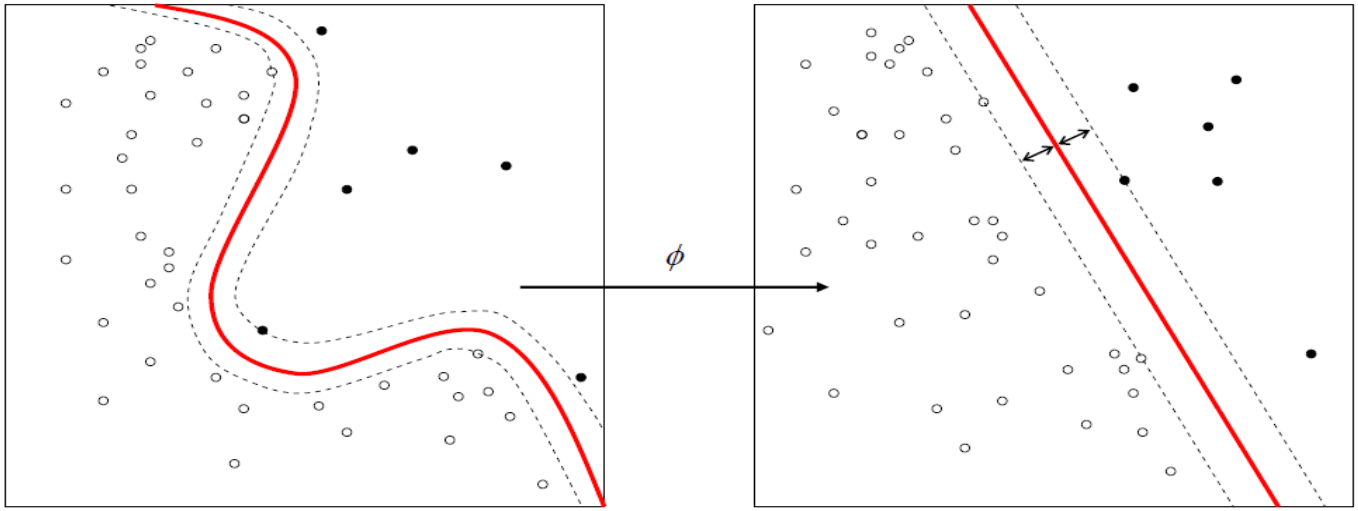
초평면 마진($\frac{2}{\|\mathbf{w}\|}$) : 두 초평면 사이의 거리
(각 서포트 벡터를 지나는 초평면 사이의 거리)

SVM은 이러한 마진을 최대로 만드는 알고리즘

● Non-Linear Classifier Margin

1992년 Bernhard E. Boser, Isabelle M. Guyon and Vladimir N. Vapnik 제안
최대 마진 초평면 문제에 커널 트릭(Kernel Trick) 적용 비선형 분류 제안

- ➢ 기존의 선형 분류 알고리즘과 형태상 비슷하지만 내적 연산이 비선형 커널 함수로 대체
- ➢ 커널 함수 이용 변환된 특징 공간(feature space)의 최대 마진 문제 해결
  - ✓ 변환 : 비선형 변환이거나 차원을 높이는 변환
- ➢ kernel : 비선형(non linear) 관계를 선형적(linear)으로 변환하는 역할
- ➢ kernel 종류 : linear, polynomial, radial, sigmoid



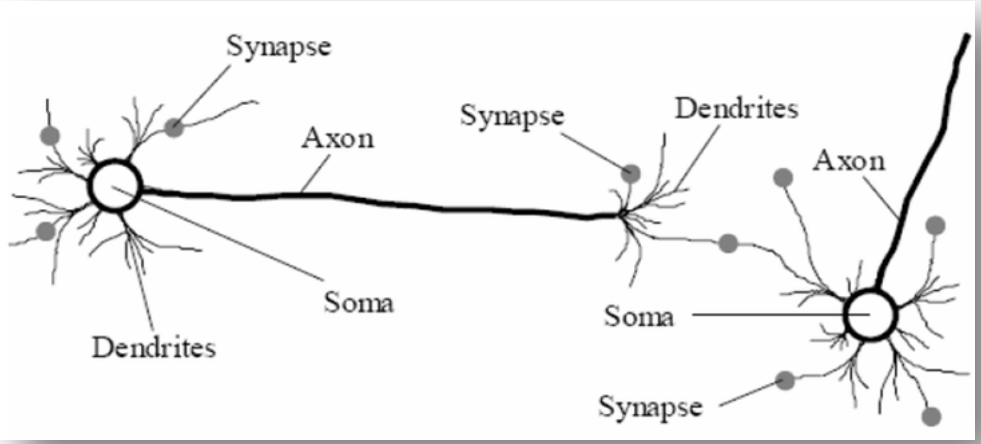
5. Artificial Neural Network 알고리즘

Chap05_ANN 수업내용

1) 인공신경망 개요
2) 인공신경망 구성요소
3) 생물학적 신경망 vs 인공신경망
4) 시냅스 가중치 결정
5) 신경망 활성함수
6) 신경망 학습과정
7) 신경망 주요 특징

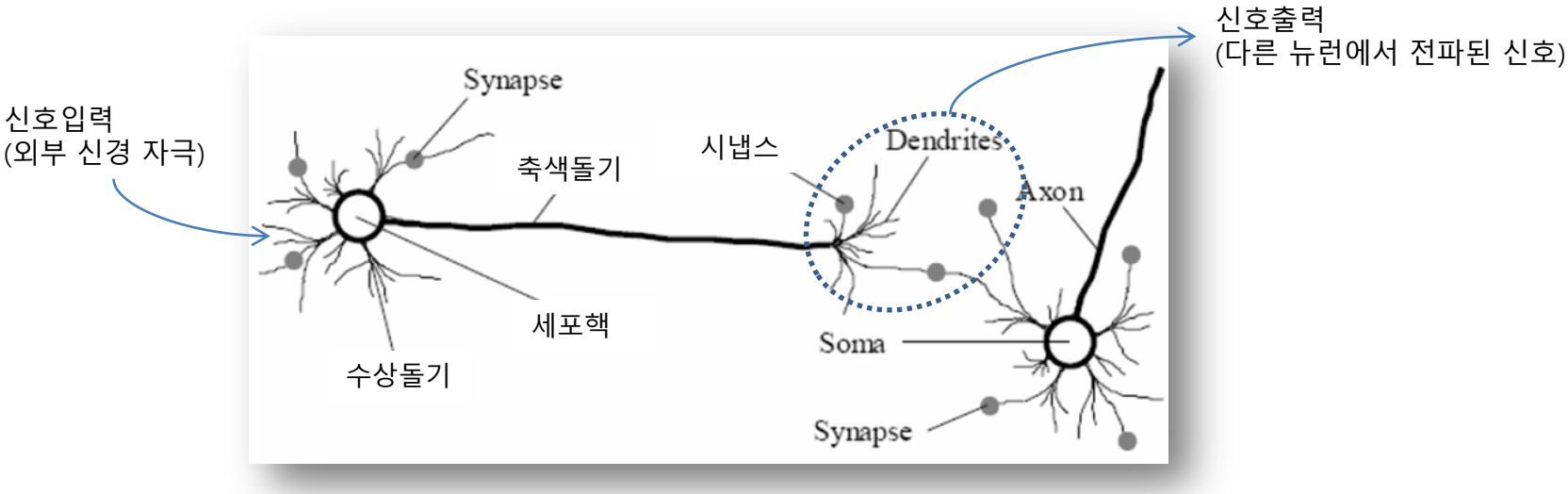
1) 인공신경망 개요

- 컴퓨터가 사람처럼 스스로 학습할 수 있도록 인공 신경망 (ANN: artificial neural network)을 기반으로 한 기계 학습
- 인간의 두뇌가 수많은 데이터 속에서 패턴을 발견한 뒤 사물을 구분하는 정보처리 방식 모방
- 컴퓨터가 사물을 분별하는 기계 학습
  - ✓ 인간 개입 없이 컴퓨터가 스스로 인지·추론·판단
- 적용분야
  - ✓문자 인식, 필기체 인식, 음성 인식, 석유 연간 소비량 예측 등
  - ✓음성, 이미지 인식, 증권시장 예측, 날씨 예보 등 활용, 구글 알파고(딥 러닝)



인간 신경망 구조

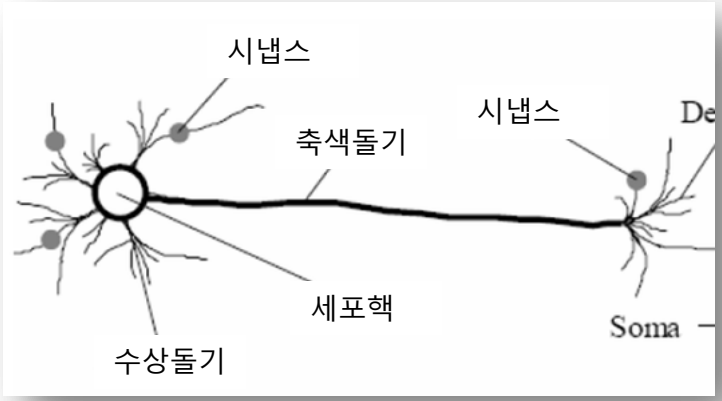
2) 신경망(Neural Network) 구성 요소



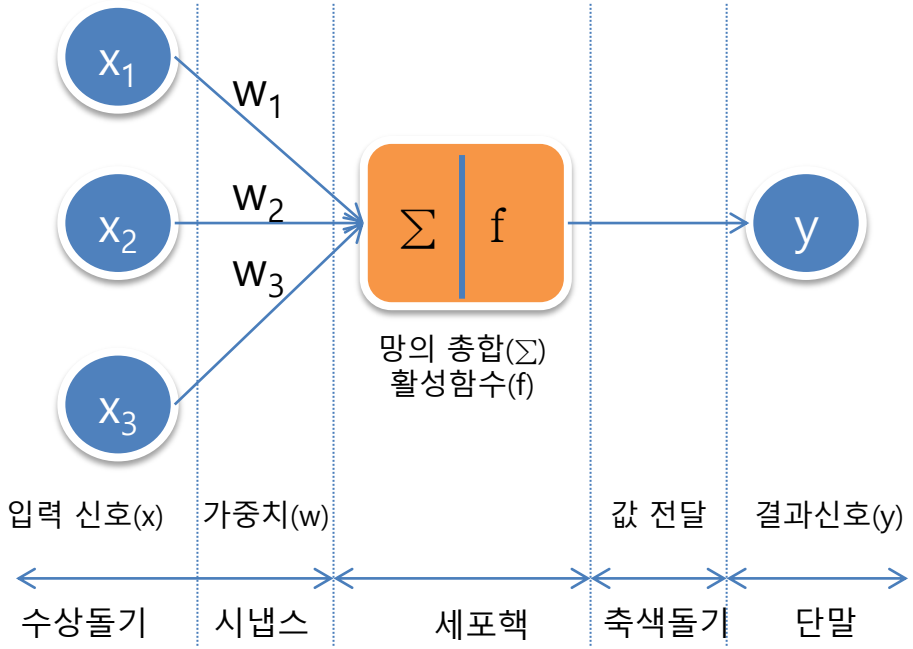
- 수상돌기(Dendrites) : 외부로부터 신경 자극을 받아들이는 역할
- 세포핵(Soma) : 여러 뉴런으로부터 전달되는 신경 자극에 대한 판정 및 다른 뉴런으로 신호 전달 여부 결정
- 축색돌기(Axon) : 전류와 비슷한 형태로 다른 뉴런으로 신호 전달 기능
- 시냅스(Synapse) : 축색돌기 말단과 또 다른 뉴런의 수상돌기 연결 기능
  - ✓ 다른 뉴런의 축색돌기로부터 전달 받은 신호를 어느 정도의 세기(Weight)로 전달할 것인지를 결정

3) 생물학적 신경망 vs 인공신경망

- **생물학적 Neural Network**



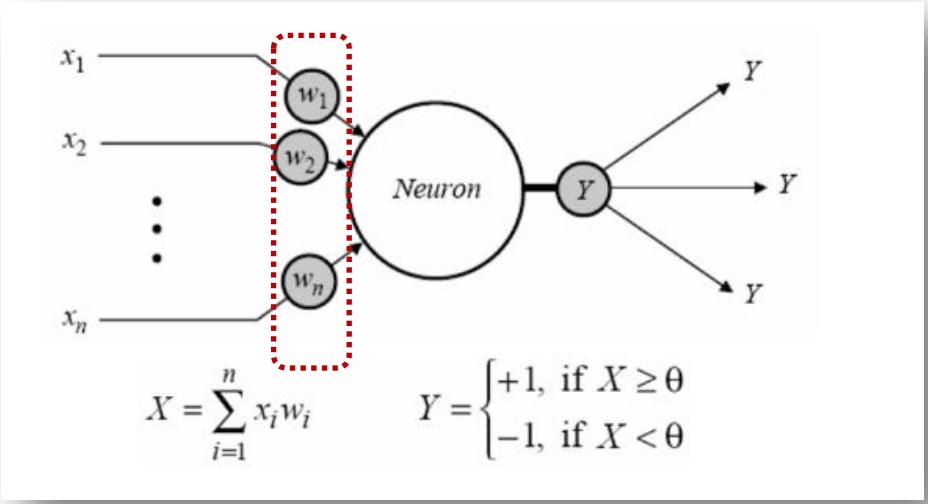
- **Artificial Neural Network**



망의 총합(∑) : 입력신호(X)와 가중치(w) 곱의 합 $\sum_{i=1}^{n} w_i x_i$

활성함수(f) : 망의 총합을 받아서 축색돌기에 결과 신호(y) 출력 $y = f\left[\sum_{i=1}^{n} w_i x_i\right]$

4) 시냅스(Synapse) : 가중치 결정

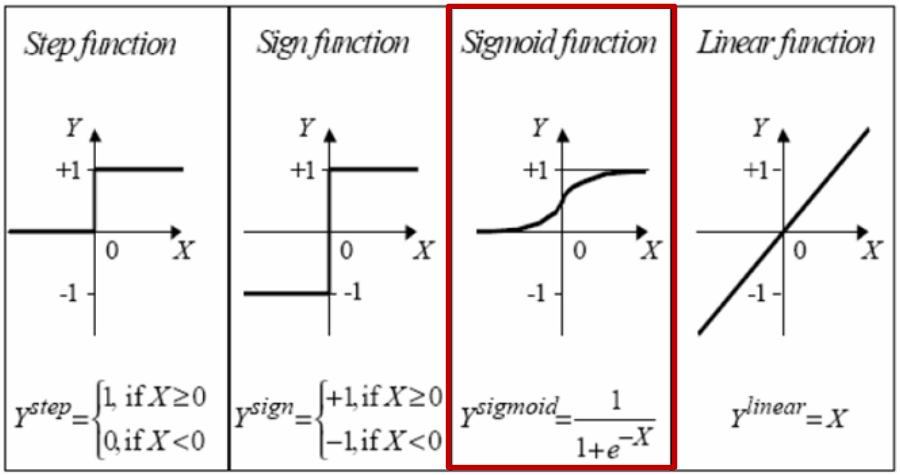


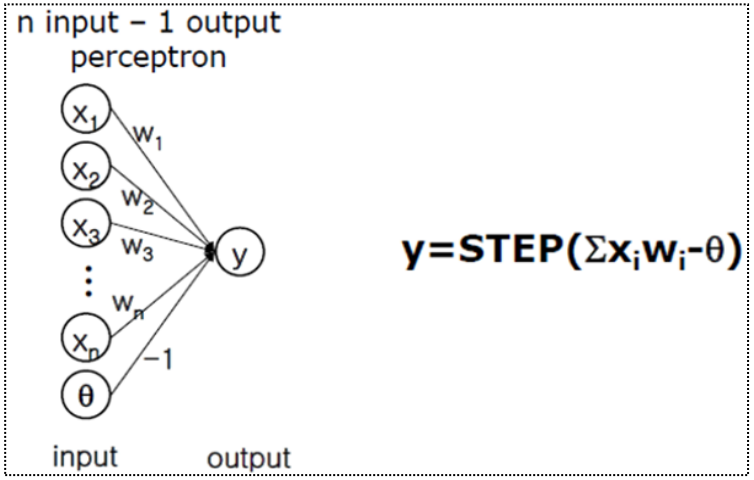
- 입력 값(x1, x2, … x_n) : 수상돌기에 해당하는 외부 신경 자극
- 가중치 (w1, w2, … w_n) : 한 신경 세포에서 다른 신경 세포를 연결하는 시냅스에서 신호 세기 결정
  - ✓ 신호 세기 = 가중치(w1, w2, … w_n)
- 세포핵에서 활성함수에 따라 최종 출력 Y 결정
  - ✓ 특정 경계값(Threshold)과 비교하여 Y의 출력 값 결정 : (0,1), (-1,+1), (-inf, +inf)

5) 신경망 활성함수

➢신경망 활성함수로 사용되는 함수

✓Step function, Sign function, Sigmoid function, Linear function





➢ Step function

✓ 0과 1로 극단적인 상황만 있기 때문에 현재 다중 Layer 신경망에서 사용되지 않음

➢ Sigmoid function

✓0~1 까지 연속적으로 변화하는 출력값을 갖기 때문에 가중치나 바이어스(bias) 변화 시 출력에 변화를 준다.(현재 활성함수로 가장 많이 사용)

6) 신경망 학습과정

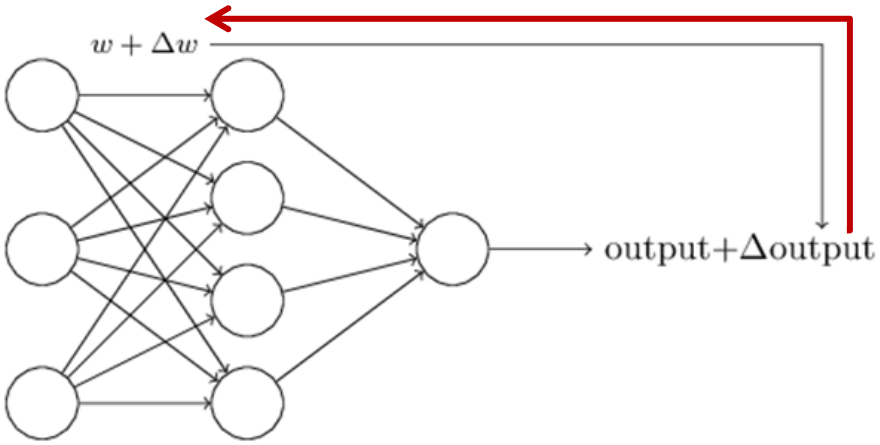
➢훈련(Training)

✓ 역전파 알고리즘을 통해 입력 값을 대상으로 시냅스의 가중치를 조절하는 과정

✓ 훈련 데이터를 이용하여 반복 훈련을 통해서 최적의 가중치 (w1, w2, … w_n) 결정 과정

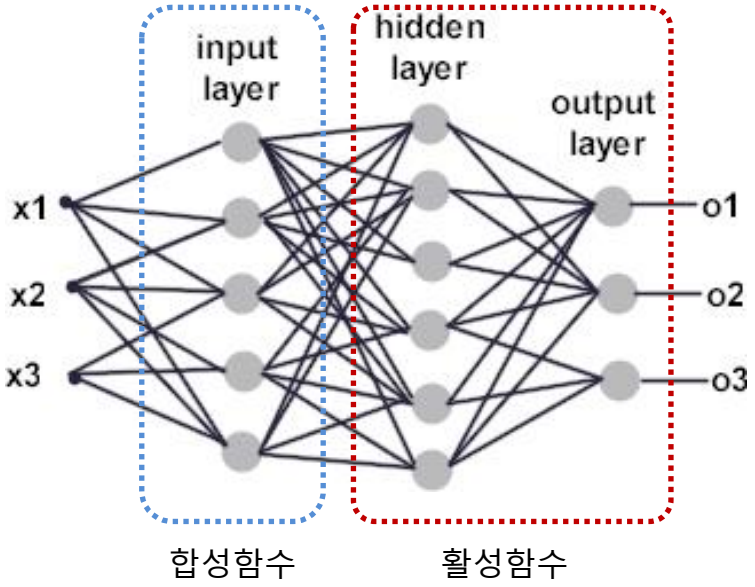
➢ 역전파(Backpropagation) 알고리즘

✓출력에서 생긴 오차를 입력 쪽(역방향)으로 전파시켜 순차적으로 편미분을 수행하면서 w와 b 등을 갱신하여 훈련 데이터에 최적화된 w와 b값을 얻는 알고리즘



7) 인공 신경망(ANN) 주요 특징

인공 신경망의 입력층, 은닉층, 출력층



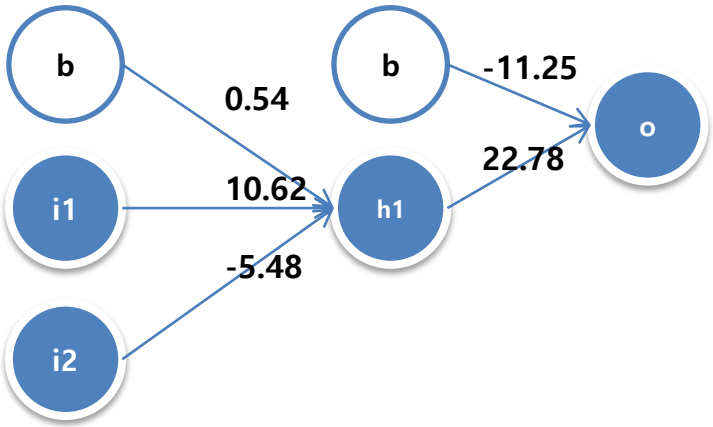
- ➢ 다수 입력 값에 학습을 부여하여 다수의 출력 값 산출 모형
  - ✓ 학습 : 입력 값에 가중치를 정하는 과정
  - ✓ 모형의 출력값과 실제값과의 차이를 최소화하기 위해 최적의 가중치를 찾는 과정
  - ✓ 가중치 : 입력 신호의 중요도/빈도(신경망 연결강도)
- ➢입력/출력 값 : 일반적으로 0~1 사이의 연속 값
- ➢각 계층 노드 수 : 분석자가 지정
  - ✓입력 노드 : 입력 데이터 속성 수
  - ✓출력 노드 : 분류 개수 또는 모델 결과 수
  - ✓ 은닉 노드 : 입력노드, 데이터 수 등을 고려하여 지정
- ➢ 합성함수(Combination Function)
  - ✓ 입력정보들을 하나의 정보로 결합(가중평균 이용)
- ➢활성함수(Activation Function)
  - ✓ 합성값을 일정범위 값으로 축약(은닉층/출력층 전달)

➢ 다중 은닉층(Multi Hidden Layer)

✓ 1개 이상의 은닉층을 갖는 모델을 딥 런닝(Deep Learning)이라고 한다.

● 가중치(입력노드, 은닉노드, 출력노드, 바이어스)

summary(model_net) # 입력노드(2), 상수(1) -> 은닉노드(1), 상수(1) -> 출력 노드(1)
#a 2-1-1 network with 5 weights
#options were - entropy fitting
#b->h1 i1->h1 i2->h1 - bias -> 은닉노드(h1) 입력노드(i1) -> 은닉노드(h1) 입력노드(i2) -> 은닉노드(h1)
#0.54 10.62 -5.48
#b->o h1->o - bias(b) -> 출력노드(o) 은닉노드(h1) -> 출력노드(o)
#-11.25 22.78

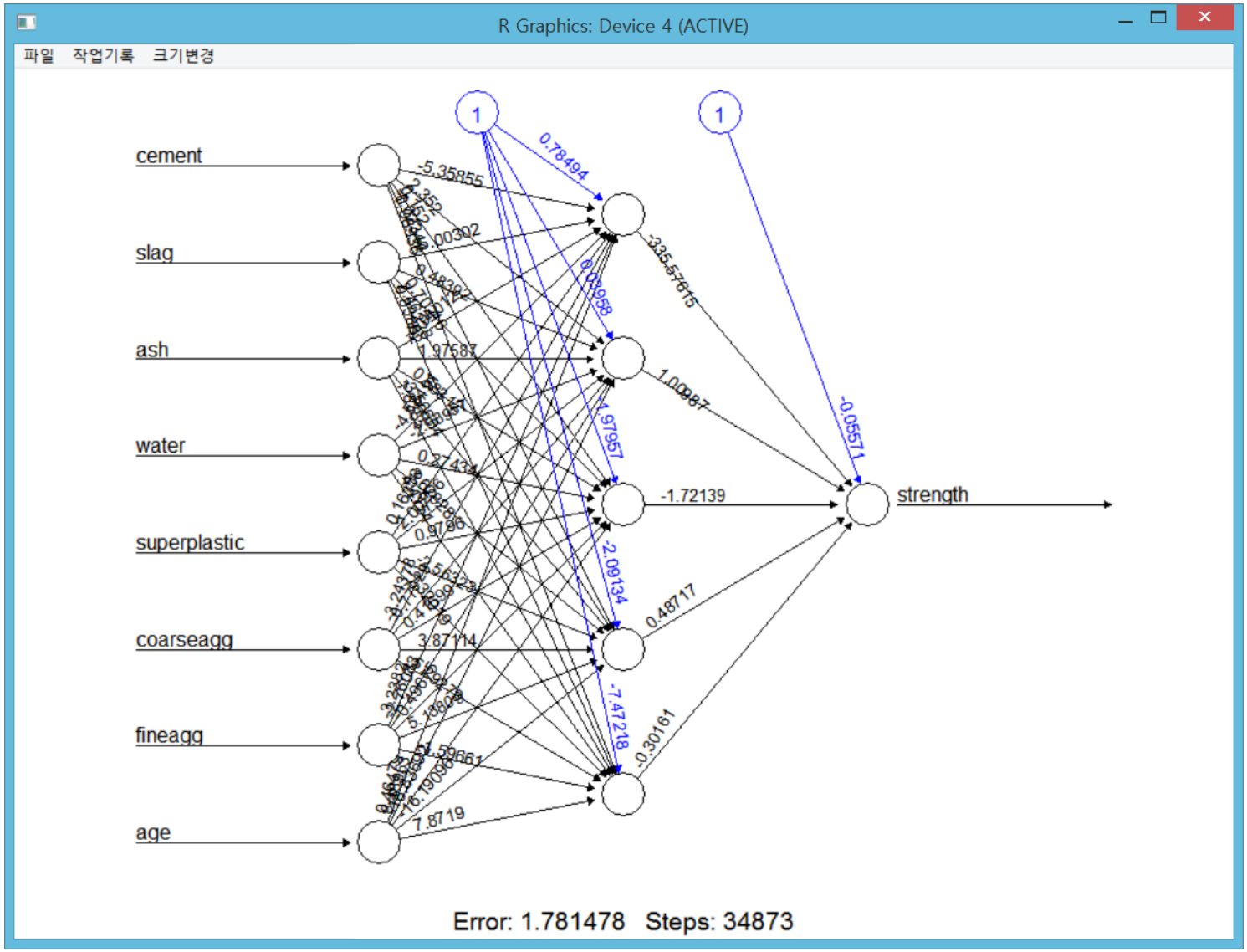


바이어스(bias) : 시그모이드 함수에 의해서 생성된 출력 경계값(0과 1 사이)으로 신호 전달 역할

h1 = 10.62(x1) + -5.48(x2) + 0.54

o = 22.78(h1) + -11.25

● 콘크리트 내구력 예측 모델 예(건축물의 내구성 추정 응용분야)



6. 군집 & 연관 예측 모형

Chap06_KM_LM 수업내용

1) 군집 예측 모형

2) 연관 예측 모형

1) 군집 예측 모형

- 예제 : 10대들의 성향을 분석하여 마케팅에 활용
- SNS 계정을 가진 30,000명 대상 36개 관심분야 데이터 수집

1그룹(외모지향) : 수영, 치어 점수, 성적 관심, 귀여움 높음

2그룹(브레인) : 대체적으로 스포츠 약함, 외모지향 평균 이하

3그룹(범죄자) : 대체적으로 평균 이상 칼럼이 많음

4그룹(운동선수) : cheerleading을 제외한 스포츠 평균 이상

5그룹(무기력) : 모든 칼럼 값이 음수(대부분 평균 이하 칼럼)

1) 군집 예측 모형

```
> teens[1:5, c("cluster", "gender", "age", "friends", "cute")]
  cluster gender     age friends cute
1       5      M 18.982       7    0
2       1      F 18.801       0    1
3       5      M 18.335      69    0
4       5      F 18.875       0    1
5       2   <NA> 18.995      10    0
```

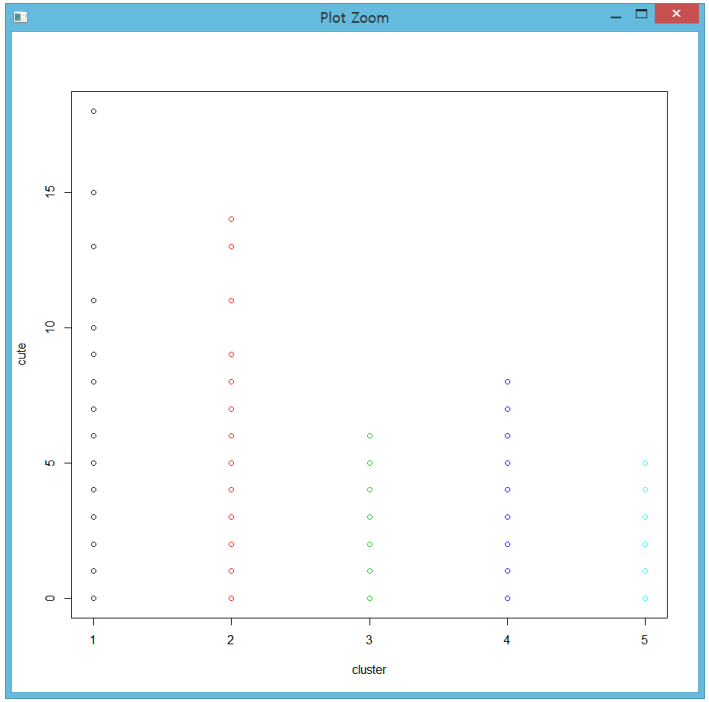
1) 군집 예측 모형

```
> aggregate(data = teens, age ~ cluster, mean)
  cluster      age
1       1 17.06780
2       2 17.11104
3       3 17.35022
4       4 16.86294
5       5 17.30259
```

1) 군집 예측 모형

```
# 차트에 곡선과 별표 추가
install.packages("PerformanceAnalytics")
library(PerformanceAnalytics)

# 비계층적 군집시각화
plot(teens[c("cluster", "cute")], col=teens$cluster)
summary(teens)
```



2) 연관 예측 모형

1. data 가져오기

- # 주의 : sep=',' 생략하면 error 발생
- # groceries.csv는 희소행렬구조(행: 거래, 열 : 식품) - 해당 거래의 식품이면 1, 아니면 0
- groceries <- read.transactions(file.choose(), sep = ',') # groceries.csv
- groceries # 거래와 식품 정보 제공
- #transactions in sparse format with
- #9835 transactions (rows) and
- #169 items (columns)

- summary(groceries) # 거래수/상품수, 주요 식품목록, 식품에 대한 거래수 제공
- #transactions as itemMatrix in sparse format with
- #9835 rows (elements/itemsets/transactions) and -> 거래수
- #169 columns (items) and a density of 0.02609146 -> 식품 수

2. data 탐색/특성 보기

```
# transaction -> data.frame으로 변환하여 데이터 보기
gdf <- as(groceries, 'data.frame') # as.data.frame() 안됨
head(gdf)
inspect(groceries[1:10]) # transction 거래 식품 보기
#    items
#1  {citrus fruit,margarine,ready soups,semi-finished bread}
#  ...
#10 {cereals,whole milk}
itemFrequency(groceries[, 1:5]) # 특정 식품 보기  - 5개 식품보기
# 5개 식품에 대한 지지도 제공

itemFrequencyPlot(groceries, support = 0.1)# 10%지지도 시각화
itemFrequencyPlot(groceries, topN = 15) # 지지도 상위 15개 상품 시각화
image(groceries[1:5]) # 5개 거래에 대한 거래 식품 내역 시각화
# 희소행렬에서 식품 거래 내역 시각화
image(sample(groceries, 50)) # 50개 거래 샘플의 식품 시각화
```

3. 연관모델 생성

➢arules <- apriori(groceries, parameter = list(support=0.1, confidence=0.8, minle=1))
➢# parameter default : support=0.1, confidence=0.8, minle=1
➢arules <- apriori(groceries, parameter = list(support=0.01, confidence=0.08, minle=1))
472 rule
➢#arules <- apriori(groceries, parameter = list(support=0.05, confidence=0.08, minle=1))
19 rule

➢summary(arules) # 규칙이 갖고 있는 상품 수, 연관규칙 척도 제공
➢#rule length distribution (lhs + rhs):sizes
➢#1 2 3
➢#13 363 96 -> 13개 규칙(1개 상품), 363개 규칙(2개 상품), 96개 규칙(3개 상품)

➢inspect(arules) # 규칙 보기 - 14 ~ 19

4. 연관모델 규칙 분석

- inspect(sort(arules, by = 'lift')[1:10]) # 최상위 향상도 보기

- # butter 식품과 자주 구매하는 식품 검색
- butter <- subset(arules, lhs %in% 'butter')
- butter # set of 8 rules
- inspect(butter)

7. Shiny 프로젝트

Chap07_Shiny 수업내용

1) Shiny 프로젝트

2) 반응형 인터페이스