

2장 목차

2018년 5월 23일 수요일 오전 10:05

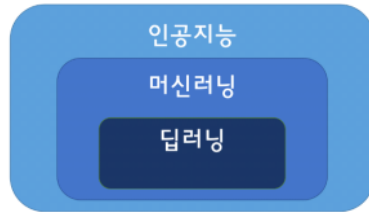
■ 목차

- 1.자료구조
- 2.R에 데이터 로드하는 방법 4가지
- 3.중심경향 측정
- 4.히스토그램과 막대그래프
- 5.정규분포
- 6.분산과 표준편차
- 7.범주형 변수 살펴보기
- 8.최빈값
- 9.산포도
- 10.이원 교차표

1.머신러닝이란?

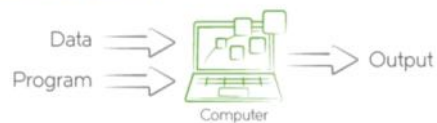
2018년 5월 23일 수요일 오전 10:05

■ 머신러닝과 딥러닝이란?

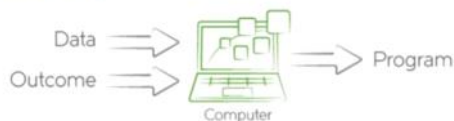


■ 머신러닝이란?

Traditional Programming



Machine Learning



- 기존의 프로그램은 데이터와 프로그램 (소스)을 입력하여 결과를 출력
- 머신 러닝은 데이터와 그 데이터의 결과를 입력하여 (기계가)프로그램을 만들어낸다.

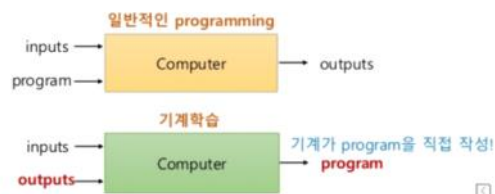
기계학습(Machine Learning)

14

• 기계학습(Machine Learning)

Machine learning is the subfield of [computer science](#) that "gives computers the ability to learn without being explicitly programmed"

기계 학습은 "컴퓨터에 명시적으로 프로그래밍하지 않고 학습 할 수 있는 능력을 부여하는" 컴퓨터 과학의 하위 분야입니다.

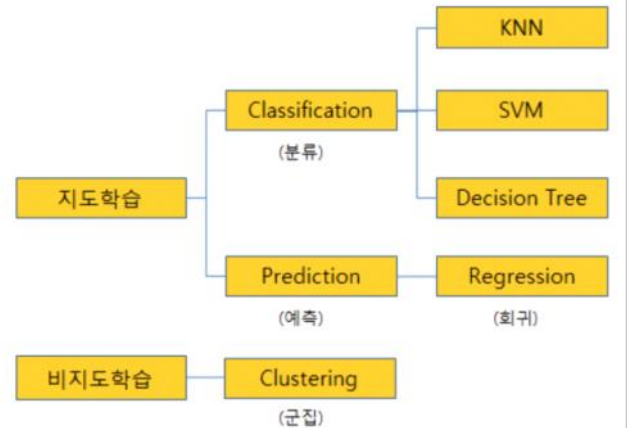


■ 머신러닝의 종류?

How Much Information Does the Machine Need to Predict? Y LeCun

- "Pure" Reinforcement Learning (cherry)
 - ▶ The machine predicts a scalar reward given once in a while.
 - ▶ **A few bits for some samples**
- Supervised Learning (icing)
 - ▶ The machine predicts a category or a few numbers for each input
 - ▶ Predicting human-supplied data
 - ▶ **10→10,000 bits per sample**
- Unsupervised/Predictive Learning (cake)
 - ▶ The machine predicts any part of its input for any observed part.
 - ▶ Predicts future frames in videos
 - ▶ **Millions of bits per sample**

■ (Yes, I know, this picture is slightly offensive to RL folks. But I'll make it up)



구분	설명	해당 알고리즘
지도학습	훈련 데이터와 정답을 가지고 데이터를 분류/예측하는 함수를 만들어내는 기계학습의 한 방법	분류 : Knn, 나이브베이즈, 결정트리, rule-base 알고리즘, 서포트 벡터머신 회귀 (예측) : 선형회귀, 신경망
비지도학습	정답 없이 훈련 데이터만 가지고 데이터로부터 숨겨진 패턴/규칙을 탐색하는 기계학습의 한 방법	클러스터링 : k-menas, 연관규칙 (아프리오 알고리즘)
강화학습	어떤 환경에서 정의된 에이전트가 현재의 상태를 인식하여 선택 가능한 행동들 중 보상을 최대화 하는 행동 혹은 행동 순서를 선택하는 방법	

Machine Learning Types	Tasks	Analysis methods/Algorithms
지도학습 (Supervised Learning)	예측, 추정 (Prediction, Estimation)	<ul style="list-style-type: none"> Linear Regression Regression Tree, Model Tree SVM(Support Vector Machine) Neural Network, Deep Learning ARIMA, Exponential Smoothing
	분류 (Classification)	<ul style="list-style-type: none"> Decision Tree Logistic Regression, Discriminant Analysis k-NN(k-Nearest Neighbor), CBR(Case-Based Reasoning) Naive Bayes Classification SVM, Neural Network Ensemble (Bagging, Boosting, Random Forest)
비지도학습 (Unsupervised Learning)	패턴/구조 발견 (Pattern/Rule)	<ul style="list-style-type: none"> Association Rule Analysis, Sequence Analysis Network Analysis, Link Analysis, Graph theory Structural Equation Modeling, Path Analysis
	그룹화 (Grouping)	<ul style="list-style-type: none"> k-Means Clustering, Hierarchical Clustering, Density-based Clustering, Fuzzy Clustering SOM(Self-Organizing Map)
	차원 축소 (Dimension Reduction)	<ul style="list-style-type: none"> PCA(Principal Component Analysis), Factor Analysis, SVD(Singular Value Decomposition)
	영상, 이미지, 문자 (Video, Image, Text, Signal processing)	<ul style="list-style-type: none"> Wavelet/Fast Fourier Transformation, DTW(Dynamic Time Warping), SAX(Symbolic Aggregate Approximation), Line/Circular Hough Transformation Text mining, Sentiment Analysis

[R 분석과 프로그래밍] <http://rfriend.tistory.com>

정규분포

2018년 5월 23일 수요일 오후 4:09

<https://terms.naver.com/entry.nhn?docId=3569149&cid=58944&categoryId=58970>

많은 경영, 경제, 사회현상, 자연현상들이 정규분포의 형태를 띠고 있는데 예를 들면 한국 성인 남자의 평균 키가 173cm 라고 하면 평균키가 173cm라는 것은, 곧 키가 평균 173cm에서 크게 벗어나지 않는 사람들이 많고 상대적으로 벗어난 150cm 또는 190cm인 사람들은 별로 없다는 소리이다.

평균에서 떨어질수록 데이터 분포가 감소하여 종모양의 형태를 띠며 정규분포 그래프를 3등분하면 평균 근처의 비율이 68% 정도 된다.

문제 208. 정규 분포 그래프를 그리시오.

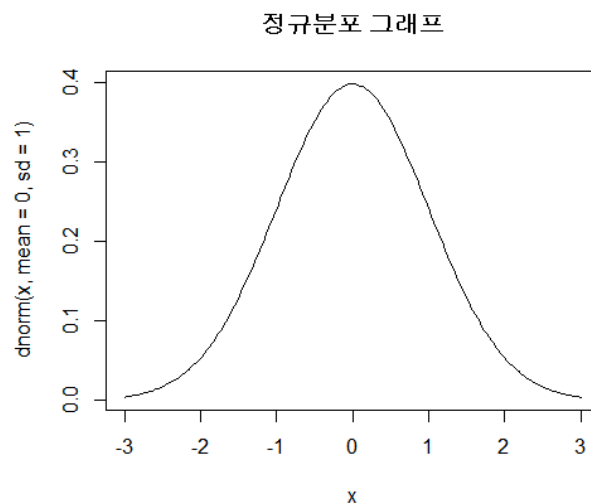
1. -3 ~ 3까지의 데이터 200개를 만든다.

```
x<-seq(-3,3,length=200)
```

2. 위에서 만든 200개의 데이터로 plot 그래프를 그린다.

```
plot(x,dnorm(x,mean=0,sd=1),type='l',main='정규분포 그래프')
```

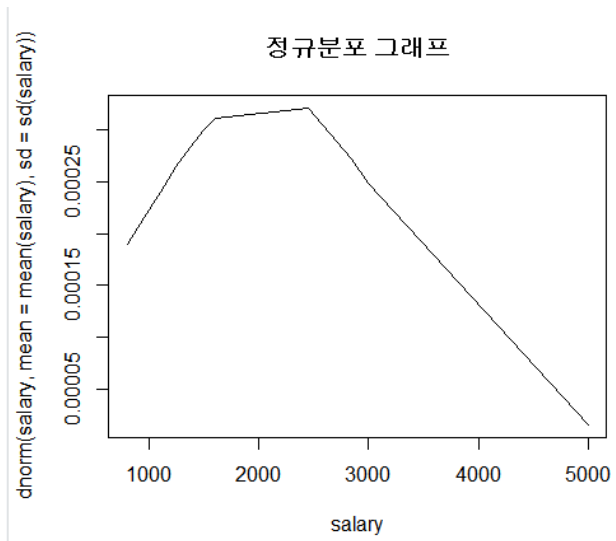
```
# dnorm (~) : y축 , dnorm : 밀도함수 , sd : 표준편차, type="l" : 직선,
```



문제 209. Emp 테이블의 월급을 정규분포 그래프로 그리시오.

```
salary<-sort(emp$sal)
```

```
plot(salary,dnorm(salary,mean=mean(salary),sd=sd(salary)),type='l',main='정규분포 그래프')
```



문제 210. Emp 테이블의 월급에서 정규분포내의 평균근처의 68%에 해당하는 월급이 얼마부터 얼마인지를 출력 하시오.

데이터 타입

2018년 5월 24일 목요일 오전 9:50

1. 데이터의 종류

1. 범주형 데이터 (명목형, 순서형) [naïve bayes 알고리즘 ..]

- a. 범주(값의 목록)를 갖는 vector
- b. factor() 함수를 통해서 생성
- c. Factor는 **명목형(nominal)**, **순서형(ordinal)** 형식 2가지가 존재
- d. Nominal은 level의 순서 값이 무의미 하며 알파벳 순서로 정의
- e. Ordinal은 level 순서의 값을 직접 정의해서 원하는 순서로 정의할 수 있다.

2. 수치형 데이터 (이산형 데이터, 연속형 데이터) [knn알고리즘 ..]

- o **이산형 데이터** (discrete : 뚜렷이 구별된다. 불연속적이다)

Ex) 1 아니면 0 이다. 1.2354.. X

신호가 있거나 없거나

2016년 음주운전 적발건수가 22만 6599건 (계수 : 헤아려 얻는 것)

- o **연속형 데이터** : 연속적인 값을 갖는 데이터

*연속형 데이터에 대한 기술적인 통계를 이용한 자료 요약 3가지

1.데이터의 중심화 경향 :

중앙값, 평균값, 최빈값(가장 많이 출현되는 값)

2.데이터의 퍼짐 정도 :

분산(데이터의 퍼짐 정도), 표준편차(평균에 대한 오차), 범위

3.데이터의 분포와 대칭정도 :

왜도(좌우로 기울어진 정도), 첨도(위아래로 뾰족한 정도)

Ex) 신장, 체중 (82.31) (계량 : 측정해서 얻는 것)

-----> 이산형 데이터보단 연속형 데이터가 얻을 수 있는 데이터가 많다.

2. 범주형 데이터 살펴보기

범주형 데이터 보는 방법 2가지

- 1. table 함수 # 건수를 출력
- 2. prop.table 함수 # 비율을 출력

예제 : 사원 테이블의 부서번호, 부서번호별 인원수를 출력 하시오.

```
table(emp$deptno)
```

```
> table(emp$deptno)
```

```
10 20 30  
3  5  6
```

prop.table(table(emp\$deptno)) #prop.table(x) x값의 비율이 나온다

```
> prop.table(table(emp$deptno))
```

```
      10      20      30  
0.2142857 0.3571429 0.4285714
```

문제 232. 중고차의 색깔과 색깔별 비율이 어떻게 되는지 출력 하시오.

```
head(usedcars)
```

```
> head(usedcars)
  year model price mileage color transmission
1 2011   SEL 21992   7413 Yellow          AUTO
2 2011   SEL 20995  10926   Gray          AUTO
3 2011   SEL 19995   7351 silver          AUTO
4 2011   SEL 17809  11613   Gray          AUTO
```

```
round(prop.table(table(usedcars$color))*100,1)
```

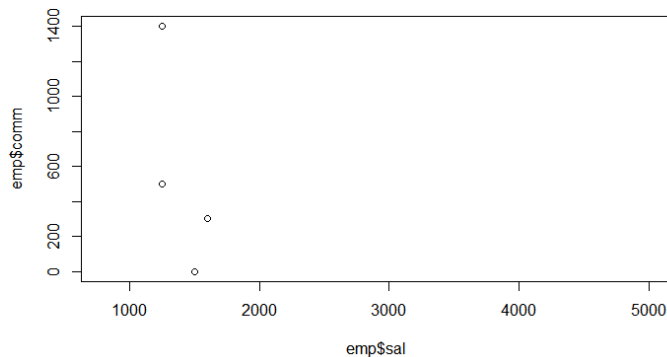
```
> round(prop.table(table(usedcars$color))*100,1)

Black   Blue   Gold   Gray   Green   Red Silver white Yellow
 23.3   11.3    0.7  10.7    3.3   16.7  21.3  10.7    2.0
```

■ 산포도 그래프

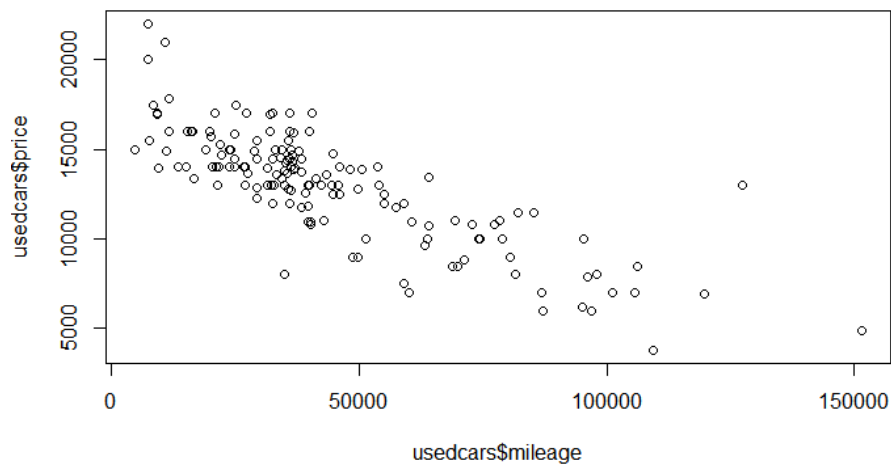
문제 233. 커미션을 받는 직원들의 월급의 분포도가 어떻게 되는지 산포도 그래프로 확인하시오.

```
plot(emp$sal, emp$comm)
```



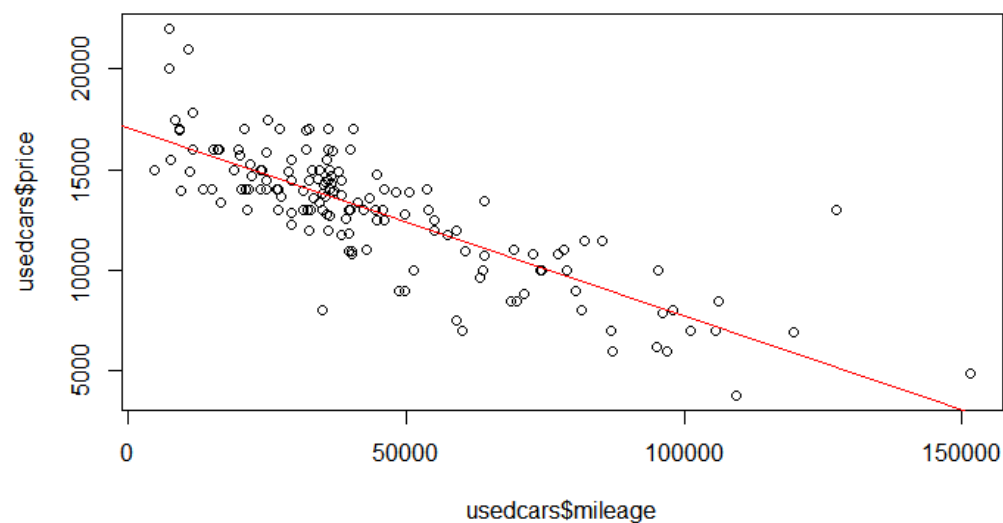
문제 234. 중고차의 주행거리가 높으면 중고차의 가격이 낮아진다는 것을 plot 그래프로 확인해 보시오.

```
plot(usedcars$mileage, usedcars$price)
```



문제 235. 위의 plot 그래프에 직선을 하나 그으시오.

```
attach(usedcars)
model<-lm(price~mileage)
model
plot(usedcars$mileage, usedcars$price)
abline(model,col="red")
```

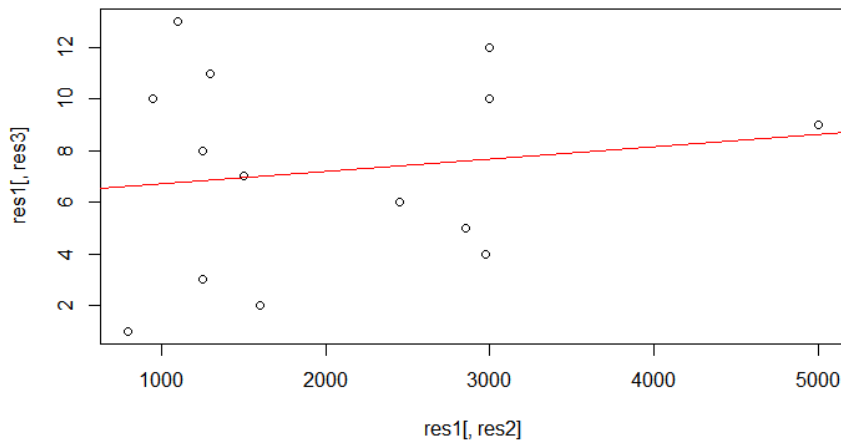


문제 236. 산포도 그래프와 평균 회귀직선을 그리는 함수를 생성 하시오.

```
plot_regression <- function(){
  res1<-get(readline(prompt = '산포도 그래프를 그릴 테이블명을 입력 하세요 : '))
  res2<-menu(colnames(res1),title = 'x축 데이터 입력 : ')
  res3<-menu(colnames(res1),title = 'y축 데이터 입력 : ')

  graphics.off()
  model<-lm(res1[,res3]~res1[,res2])
  plot(res1[,res2],res1[,res3])
  abline(model,col="red")
}
```





문제 237. 오늘 만든 3개의 함수를 그래프 자동화 스크립트에 추가 하시오.

```
baek_func <- function() {

  # 현재 컴퓨터에 필요한 패키지 설치
  # 만약 패키지가 존재하지 않는 경우에만 설치

  packages <- c("XML","stringr","rJava","KoNLP","wordcloud","wordcloud2","lubridate")

  if (length(setdiff(packages, rownames(installed.packages()))) > 0) {
    install.packages(setdiff(packages, rownames(installed.packages())))
  }

  graphics.off()
  x1 <- menu( c("막대그래프","원형그래프","산포도그래프","워드클라우드","사분위수그래프","분산시각화","정규
분포&히스토그램","산포도그래프&회귀직선"),title='원하는 그래프의 숫자를 선택하세요 ')

  res0<- get(readline(prompt = '테이블명 입력 : '))
  if (x1 == 1 | x1 == 2){
    res1<- menu(colnames(res0), title='토탈 값을 구할 컬럼 선택 : ')
    res2<- menu(colnames(res0), title='그룹핑할 컬럼 선택 : ')
    q<-tapply(res0[,res1], res0[,res2],sum)
  }

  switch(x1,
    {#막대그래프
      q[is.na(q)] <- 0
      barplot(q, col = rainbow(nrow(q)), main = paste( colnames(res0)[res2], '별', colnames(res0)[res1],'총합' ), beside =
T, ylim = c(0,max(q)*1.4))
      legend("topright", rownames(q),title = paste(colnames(res0)[res2],' 구분' ),inset = 0,fill = rainbow(nrow(q)),cex=
0.8)
    },
    {#원형 그래프
      label<-paste(unique(res0[,res2]), round(q/sum(q) * 100,1),'%')
      pie(q,col=rainbow(nrow(q)),label=label,main = paste( colnames(res0)[res2], '별',colnames(res0)[res1],'총합' ))
    },
    {#산포도 그래프
      x <- menu(colnames(res0), title='x축 컬럼 선택 : ')
      y <- menu(colnames(res0), title='y축 컬럼 선택 : ')
      plot(res0[,x],res0[,y],pch=16, col=blues9,xlab = colnames(res0)[x] ,ylab = colnames(res0)[y],main =
```

```

paste(colnames(res0)[x], '와 ', colnames(res0)[y], '의 상관 관계 ')

},
{#워드클라우드

library(wordcloud)
library(KoNLP)
library(plyr)
useSejongDic()          # 370957개의 한글 단어가 추가 (전희원 선생님이 만듦)

graphics.off()

res<-readline(prompt = 'c:\\data 경로에 위치한 txt 파일명 입력 : ')
word<-readLines(gsub(' ', '', paste('c:\\\\data\\\\', res, '.txt'))))

nouns <- extractNoun(word) # 연설문에서 명사만 출력
nouns <- nouns[nchar(nouns)>=2] # 두글자 이상인 명사만 추출
cnouns <- count(unlist(nouns)) # 단어와 건수 출력

pal <- brewer.pal(6, "Dark2") # Dark2라는 색깔을 추가하는 작업
pal <- pal[-1]
windowsFonts(malgun=windowsFont("맑은 고딕")) # 맑은 고딕 폰트 추가
wordcloud(words=cnouns$x, freq=cnouns$freq, colors=pal, min.freq=3,
          random.order=F, family="malgun")
},
{#사분위수 그래프
res1<- menu(colnames(res0), title='컬럼 선택 : ')
boxplot(res0[,res1], horizontal = T, col = blues9)
},
{#분산시각화
res1<- menu(colnames(res0), title='컬럼 선택 : ')

plot(res0[,res1], main=paste('분산= ', round(var(res0[,res1]), 4), '표준편차= ', round(sd(res0[,res1]), 4), col='blue')
abline(h=mean(res0[,res1]), lty=2, col='red')

},
{#정규분포&히스토그램
library(fBasics)
res1<-menu(colnames(res0), title='컬럼 선택 : ')

x<-sort(res0[,res1])

hist(x, col=blues9, axes = F, ann=F)
par(new=T)
plot(x, dnorm(x, mean=mean(x), sd=sd(x)), type='l', lwd=3, col='red', main=paste('왜도값 : ', round(skewness(x), 4)))
},
{#산포도 그래프&회귀직선
res2<-menu(colnames(res0), title = 'x축 데이터 입력 : ')
res3<-menu(colnames(res0), title = 'y축 데이터 입력 : ')

graphics.off()
model<-lm(res0[,res3]~res0[,res2])
plot(res0[,res2], res0[,res3])
abline(model, col="red")
}
)

```

}

데이터 중심화 경향

2018년 5월 23일 수요일 오후 3:41

■ 최빈값 104p

가장 빈번히 발생한 값, 출현 빈도가 높은 데이터
평균값과 중앙값 외에 반드시 알아야 되는 값이 최빈값이다.

■ 데이터에 대한 일반적인 통계결과 확인하는 방법

```
summary(emp$sal)
```

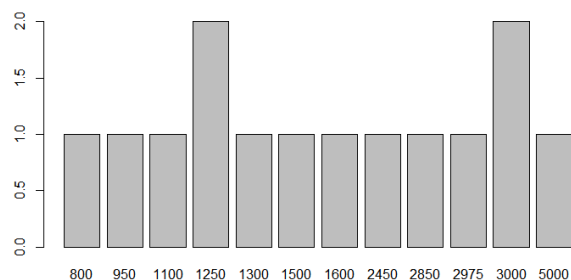
```
> summary(emp$sal)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   800    1250    1550    2073    2944    5000
```

중앙값과 평균값을 확인한다.

■ 막대 그래프와 히스토그램 그래프 97p

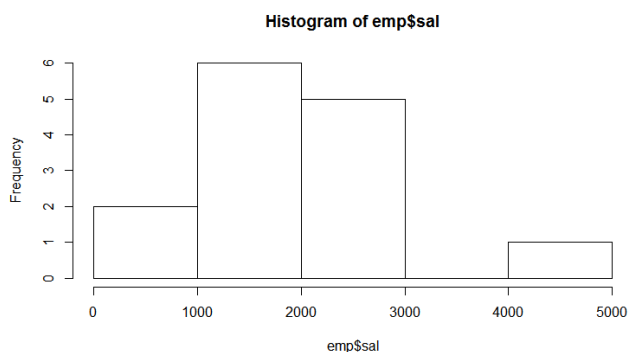
1. 막대 그래프 : 데이터안의 특정 항목들의 수량이 많고 적음을 나타낼 때 용이

```
barplot(table(emp$sal))
```

 # 이 데이터는 barplot 으로 표현하기엔 부적절해 보인다.

2. 히스토그램 그래프 : 데이터안에 데이터의 분포의 상태를 파악하거나 비교할 때 용이하다.

```
hist(emp$sal)
```



1. 예제

예제	<p>17살인 나는 10대를 위한 수영교실을 수강하려고 한다. 해당 수영교실에 평균나이를 확인해서 내가 들어 가도 되는 교실인지 확인해보려한다. 아래의 데이터를 만들고 아래의 데이터의 나이 평균을 출력 하시오.</p> <p>나이 1 2 3 31 32 33 도수 3 4 2 2 4 3</p>
-----------	--

```
x<-c(rep(c(1,2,3,31,32,33),c(3,4,2,2,4,3)))
mean(x)
```

```
> x<-c(rep(c(1,2,3,31,32,33),c(3,4,2,2,4,3)))
> x
[1] 1 1 1 2 2 2 2 3 3 31 31 32 32 32 32 33 33 33
> mean(x)
[1] 17
```

나이 평균이 17이라서 자신과 비슷한 연령대가 있는 수영 교실인줄 알았는데 사실은 엄마와 아이가 함께 하는 수영교실 이였다..!

-----> 그래서 평균 값만 알아서는 안되고 최빈값을 알아야 된다.

문제 211.	<p>아래의 데이터 x의 최빈값을 구하시오.</p> <pre>> x [1] 1 1 1 2 2 2 2 3 3 31 31 32 32 32 32 33 33 33</pre>
----------------	---

```
x<-c(rep(c(1,2,3,31,32,33),c(3,4,2,2,4,3)))
```

```
y<-data.frame(table(x))
```

```
y[y$Freq==max(y$Freq),]
```

```
> x<-c(rep(c(1,2,3,31,32,33),c(3,4,2,2,4,3)))
>
> y<-data.frame(table(x))
>
> y[y$Freq==max(y$Freq),]
  x Freq
2  2    4
5 32    4
```

문제 212.	우리반의 나이 중에 최빈값을 구하시오.
----------------	-----------------------

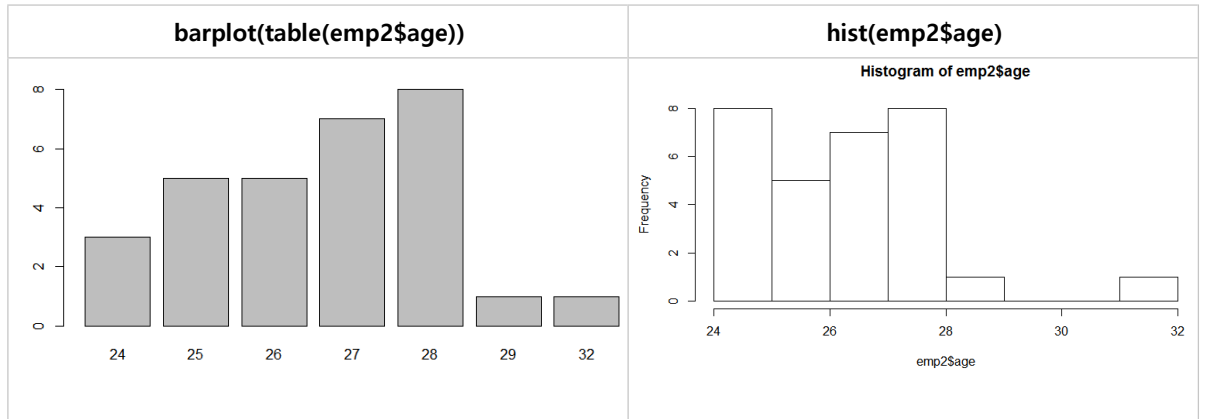
```
a <- table(emp2$age)
```

```
b <- data.frame(a)
```

```
b[b$Freq == max(b$Freq), ]
```

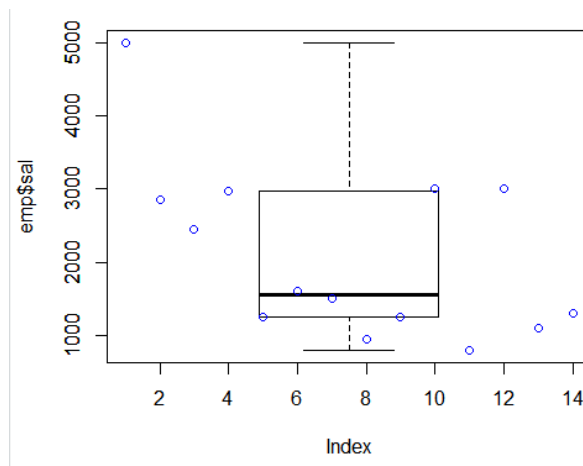
```
> a <- table(emp2$age)
> b <- data.frame(a)
> b[b$Freq == max(b$Freq), ]
  Var1 Freq
5    28    8
```

문제 213. 우리반의 나이 데이터를 막대,히스토그램 그래프로 그려보시오.



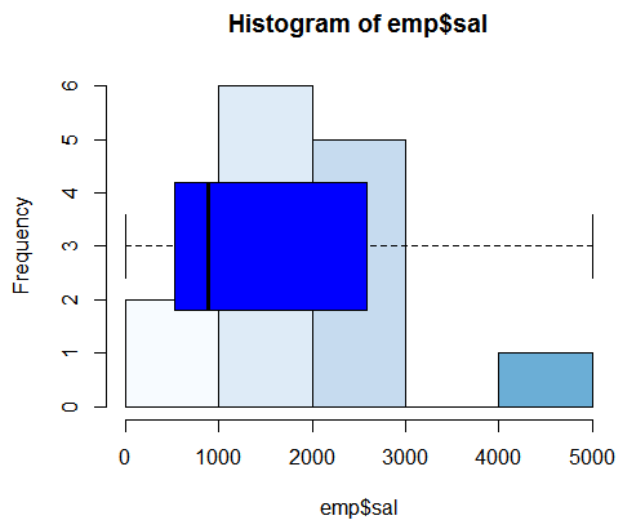
문제 206. 사원 테이블의 월급에 대하여 박스 그래프와 산포도(plot) 그래프를 겹쳐서 보이게 출력 하시오.

```
graphics.off()
boxplot(emp$sal)
par(new=T)
plot(emp$sal, col='blue')
```



문제 207. Emp 테이블의 월급을 사분위수 그래프로 그리는데 옆으로 눕혀서 출력하고 histogram 그래프로 그리시오.

```
graphics.off()
hist(emp$sal, col=blues9)
par(new=T)
boxplot(emp$sal, col='blue',horizontal = T, axes=F)
```



데이터 퍼짐 정도

2018년 6월 23일 토요일 오후 5:56

1. 분산과 표준편차

- 분산 : 데이터의 퍼짐 정도
- 표준편차 : 평균에 대한 오차

■ 왜 분산과 표준편차를 알아야 하는가?

실제 데이터 값이 평균을 기준으로 얼마나 둘썹날썹 하느냐를 나타낸 것이 표준편차이다.

편차란? 원래의 값에서 평균을 뺀 것인데, 편차는 +, - 가 될 수도 있다.

예를 들어 4개의 데이터가 있을 때 평균을 m이라고 가정하고 각 값이 m+1, m-2, m+3, m-4 라고 할 때 편차의 합은 실제로 1+2+3+4 = 10 10이어야 하지만 실제값이 -2, -4 이기 때문에 계산해보면 10이 아니라 영똥한 값이다.

$$1 - 2 + 3 - 4 = -2$$

그래서 이 음수를 양수화 해야 되는데 그러한 방법 중 하나가 제곱이다 편차를 합하기 전에 제곱을 해서 합한다.

$$1^2 + (-2^2) + 3^2 + (-4^2) = 30$$

$$30/4 = 7.5 <---- \text{분산이라고 한다. (편차 제곱의 합/n)}$$

■ 분산을 바로 쓰지 않고 표준편차를 쓰는 이유는?

분산은 편차에 제곱을 하여 계산했기 때문에 실제 값과 너무 멀어져 있다. 그래서 실제 값에 근접시키기 위해서 분산에 루트를 씌워 준게 **표준편차**이다.

표준편차 : 분산(편차 제곱의 합/n)에 루트를 씌운것

문제 213+ 아래의 수학점수를 갖고있는 두개의 학습을 각각 x1,x2,로 만들고 분산과 표준편차를 각각 구하시오.

```
x1 <- c(25,55,60,70,100)
x2 <- c(55,60,60,65,70)
```

```
x1 <- c(25,55,60,70,100)
x2 <- c(55,60,60,65,70)
```

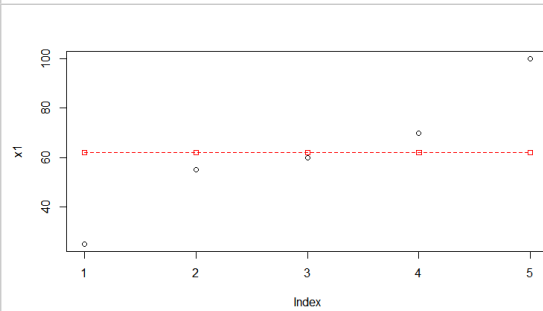


```
sd(x1)    # sd : 표준편차
sd(x2)
var(x1)   # var : 분산
var(x2)
```

문제 214. 산포도 그래프를 이용해서 두 클래스의 데이터를 시각화 하시오.

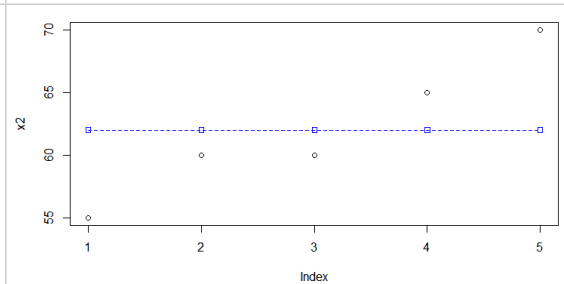
```
x1_mean<-rep(mean(x1),5)
```

```
plot(x1)
lines(x1_mean,type="o",pch=22,lty=2, col='red')
```



```
x2_mean<-rep(mean(x2),5)
```

```
plot(x2)
lines(x2_mean,type="o",pch=22,lty=2, col='blue')
```



화면을 분할하여 한번에 여러 그래프 출력

```
graphics.off()
```

```
par(mfrow=c(1,2))    # 화면 분할 (행,열)
```

```
par(mar=c(1,1,1,5))  # 여백주기 (아래, 왼쪽, 위, 오른쪽)
```

```
plot(x1,ylim=c(1,100))
```

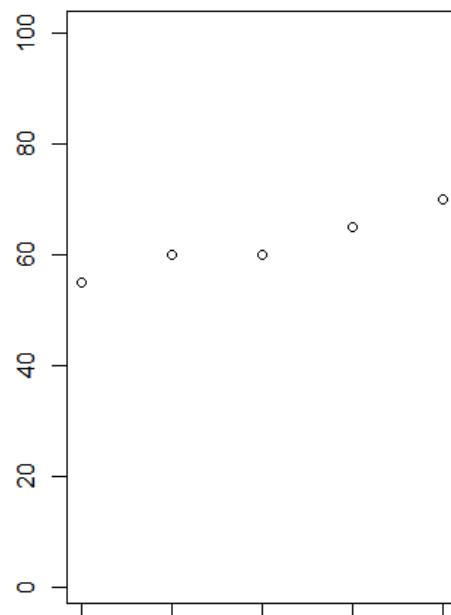
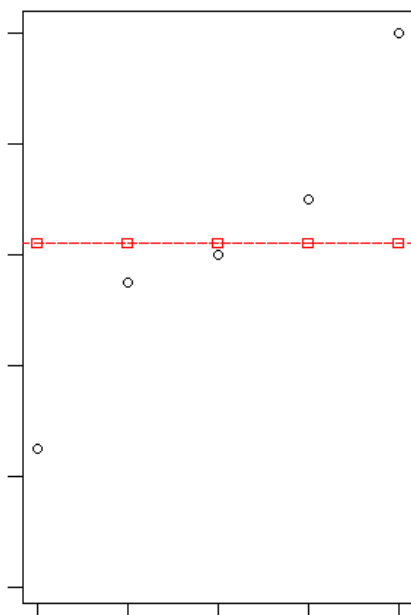
```
lines(x1_mean,type="o",pch=22,lty=2, col='red')
```

```
abline(h=mean(x1),lty=2,col='red')
```

```
x2_mean<-rep(mean(x2),5)
```

```
plot(x2,ylim=c(1,100))
```

```
lines(x2_mean,type="o",pch=22,lty=2, col='blue')
```

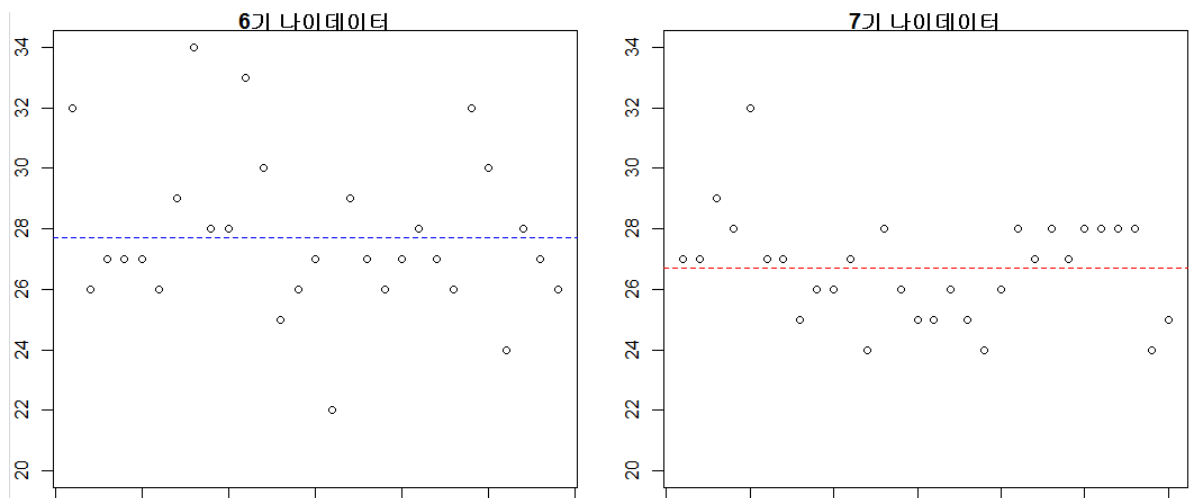


문제 215. 7기의 나이와 6기의 나이의 분산과 표준편차를 각각 구하고 시각화 하시오.

```
emp99<-read.csv("c:\\data\\emp99.csv",header = T)
graphics.off()
par(mfrow=c(1,2))
par(mar=c(1,1,1,5))

plot(emp99$age, ylim=c(20,34),main='6기 나이데이터')
abline(h=mean(emp99$age),lty=2,col='blue')      # h : 가로 선 , v : 세로 선

plot(emp2$age,ylim=c(20,34),main='7기 나이데이터')
abline(h=mean(emp2$age),lty=2,col='red')
```



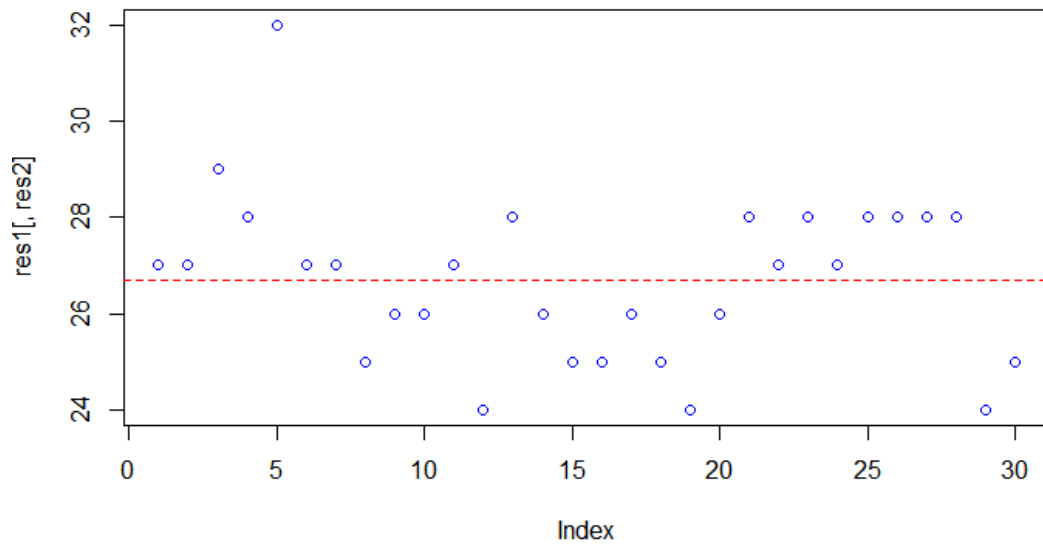
문제 216. 분산과 표준편차를 제목으로 두고 데이터의 분산을 시각화 하는 함수를 생성 하시오.

```
data_var_sd <- function(){

  graphics.off()
  res1<-get(readline(prompt='분산을 시각화할 테이블명을 입력하세요'))
  res2<-menu(colnames(res1),title='컬럼을 선택하세요')

  plot(res1[,res2],main=paste('분산= ',round(var(res1[,res2]),4),'표준편차= ',round(sd(res1[,res2])),4),col='blue')
  abline(h=mean(res1[,res2]),lty=2,col='red')
}
```

분산= 2.97586206896552 표준편차= 1.72506871427358



문제 217. emp 테이블의 월급의 범위를 구하시오.

```
> range(emp$sal)
[1] 800 5000
```

문제 218. 위에서 출력한 두 값의 차이를 구하시오.

```
range(emp$sal)[2]-range(emp$sal)[1]

> range(emp$sal)[2]-range(emp$sal)[1]
[1] 4200
```

데이터의 분포와 대칭정도

2018년 6월 23일 토요일 오후 5:54

■ 데이터의 분포와 대칭정도

- 왜도 : 좌우로 기울어짐의 정도 (skewness 함수)

왜도값 > 0 : 오른쪽 꼬리가 길다

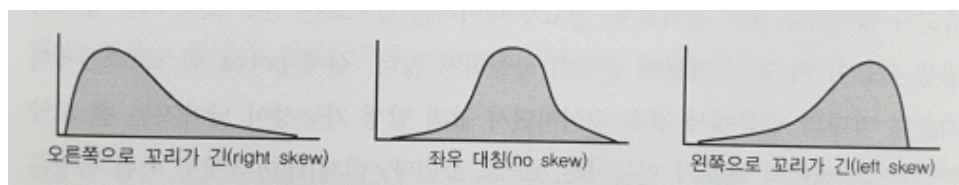
왜도값 < 0 : 왼쪽 꼬리가 길다

- 첨도 : 위아래 뾰족한 정도 (kurtosis 함수)

첨도값이 3에 가까울수록 정규분포에 해당

첨도값 < 3 : 완만한 곡선

첨도값 > 3 : 뾰족한 곡선



오른쪽 꼬리가 긴 경우 : 데이터들이 대부분 값이 작은 쪽에 모여있다.

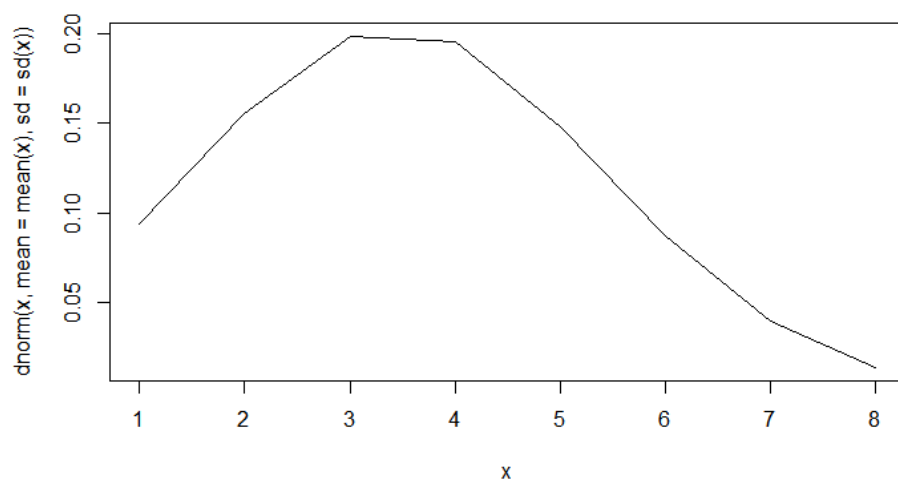
문제 219. 아래 데이터의 정규분포 곡선을 그리시오.

값	1	2	3	4	5	6	7	8
도수	4	6	4	4	3	2	1	1

```
x<-c(rep(c(1,2,3,4,5,6,7,8),c(4,6,4,4,3,2,1,1)))
```

```
plot(x,dnorm(x,mean=mean(x),sd=sd(x)),type='l',main='정규분포 그래프')
```

정규분포 그래프



문제 220. x데이터의 왜도값을 측정 하시오.

```
install.packages('fBasics')
library(fBasics)
skewness(x)
```

```
> skewness(x)
[1] 0.5805801
attr(,"method")
[1] "moment"
```

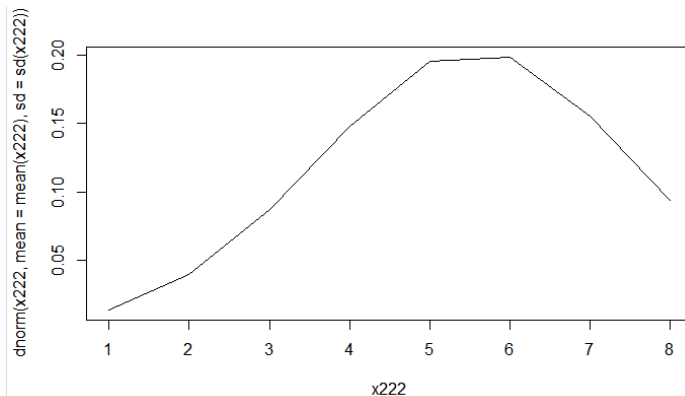
문제 221. 사원 테이블의 월급의 왜도 값을 출력 하시오,

```
> skewness(emp$sal)
[1] 0.9349611
attr(,"method")
[1] "moment"
```

문제 222. 아래의 데이터를 만들고 정규분포 곡성을 그리시오.

```
x222<-c(rep(c(1,2,3,4,5,6,7,8), c(1,1,2,3,4,4,6,4)))
```

```
x222<-c(rep(c(1,2,3,4,5,6,7,8), c(1,1,2,3,4,4,6,4)))
plot(x222,dnorm(x222,mean=mean(x222),sd=sd(x222)),type='l')
```



문제 223. x222의 왜도값을 출력 하시오.

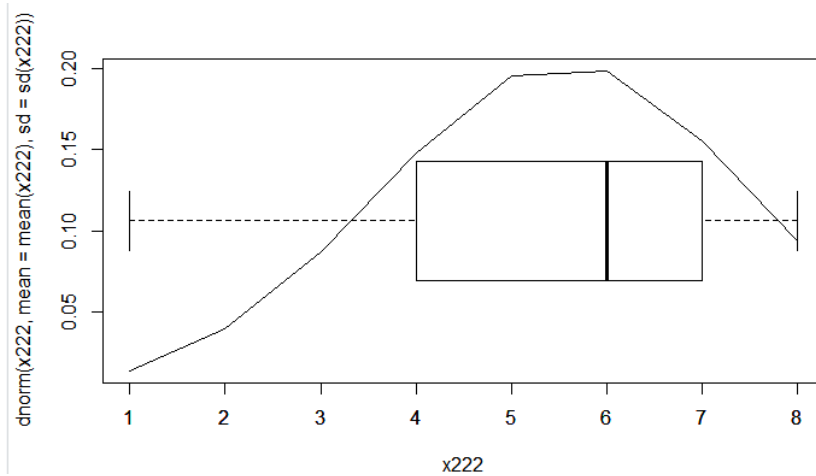
```
library(fBasics)
skewness(x222)
```

```
> skewness(x222)
[1] -0.5805801
attr(,"method")
[1] "moment"
```

문제 224. x222의 정규분포 그래프에 박스 그래프가 겹쳐서 나오게 시각화 하시오. (박스 그래프를 겹쳐서 출력 하시오)

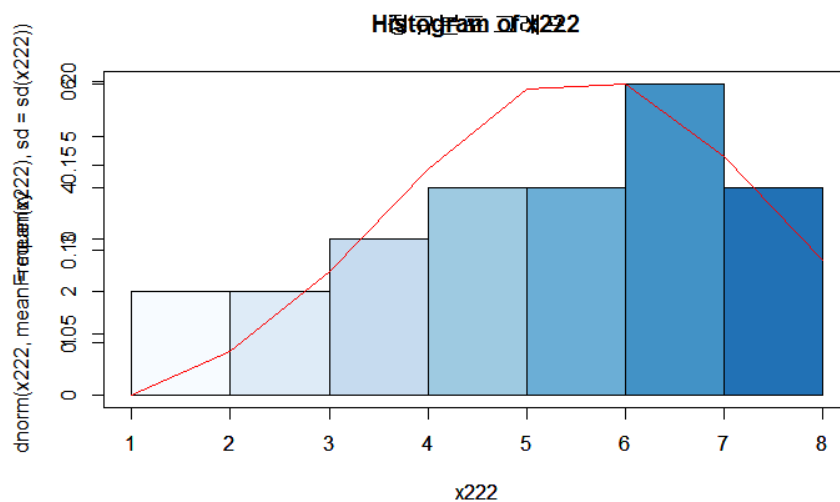
```
library(fBasics)
skewness(x222)
```

```
plot(x222,dnorm(x222,mean=mean(x222),sd=sd(x222)),type='l', main='정규분포
그래프')
par(new=T)
boxplot(x222,horizontal = T)
```



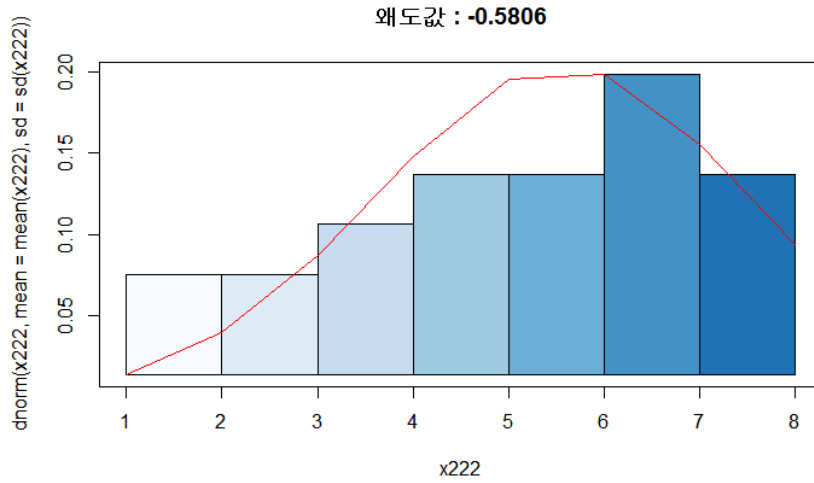
문제 225. x222의 정규분포 그래프에 히스토그램 그래프가 겹쳐서 나오게 시각화 하시오.

```
hist(x222,horizontal = T,col=blues9)
par(new=T)
plot(x222,dnorm(x222,mean=mean(x222),sd=sd(x222)),type='l', main='정규분포 그래
프', col='red')
```



문제 226. 위의 그래프 제목에 왜도값도 출력되게 하시오.

```
graphics.off()
hist(x222,horizontal = T,col=blues9, axes = F, ann=F)
par(new=T)
plot(x222,dnorm(x222,mean=mean(x222),sd=sd(x222)),type='l',
col='red',main=paste('왜도값 : ',round(skewness(x222),4)))
```



문제 227. x222의 첨도값을 확인 하시오.

```
library(fBasics)
kurtosis(x222)

> kurtosis(x222)
[1] -0.6541229
attr(,"method")
[1] "excess"
```

문제 228. 히스토그램 그래프와 정규분포 그래프와 왜도값을 출력하는 함수를 생성 하시오.

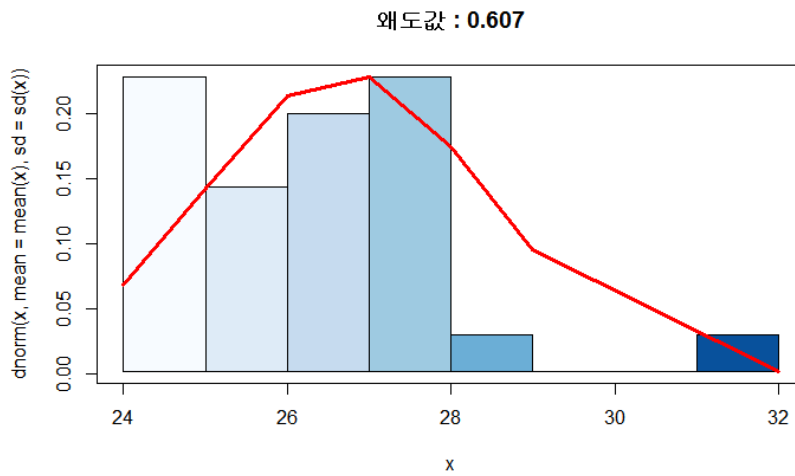
```
plot_hist<-function(){

  res1<-get(readline(prompt = '정규분포 곡선을 그릴 테이블명 입력 : '))
  res2<-menu(colnames(res1),title='컬럼명 입력 : ')

  x<-sort(res1[,res2])

  graphics.off()
  hist(x,col=blues9, axes = F, ann=F)
  par(new=T)
  plot(x,dnorm(x,mean=mean(x),sd=sd(x)),type='l', lwd=3, col='red',main=paste('왜도
값 : ',round(skewness(x),4)))

}
```



문제 229. 정규분포 그래프의 68%에 해당하는 데이터는 구하는데 사원 테이블의 월급이 얼마부터 얼마까지가 정규분포 68%에 해당하는지 알아내시오.

```
mean(emp$sal) # 2073.214
sd(emp$sal) #1182.503
```

```
2073-1182 = 891
2073+1182 = 3255
```

-----> 891~3255값이 68%에 속한다.

`pnorm(3255,2073,1182)` # 0~3255는 평균이 2073이고 표준편차가 1182일때 84%의 값에 해당한다.

0~3225값은 84%(68+16)

```
> mean(emp$sal) # 2073.214
[1] 2073.214
> sd(emp$sal) #1182.503
[1] 1182.503
> pnorm(3255,2073,1182)
[1] 0.8413447
```

문제 230. 우리반 나이의 정규분포 68%에 해당하는 나이가 어디서부터 어디까지 인지 출력 하시오.

- 1.우리반 나이의 평균 : `mean(emp2$age)` #26.7
- 2.우리반 나이의 표준편차 : `mean(emp2$age)` #1.73
3. 26.7-1.7 ~ 26.7+1.7 -----> 25 ~ 28.4 가 우리반 나이 68%에 해당한다.

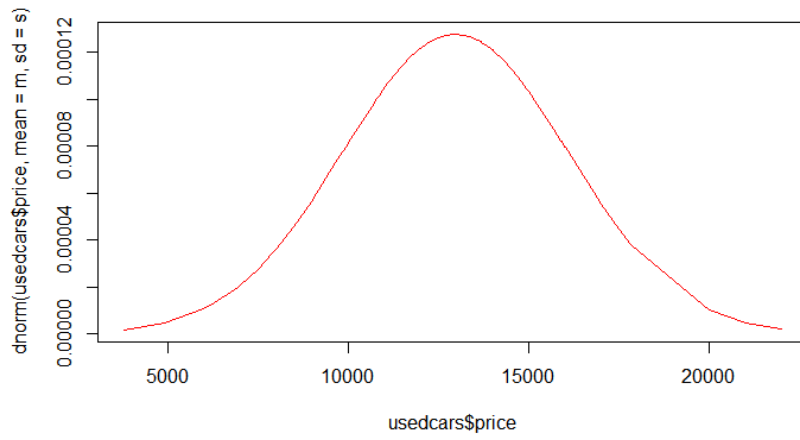
문제 231. 중고차 가격데이터를 R로 로드하고 중고차 가격 9,840달러 ~16,084 달러 사이의 가격이 전체 데이터 중 68%에 해당하는지 R로 확인해 보시오.

```
usedcars<-read.csv("c:\\data\\usedcars.csv",header = T)
m<-mean(usedcars$price) #12961.93
s<-sd(usedcars$price) # 3122.482
```


m-s #9839
m+s #16084

----> 9839~16084 값이 68%에 해당한다

```
plot(usedcars$price,dnorm(usedcars$price,mean=m,sd=s),type='l',col='red')
```



이원교차표

2018년 5월 25일 금요일 오전 9:59

■ 이원 교차표

이원 교차표는 머신러닝 학습 알고리즘의 정확도를 확인하고 제대로 데이터를 분류하는 머신러닝인지 확인하고자 할 때 사용하는 도구이다.

		model prediction	
		no default (0)	default (1)
actual loan status	no default (0)	TN	FP
	default (1)	FN	TP

#참 긍정(TP, true positive) : 관심범주를 정확히 분류 (정상을 정상으로 예측)

#참 부정(TN, true Negative) : 관심 범주가 아닌 것을 정확하게 분류 (폐암을 예측했는데 폐암이 맞음)

#거짓 긍정(FP, False Positive) : 관심 범주로 잘못 분류 (정상인줄 알았는데 폐암)

#거짓 부정(FN, False Negative) : 관심 범주가 아닌 것으로 잘못 분류 (폐암을 예측했는데 정상임)

유방암 조직검사		result1		약성	
wbcd[470:569, 2]		양성	Benign	Malignant	Row Total
Benign			61	10	61
			1.000	0.000	0.610
			0.968	0.000	
			0.610	0.000	
Malignant			2	37	39
			0.051	0.949	0.390
			0.032	1.000	
			0.020	0.370	
Column Total			63	37	100
			0.630	0.370	

100건중 2건만 틀렸다 ----> 98%의 정확도

Cross table (x,y) # x : 예측, y : 실제

문제 238. 직업(세로), 부서번호(가로), 직업별 부서번호별 인원수를 출력 하시오.

`tapply(emp$ename,list(emp$deptno,emp$job),length, default = 0)` # default : na값 0 출력

```
> tapply(emp$ename, list(emp$deptno, emp$job), length, default = 0)
      ANALYST CLERK  MANAGER  PRESIDENT  SALESMAN
10           0      1         1           1         0
20           2      2         1           0         0
30           0      1         1           0         4
```

문제 239. 이원교차표를 출력하는 crosstable 함수를 이용해서 위의 결과를 출력 하시오.

```
install.packages("gmodels")
library(gmodels)
```

```
CrossTable(x=emp$deptno, y=emp$job)
```

emp\$deptno	emp\$job ANALYST	CLERK	MANAGER	PRESIDENT	SALESMAN	Row Total
10	0	1	1	1	0	3
	0.429	0.024	0.198	2.881	0.857	0.214
	0.000	0.333	0.333	0.333	0.000	
	0.000	0.250	0.333	1.000	0.000	
	0.000	0.071	0.071	0.071	0.000	
20	2	2	1	0	0	5
	2.314	0.229	0.005	0.357	1.429	0.357
	0.400	0.400	0.200	0.000	0.000	
	1.000	0.500	0.333	0.000	0.000	
	0.143	0.143	0.071	0.000	0.000	
30	0	1	1	0	4	6
	0.857	0.298	0.063	0.429	3.048	0.429
	0.000	0.167	0.167	0.000	0.667	
	0.000	0.250	0.333	0.000	1.000	
	0.000	0.071	0.071	0.000	0.286	
Column Total	2	4	3	1	4	14
	0.143	0.286	0.214	0.071	0.286	

문제 240. 직업별로 월급의 차이가 존재하는지 이원 교차표로 확인 하시오.
월급 2500을 기준으로 직업별로 각각 월급이 2500 이상인 사원과 2500보다 작은 사원들이 어떻게 분포 되어있는지 확인하시오.

```
CrossTable (x축, 조건 )
```

```
CrossTable(emp$job, emp$sal >= 2500)
```

emp\$job	emp\$sal >= 2500 FALSE	TRUE	Row Total
ANALYST	0	2	2
	1.286	2.314	0.143
	0.000	1.000	
	0.000	0.400	
	0.000	0.143	
CLERK	4	0	4
	0.794	1.429	0.286
	1.000	0.000	
	0.444	0.000	
	0.286	0.000	
MANAGER	1	2	3
	0.447	0.805	0.214
	0.333	0.667	
	0.111	0.400	
	0.071	0.143	
PRESIDENT	0	1	1
	0.643	1.157	0.071
	0.000	1.000	
	0.000	0.200	
	0.000	0.071	
SALESMAN	4	0	4
	0.794	1.429	0.286
	1.000	0.000	
	0.444	0.000	
	0.286	0.000	
Column Total	9	5	14
	0.643	0.357	

문제 241. 중고차의 모델의 종류가 어떻게 되는지 출력 하시오.

```
> usedcars<-read.csv("c:\\data\\usedcars.csv",header = T)
> unique(usedcars$model)
[1] SEL SE SES
Levels: SE SEL SES
```

문제 242. 차 모델별로 보수적인 색을 가진 자동차의 비율을 비교해보기 위해 이원 교차표를 살펴 보시오.

```
CrossTable(usedcars$model,usedcars$color %in% c('Black','Gray','Silver','White'))
```

usedcars\$model	usedcars\$color %in% c("Black", "Gray", "Silver", "White")		
	FALSE	TRUE	Row Total
SE	27	51	78
	0.009	0.004	
	0.346	0.654	0.520
	0.529	0.515	
	0.180	0.340	
SEL	7	16	23
	0.086	0.044	
	0.304	0.696	0.153
	0.137	0.162	
	0.047	0.107	
SES	17	32	49
	0.007	0.004	
	0.347	0.653	0.327
	0.333	0.323	
	0.113	0.213	
Column Total	51	99	150
	0.340	0.660	

■ 목차

1. knn 이란 무엇인가?
2. knn 이 필요한 이유
3. knn 의 분류 실습1(유방암 데이터)

1. Knn이란 무엇인가?

" K nearest neighbor의 약자로 머신러닝의 **지도학습-분류**에 해당하는 알고리즘이다. "

새로 들어온 데이터가 기존 데이터의 그룹 중 어느 그룹에 속하는지 찾을 때 거리가 가장 가까운 데이터의 그룹을 자기 그룹으로 선택하는 아주 간단한 알고리즘이다.

(같은 거리 안에 여러 개의 그룹이 있으면 다수결에 의해서 가장 많은 그룹을 선택하는 분류 방법이다.)

머신러닝의 종류 3가지

- a. 지도학습 : 정답이 있는 경우
 - 분류 : **knn (수치형 데이터 사용)**
 - 예측
- b. 비지도 학습 : 정답이 없는 경우
- c. 강화학습 : 스스로 학습해서 데이터를 생성

Knn 알고리즘은 지도학습(분류)에 속한다.

*분류문제란 새로운 데이터가 들어왔을 때 이 데이터가 기존에 있던 데이터의 그룹 중에 어느 그룹에 속한건지를 분류 하는 것을 말한다.

■ knn이 필요한 이유

"유방암의 종양의 크기에 대한 데이터 (반지름, 둘레, 면적 등) 만 가지고 이 종양이 악성인지 양성인지를 예측할 수 있다면 환자에 대한 치료 스케줄에 큰 영향을 미칠수 있기 때문에 잘 분류하고 훈련된 모델이 필요하다 "

■ Knn의 여러 시점

1 knn을 사회적인 관계로 보면?

사회적으로 대략적으로 비슷한 사람들끼리 모이는 성질이 있다. 비슷한 취향의 사람들끼리 모여서 동호회를 만들고 비슷한 부류의 계층의 사람들끼리 친분을 맺기도 한다.

2.knn을 공간적 관례로 관찰해 보면?

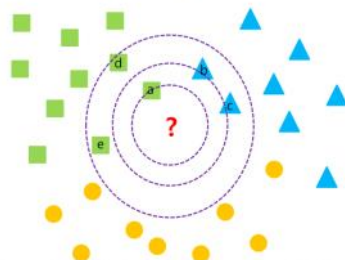
가구점이 모여있는 상가지역이 따로 형성 되어있거나 가전제품이 밀집되어있는 지역이 존재한다.

이러한 특성을 가진 데이터를 겨냥해서 만들어진 알고리즘이 knn이다.

knn 은 그냥 새로 들어온 ?가 ■에 더 가까우니
■로 분류하겠다는 알고리즘입니다



k가 5이면 ?의 5번째로 가까운 이웃을 a,b,c,d,e로
보고 신규 데이터를 분류 하는데



여기서는 ■가 3, ▲가 2이므로 ■로 구분 합니다

K개의 가장 가까운 이웃이 누구냐 ??? 항상 홀수여야 됨 (근접한 이웃의 개수로 분류하기 때문에)

K의 값을 어떻게 선정하느냐가 중요하다. -----> 적절한 k값을 찾기 위한 실험을 해야함

2. 최적의 k값을 구하는 방법

가장 좋은 K값을 구하기 위해서 테스트를 수행하는 과정

```
library(caret)
library(e1071)
```

```
# Setting up train controls k값을 알아내기 위한 테스트 반복횟수 등을 지정
repeats = 3
numbers = 10
tunel = 10
```

```
set.seed(1234)
```

```
#최적의 k값을 알기 위해 test를 어떻게 반복 시킬것 인지 반복 방법을 정하는 문법
```

```
x = trainControl(method = "repeatedcv",
  number = numbers,
  repeats = repeats,
  classProbs = TRUE,
  summaryFunction = twoClassSummary)
```

```

model1 <- train(diagnosis~., data = wbcd_train, method = "knn", #라벨, 훈련데이터, 훈련방식
  preProcess = c("center","scale"),          # 정규화 하겠다
  trControl = x,
  metric = "ROC",                             # 그래프에서 확인
  tuneLength = tune1)                        #10 그래프에서 확인

```

```
# Summary of model
```

```
model1
```

```
plot(model1)
```

8. knn 모델로 훈련시켜서 모델을 만들고 바로 그 모델에 test 데이터를 넣어서 예측 결과(양성,악성)를 추출한다.

```
install.packages("class")
```

```
library(class)
```

```
wbcd_train <- wbcd_train[-1]
```

```
wbcd_test <- wbcd_test[-1]
```

```
result1 <- knn(train=wbcd_train, test=wbcd_test,
  cl=wbcd_train_label, k=16)
```

```
result1 <--- 테스트 데이터를 가지고 예측한 결과
              53명의 환자들의 예측 결과
```

9. 예측과 실제 데이터를 먼저 본다

```
library(data.table)
```

```
data.table("예측"=result1, "실제"=wbcd_test_label)
```

10. 몇개 맞췄고 몇개 못맞췄는지 확인해본다

```
table(ifelse(wbcd_test_label==result1,"o","x"))
```

```
o x
```

```
55 2
```

11. 이 모델의 정확도가 몇 % 인지 확인하시오 !

```
prop.table(table(ifelse(wbcd_test_label==result1,"o","x"))) )
```

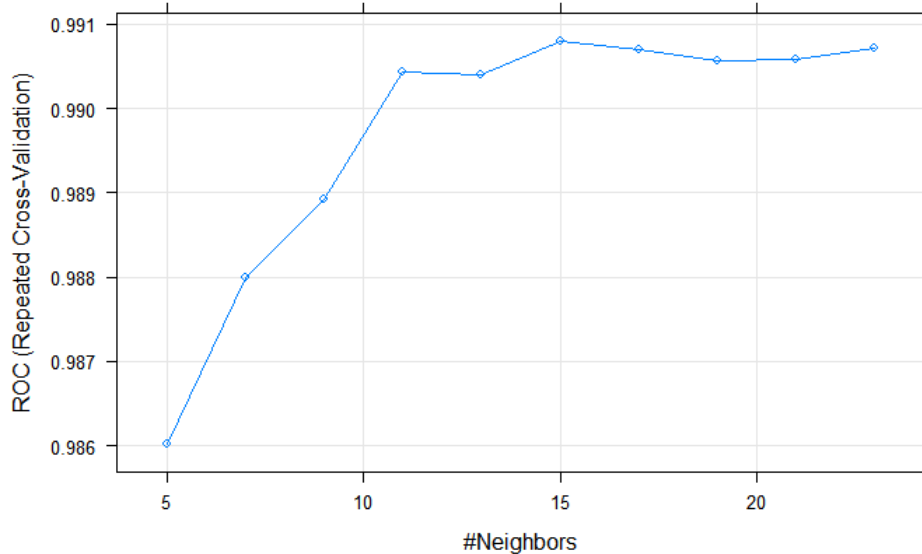
```
o      x
0.96491228 0.03508772
```

```
>
```

12. 이원 교차표를 그려서 TP,TN,FP,FN 을 확인해보시오

```
library(gmodels)
```

```
CrossTable( x= wbcd_test_label, y= result1,
  prop.chisq=FALSE)
```



오버피팅 : 훈련데이터에만 최적화된 상태 , 테스트 데이터는 잘 못함

언더피팅 : k를 크게 주면 훈련데이터 조차분류를 잘 못함

4. knn의 분류 실습1 (유방암 데이터)

1. 유방암 데이터를 내려받고 wbcd 변수에 로드 하시오.

```
> wbcd <- read.csv("c:\\data\\wisc_bc_data.csv", header = T, stringsAsFactors = F)
> nrow(wbcd)
[1] 569
> str(wbcd)
'data.frame': 569 obs. of 32 variables:
 $ id      : int  87139402 8910251 905520 868871 9012568 906539 925291 87880 862989 89827 .
 $ diagnosis 정답 : chr  "B" "B" "B" "B" ...
 $ radius_mean : num  12.3 10.6 11 11.3 15.2 ...
 $ texture_mean : num  12.4 18.9 16.8 13.4 13.2 ...
 $ perimeter_mean : num  78.8 69.3 70.9 73 97.7 ...
 $ area_mean : num  464 346 373 385 712 ...
 $ smoothness_mean : num  0.1028 0.0969 0.1077 0.1164 0.0796 ...
 $ compactness_mean : num  0.0698 0.1147 0.078 0.1136 0.0693 ...
 $ concavity_mean : num  0.0399 0.0639 0.0305 0.0464 0.0339 ...
 $ points_mean : num  0.037 0.0264 0.0248 0.048 0.0266 ...
 $ symmetry_mean : num  0.196 0.192 0.171 0.177 0.172 ...
 $ dimension_mean : num  0.0595 0.0649 0.0634 0.0607 0.0554 ...
 $ radius_se : num  0.236 0.451 0.197 0.338 0.178 ...
 $ texture_se : num  0.666 1.197 1.387 1.343 0.412 ...
 $ perimeter_se : num  1.67 3.43 1.34 1.85 1.34 ...
 $ area_se : num  17.4 27.1 13.5 26.3 17.7 ...
 $ smoothness_se : num  0.00805 0.00747 0.00516 0.01127 0.00501 ...
 $ compactness_se : num  0.0118 0.03581 0.00936 0.03498 0.01485 ...
 $ concavity_se : num  0.0168 0.0335 0.0106 0.0219 0.0155 ...
 $ points_se : num  0.01241 0.01365 0.00748 0.01965 0.00915 ...
 $ symmetry_se : num  0.0192 0.035 0.0172 0.0158 0.0165 ...
 $ dimension_se : num  0.00225 0.00332 0.0022 0.00344 0.00177 ...
 $ radius_worst : num  13.5 11.9 12.4 11.9 16.2 ...
 $ texture_worst : num  15.6 22.9 26.4 15.8 15.7 ...
 $ perimeter_worst : num  87 78.3 79.9 76.5 104.5 ...
 $ area_worst : num  549 425 471 434 819 ...
 $ smoothness_worst : num  0.139 0.121 0.137 0.137 0.113 ...
 $ compactness_worst : num  0.127 0.252 0.148 0.182 0.174 ...
 $ concavity_worst : num  0.1242 0.1916 0.1067 0.0867 0.1362 ...
 $ points_worst : num  0.0939 0.0793 0.0743 0.0861 0.0818 ...
 $ symmetry_worst : num  0.283 0.294 0.3 0.21 0.249 ...
 $ dimension_worst : num  0.0677 0.0759 0.0788 0.0678 0.0677 ...
```

컬럼 설명 : 127p

문제 243. Wbcd 536건의 유방암 데이터를 훈련 데이터와 테스트 데이터로 나누시오.

훈련 데이터 : 머신러닝 모델을 훈련시킨 데이터
테스트 데이터 : 훈련된 모델을 테스트 해보는 데이터

훈련 데이터를 80%, 테스트 데이터를 20%

wbcd_train (훈련데이터)
wbcd_test (테스트 데이터)

```
wbcd_train <- wbcd[1:floor(nrow(wbcd)*0.8),]  
nrow(wbcd_train)  
wbcd_test <- wbcd[ceiling(nrow(wbcd)*0.8):nrow(wbcd),]  
nrow(wbcd_test)  
  
> wbcd_train <- wbcd[1:floor(nrow(wbcd)*0.8),]  
> nrow(wbcd_train)  
[1] 455  
> wbcd_test <- wbcd[ceiling(nrow(wbcd)*0.8):nrow(wbcd),]  
> nrow(wbcd_test)  
[1] 114
```

문제 244.	StringAsFactors = F 옵션을 사용해서 데이터를 로드한 wbcd 변수와 이 옵션을 사용하지 않고 로드한 변수의 차이를 str로 비교 하시오.
---------	---

```
wbcd <- read.csv("c:\data\wisc_bc_data.csv", header = T, stringsAsFactors = F)  
wbcd2 <- read.csv("c:\data\wisc_bc_data.csv", header = T, stringsAsFactors = T)
```

```
str(wbcd)  
str(wbcd2)
```

```
> str(wbcd)  
'data.frame': 569 obs. of 32 variables:  
 $ id : int 87139402 8910251 905520 868871 9012568 906539 925291 87880 862989 89827 ...  
 $ diagnosis : chr "B" "B" "B" "B" ...  
 $ radius_mean : num 12.3 10.6 11 11.3 15.2 ...  
 $ texture_mean : num 12.4 18.9 16.8 13.4 13.2 ...  
 $ perimeter_mean : num 78.8 69.3 70.9 73 97.7 ...  
 $ area_mean : num 464 346 373 385 712 ...  
 $ compactness mean : num 0.1028 0.0969 0.1077 0.1164 0.0796 ...  
  
> str(wbcd2)  
'data.frame': 569 obs. of 32 variables:  
 $ id : int 87139402 8910251 905520 868871 9012568 906539 925291 87880 862989 89827 ...  
 $ diagnosis : Factor w/ 2 levels "B","M": 1 1 1 1 1 1 1 2 1 1 ...  
 $ radius_mean : num 12.3 10.6 11 11.3 15.2 ...  
 $ texture_mean : num 12.4 18.9 16.8 13.4 13.2 ...  
 $ perimeter_mean : num 78.8 69.3 70.9 73 97.7 ...  
 $ area_mean : num 464 346 373 385 712 ...  
 $ smoothness_mean : num 0.1028 0.0969 0.1077 0.1164 0.0796 ...  
 $ compactness mean : num 0.0698 0.1147 0.078 0.1136 0.0693 ...
```

문제 245.	유방암 데이터의 양성(B)과 악성(M)의 건수가 각각 어떻게 되는지 확인 하시오.
---------	---

```
table(wbcd$diagnosis)
```

```
> table(wbcd$diagnosis)  
  
 B    M  
357 212
```

문제 246. 유방암 데이터의 양성(B)과 악성(M)의 비율이 어떻게 되는지 확인 하시오.

```
prop.table(table(wbcd$diagnosis))

> prop.table(table(wbcd$diagnosis))

      B      M 
0.6274165 0.3725835
```

문제 247. 유방암 데이터 wbcd2의 diagnosis 컬럼을 팩터로 변환하는데 labels 옵션을 줘서 b는 Benign이고 m은 Maligant로 지정 하시오.

```
wbcd$diagnosis <- factor(wbcd$diagnosis,levels = c("B","M"),labels=c("Benign","Maliganant"))
prop.table(table(wbcd$diagnosis))*100

> prop.table(table(wbcd$diagnosis))*100

  Benign Maliganant 
62.74165  37.25835
```

문제 248. emp 데이터 프레임에서 랜덤으로 5개 데이터를 가져오시오.

```
emp[sample(5),] # 실행할 때마다 다른 값을 가져옴

> emp[sample(5),]
  empno ename      job mgr  hiredate  sal comm deptno
5  7654 MARTIN SALESMAN 7698 1981-09-10 1250 1400     30
2  7698 BLAKE  MANAGER 7839 1981-05-01 2850    NA     30
1  7839 KING PRESIDENT   NA 1981-11-17 5000    NA     10
4  7566 JONES  MANAGER 7839 1981-04-01 2975    NA     20
3  7782 CLARK  MANAGER 7839 1981-05-09 2450    NA     10
> emp[sample(5),]
  empno ename      job mgr  hiredate  sal comm deptno
2  7698 BLAKE  MANAGER 7839 1981-05-01 2850    NA     30
3  7782 CLARK  MANAGER 7839 1981-05-09 2450    NA     10
4  7566 JONES  MANAGER 7839 1981-04-01 2975    NA     20
1  7839 KING PRESIDENT   NA 1981-11-17 5000    NA     10
5  7654 MARTIN SALESMAN 7698 1981-09-10 1250 1400     30
```

문제 249. Emp 데이터 14개를 random으로 shuffle 해서 변수에 넣으시오.

```
emp_shuffle <- emp[sample(14),]
```

문제 250. Wbcd 데이터를 shuffle 해서 wbcd_shuffle 변수에 넣으시오.

```
wbcd_shuffle <- wbcd[sample(nrow(wbcd)),]
```

문제 251. Wbcd_shuffle 변수의 데이터를 9:1 비율로 wbcd_shuffle_train (훈련데이터), wbcd_shuffle_test (테스트 데이터) 변수에 각각 넣으시오.

```
wbcd_shuffle_train <- wbcd[1:floor(nrow(wbcd_shuffle)*0.9),]
wbcd_shuffle_test <- wbcd[ceiling(nrow(wbcd_shuffle)*0.9):nrow(wbcd_shuffle),]
```

문제 252. Wbcd_train과 wbcd_test 데이터에서 id 컬럼을 제외 시키시오.

```
wbcd_train<-wbcd_train[,-1]
wbcd_test<-wbcd_test[,-1]
```

문제 253. Wbcd_train과 wbcd_test 데이터를 정규화 시키시오.

❖ **최소-최대 정규화**

$$X_{new} = \frac{X - \min(X)}{\max(X) - \min(X)}$$

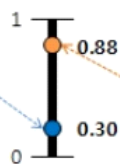
❖ **Z점수 표준화**

$$X_{new} = \frac{X - \mu}{\sigma} = \frac{X - \text{Mean}(X)}{\text{StdDev}(X)}$$

한국 성인 남성 키



[0-1] 변환



부시맨 성인 남성 키



```
wbcd_train_n <- as.data.frame(lapply(wbcd_train[2:31],normalize))
wbcd_test_n <- as.data.frame(lapply(wbcd_test[2:31],normalize))
```

모두 0~1 사이 값을 가진다.

```
> summary(wbcd_train_n)
  radius_mean  texture_mean  perimeter_mean  area_mean  smoothness_mean
Min.   :0.0000  Min.   :0.0000  Min.   :0.0000  Min.   :0.0000  Min.   :0.0000
1st Qu.:0.1941  1st Qu.:0.2185  1st Qu.:0.1924  1st Qu.:0.1033  1st Qu.:0.2071
Median :0.2782  Median :0.3064  Median :0.2730  Median :0.1610  Median :0.3062
Mean   :0.3128  Mean   :0.3198  Mean   :0.3116  Mean   :0.2040  Mean   :0.3118
3rd Qu.:0.3943  3rd Qu.:0.4048  3rd Qu.:0.3969  3rd Qu.:0.2579  3rd Qu.:0.4009
Max.   :1.0000  Max.   :1.0000  Max.   :1.0000  Max.   :1.0000  Max.   :1.0000
compactness_mean  concavity_mean  points_mean  symmetry_mean  dimension_mean
Min.   :0.0000  Min.   :0.00000  Min.   :0.00000  Min.   :0.0000  Min.   :0.0000
1st Qu.:0.1433  1st Qu.:0.06687  1st Qu.:0.09883  1st Qu.:0.3301  1st Qu.:0.1592
Median :0.2315  Median :0.14056  Median :0.16571  Median :0.4302  Median :0.2409
Mean   :0.2749  Mean   :0.20385  Mean   :0.23932  Mean   :0.4398  Mean   :0.2649
3rd Qu.:0.3692  3rd Qu.:0.29674  3rd Qu.:0.36827  3rd Qu.:0.5270  3rd Qu.:0.3321
Max.   :1.0000  Max.   :1.00000  Max.   :1.00000  Max.   :1.0000  Max.   :1.0000
  radius_se  texture_se  perimeter_se  area_se  smoothness_se
Min.   :0.00000  Min.   :0.0000  Min.   :0.00000  Min.   :0.00000  Min.   :0.0000
1st Qu.:0.04375  1st Qu.:0.1037  1st Qu.:0.04015  1st Qu.:0.02131  1st Qu.:0.1149
Median :0.07446  Median :0.1620  Median :0.07171  Median :0.03417  Median :0.1603
Mean   :0.10429  Mean   :0.1903  Mean   :0.09799  Mean   :0.06306  Mean   :0.1818
3rd Qu.:0.13175  3rd Qu.:0.2465  3rd Qu.:0.12413  3rd Qu.:0.07329  3rd Qu.:0.2227
Max.   :1.00000  Max.   :1.0000  Max.   :1.00000  Max.   :1.00000  Max.   :1.0000
compactness_se  concavity_se  points_se  symmetry_se  dimension_se
Min.   :0.00000  Min.   :0.00000  Min.   :0.0000  Min.   :0.0000  Min.   :0.00000
1st Qu.:0.07386  1st Qu.:0.03797  1st Qu.:0.1449  1st Qu.:0.1329  1st Qu.:0.04528
```

문제 254. 훈련 데이터를 데이터와 라벨로 나누고 테스트 데이터를 데이터와 라벨로 나누시오.

```
wbcd_train<- wbcd_train_n
wbcd_train_label<-wbcd_train[1]
```

■ knn 유방암 데이터 학습 총정리

1. 데이터를 로드 한다.

```
wbcd <- read.csv("C:\\data\\wisc_bc_data.csv", header=T, stringsAsFactors=FALSE)
```

2. diagnosis 를 factor 로 변환한다

```
wbcd$diagnosis <- factor(wbcd$diagnosis,
  levels =c("B","M"),
  labels = c("Benign","Maliganant"))
```

3. 데이터를 shuffle 시킨다.

```
wbcd_shuffle <- wbcd[sample(nrow(wbcd)), ]
```

4. 데이터에서 id 를 제외 시킨다 (숫자 값만 남김)

```
wbcd2 <- wbcd_shuffle[-1]
```

5. 데이터를 정규화 한다.

```
normalize <- function(x) {
  return ( (x-min(x)) / (max(x) - min(x)) )
}
wbcd_n <- as.data.frame(lapply(wbcd2[2:31],normalize))
```

6. train 데이터와 test 데이터로 9 대 1로 나눈다

```
train_num<-round(0.9*nrow(wbcd_n),0)
wbcd_train<-wbcd_n[1:train_num,]
wbcd_test<-wbcd_n[(train_num+1):nrow(wbcd_n),]
```

7. train 데이터를 데이터와 라벨로 나누고 test 데이터를 데이터와 라벨로 나누시오 ~

```
wbcd_train_label <- wbcd2[1:train_num,1]
wbcd_test_label <- wbcd2[(train_num+1):nrow(wbcd_n),1]
```

8. knn 모델로 훈련시켜서 모델을 만들고 바로 그 모델에 test 데이터를 넣어서 정확도를 확인한다

```
result1 <- knn(train=wbcd_train, test=wbcd_test, cl=wbcd_train_label, k=21)
```

```
result1
```

9. 예측과 실제 데이터를 먼저 본다.

```
install.packages("data.table")
library(data.table)
```

```
data.table("예측"=result1, "실제"=wbcd_test_label)
```

```
> data.table("예측"=result1, "실제"=wbcd_test_label)
   예측   실제
1:  Benign  Benign
2: Malignant Malignant
3:  Benign  Benign
4:  Benign  Benign
5:  Benign  Benign
6: Malignant Malignant
7:  Benign  Benign
8:  Benign  Benign
9: Malignant Malignant
10: Benign  Benign
11: Malignant Malignant
12: Benign  Benign
```

10. 몇 개 맞췄고 몇 개 못맞췄는지 확인해 본다.

```
table(ifelse(wbcd_test_label==result1,"o","x"))
```

```
> table(ifelse(wbcd_test_label==result1,"o","x"))
```

```
o  x
56 1
```

```
prop.table(table(ifelse(wbcd_test_label==result1,"o","x")))
```

```
> prop.table(table(ifelse(wbcd_test_label==result1,"o","x")))
```

```
      o      x
0.98245614 0.01754386
```

11. 이원 교차표를 그려서 TP,TN,FP,FN 을 확인해 보시오.

```
library(gmodels)
CrossTable(x=wbcd_test_label,y=result1,prop.chisq = F)
```

wbc_d_test_label	result1		Row Total
	Benign	Malignant	
Benign 나쁜 오진	33	0	33
	1.000	0.000	0.579
	0.971	0.000	
	0.579	0.000	
Malignant	1	23	24
	0.042	0.958	0.421
	0.029	1.000	
	0.018	0.404	
Column Total	34	23	57
	0.596	0.404	

문제 255. FN이 0으로 나오게끔 모델의 성능을 높이시오.

- 방법 1. 서플 다시 수행
2. K값을 조정

```
result1 <- knn(train=wbcd_train, test=wbcd_test, cl=wbcd_train_label, k=19)
```

wbc_d_test_label	result1		Row Total
	Benign	Malignant	
Benign	33	0	33
	1.000	0.000	0.579
	1.000	0.000	
	0.579	0.000	
Malignant	0	24	24
	0.000	1.000	0.421
	0.000	1.000	
	0.000	0.421	
Column Total	33	24	57
	0.579	0.421	

문제 256. 한국인 신체 데이터를 내려받아 R로 로드하시오.

```
kbody<-read.csv("c:\wwwdata\wwwkbody2.csv",header = T)
```

```
library(class)
```

```
library(gmodels)
```

```
head(kbody,4)
```

```
head(kbody_la,4)
```

```
normalize <- function(x) {
  return ( (x-min(x)) / (max(x) - min(x)) )
}
```

```
length(kbody)
```

```
nrow(kbody) #14016
```

```
kbody<-na.omit(kbody) #12894
```

```
kbody$성별<-ifelse(kbody$성별=='남',1,0)
```

```
kbody<-kbody[sample(nrow(kbody)),] #순서 랜덤
```

```
kbody_la<-kbody[,19:20]
```

```
kbody_la_train <- kbody_la[1:floor(nrow(kbody)*0.8),1]
```

```
kbody_la_test <- kbody_la[ceiling(nrow(kbody)*0.8):nrow(kbody),1]
```

```
kbody<-as.data.frame(lapply(kbody[,1:18],normalize))
```

```
kbody_train <- kbody[1:floor(nrow(kbody)*0.8),]
```

```
kbody_test <- kbody[ceiling(nrow(kbody)*0.8):nrow(kbody),]
```

```
result2<- knn(test=kbody_test, train = kbody_train, cl=kbody_la_train, k = 20)
```

```
data.table("예측"=result2,"실제"=kbody_la_test)
```

```
table(ifelse(kbody_la_test==result2,"o","x"))
```

```
prop.table(table(ifelse(wbcd_test_label==result1,"o","x")))
```

```
prop.table(table(result2))*100
```

```
CrossTable(x=kbody_la_test,y=result2,prop.chisq = F)
```

kbody_la_test	result2 경계	복비	표준	Row Total
	0 0.000 0.000 0.000	1 1.000 0.002 0.000	0 0.000 0.000 0.000	1 0.000
경계	2 0.019 0.167 0.001	36 0.340 0.077 0.014	68 0.642 0.032 0.026	106 0.041
복비	8 0.018 0.667 0.003	408 0.903 0.872 0.158	36 0.080 0.017 0.014	452 0.175
표준	2 0.001 0.167 0.001	23 0.011 0.049 0.009	1995 0.988 0.950 0.774	2020 0.783
Column Total	12 0.005	468 0.181	2099 0.814	2579

문제 257. set.seed가 어떤 함수인지 확인해보자.

컴퓨터 프로그램에서 무작위와 관련된 모든 알고리즘은 사실 무작위가 아니라 시작 숫자를 정해 주면 그 다음에는 정해진 알고리즘에 의해 마치 난수처럼 보이는 수열을 생성한다. 다만 출력되는 숫자들 간의 상관관계가 없어 보일 뿐이다.

또한 같은 알고리즘을 여러번 실행하더라도 다른 숫자가 나오도록 시작 숫자는 현재 시간 등을 사용해서 매번 바꿔준다. 이런 시작 숫자를 시드(seed)라고 한다.

따라서 시드를 사람이 수동으로 설정한다면 그 다음에 만들어지는 난수들은 예측할 수 있다.

R에서 시드 설정하기

```
set.seed(0)

sample(5, replace=TRUE)

sample(10, replace=TRUE)

sample(10, replace=TRUE)

# 이제 시드를 0으로 재설정하고 다시 난수를 발생시켜 본다.

set.seed(0)

sample(5, replace=TRUE)

sample(10, replace=TRUE)

sample(10, replace=TRUE)

# 위와 같이 같은 숫자가 샘플링 되는 것을 확인할 수 있다.

set.seed(1)

sample(5, replace=TRUE)

sample(10, replace=TRUE)

sample(10, replace=TRUE)
```

시드 값을 1로 변경하면 다르게 샘플링된다.

```
> set.seed(1)
> sample(5,replace = T)
[1] 2 2 3 5 2
> sample(10,replace = T)
[1] 9 10 7 7 1 3 2 7 4 8
> sample(10,replace = T)
[1] 5 8 10 4 8 10 3 7 2 3
> set.seed(1)
> sample(5,replace = T)
[1] 2 2 3 5 2
> sample(10,replace = T)
[1] 9 10 7 7 1 3 2 7 4 8
> sample(10,replace = T)
[1] 5 8 10 4 8 10 3 7 2 3
```

문제 258.	위의 스크립트를 묶어서 함수로 만드는데 데이터를 물어보게 하고 라벨을 물어보게 해서 가장 적절한 k 값이 무엇인지 출력하시오.
----------------	--

```
paste('최적의 k값 : ',model1$bestTune[,1])

> paste('최적의 k값 : ',model1$bestTune[,1])
[1] "최적의 k값 : 15"
```

문제 260.	Knn 머신러닝 알고리즘을 사용하는 코드가 자동화 되게끔 아래와 같은 함수를 생성 하시오. ex. >knn_func()
----------------	---

분석할 csv 파일명을 입력 하세요 ~ wisc_bc_data.csv
라벨이 될 컬럼 이름을 입력하세요 ~ diagnosis
K값을 입력 하세요~ 15

결과 : 이원 교차표 출력

```
knn_fun<-function(){

  setwd("c:\\data")

  x <- readline("분석할 csv 파일명을 입력하세요~ ")
  y <-readline("라벨의 컬럼의 이름을 입력하세요~ ")
  #입력값받는 함수
  k_n<-readline("k값을 입력하시오~ ")

  wbcd <- na.omit(read.csv(x, stringsAsFactors=FALSE))

  # set.seed(26)
  # wbcd <- wbcd[sample(nrow(wbcd)), ]

  normalize<-function(x) {
    return( (x-min(x))/ ( max(x)-min(x)))
  }

  wbcd <- wbcd[-1]
  ncol1 <- which(colnames(wbcd)==y)

  wbcd_n <- as.data.frame(lapply(wbcd[, -ncol1], normalize) )

  mm<-round(nrow(wbcd_n)*9/10)

  wbcd_train <- wbcd_n[1:mm, ]
  wbcd_test <- wbcd_n[(mm+1):nrow(wbcd_n), ]

  wbcd_train_label <- wbcd[1:mm,y]
  wbcd_test_label <- wbcd[(mm+1):nrow(wbcd_n),y]

  library(class)

  result1 <- knn(train=wbcd_train, test=wbcd_test,
                 cl= wbcd_train_label, k = k_n )

  # prop.table( table(ifelse(wbcd[(mm+1):nrow(wbcd_n),y]==result1,"o","x" )))

  library(gmodels)
  CrossTable( x= wbcd_test_label, y= result1,prop.chisq=FALSE)

}
```

knn_fun()

문제 261.	위의 knn_func() 함수를 수정하는데 knn_func()를 치고 엔터 누르면 윈도우 탐색창이 열리면서 파일을 선택 할 수 있도록 수정 하시오.
----------------	--

```
knn_fun<-function(){

  x<-read.csv(file.choose(),stringsAsFactors=FALSE)
  y <-readline("라벨의 컬럼의 이름을 입력하세요~ ")
  k_n<-readline("k값을 입력하시오~ ")

  wbcd <- na.omit(x)
  set.seed(26)
  wbcd <- wbcd[sample(nrow(wbcd)), ]

  normalize<-function(x) {
    return( (x-min(x))/ ( max(x)-min(x)))
  }

  wbcd <- wbcd[-1]
  ncol1 <- which(colnames(wbcd)==y)

  wbcd_n <- as.data.frame(lapply(wbcd[,-ncol1], normalize) )

  mm<-round(nrow(wbcd_n)*9/10)
  wbcd_train <- wbcd_n[1:mm, ]
  wbcd_test  <- wbcd_n[(mm+1):nrow(wbcd_n), ]

  wbcd_train_label <- wbcd[1:mm,y]
  wbcd_test_label  <- wbcd[(mm+1):nrow(wbcd_n),y]

  library(class)

  result1 <- knn(train=wbcd_train, test=wbcd_test,
                 cl= wbcd_train_label, k = k_n )

  library(gmodels)
  CrossTable( x= wbcd_test_label, y= result1,prop.chisq=FALSE)

}

knn_fun()
```

wbcd_test_label	result1		Row Total
	B	M	
B	41	0	41
	1.000	0.000	0.719
	1.000	0.000	
	0.719	0.000	
M	0	16	16
	0.000	1.000	0.281
	0.000	1.000	
	0.000	0.281	
Column Total		41	16
		0.719	0.281
			57

문제 262. Knn_fun() 함수의 라벨을 선택하는 부분을 컬럼번호로 입력하게끔 코드를 수정 하시오.

```
knn_fun<-function(){

  x<-read.csv(file.choose(),stringsAsFactors=FALSE)
  y<-menu(colnames(x),title = '라벨의 컬럼 이름을 선택 하시오.')
  y<-colnames(x[2])

  #입력값받는 함수
  k_n<-readline("k값을 입력하시오~ ")

  wbcd <- na.omit(x)
  set.seed(26)
  wbcd <- wbcd[sample(nrow(wbcd)), ]

  normalize<-function(x) {
    return( (x-min(x))/( max(x)-min(x)))
  }

  wbcd <- wbcd[-1]
  ncol1 <- which(colnames(wbcd)==y)

  wbcd_n <- as.data.frame(lapply(wbcd[, -ncol1], normalize) )

  mm<-round(nrow(wbcd_n)*9/10)
  wbcd_train <- wbcd_n[1:mm, ]
  wbcd_test  <- wbcd_n[(mm+1):nrow(wbcd_n), ]

  wbcd_train_label <- wbcd[1:mm,y]
  wbcd_test_label  <- wbcd[(mm+1):nrow(wbcd_n),y]

  library(class)

  result1 <- knn(train=wbcd_train, test=wbcd_test,
                 cl= wbcd_train_label, k = k_n )

  library(gmodels)
```

```
CrossTable( x= wbcd_test_label, y= result1,prop.chisq=FALSE)
}
knn_fun()
```

문제 263. Knn_fun() 에서 이원 교차표까지 출력되게 하시오.

```
knn_fun<-function(){

  x<-read.csv(file.choose(),stringsAsFactors=FALSE)
  y<-menu(colnames(x),title = '라벨의 컬럼 이름을 선택 하시오.')
  y<-colnames(x[2])

  #입력값받는 함수
  #k_n<-readline("k값을 입력하시오~ ")

  wbcd <- na.omit(x)
  set.seed(26)
  wbcd <- wbcd[sample(nrow(wbcd)), ]

  normalize<-function(x) {
    return( (x-min(x))/ ( max(x)-min(x)))
  }

  wbcd <- wbcd[-1]
  ncol1 <- which(colnames(wbcd)==y)

  wbcd_n <- as.data.frame(lapply(wbcd[, -ncol1], normalize) )

  mm<-round(nrow(wbcd_n)*9/10)
  wbcd_train <- wbcd_n[1:mm, ]
  wbcd_test  <- wbcd_n[(mm+1):nrow(wbcd_n), ]

  wbcd_train_label <- wbcd[1:mm,y]
  wbcd_test_label  <- wbcd[(mm+1):nrow(wbcd_n),y]

  library(caret)
  library(e1071)

  repeats = 29   # 반복 횟수를 늘릴수록 데이터가 늘어남으로 더 정확한 k값을 구할 수 있다.
  numbers = 10   # 데이터를 10등분 함 10개 중 하나를 validation 나머지 train 을 순서대로 10번 반복
  tunel = 10

  set.seed(1234)
  t = trainControl(method = "repeatedcv",
                    number = numbers,
                    repeats = repeats,
                    classProbs = TRUE,
                    summaryFunction = twoClassSummary)
```

```

model1 <- train(wbcd_train_label~. , data = data.frame(wbcd_train,wbcd_train_label), method = "knn", #라벨, 훈련
데이터, 훈련방식
               preprocess = c("center","scale"),          # 정규화 하겠다
               trControl = t,
               metric = "ROC",                             # 그래프에서 확인
               tuneLength = tune1)                         #10 그래프에서 확인

# Summary of model
model1
plot(model1)

library(class)
result1 <- knn(train=wbcd_train, test=wbcd_test,
               cl= wbcd_train_label, k = model1$bestTune[1] )

library(gmodels)
CrossTable( x= wbcd_test_label, y= result1,prop.chisq=FALSE)
}

knn_fun()

```

wbcd_test_label	result1		Row Total
	B	M	
B	41	0	41
	1.000	0.000	0.719
	1.000	0.000	
	0.719	0.000	
M	0	16	16
	0.000	1.000	0.281
	0.000	1.000	
	0.000	0.281	
Column Total		16	57
	41	0.281	

문제 264. 아산병원에 온 환자가 유방암이 악성인지 양성인지를 예측하는 자동화 knn 모델을 완성 시키시오.

```

knn_fun<-function(){
  fname<-file.choose()
  wbcd <- na.omit(read.csv(fname, stringsAsFactors=FALSE))
  y <- menu(colnames(wbcd), title = " 라벨 컬럼을 선택하세요")
  y <- colnames(wbcd[y])
  z <- menu(colnames(wbcd), title = " 삭제 할 컬럼을 선택하세요")
  normalize<-function(x) {
    return((x-min(x))/ (max(x)-min(x)))
  }

  print('환자 정보가 담긴 파일을 넣어주세요')
  patient<-read.csv(file.choose(),stringsAsFactors=F, header=T)

```

```

wbcd <- wbcd[-z]
ncol1 <- which(colnames(wbcd)==y)
wbcd_n <- as.data.frame(lapply(rbind(wbcd[, -ncol1], patient), normalize))
pp<-wbcd_n[nrow(wbcd_n),]
wbcd_n<-wbcd_n[-nrow(wbcd_n),]

mm<-round(nrow(wbcd_n)*9/10)
wbcd_train <- wbcd_n[1:mm, ]
wbcd_test  <- wbcd_n[(mm+1):nrow(wbcd_n), ]

wbcd_train_label <- wbcd[1:mm,y]
wbcd_test_label  <- wbcd[(mm+1):nrow(wbcd_n),y]

library(class)
repeats = 3
numbers = 10
tunel = 10

set.seed(1234)

x = trainControl(method = "repeatedcv",
                  number = numbers,
                  repeats = repeats,
                  classProbs = TRUE,
                  summaryFunction = twoClassSummary)

model1 <- train( wbcd_train_label~. , data = data.frame(wbcd_train,wbcd_train_label), method = "knn",
                 preProcess = c("center","scale"),
                 trControl = x,
                 metric = "ROC",
                 tuneLength = tunel)

k_n<-model1$bestTune

result1 <- knn(train=wbcd_train, test=wbcd_test,
               cl= wbcd_train_label, k = k_n)
library(gmodels)
CrossTable(x = wbcd_test_label, y= result1, prop.chisq=FALSE)
result2<-knn(train=wbcd_train, test=pp,cl=wbcd_train_label,k=k_n)

if(result2=='M')print('환자의 종양은 악성입니다.')
else print('환자의 종양은 양성입니다.')
}

knn_fun()

```

■ 추가 예제

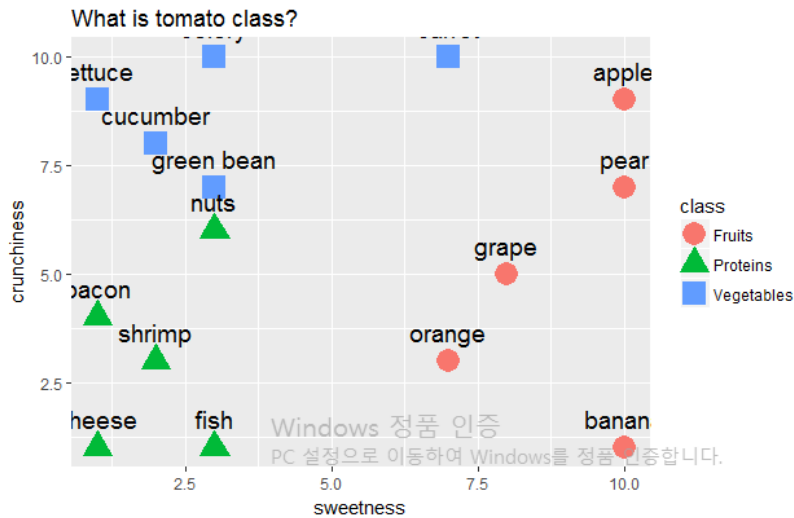
```

# food dataframe
food <- data.frame(ingredient = c("apple", "bacon", "banana", "carrot",
                                "celery", "cheese", "cucumber", "fish",
                                "grape", "green bean", "lettuce",
                                "nuts", "orange", "pear", "shrimp"
                                ),
sweetness = c(10,1,10,7,3,1,2,3,8,3,1,3,7,10,2),
crunchiness = c(9,4,1,10,10,1,8,1,5,7,9,6,3,7,3),
class = c("Fruits","Proteins","Fruits","Vegetables",
          "Vegetables","Proteins","Vegetables",
          "Proteins","Fruits","Vegetables",
          "Vegetables","Proteins","Fruits",
          "Fruits","Proteins"))
food
# 토마토 데이터 만들기
tomato <- data.frame(ingredient = "tomato",
                     sweetness = 6,
                     crunchiness = 4)
tomato
#####
##### 그래프 그리기 비교 #####
#####
# ggplot2 : 그래프 만드는 패키지

install.packages("ggplot2")

library(ggplot2)
# par : 파라미터 지정 / pty : plot모형을 "square" 정사각형
par(pty="s")
# 그래프 그리기(version : ggplot)
#par:파라미터/xpd:모형웁기기/mar:여백설정(아래,왼쪽,위,오른쪽)
par(xpd=T, mar=par())$mar+c(0,0,0,15))
plot(food$sweetness,food$crunchiness,
     pch=as.integer(food$class),
     #pch=food$class, # pch는 모형 지정
     xlab = "sweetness", ylab = "crunchiness",
     main = "What is tomato class?")
legend(10.5,10, # legend 위치 지정
      c("Fruits", "Proteins", "Vegetables", "X"),
      pch=as.integer(food$class))
text(food$sweetness, food$crunchiness,
     labels=food$ingredient,
     pos = 3, # 글자위치position(1:below/2:left/3:above/4:right)
     offset = 0.3, # 좌표와 얼마나 띄어쓰기 할것인지
     cex = 0.7 ) # 문자크기
# 그래프 그리기(version : ggplot2)
ggplot(data=food,aes(x=sweetness,y=crunchiness))+
  labs(title="What is tomato class?")+ # 타이틀 명
  geom_point(aes(color=class, shape=class),size=6)+
  geom_text(aes(label=ingredient), # 라벨링 표시
            vjust=-1, # 수직으로 움직일 거리 (위는 -, 아래는 +)
            size = 5) # 문자크기

```



```
#####
##### Knn #####
#####
# dplyr : 큰 데이터를 다룰 때, split-apply-combine 논리로 접근 작업가능
install.packages("dplyr")
#contains knn function
library(class)
library(dplyr)
# k=1로 두었을 때, 토마토만 예측
# 유클리디안 측정 1nn 분류
tmt <- knn(select(food, sweetness, crunchiness),
           select(tomato, sweetness, crunchiness),
           food$class, k=1)
tmt
# 데이터프레임 만들기
# 포도, 완두콩, 오렌지, 토마토를 통해서 예측하기
unknown <- data.frame(ingredient = c("grape", "green bean", "orange", "tomato"),
                      sweetness = c(8, 3, 7, 6),
                      crunchiness = c(5, 7, 3, 4))
unknown
# 포도, 완두콩, 오렌지, 토마토를 통해서 k=3으로 예측
pred <- knn(select(food, sweetness, crunchiness),
            select(unknown, sweetness, crunchiness),
            food$class, k=3)
Pred
```


나이브 베이즈 분류

2018년 5월 29일 화요일 오전 9:57

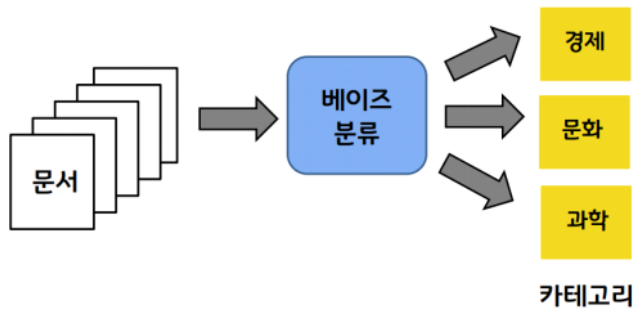
4장 목차

1. 확률에 대한 기본적인 이해
2. 나이브 베이즈 알고리즘
3. 나이브 베이즈 실습
 - 독버섯과 정상버섯의 분류
 - 영화장르 분류
 - 스팸메일과 햄메일의 분류
4. 머신러닝 자동화 스크립트에 나이브 베이즈 추가

1. 확률에 대한 기본적인 이해

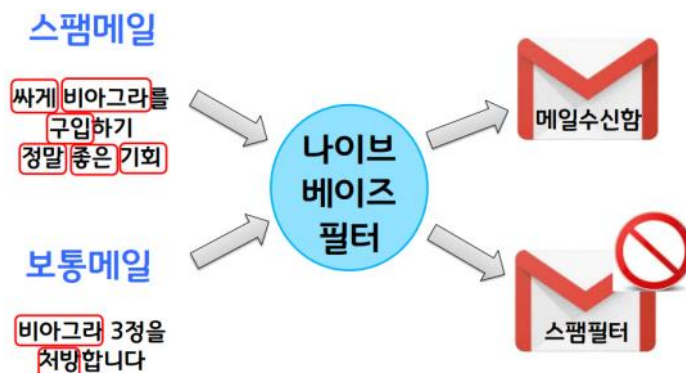
베이즈 분류

베이즈 이론을 이용해서 주어진 **대상을 원하는 카테고리**로 분류하는 방법을 말합니다.



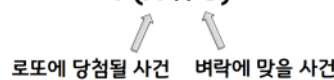
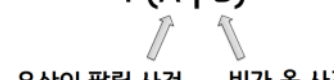
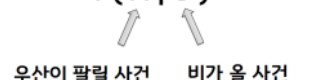
사용예시

스팸성 단어들이 더 나왔을때 스팸일 확률이 높아짐으로 스팸으로 분류합니다.



나이브 베이즈 분류를 이해하려면 아래의 내용을 순서대로 이해하면 됩니다

1. 확률의 시행과 사건	주사위를 던지는 것을 시행 이라고 하고 주사위를 던져서 6이 나온 걸 사건 이라고 합니다
---------------	---

<p>2. 결합 확률과 조건부 확률</p>	<p>확률은 결합 확률과 조건부 확률이 있습니다.</p> <p>결합 확률은 서로 배반되는 두 사상 A와 B가 있을 때 두 사상이 연속적으로 또는 동시에 일어나는 확률을 말합니다. 예를 들면 로또에 당첨될 사건과 벼락에 맞을 사건이 동시에 일어날 확률을 말합니다</p> <div style="text-align: center;"> $P(A \cap B)$  <p>로또에 당첨될 사건 벼락에 맞을 사건</p> </div> <p>조건부 확률은 어떠한 상황이 주어졌을 때 그 상황속에서 다른 상황이 일어날 확률을 말합니다. 예를 들면 비가 오는 사건이 일어나는 경우 하에 우산이 팔리는 사건이 일어나는 경우를 말합니다.</p> <div style="text-align: center;"> $P(A B)$  <p>우산이 팔릴 사건 비가 올 사건</p> </div>
<p>3. 독립사건과 종속사건</p>	<p>사건은 독립사건과 종속사건으로 나뉘는데 예를 들면 동전던지기의 결과와 화창한 날씨와는 서로 독립적입니다 독립사건을 조건부 확률로 나타내면 아래와 같습니다.</p> <div style="text-align: center;"> $P(A B) = P(A)$ $P(\text{👉} \text{🌤}) = P(\text{👉})$ </div> <p>종속사건은 사건 B 가 일어났을 경우와 일어나지 않았을 경우에 따라서 사건 A 가 일어날 확률이 다를 때 A 는 B 의 종속사건</p> <div style="text-align: center;"> $P(A B)$  <p>우산이 팔릴 사건 비가 올 사건</p> <p style="text-align: right;">사건A와 사건B가 동시에 일어날 결합확률</p> $P(A B) = \frac{P(A \cap B)}{P(B)}$ <p style="text-align: right;">사건 B가 일어날 확률</p> <p>정리하면 화창한 날씨에 동전던지기</p> <p>독립사건의 조건부 확률 :</p> $P(A B) = P(A)$ <p style="text-align: right;">비가올때 우산 팔기</p> <p>종속사건의 조건부 확률 :</p> $P(A B) = \frac{P(A \cap B)}{P(B)}$ </div>
<p>4. 베이즈 정리</p>	<p>베이즈 분류의 조건부 확률 공식</p> $P(A B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B A) * P(A)}{P(B)}$ <p>Ex. 흡연을 하는 사람이 간암에 걸릴 확률</p>

	$P(\text{간암} \text{흡연}) = \frac{P(\text{흡연} \text{간암}) * P(\text{간암})}{P(\text{흡연})}$ <p>나이브 베이즈의 장점이 간암의 요인을 흡연이외에 여러 개로 나열해서 추론할 수 있다</p> $P(\text{간암} \text{흡연, 음주, 직업, 성별, ...})$
--	---

2. 나이브 베이즈에 의한 분류

---> 범주형 데이터 사용

*스팸 메일과 햄메일을 정확하게 구분하기 위해서는?

비아그라 단어 하나만 가지고 스팸 메일인지를 분류하면 처방전도 스팸이 될 확률이 높아지니 다른 단어들도 같이 포함시켜서 확률을 구해야 한다.

예 : 비아그라 = yes, 돈 = no, 식료품 = no, 주소삭제 = yes

용어
¬ : 존재하지 않는다 (부정)
∩ : 존재한다 (긍정)

예제	비아그라와 주소 삭제는 포함하고 돈과 식료품은 포함하지 않는 메시지가 스팸일 확률은 어떻게 되는가?
	<div> <div>비아그라(w1)</div> <div>돈(w2)</div> <div>식료품(w3)</div> <div>주소삭제(w4)</div> </div> <div> <div>우도</div> <div>Yes</div> <div>No</div> <div>Yes</div> <div>No</div> <div>Yes</div> <div>No</div> <div>Yes</div> <div>No</div> </div> <div> <div>스팸</div> <div>4/20</div> <div>16/20</div> <div>10/20</div> <div>10/20</div> <div>0/20</div> <div>20/20</div> <div>12/20</div> <div>8/20</div> <div>20</div> </div> <div> <div>햄</div> <div>1/80</div> <div>79/80</div> <div>14/80</div> <div>66/80</div> <div>8/80</div> <div>71/80</div> <div>23/80</div> <div>57/80</div> <div>80</div> </div> <div> <div>총합</div> <div>5/100</div> <div>95/100</div> <div>24/100</div> <div>76/100</div> <div>8/100</div> <div>91/100</div> <div>35/100</div> <div>65/100</div> <div>100</div> </div>

$p(A|B) = P(A \cap B) / P(B) = (P(B|A) * P(A)) / P(B)$ 공식을 이용해서

$p(\text{스팸} | \text{비아그라} \cap \neg \text{돈} \cap \neg \text{식료품} \cap \text{주소삭제})$

$= \frac{P(\text{비아그라} \cap \neg \text{돈} \cap \neg \text{식료품} \cap \text{주소삭제} | \text{스팸}) * P(\text{스팸})}{P(\text{비아그라} \cap \neg \text{돈} \cap \neg \text{식료품} \cap \text{주소삭제})}$

$= \frac{P(\text{비아그라} | \text{스팸}) \cap P(\neg \text{돈} | \text{스팸}) \cap P(\neg \text{식료품} | \text{스팸}) \cap P(\text{주소삭제} | \text{스팸}) * P(\text{스팸})}{P(\text{비아그라} \cap \neg \text{돈} \cap \neg \text{식료품} \cap \text{주소삭제})}$

스팸일 확률 ?

$P(\text{비아그라} | \text{스팸}) \cap P(\neg \text{돈} | \text{스팸}) \cap P(\neg \text{식료품} | \text{스팸}) \cap P(\text{주소삭제} | \text{스팸}) * P(\text{스팸})$

스팸 우도 + 햄 우도

스팸일 우도 (분자값) : $(4/20) * (10/20) * (20/20) * (12/20) * (20/100) = 0.012$

스팸일 확률 : $\text{우도} / (0.012 + 0.002) = 0.85$

햄일 확률 ?

$$\frac{P(\text{비아그라} | \text{햄}) \cap P(\neg \text{돈} | \text{햄}) \cap P(\neg \text{식료품} | \text{햄}) \cap P(\text{주소삭제} | \text{햄}) * P(\text{햄})}{\text{스팸 우도} + \text{햄 우도}}$$

햄일 우도 (분자값) : $(1/80) * (66/80) * (71/80) * (23/80) * (80/100) = 0.0021$

햄일 확률 : $\text{우도} / (0.012 + 0.002) = 0.15$

문제 265. 아래 메일의 메시지를 보고 비아그라와 쿠폰이라는 메시지가 메일에 포함되어져 있으면 스팸일 확률이 어떻게 되는지 확인 하시오.

No	Email
1	I got viagra from my friend
2	viagra coupon from xx.com
3	watch viagra new product from viagra.com
4	Best deal, prommo code here
5	There will be viagra consumer meeting today 2pm
6	Scheduled meeting tomorrow
7	Can we have lunch today?
8	I miss you
9	thanks my friend
10	It was good to see you today
11	viagra coupon , last deal
12	viagra sale coupon
13	I sent the coupon you asked, it is viagra
14	Hurry up amazing deal!

$(4/6) * (3/6) * (6/14) = 0.1429$

문제 266. 위의 문제를 R로 구현 하시오.

```
x<-matrix(c(4,3,2,5),nrow = 2)
y<-matrix(c(3,1,3,7),nrow = 2)
```

```

r1 <- (x[1,1]/sum(x[1,])) * (y[1,1]/sum(x[1,])) * (sum(x[1,])/sum(x)) #스팸 우도
r2 <- (y[1,1]/sum(y[2,])) * (y[2,1]/sum(y[2,])) * (sum(y[2,])/sum(y)) # 햄 우도

r3 <- r1 / ( r1 + r2)
r3

> r3 <- r1 / ( r1 + r2)
> r3
[1] 0.8421053

```

3. 나이브 베이즈 실습

1.영화데이터 실습

" 나이브 베이즈 분류로 영화 장르 선호도를 분석해본다. "

```

movie <- read.csv("c:\data\movie.csv", header = T) # 데이터 불러온다.

colnames(movie)<-c("age","gender","job","marry","friend","m_type") # 컬럼명을 영어로 바꿈

unique(movie[, "m_type"]) # 영화 장르 종류 확인

head(movie,10)

str(movie) # 39

movie[movie$age == '20대' & movie$gender == '여' & movie$job == 'IT' & movie$marry == 'NO', ]

# 20대의 IT에 종사하는 미혼 여성이 즐겨보는 영화장르는?
# P (공포 | '20대' n n n n

nrow(movie[movie$m_type=="스릴러"&movie$age=="20대",])/nrow(movie[movie$age=="20대",])

nrow(movie[movie$m_type=="스릴러"& movie$gender=="여자",])/nrow(movie[movie$gender=="여자",])

nrow(movie[movie$m_type=="스릴러"&movie$marry=="NO",])/nrow(movie[movie$marry=="NO",])

nrow(movie[movie$m_type=="스릴러"&movie$job=="IT",])/nrow(movie[movie$job=="IT",])

```

나이브 베이즈 함수를 이용해서 예측 결과를 출력한다

(나이 , 성별, 직업, 결혼여부, 이성친구에 따라서 선호하는 영화장르가 어떻게 되는지 예측)

```

install.packages("e1071")
library(e1071)
model<-naiveBayes(movie[1:nrow(movie)-1,1:5], movie[1:nrow(movie)-1,"m_type"], laplace=0) # 훈련데이터, 훈련데이
터 라벨(정답),

```

```
model
```

```
result<-predict(model,movie[nrow(movie),1:5])    #훈련된 모델, 테스트 데이터
```

```
Result
```

2.독버섯 실습

"독버섯과 정상 버섯 분류 테스트"

1. 버섯 데이터를 R로 로드한다.
2. ? 표시의 데이터를 NA로 변경한다.
3. "비어있는 데이터를 NA로 변경한다.
4. Mushroom 전체 데이터를 factor로 변환한다.
5. Mushroom 데이터를 훈련 데이터와 테스트 데이터로 나눈다. (75%는 훈련 데이터, 25%는 테스트 데이터)
6. 나이브 베이즈 함수를 이용해서 독버섯과 일반 버섯을 분류하는 모델을 생성한다.
7. 위에서 만든 모델로 테스트 데이터를 가지고 독버섯과 일반버섯을 잘 맞추는지 확인한다.
8. 이원 교차표를 그려서 최종 분류 결과를 확인한다.

```
murshroom<-read.csv("c:\\data\\murshroom.csv",header = F, stringsAsFactors = F)
```

```
murshroom[which(murshroom$V12=="?"), "V12"] <- NA
```

```
length(murshroom)
```

```
head(murshroom)
```

```
for (i in 1:length(murshroom)){  
  murshroom[which(murshroom[,i]==""), i] <- NA  
}
```

#4 murshroom 전체 데이터를 factor로 변환

```
dim(murshroom)
```

```
for(i in 1 : length(murshroom)){  
  murshroom[, i] <- factor(murshroom[,i])  
}
```

#5 murshroom 데이터를 훈련 데이터와 테스트 데이터로 나눈다 (75% 훈련데이터, 25%는 테스트 데이터)

```
set.seed(123450)
```

```
dim(murshroom)
```

```
train_cnt<- round (0.75*dim(murshroom)[1])
```

```
train_indx<-sample(1:dim(murshroom)[1],train_cnt,replace = F)
```

```
murshroom_train <- murshroom[train_indx,]
```

```
murshroom_test <- murshroom[-train_indx,]
```

#6. 나이브 베이즈 함수를 이용해서 독버섯과 일반버섯을 분류하는 모델을 생성한다.

```
library(e1071)
model1<-naiveBayes(V1~.,data=murshroom_train)

#7. 위에서 만든 모델로 테스트 데이터를 가지고 독버섯과 일반 버섯을 잘 맞추는지 확인한다.
result1<-predict(model1, murshroom_test[,-1])
result1

#8. 이원 교차표를 그려서 최종 분류 결과를 확인한다.
library(gmodels)
CrossTable(murshroom_test[,1], result1) # 실제(y축), 예측(x축)

#9. 위의 모델 성능을 올리시오. # 라플라스
library(e1071)
model2<-naiveBayes(V1~. , data = murshroom_train, laplace = 0.001)
result2<-predict(model2,murshroom_test[,-1])
CrossTable(murshroom_test[,1],result2)
```

4. 라플라스 추정기 (p 155)

아까 문제는 비아그라와 주소 삭제는 포함하면서 돈과 식료품은 포함하지 않는 메세지가 스팸일 확률을 구했는데 이번에는 비아그라, 주소삭제, 식료품, 돈이 전부 포함되어져 있는 메세지가 스팸일 확률을 구한다면 어떻게 해야할까?

스팸의 우도 ?

	비아그라(w1)		돈(w2)		식료품(w3)		주소삭제(w4)		
우도	Yes	No	Yes	No	Yes	No	Yes	No	
스팸	4/20	16/20	10/20	10/20	0/20	20/20	12/20	8/20	20
햄	1/80	79/80	14/80	66/80	8/80	71/80	23/80	57/80	80
총합	5/100	95/100	24/100	76/100	8/100	91/100	35/100	65/100	100

$P(\text{비아그라}|\text{스팸}) * P(\text{돈}|\text{스팸}) * P(\text{식료품}|\text{스팸}) * P(\text{주소삭제} | \text{스팸}) * P(\text{스팸})$

= $4/20 * 10/20 * 0/20 \dots$ 식료품으로 인해 다른 증거들이 모두 무효(0값)가 된다. 이를 해결 하기 위해서 프랑스의 수학자 피에르 시몬 라플라스가 확률이 0이 되지 않기 위해서 빈도표의 각 값에 작은 수를 추가했다.

위의 각각의 값이 1을 더해본다.

라플라스 추정기를 사용한 스팸의 우도?

---> 전체 값에 1씩 더해준다. (0값을 없애기 위해서)

$P(\text{비아그라}|\text{스팸}) * P(\text{돈}|\text{스팸}) * P(\text{식료품}|\text{스팸}) * P(\text{주소삭제} | \text{스팸}) * P(\text{스팸})$

= $5/24 * 11/24 * 1/24 * 13/24 * 24/104 = ?$

문제 267. Knn_fun() 함수를 만들었는데 이번에는 naive_bayes_fun() 함수를 아래와 같이 생성 하시오.

```
naive_bayes_fun()
```

윈도우 탐색기가 열리면서 csv 파일을 선택
라벨 선택 , 라플라스값 입력

```
naive_bayes_fun<-function(){
```

```
  dt<-read.csv(file.choose(),header = F, stringsAsFactors = F)
```

```
  res1 <- menu(colnames(dt), title='라벨이 될 컬럼번호 선택 : ')
```

```
  lplc <- as.numeric(readline(prompt = '라플라스 값 입력 (사용하지 않으려면 0 입력): '))
```

```
  for (i in 1:length(dt)){
```

```
    dt[which(dt[,i]=='|dt[,i]=='?'), i] <- NA
```

```
  }
```

```
  dt<-na.omit(dt) # na값 생략 ( ? or '')
```

```
  for(i in 1 : length(dt)){
```

```
    dt[, i] <- factor(dt[,i])
```

```
  }
```

```
  set.seed(123450)
```

```
  train_cnt<- round (0.75*nrow(dt))
```

```
  train_indx<-sample(1:nrow(dt),train_cnt,replace = F)
```

```
  dt_train <- dt[train_indx,]
```

```
  dt_test <- dt[-train_indx,]
```

```
  library(gmodels)
```

```
  library(e1071)
```

```
  model2<-naiveBayes(dt_train[,res1]~. , data = dt_train, laplace = lplc) #라플라스 사용 o
```

```
  result2<-predict(model2,dt_test[,1])
```

```
  CrossTable(dt_test[,1],result2)
```

```
}
```

```
naive_bayes_fun()
```


dt_test[, 1]	result2		Row Total
	e	p	
e	847	0	847
	150.700	289.780	
	1.000	0.000	0.657
	0.999	0.000	
	0.657	0.000	
p	1	441	442
	288.784	555.303	
	0.002	0.998	0.343
	0.001	1.000	
	0.001	0.342	
Column Total	848	441	1289
	0.658	0.342	

의사결정 트리

2018년 5월 30일 수요일 오전 9:53

5. 의사결정 트리 목차

1. 의사결정트리가 무엇인가?
2. 엔트로피와 정보 획득량
3. 의사결정트리 실습
 - 심장질환 데이터
 - 부도 예측 데이터
 - 지방간 데이터 독일 은행의 대출여부 데이터
4. 의사결정 자동화 스크립트

의사결정 트리에 필요한 수학적 지식 ? 확률, 로그함수

1. 의사결정트리란?

" 의사결정트리 알고리즘은 데이터 사이에 존재하는 패턴을 예측 가능한 규칙들의 조합으로 나타내는 알고리즘 "

- decision rule (의사결정 룰)을 계층적 나무구조로 도표화 하여 분류 및 예측할 수 있는 분석 방법

*의사결정 트리의 장점

결정트리를 통한 데이터 분석의 결과는 나무구조로 표현되기 때문에 분석가가 결과를 쉽게 이해하고 설명할 수 있다.

분류율에 대한 정확도만 따지면 신경망, 회귀분석 등의 분류 방법들보다 낮게 평가되지만 결과를 쉽게 이해하고 설명할 수 있으며 의사결정을 하는데 직접적으로 사용할 수 있는 장점이 있다.

의사결정 알고리즘의 종류

- ID3 알고리즘 (엔트로피)
- C4.5 알고리즘 (정보획득량)
- C5.0 알고리즘 (정보획득량) : C4.5 + trials 옵션 + ..
- CART 알고리즘 (카이제곱)
- CHAID 알고리즘 (지니계수)

의사결정트리 구현 시 고려할 3가지 문제

1. 훈련 데이터를 어떤 속성(컬럼)으로 분할해야 하느냐?

---> 정보 획득량이 가장 높은 컬럼을 우선순위로 두고 분할하면 된다.(information.gain함수 사용)

2. 트리의 성장(분할과정)은 언제, 어떻게 정지해야 하는가?

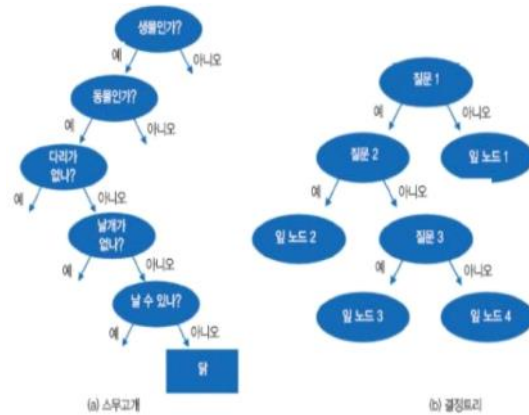
---> 의사결정 트리를 시각화해서 순도가 높게 분류되는지를 확인해서 정지

3. 과적합(overfitting) 문제는 어떻게 해결해야 하느냐? (과적합 : train 데이터는 잘 맞추지만, test 데이터는 못맞춤)

---> tree pruning(가지치기)를 이용해서 불필요한 가지(질문)를 제거한다.

(질문이 너무 많으면 배가 산으로 간다..)

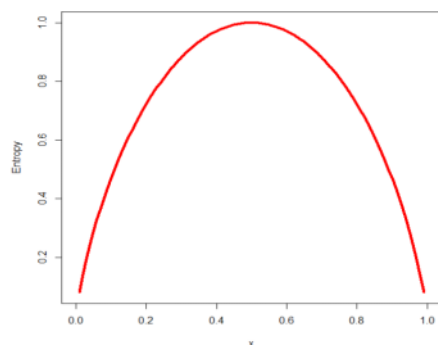
Ex. 예를 들면 질문을 던져서 대상을 좁혀 나가는 스무 고개 놀이와 비슷한 개념



질문을 할 때 중요한 질문들을 우선적으로 해야 더 빨리 정답을 얻을 수 있다.

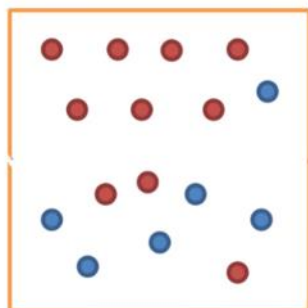
우선순위를 정할때 필요한게 엔트로피(entropy)이다. 엔트로피는 불확실성의 정도를 말한다.

x 축이 확률이고 y 축이 엔트로피입니다



엔트로피 구하는 예시

아래의 박스의 엔트로피를 구해보면 ?
(불순도)

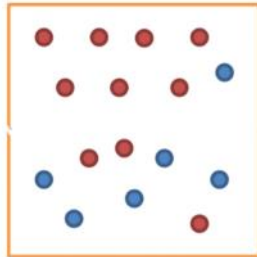


엔트로피 공식에 대입해서

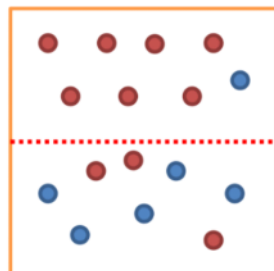
$$Entropy(A) = - \sum_{k=1}^m p_k \log_2(p_k)$$

전체 16개 가운데 빨간색 동그라미(범주, k=1)는 10개, 파란색(범주, k=2)은 6개이군요. 그럼 A 영역의 엔트로피는 다음과 같습니다.

$$Entropy(A) = -\frac{10}{16} \log_2\left(\frac{10}{16}\right) - \frac{6}{16} \log_2\left(\frac{6}{16}\right) \approx 0.95$$

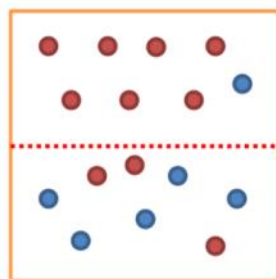


이번에는 아래와 같이 분할한 상태에서
의 불순도를 각각 구해보겠습니다



$$Entropy(A) = \sum_{i=1}^d R_i \left(- \sum_{k=1}^m p_k \log_2(p_k) \right)$$

$$Entropy(A) = 0.5 \times \left(-\frac{7}{8} \log_2\left(\frac{7}{8}\right) - \frac{1}{8} \log_2\left(\frac{1}{8}\right) \right) + 0.5 \times \left(-\frac{3}{8} \log_2\left(\frac{3}{8}\right) - \frac{5}{8} \log_2\left(\frac{5}{8}\right) \right) \approx 0.75$$



그래서 정보 획득량은 ?

분할전 엔트로피 - 분할후 엔트로피

이므로

정보 획득량 ? $0.95 - 0.75 = 0.2$ 입니다

****함수를 사용해서 정보 획득량을 얻을 수 있다**

```
library(FSelector)
x<-information.gain(coupon_react~,skin) # coupon_react : 라벨컬럼, ~. : 나머지 컬럼 전체
```

문제 268. 화장품 구매 고객 데이터 컬럼들에 대한 정보 획득량을 출력 하시오.

```
skin<-read.csv("c:\\data\\skin.csv",header = T)

install.packages("FSelector") # r 3.4 버전에서는 안되서 3.5버전으로 변경함.
library(FSelector)

skin<-skin[,-1] #불필요한 id 컬럼 삭제

x<-information.gain(coupon_react~,skin) # coupon_react : 라벨컬럼, ~. : 나머지 컬럼 전체
x

> x
      attr_importance
cust_no      0.000000000
gender       0.080610238
age          0.019525931
job          0.013737789
marry        0.224337222
car          0.006023806
```

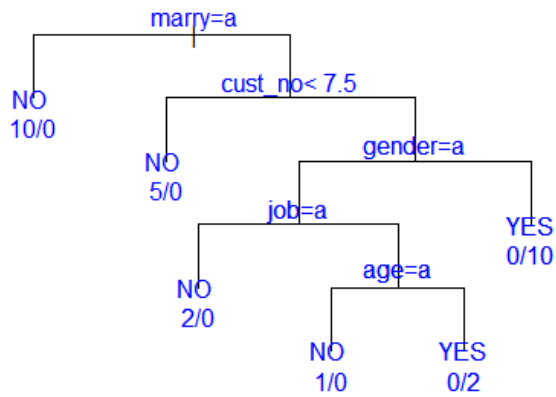
문제 269. 화장품 고객 데이터를 가지고 의사결정 트리를 시각화 하시오.

```
install.packages("rpart")
library(rpart)

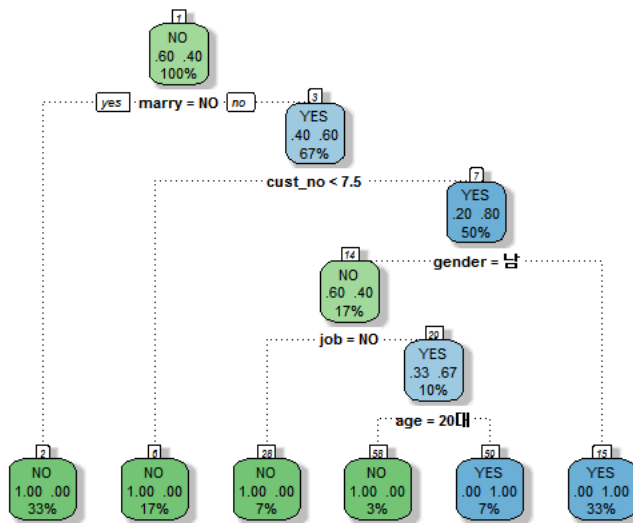
tree1<-rpart(coupon_react ~., data=skin,control = rpart.control(minsplit = 2))

# minsplit=5: 한 노드를 분할하기 위해 필요한 데이터의 개수.
# 이 값보다 적은 수의 관측치가 있는 노드는 분할하지 않는다. 디폴트는 20

plot(tree1,compress = T, uniform = T, margin = 0.1)
text(tree1,use.n = T, col="blue")
```



```
install.packages("rattle")
library(rattle)
fancyRpartPlot(tree1)
```



Rattle 2018-5-30 11:15:48 Administrator

문제 270. 지방간 데이터의 정보 획득량을 구해서 지방간을 일으키는데 가장 영향력이 큰 변수가 무엇인지 알아내시오.

```
fatliver <- read.csv(file.choose(),header = T)
x<-information.gain(FATLIVER~,fatliver)
x
```

	attr_importance
AGE	0.022358781
GENDER	0.019859086
DRINK	0.008449112
SMOKING	0.006801213

문제 271. 백화점 고객 구매 여부 데이터를 가지고 각 컬럼에 대해서 정보 획득량을 구하시오.

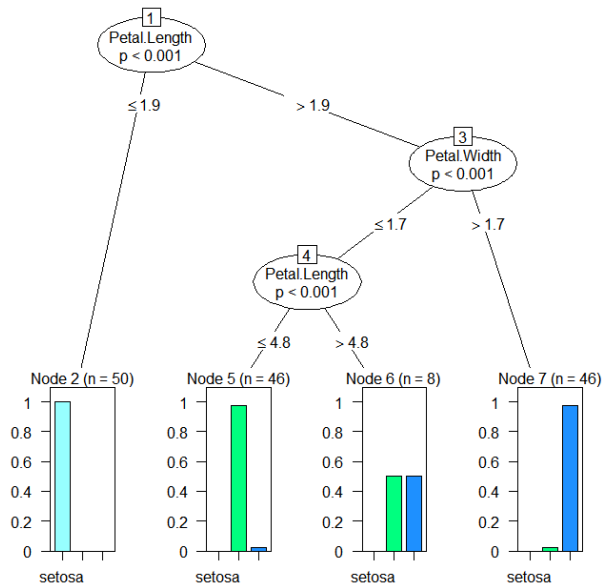
```
buy<-read.csv("c:\\data\\buy2.csv",header = T)
inf<-information.gain(buy_yn~,buy[,c(-1,-2)])
inf
```

> inf

```
> inf
      attr_importance
card_yn      0.50040242
review_yn    0.22314355
before_buy_yn 0.05053431
```

문제 272. 좋은 패키지를 찾아서 가장 예쁜 의사 결정트리를 그린 라인만 식사

```
library(party)
irisct <- ctree(Species ~ ., data = iris)
plot(irisct, tp_args = list(fill = c("Dark Slate Gray 1", "Spring Green 1", "Dodger Blue 1")))
```



■ C5.0 패키지를 이용해서 분류 모델 생성

1. 의사결정 패키지인 C50 패키지를 설치한다.

```
install.packages("C50")
library(C50)
```

2. 화장품 고객 데이터를 shuffle 한다

```
skin <- read.csv("skin.csv", header=T)
skin <- skin[, -1]
set.seed(1)
skin_shuffle <- skin[sample(nrow(skin)), ]
```

3. 화장품 고객 데이터를 9대 1로 트레인과 테스트로 나눈다

```
train_num <- round(0.7*nrow(skin_shuffle), 0)
skin_train <- skin_shuffle[1:train_num,]
skin_test <- skin_shuffle[(train_num+1):nrow(skin_shuffle),]
```

4. C50 패키지와 train data 로 분류 모델을 생성한다

```
skin_model <- C5.0(skin_train[, -6], skin_train$cupon_react )
```

↑ ↑
 라벨을 뺀 전체 train data train data 라벨

5. 분류모델과 test data 로 분류결과를 예측한다

```
skin_result <- predict(skin_model, skin_test[, -6])
```

↑
 테스트 데이터

6. 이원교차표를 그려서 결과를 확인한다.

```
library(gmodels)
CrossTable( skin_test[, 6], skin_result )
```

↑ ↑
 실제 예측

skin_test[, 6]	skin_result		Row Total
	NO	YES	
NO	4	3	7
	0.003	0.004	0.778
	0.571	0.429	
	0.800	0.750	
	0.444	0.333	
YES	1	1	2
	0.011	0.014	0.222
	0.500	0.500	
	0.200	0.250	
	0.111	0.111	
Column Total	5	4	9
	0.556	0.444	

skin_test[, 6]	skin_result		Row Total
	NO	YES	
NO	6	0	6
	1.000	2.000	0.667
	1.000	0.000	
	1.000	0.000	
	0.667	0.000	
YES	0	3	3
	2.000	4.000	0.333
	0.000	1.000	
	0.000	1.000	
	0.000	0.333	
Column Total	6	3	9
	0.667	0.333	

문제 273. 위의 의사결정 트리의 성능을 높이시오

```
install.packages("C50")
```

```
library(C50)
```

```
skin <- read.csv("skin.csv", header=T)
```

```
skin <- skin[, -1]
```

```
set.seed(10)
```

```
skin_shuffle <- skin[sample(nrow(skin)), ]
```



```
train_num<-round(0.7*nrow(skin_shuffle),0)
```

```
skin_train <- skin_shuffle[1:train_num,]
```

```
skin_test <-skin_shuffle[(train_num+1):nrow(skin_shuffle),]
```

```
skin_model2 <- C5.0(skin_train[,6],skin_train$cupon_react, trials = 10) # trials 옵션을 주거나 set.seed로 다시 섞음
```

```
skin_result2 <- predict(skin_model2,skin_test[,6])
```

```
library(gmodels)
```

```
CrossTable( skin_test[ , 6], skin_result2 )
```

skin_test[, 6]	skin_result2		Row Total
	NO	YES	
NO	3	1	4
	2.083	1.042	0.444
	0.750	0.250	
	1.000	0.167	
	0.333	0.111	
YES	0	5	5
	1.667	0.833	0.556
	0.000	1.000	
	0.000	0.833	
	0.000	0.556	
Column Total			
		3	6
		0.333	0.667

trial 옵션 : 트레이닝 데이터를 반복 추출하여 표본을 여러 개로 만든 후에 각 표본마다의 분류 모델을 생성해서 가장 좋은 모델을 고르는 방법

■ 독일 은행의 대출 여부 데이터로 의사결정 트리 실습

데이터 : credit.csv

1. 데이터를 로드한다.
2. 데이터에 각 컬럼들을 이해한다.
3. 데이터가 명목형 데이터인지 확인해 본다.
4. 데이터를 shuffle 시킨다.
5. 데이터를 9대1로 나눈다.
6. C5.0패키지와 훈련 데이터를 이용해서 모델을 생성한다.
7. 모델과 테스트 데이터로 결과를 예측한다.
8. 이원 교차표로 결과를 살펴본다.

#과거의 데이터를 분석해보니 대출금 상환 불이행자가

#30%나 되어서 앞으로는 30% 이내로떨어뜨리는것이 은행의 목표 되게끔 model을 짜시오.

#1. 데이터를 로드한다.

```
credit<-read.csv("c:\\data\\credit.csv",header = T)
```

#2. 데이터에 각 컬럼들을 이해한다.

```
head(credit,4)
```

라벨 : default (yes:대출금 상환 안함, no:대출금 상환)

checking_balance : 예금계좌 , saving_balance : 적금계좌, amount : 대출금액,

```

#3. 데이터가 명목형 데이터인지 확인해 본다.
#4. 데이터를 shuffle 시킨다.
set.seed(11)
credit_shuffle <- credit[sample(nrow(credit)),]

#5. 데이터를 9대1로 나눈다.
train_num<-round(0.9*nrow(credit_shuffle),0)

credit_train <- credit_shuffle[1:train_num,]
credit_test <- credit_shuffle[(train_num+1):nrow(credit_shuffle),]

#6. C5.0패키지와 훈련 데이터를 이용해서 모델을 생성한다.
library(C50)
credit_model <- C5.0(credit_train[,-17],credit_train[,17])

#7. 모델과 테스트 데이터로 결과를 예측한다.
credit_result <- predict(credit_model,credit_test[,-17])

# 이원 교차표로 결과를 살펴본다.
library(gmodels)
CrossTable(credit_test[,17], credit_result)
summary(credit$amount)
prop.table(table(credit$default))

```

credit_test[, 17]	credit_result		Row Total
	no	yes	
no	66	9	75
	1.179	3.946	0.750
	0.880	0.120	
	0.857	0.391	
	0.660	0.090	
yes	11	14	25
	3.536	11.837	0.250
	0.440	0.560	
	0.143	0.609	
	0.110	0.140	
Column Total	77	23	100
	0.770	0.230	

문제 274. 위의 모델의 정확도를 90% 이상 올리시오.

```

credit<-read.csv("c:\\data\\credit.csv",header = T)

#2. 데이터에 각 컬럼들을 이해한다.
head(credit,4)

# 라벨 : default (yes:대출금 상환 안함, no:대출금 상환)
# checking_balance : 예금계좌 , saving_balance : 적금계좌, amount : 대출금액,

#3. 데이터가 명목형 데이터인지 확인해 본다

```

#4. 데이터를 shuffle 시킨다.

```
set.seed(12341)
```

```
credit_shuffle <- credit[sample(nrow(credit)),]
```

#5. 데이터를 9대1로 나눈다.

```
train_num<-round(0.9*nrow(credit_shuffle),0)
```

```
credit_train <- credit_shuffle[1:train_num,]
```

```
credit_test <- credit_shuffle[(train_num+1):nrow(credit_shuffle),]
```

#6. C5.0패키지와 훈련 데이터를 이용해서 모델을 생성한다.

```
library(C50)
```

```
credit_model <- C5.0(credit_train[,-17],credit_train[,17], trials = 50)
```

#7. 모델과 테스트 데이터로 결과를 예측한다.

```
credit_result <- predict(credit_model,credit_test[,-17])
```

이원 교차표로 결과를 살펴본다.

```
library(gmodels)
```

```
CrossTable(credit_test[,17], credit_result)
```

```
summary(credit$amount)
```

```
prop.table(table(credit$default))
```

credit_test[, 17]	credit_result		Row Total
	no	yes	
no	65	1	66
	8.316	18.509	
	0.985	0.015	0.660
	0.942	0.032	
	0.650	0.010	
yes	4	30	34
	16.142	35.929	
	0.118	0.882	0.340
	0.058	0.968	
	0.040	0.300	
Column Total	69	31	100
	0.690	0.310	

문제 275. Skin_model을 이용해서 고객이 쿠폰반응이 있는 고객인지 없는 고객인지 예측하는 프로그램을 생성하시오.

```
skin_func()
```

성별을 입력하세요 (예 : male, female) :

나이대를 선택하세요 (예 : 20,30,40) :

직업 유무를 입력하세요 (예 : YES,NO) :
 결혼 여부를 입력하세요 (예 : YES,NO) :
 차 소유 여부를 입력하세요 (예 : YES,NO) :

```
skin_func <- function(){

  i_gender <- switch(menu(c('남성','여성'),title='성별을 입력 하세요'),'male','female')
  i_age <- switch(menu(c('20대','30대','40대'),title='나이대를 입력 하세요'),'20','30','40')
  i_job <- switch(menu(c('직장인','백수'),title='직업유무를 입력 하세요'),'YES','NO')
  i_marry <- switch(menu(c('기혼','미혼'),title='결혼여부를 입력 하세요'),'YES','NO')
  i_car <- switch(menu(c('소유','무소유'),title='차 소유 여부를 입력 하세요'),'YES','NO')
  i_cust_data1<-data.frame(gender=i_gender,
                           age =i_age,
                           job=i_job,
                           marry=i_marry,
                           car=i_car)

  library(C50)
  skin <- read.csv("c:\wwdata\wwskin2.csv", header=T)
  skin <- skin[, -1]
  set.seed(11)
  skin_shuffle <- skin[sample(nrow(skin)), ]
  train_num<-round(0.7*nrow(skin_shuffle),0)
  skin_train <- skin_shuffle[1:train_num,]
  skin_test <-skin_shuffle[(train_num+1):nrow(skin_shuffle),]
  skin_model <- C5.0(skin_train[, -6], skin_train$cupon_react , trials = 10)
  skin_result <- predict(skin_model, skin_test[, -6])

  library(gmodels)
  CrossTable( skin_test[, 6], skin_result )
  skin_result2<-predict(skin_model,cust_data1)
  print(paste('예측결과 : ',skin_result2))
}
```

skin_func()

skin_test[, 6]	skin_result		Row Total
	NO	YES	
NO	6	0	6
	1.000	2.000	0.667
	1.000	0.000	
	1.000	0.000	
	0.667	0.000	
YES	0	3	3
	2.000	4.000	0.333
	0.000	1.000	
	0.000	1.000	
	0.000	0.333	
Column Total	6	3	9
	0.667	0.333	

[1] "예측결과 : YES"

문제 276.	<p>(의사결정트리 자동화 스크립트) 아래와 같이 물어보게 하고 위원 교차표가 출력되는 의사 결정 트리 자동화 스크립트를 생성 하시오.</p> <p>조건</p> <ol style="list-style-type: none"> 1. 윈도우 탐색기로 csv 파일선택 2. 삭제할 컬럼번호 선택 3.라벨이 될 컬럼 선택 4. 이원교차표와 의사결정 트리 같이 출력
---------	--

```

decision_func <- function(){
  library(C50)
  library(gmodels)
  library(rattle)
  library(rpart)
  library(RColorBrewer)
  library(FSelector)

  slc <- switch(menu(c('T','F'),title='header 옵션 선택 : '),!0,!1)

  dt <- read.csv(file.choose(), header = slc)
  print(str(dt))

  slc <- switch(menu(c('yes','no'),title='삭제할 컬럼이 존재합니까? : '),!0,!1)

  if(slc){
    stop<-T
    while(stop){
      del_col <- menu(colnames(dt),title='삭제할 컬럼 선택 : ')
      dt<-dt[,-del_col]
      stop <- switch(menu(c('yes','no'),title='삭제할 컬럼이 더 존재 합니까?'),!0,!1)
    }
  }

  lb <- menu(colnames(dt),title = '라벨이 될 컬럼 선택 : ')

  tp <- as.numeric(readline(prompt='train data의 비율 입력 ex.0.9(90%일 경우) : '))

  set.seed(11)
  dt_shuffle <- dt[sample(nrow(dt)), ] # 데이터 셔플

  train_num<-round(tp*nrow(dt_shuffle),0)

  dt_train <- dt_shuffle[1:train_num,]
  dt_test <-dt_shuffle[(train_num+1):nrow(dt_shuffle),]
  dt_model <- C5.0(dt_train[, -lb], dt_train[,lb] , trials = 100)
  dt_result <- predict(dt_model, dt_test[, -lb])

  CrossTable(dt_test[, lb], dt_result ,prop.chisq = F)

```

```
tree1<-rpart( dt_train[,lb]~., data = dt_train[, -lb], method='class',control = rpart.control(minsplit = 3))

graphics.off()
fancyRpartPlot(tree1,type=2,palette=c('Spectral','RdYlGn'),caption = '의사결정나무 그래프')
}

decision_func()
```

규칙기반 분류 (oneR, Riper)

2018년 5월 31일 목요일 오전 9:50

1. 규칙기반 분류란?

" if ~ then .. " 규칙들의 집합을 사용하여 항목들을 분류하는 방법

Ex. 동물을 포유류와 양서류, 조류 등으로 분류할 때, 아래의 규칙으로 분류한다.

규칙 1. 땅으로 걷고 꼬리가 있는 동물은 포유류 --> 곰, 고양이, 개, 코끼리, 돼지, 토끼, 쥐, 코뿔소

규칙 2. 동물이 털이 없다면 포유류가 아니다. --> 물고기, 개구리, 곤충

규칙 3. 그렇지 않으면 동물은 포유류다. --> 박쥐

2. 규칙기반 알고리즘의 종류

- oneR 알고리즘 (p223)

하나의 사실(규칙)만 가지고 간단하게 분류하는 알고리즘. 간단하긴 하지만 오류가 많아진다.

Ex. 심장질환 데이터를 보면 가슴 통증이 있는가에 대한 컬럼이 있는데 가슴 통증 하나만 보고 심장 질환이 있다고 분류하기에는 오류가 많아진다. 식도염, 폐질환도 가슴통증이 있기 때문이다.

- Riper 알고리즘 (p226) - (Jrip)

복수개의 사실(조건)을 가지고 분류하는 알고리즘.

Ex. 하늘을 날고 털이 있다면 그것은 포유류이다.

땅을 걷고 털이 있다면 그것은 포유류이다.

■ 규칙기반 분류 알고리즘 (oneR 실습)

"독버섯과 정상 "

#1. 버섯 데이터를 R로 로드한다.

#2. ? 표시의 데이터를 NA로 변경한다.

#3. "비어있는 데이터를 NA로 변경한다.

```
murshroom<-read.csv("c:\wwdata\ww\murshroom.csv",header = F, stringsAsFactors = F)
```

```
murshroom[which(murshroom$V12=='?'), "V12"] <- NA
```

```
length(murshroom)
```

```
head(murshroom)
```

```
for (i in 1:length(murshroom)){
```

```
  murshroom[which(murshroom[,i]==""), i] <- NA
```

```
}
```

#4 murshroom 전체 데이터를 factor로 변환

```
dim(murshroom)
for(i in 1 : length(murshroom)){
  murshroom[, i] <- factor(murshroom[,i])
}
```

#5 murshroom 데이터를 훈련 데이터와 테스트 데이터로 나눈다 (75% 훈련데이터, 25%는 테스트 데이터)

```
set.seed(123450)
```

```
dim(murshroom)
```

```
train_cnt<- round (0.75*dim(murshroom)[1])
```

```
train_indx<-sample(1:dim(murshroom)[1],train_cnt,replace = F)
```

```
murshroom_train <- murshroom[train_indx,]
```

```
murshroom_test <- murshroom[-train_indx,]
```

#6. 규칙기반 알고리즘인 oneR을 이용해서 독버섯과 일반버섯을 분류하는 모델을 생성한다.

```
install.packages("OneR")
```

```
library(OneR)
```

```
model1<-OneR(V1~.,data=murshroom_train)
```

```
model1
```

```
summary(model1)
```

#7. 위에서 생성한 모델을 가지고 테스트 데이터를 사용해 결과를 확인한다.

```
result1<-predict(model1,murshroom_test[, -1])
```

```
library(gmodels)
```

```
CrossTable(result1,murshroom_test[,1])
```

result1	murshroom_test[, 1]		Row Total
	e	p	
e	887	11	898
	141.837	280.199	
	0.988	0.012	0.672
	1.000	0.024	
	0.664	0.008	
p	0	426	426
	282.831	558.733	
	0.000	1.000	0.319
	0.000	0.949	
	0.000	0.319	
UNSEEN	0	12	12
	7.967	15.739	
	0.000	1.000	0.009
	0.000	0.027	
	0.000	0.009	
Column Total	887	449	1336
	0.664	0.336	

■ 규칙기반 분류 알고리즘 (JRip 실습)

"독버섯과 정상 "


```

#1. 버섯 데이터를 R로 로드한다.
#2. ? 표시의 데이터를 NA로 변경한다.
#3. "비어있는" 데이터를 NA로 변경한다.
murshroom<-read.csv("c:\wwdata\ww\murshroom.csv",header = F, stringsAsFactors = F)
murshroom[which(murshroom$V12=="?"), "V12"] <- NA
length(murshroom)
head(murshroom)

for (i in 1:length(murshroom)){
  murshroom[which(murshroom[,i]==""), i] <- NA
}

#4 murshroom 전체 데이터를 factor로 변환
dim(murshroom)
for(i in 1 : length(murshroom)){
  murshroom[, i] <- factor(murshroom[,i])
}

#5 murshroom 데이터를 훈련 데이터와 테스트 데이터로 나눈다 (75% 훈련데이터, 25%는 테스트 데이터)
set.seed(123450)
dim(murshroom)

train_cnt<- round (0.75*dim(murshroom)[1])
train_idx<-sample(1:dim(murshroom)[1],train_cnt,replace = F)

murshroom_train <- murshroom[train_idx,]
murshroom_test <- murshroom[-train_idx,]

#6. 규칙기반 알고리즘인 oneR을 이용해서 독버섯과 일반버섯을 분류하는 모델을 생성한다.
install.packages("OneR")
library(OneR)

model1<-OneR(V1~.,data=murshroom_train)
model1
summary(model1)

#7. 위에서 생성한 모델을 가지고 테스트 데이터를 사용해 결과를 확인한다.
result1<-predict(model1,murshroom_test[,-1])
library(gmodels)
CrossTable(result1,murshroom_test[,1])

```

result1	murshroom_test[, 1]		
	e	p	Row Total
e	887	11	898
	141.837	280.199	
	0.988	0.012	0.672
	1.000	0.024	
	0.664	0.008	
p	0	426	426
	282.831	558.733	
	0.000	1.000	0.319
	0.000	0.949	
	0.000	0.319	
UNSEEN	0	12	12
	7.967	15.739	
	0.000	1.000	0.009
	0.000	0.027	
	0.000	0.009	
Column Total	887	449	1336
	0.664	0.336	

문제 277. 다시 배포한 mushrooms.csv 데이터를 가지고 아래의 3가지 알고리즘 별로 이원교차표를 각각 출력 하시오.

- 1.나이브베이지
- 2.OneR
- 3.Jrip

```

mushroom<-read.csv("c:\ww\data\ww\mushrooms.csv",header = F, stringsAsFactors = T)
mushroom[which(mushroom$V12=="?"), 12] <- NA

for (i in (15:23) ) {
  mushroom[which(mushroom[, i] ==""), i] <- NA
}

for (i in (15:23) ) {
  mushroom[which(mushroom[, i] ==""), i] <- NA
}

set.seed(11)
dim(mushroom)

train_cnt <- round( 0.75*dim(mushroom)[1])
train_index <- sample(1:dim(mushroom)[1], train_cnt,replace=F)

mushroom_train <- mushroom[train_index, ]
mushroom_test <- mushroom[-train_index, ]

#나이브 베이지
library(e1071)
model0<-naiveBayes(V1~. , data = mushroom_train, laplace = 0.000001) #라플라스 사용 o
result0<-predict(model0,mushroom_test[, -1])

CrossTable(mushroom_test[,1],result0)

```

```
#OneR
library(OneR)
model1 <- OneR(V1~., data=mushroom_train)
result1 <- predict(model1, mushroom_test[, -1])
library(gmodels)
```

```
CrossTable( mushroom_test[, 1], result1)
```

```
#JRip
```

```
library(RWeka)
model2 <- JRip(V1~., data=mushroom_train)
result2 <- predict(model2, mushroom_test[, -1])
```

```
CrossTable( mushroom_test[, 1], result2)
```

문제 278. 규칙기반 알고리즘을 활용하는 자동화 스크립트 함수를 생성 하시오.

```
oneR_fun() & Jrip_fun()
- 윈도우 탐색기 열리면서 csv 파일 선택
- 라벨이 될 컬럼 선택
- 결과 : 이원 교차표
```

```
rule_based<-function(){
  setwd('c:\ww\data')
  library(RWeka)
  library(OneR)
  library(gmodels)
  print('규칙기반 알고리즘을 이용할 데이터를 선택해주세요')
  files<-choose.files()
  files<-switch(menu(c('네','아니오'),title='파일에 컬럼명이 포함되어 있습니까?'),
    ,files<-read.csv(files,header=T),
    files<-read.csv(files,header=F))

  print('파일을 엽니다.')
  View(files)
  print('파일 정보입니다.')
  str(files)
  delete<-0
  while(TRUE){
    delete<-menu(c('없음',colnames(files)),title='삭제할 컬럼번호를 입력하세요')
    if(delete==1) break
    files<-files[-(delete+1)]
  }
  label_num<-menu(colnames(files),title='라벨이 될 컬럼을 선택하세요')
  files<-data.frame(label=files[,label_num],files[-label_num])
  a<-readline(prompt='train data의 비율을 어떻게 하겠습니까? ex.0.9(90%일 경우)')
  a<-as.numeric(a)
```

```

set.seed(123450)
train_cnt<-round(a * dim(files)[1])
train_indx<-sample(1:dim(files)[1],train_cnt, replace=F)
data_train<-files[train_indx,]
data_test<-files[-train_indx,]

model<-switch(menu(c('oneR','JRip'),title='어떤 알고리즘을 사용하여 진행하시겠습니까?'),
               OneR(label~.,data=data_train),JRip(label~.,data=data_train))
result<-predict(model,data_test[-1])
CrossTable(data_test[,1],result,prop.chisq = F)
}
rule_based()

```

회귀분석

2018년 5월 31일 목요일 오후 2:22

0.목차

1. 단순 선형회귀분석 이론
2. 단순 선형회귀분석 실습1 (탄닌 함유량과 애플레 성장)
3. 단순 선형회귀분석 실습2 (우주 왕복선 챌린저 폭발 원인 분석)
4. 단순 선형회귀분석 실습3 (코스피 지수 수익율과 삼성, 현대 자동차 주식 수익율의 상관관계 분석)
5. 다중 선형회귀 이론
6. 다중 선형회귀분석 실습1 (스마트폰 만족도에 미치는 영향도 분석)
7. 다중 선형회귀분석 실습2 (미국 대학 입학에 가장 영향이 높은 과목 분석)
8. 다중 선형회귀분석 실습3 (보험회사의 보험료 인상에 미치는 요소분석)

1. 회귀

유전학자 프린시스 골턴이 유전의 법칙을 연구하다가 나온 명칭이다. 아버지의 키가 아무리 크다고 할지라도 아들의 키는 아들 세대의 평균으로 접근하는 경향이 있다는 것을 발견했다. (평균으로의 회귀)

- a. 담배 판매량이 변하면(독립변수), 폐암환자수(종속변수)가 어떻게 변하는가?
- b. 공장의 기계(독립변수)를 바꾸면, 생산량(종속변수)은 어떻게 변하는가?

회귀는 독립변수와 종속변수 둘 사이에 상관관계가 있다는 것에서 끝나는게 아니라 어떤 관계인지 까지 자세히 살펴보는 것이다.

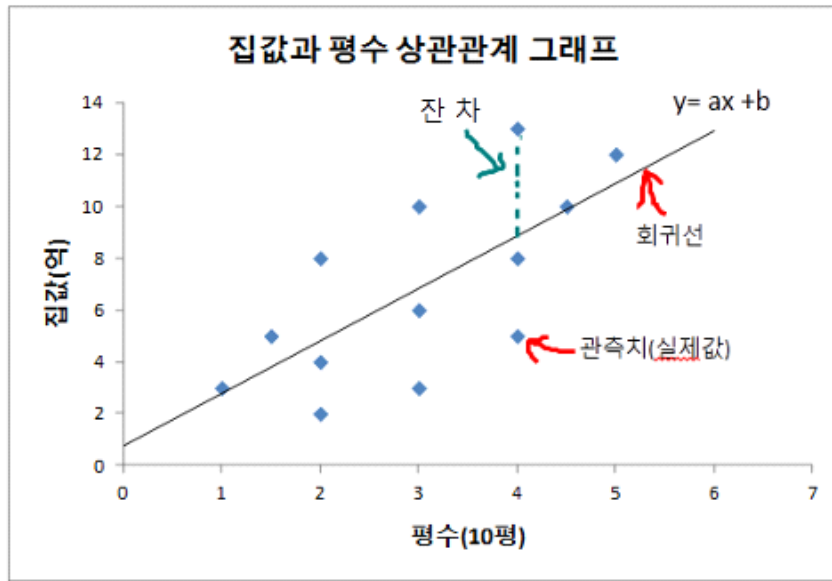
■ 회귀분석이란?

한개 또는 여러개의 독립변수들과 종속변수 사이의 관계를 수학적인 모형을 이용하여 설명, 예측 하는 것

■ 회귀분석 전제 조건 2가지

1. 종속변수 값들이 정규분포에 해당되어야 한다. Ex) 집값
2. 독립변수들이 여러 개인 경우 독립변수들 간에 서로 영향을 주는 성향이 존재 하지 않아야 함. (다중공선성)
Ex. 집값 (종속변수) , 평수 - 역세권 - 상권 - 학군 (독립변수) --> 학군이 좋다고 평수가 넓진 않아야 함

■ 회귀직선을 도출하는 과정



잔차 : 관측치 - 회귀직선값 (-가 될수도 있어서 제공)

최소자승법 : 잔차의 제곱의 합이 최소가 되도록 하는 최적의 직선식을 구하는 방법

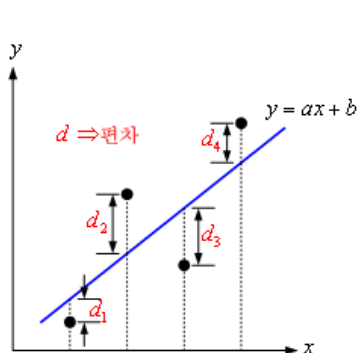
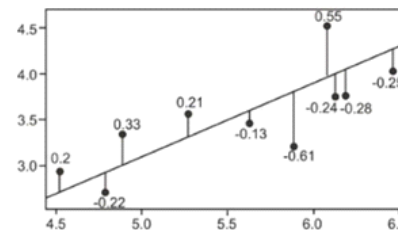
LINEAR REGRESSION

● 최소자승법

Least-squares method

관측점들과 회귀선간에 수직선을 그리고 그 거리를 각각 제공하여 더한 값
각 관측값에서 추정된 직선까지의 거리의 제곱합이 최소가 되도록 회귀계수를 구하는 것이다.

- 최소자승회귀선, 최소제곱직선, 회귀선이라고도 한다.
- 잔차 : 관측값과 예측값의 차이로, 관측값과 회귀선의 수직거리를 의미한다.



$$S = \sum_{i=1}^n d_i^2 = \sum_{i=1}^n (y_i - (ax_i + b))^2$$

↓ 최소화

$$a = \frac{n \sum_{i=1}^n x_i y_i - \left(\sum_{i=1}^n x_i \right) \left(\sum_{i=1}^n y_i \right)}{\sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2}$$

$$b = \frac{\sum_{i=1}^n y_i - a \sum_{i=1}^n x_i}{n}$$

2. 단순 선형회귀분석 이론

* 회귀분석은 하나의 변수가 나머지 다른 변수들과의 선형적 관계를 갖는가의 여부를 분석하는 방법
즉, 하나의 종속변수(예측하고자 하는 값)와 독립변수 사이의 관계를 명시하는 것

회귀분석은 **numeric** 타입을 다룬다. (명목형은 더미코딩 해야함)

- 독립 변수 : 종속변수에 영향을 주는 변수
Ex. 집가격에 영향을 주는 변수 (역세권, 지역상권, 학군, 평수)
- 종속 변수 : 서로 관계를 가지고 있는 변수들 중에서 다른 변수에 영향을 받는 변수
Ex. 집값

■ 단순선형회귀분석 과정

1. 회귀식의 추정

두 변수 X와 Y의 관계에 적합한 회귀식을 구하기 위해서 관측된 값으로 부터 회귀계수 B0과 B1 값을 추정해야 한다.
이 때 일반적으로 많이 사용하는 방법을 최소제곱법이라고 한다.

<단순선형회귀분석>

$$y = \beta_0 + \beta_1 x + \varepsilon_i$$

R에서는 lm(Y~X) 함수를 이용하면 회귀식을 쉽게 추정할 수 있다.

```
> attach(df)
> lm(overall~rides)

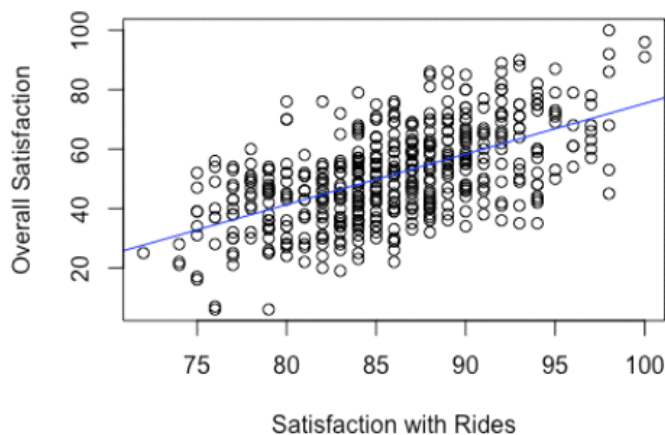
Call:
lm(formula = overall ~ rides)

Coefficients:
(Intercept)      rides
    -94.962         1.703
```

B0 = -94.962, B1 = 1.703 으로 부터 overall = -94.962 + 1.703*B1 라는 회귀식을 구할 수 있으며, 놀이기구에 대한 만족도 (rides)가 1증가할 때마다 전체만족도 (overall)이 1.703만큼 증가한다고 볼 수 있다.

위의 회귀식을 산점도 위에 회귀직선으로 그려보면

```
> m1 <- lm(overall~rides)
> plot(overall~rides, xlab="Satisfaction with Rides", ylab="Overall Satisfaction")
> abline(m1, col='blue')
```



2. 회귀모형의 검정 및 적합도 파악

위의 회귀식이 통계적으로 유의한지, 변수가 유의하게 영향을 미치는지, 그리고 얼마만큼의 설명력을 가지는지 등의 여부를 확인해야 한다.

----> summary 함수를 통해 확인 할 수 있다.

```
> summary(m1)

Call:
lm(formula = overall ~ rides)

Residuals:
    Min       1Q   Median       3Q      Max
-33.597 -10.048   0.425   8.694  34.699

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -94.9622     9.0790  -10.46  <2e-16 ***
rides         1.7033     0.1055   16.14  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12.88 on 498 degrees of freedom
Multiple R-squared:  0.3434,    Adjusted R-squared:  0.3421
F-statistic: 260.4 on 1 and 498 DF,  p-value: < 2.2e-16
```

F-statistic : 도출된 회귀식이 회귀분석 모델 전체에 대해 통계적으로 의미가 있는지 파악
0.05 보다 작으면 이 회귀식은 회귀분석 모델 전체에 대해 통계적으로 의미가 있다고 볼 수 있다.

p-value : 각 변수가 종속변수에 미치는 영향이 유의한지 파악

수정된 R 제곱 : 회귀직선에 의하여 설명되는 변동이 총 변동 중에서 차지하고 있는 상대적인 비율이 얼마인지 나타냄. 즉, 회귀직선이 종속변수의 몇 %를 설명할 수 있는지 확인

실습 1. 단순 선형회귀분석 실습1 (탄닌 함유량과 애벌레 성장)

탄닌은 애벌레 몸에 좋은데 맛이 없어서 애벌레들이 잘 먹지 않음

----> 사료회사에서 필요한 회귀분석

분석목표 : 사료회사에서 애벌레 입맛에도 맞고 몸에도 좋은 적당한 탄닌의 양이 어떻게 되는지 그 회귀직선을 알고자 한다.

1. 데이터를 R로 로드한다.
2. 데이터를 가지고 plot 그래프를 그린다. (x축 : 탄닌 함유량, y축 : 애벌레 성장율)
3. 데이터에 맞는 회귀 모델을 만든다.
4. 회귀 모델로 직선을 그린다.

#1. 데이터를 R로 로드한다.

```
reg<-read.table("c:\data\wwregression.txt",header = T)
```


#2. 데이터를 가지고 plot 그래프를 그린다. (x축 : 탄닌 함유량, y축 : 애벌레 성장율)

```
attach(reg)
```

```
plot(growth~tannin,data=reg,pch=21,col=blues9,bg='red')
```

#3. 데이터에 맞는 회귀 모델을 만든다.

```
install.packages("stats")
```

```
library(stats)
```

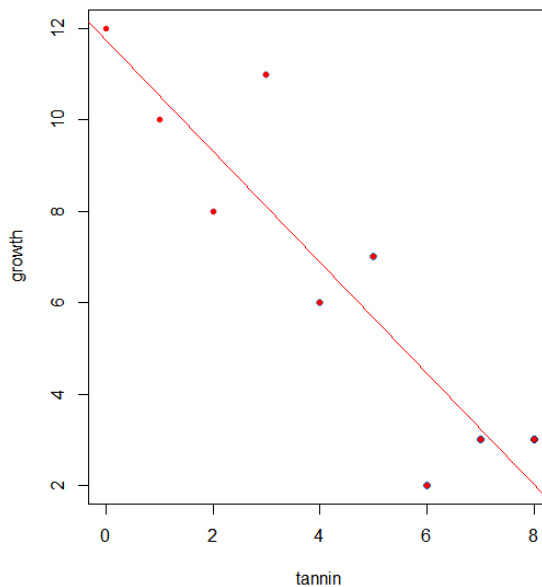
```
model<-lm(growth~tannin)
```

```
model      # 절편, 기울기 출력
```

#4. 회귀 모델로 직선을 그린다.

```
abline(model,col='red')
```

#위의 절편의 방정식 $y = -1.217x + 11.756$



문제 279. 탄닌 함유량이 200일 때 애벌레 성장률은 어떻게 되는지 R로 구현 하시오.

```
f<- function(x){  
  y<-(-1.217)*x +(11.756)  
  y}  
f(200)
```

문제 280. 위의 plot 그래프를 출력할 때 제목으로 직선의 방정식이 출력되게 하시오.

#위의 절편의 방정식 $y = -1.217x + 11.756$

```
reg<-read.table("c:\\data\\regression.txt",header = T)
```

#2. 데이터를 가지고 plot 그래프를 그린다. (x축 : 탄닌 함유량, y축 : 애벌레 성장율)

```
attach(reg)
```

```
plot(growth~tannin,data=reg,pch=21,col=blues9,bg='red',main=paste('y=',round(model$coefficients[2],3),'x + ',round(model$coefficients[1],3)))
```

#3. 데이터에 맞는 회귀 모델을 만든다.

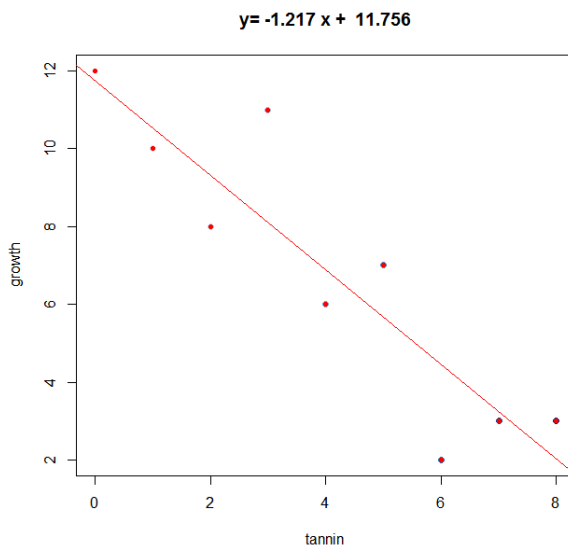
```
#install.packages("stats")
```

```
library(stats)
```

```
model<-lm(growth~tannin)
```

#4. 회귀 모델로 직선을 그린다.

```
abline(model,col='red')
```



실습 2. 단순 선형회귀분석 실습2 (우주 왕복선 챌린저 폭발 원인 분석)

1. 데이터를 R로 로드한다.

```
launch <- read.csv("c:
```

2. 데이터를 가지고 plot 그래프를 그린다.

3. 데이터를 가지고 회귀모델을 만든다.

4. 회귀모델로 회귀 직선을 그린다. (제목까지 포함)

```
launch <- read.csv("c:\\data\\challenger.csv", header = T)
```

```
attach(launch)
```

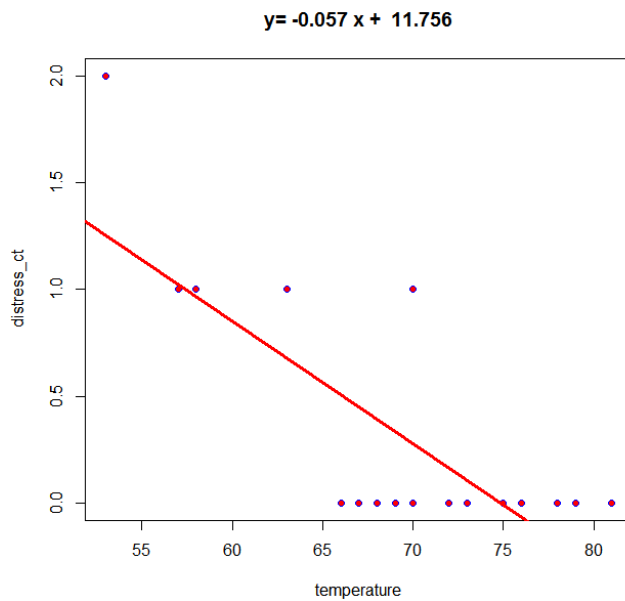
```
library(stats)
```

```
model2<-lm(distress_ct ~ temperature) # y축 x축
```

```
plot(distress_ct ~ temperature, data=launch, pch=21,col='blue',bg='red',
      main=paste('y=',round(model2$coefficients[2],3),'x + ',round(model$coefficients[1],3)))
```

```
abline(model2,col='red',lwd=3)
```

```
detach(launch)
```



문제 281. 회귀 자동화 함수를 생성 하시오.

```
regression_func()
```

- 1.윈도우 탐색기창으로 csv 파일 선택
- 2.독립변수가 될 컬럼 선택
- 3.종속변수가 될 컬럼 선택

```
regression_func <- function(){
  library(stats)

  h <- switch(menu(c('yes','no'),title='header 옵션 : '),!0,!1)
  dt <- read.csv(file.choose(),header = h, stringsAsFactors = T)

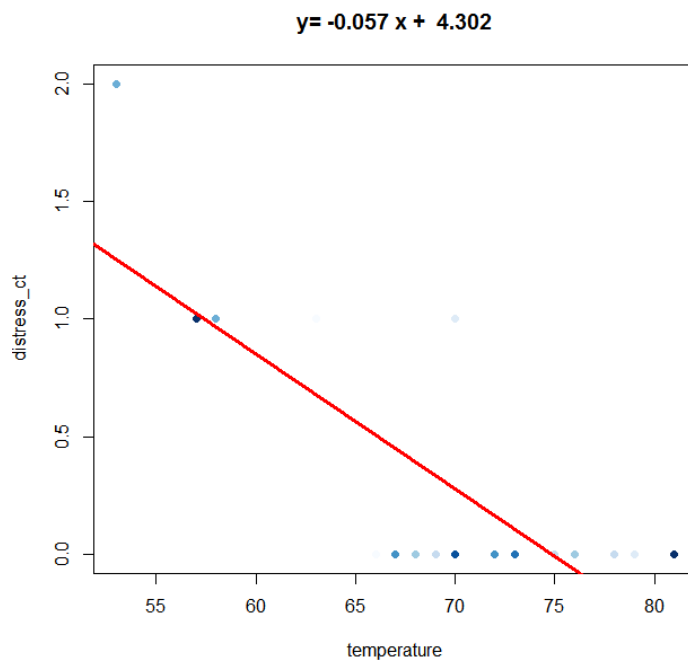
  x <- menu(colnames(dt),title = '독립변수가 될 컬럼 선택(x축): ')
  y <- menu(colnames(dt[-x]),title = '종속변수가 될 컬럼 선택(y축): ')

  model2<-lm(dt[,y] ~ dt[,x]) # y축 x축

  plot(dt[,y] ~ dt[,x], data=dt, pch=21,col=blues9,bg=blues9,
       main=paste('y=',round(model2$coefficients[2],3),'x + ',round(model2$coefficients[1],3)),
       xlab=colnames(dt[x]),ylab=colnames(dt[y]))

  abline(model2,col='red',lwd=3)
}
```

regression_func()



문제 282. 그동안 만들었던 자동화 스크립트를 하나로 통합하는 작업을 수행 하시오.

```
baek_func <- function() {  
  
  # 현재 컴퓨터에 필요한 패키지 설치  
  # 만약 패키지가 존재하지 않는 경우에만 설치  
  
  packages <- c("XML","stringr","rJava","KoNLP","wordcloud","wordcloud2","lubridate")  
  
  if (length(setdiff(packages, rownames(installed.packages()))) > 0) {  
    install.packages(setdiff(packages, rownames(installed.packages())))  
  }  
  
  graphics.off()  
  
  slc<-c('막대그래프','원형그래프','산포도 그래프','워드클라우드','사분함수 그래프','분산과 표준편차',  
        '정규분포와 히스토그램','산포도와 회귀직선','knn 머신러닝','naive bayes 머신러닝',  
        'decision tree','규칙기반 oneR','규칙기반 JRip','단순회귀')  
  
  x1<-menu(slc,title = '그래프 선택',graphics=T)  
  
  h <- switch(menu(c('TRUE','FALSE'),title='header 옵션 : ',graphics=T),!0,!1)  
  
  res0 <- read.csv(file.choose(),header = h, stringsAsFactors = T)  
  
  #res0<- get(readline(prompt = '테이블명 입력 : '))  
  
  barplot_func <-function(){#막대그래프  
    res1<- menu(colnames(res0), title='토탈 값을 구할 컬럼 선택 : ',graphics=T)
```

```

res2<- menu(colnames(res0), title='그룹핑할 컬럼 선택 : ',graphics=T)
q<-tapply(res0[,res1], res0[,res2],sum)
q[is.na(q)] <- 0
barplot(q, col = rainbow(nrow(q)), main = paste( colnames(res0)[res2], '별', colnames(res0)[res1],'총합' ),
beside = T, ylim = c(0,max(q)*1.4))
legend("topright", rownames(q),title = paste(colnames(res0)[res2],' 구분') ,inset = 0,fill =
rainbow(nrow(q)),cex=0.8)
}

pie_func <-function(){#원형 그래프
res1<- menu(colnames(res0), title='토탈 값을 구할 컬럼 선택 : ',graphics=T)
res2<- menu(colnames(res0), title='그룹핑할 컬럼 선택 : ',graphics=T)
q<-tapply(res0[,res1], res0[,res2],sum)
label<-paste(unique(res0[,res2]), round(q/sum(q) * 100,1),'%')
pie(q,col=rainbow(nrow(q)),label=label,main = paste( colnames(res0)[res2], '별',colnames(res0)[res1],'총합' ))
}

plot_func<-function(){#산포도 그래프
x <- menu(colnames(res0), title='x축 컬럼 선택 : ')
y <- menu(colnames(res0), title='y축 컬럼 선택 : ')
plot(res0[,x],res0[,y],pch=16, col=blues9,xlab = colnames(res0)[x] ,ylab = colnames(res0)[y],main =
paste(colnames(res0)[x],'와 ',colnames(res0)[y],'의 상관 관계 '))
}

wordcloud_func<-function(){#워드클라우드
library(wordcloud)
library(KoNLP)
library(plyr)
useSejongDic() # 370957개의 한글 단어가 추가 (전희원 선생님이 만듦)

graphics.off()

res<-readline(prompt = 'c:\wwwdata 경로에 위치한 txt 파일명 입력 : ')
word<-readLines(gsub(' ','',paste('c:\wwwdata\www',res,'.txt'))))

nouns <- extractNoun(word) # 연설문에서 명사만 출력
nouns <- nouns[nchar(nouns)>=2] #두글자 이상인 명사만 추출
cnouns <- count(unlist(nouns)) #단어와 건수 출력

pal <- brewer.pal(6,"Dark2") # Dark2라는 색깔을 추가하는 작업
pal <- pal[:(1)]
windowsFonts(malgun=windowsFont("맑은 고딕")) #맑은 고딕 폰트 추가
wordcloud(words=cnouns$x, freq=cnouns$freq, colors=pal, min.freq=3,
random.order=F, family="malgun")
}

boxplot_func<-function(){#사분위수 그래프
res1<- menu(colnames(res0), title='컬럼 선택 : ')
boxplot(res0[,res1], horizontal = T, col = blues9)
}

```

```

variance_func<-function(){#분산시각화
  res1<- menu(colnames(res0),title='컬럼 선택 : ')

  plot(res0[,res1],main=paste('분산= ',round(var(res0[,res1]),4),'표준편차= ',round(sd(res0[,res1]),4),col='blue')
  abline(h=mean(res0[,res1]),lty=2,col='red')
}

distribution_his_func<-function(){#정규분포&히스토그램
  library(fBasics)
  res1<-menu(colnames(res0),title='컬럼 선택 : ')

  x<-sort(res0[,res1])

  hist(x,col=blues9, axes = F, ann=F)
  par(new=T)
  plot(x,dnorm(x,mean=mean(x),sd=sd(x)),type='l', lwd=3, col='red',main=paste('왜도값 :',round(skewness(x),4)))
}

sanpodo_func<-function(){#산포도그래프&회귀직선
  res2<-menu(colnames(res0),title = 'x축 데이터 입력 : ',graphics=T)
  res3<-menu(colnames(res0),title = 'y축 데이터 입력 : ',graphics=T)

  graphics.off()
  model<-lm(res0[,res3]~res0[,res2])
  plot(res0[,res2],res0[,res3])
  abline(model,col="red")
}

knn_func<-function(){#knn_머신러닝
  library(caret)
  library(e1071)
  library(gmodels)
  library(class)
  #x <-readline("분석할 csv 파일명을 입력하세요~ ")
  y <-menu(colnames(res0),title = '라벨로 지정할 컬럼 선택 : ',graphics=T)
  y<-colnames(res0[y])

  #입력값받는 함수
  #k_n<-readline("k값을 입력하시오~ ")

  wbcd <- na.omit(res0)

  # set.seed(26)
  # wbcd <- wbcd[sample(nrow(wbcd)), ]

  normalize<-function(x) {
    return( (x-min(x))/ ( max(x)-min(x)))
  }

```

```

wbcd <- wbcd[-1]
ncol1 <- which(colnames(wbcd)==y)

wbcd_n <- as.data.frame(lapply(wbcd[, -ncol1], normalize) )

mm<-round(nrow(wbcd_n)*9/10)

wbcd_train <- wbcd_n[1:mm, ]
wbcd_test  <- wbcd_n[(mm+1):nrow(wbcd_n), ]

wbcd_train_label <- wbcd[1:mm,y]
wbcd_test_label  <- wbcd[(mm+1):nrow(wbcd_n),y]

repeats = 3
numbers = 10
tunel = 10

set.seed(1234)

x = trainControl(method = "repeatedcv",
                  number = numbers,
                  repeats = repeats,
                  classProbs = TRUE,
                  summaryFunction = twoClassSummary)

model1 <- train( wbcd_train_label~. , data = data.frame(wbcd_train,wbcd_train_label), method = "knn",
                preProcess = c("center","scale"),
                trControl = x,
                metric = "ROC",
                tuneLength = tunel)

k_n<-model1$bestTune

result1 <- knn(train=wbcd_train, test=wbcd_test,
               cl= wbcd_train_label, k = k_n )
# prop.table( table(ifelse(wbcd[(mm+1):nrow(wbcd_n),y]==result1,"o","x" )))
CrossTable( x= wbcd_test_label, y= result1,prop.chisq=FALSE)

}

naive_bayes_fun<-function(){
  library(gmodels)
  library(e1071)
  dt<-res0
  res1 <- menu(colnames(dt), title='라벨이 될 컬럼번호 선택 : ',graphics=T)
  lplc <- as.numeric(readline(prompt = '라플라스 값 입력 (사용하지 않으려면 0 입력, ex.0.001): '))

  for (i in 1:length(dt)){
    dt[which(dt[,i]=='|' |dt[,i]=='?'), i] <- NA
  }
}

```

```

dt<-na.omit(dt) # na값 생략 ( ? or '')

for(i in 1 : length(dt)){
  dt[, i] <- factor(dt[,i])
}

set.seed(123450)
train_cnt<- round (0.75*nrow(dt))
train_indx<-sample(1:nrow(dt),train_cnt,replace = F)

dt_train <- dt[train_indx,]
dt_test <- dt[-train_indx,]

model2<-naiveBayes(dt_train[,res1]~. , data = dt_train, laplace = lplc) #라플라스 사용 o 0.1~~~0.000001
result2<-predict(model2,dt_test[, -1])

CrossTable(dt_test[,1],result2)
}

decision_func <- function(){
  library(C50)
  library(gmodels)
  library(rattle)
  library(rpart)
  library(RColorBrewer)
  library(FSelector)

  dt <- res0

  slc <- switch(menu(c('yes','no'),title='삭제할 컬럼이 존재합니까? : ',graphics=T),!0,!1)

  if(slc){
    stop<-T
    while(stop){
      del_col <- menu(colnames(dt),title='삭제할 컬럼 선택 : ',graphics=T)
      dt<-dt[,-del_col]
      stop <- switch(menu(c('yes','no'),title='삭제할 컬럼이 더 존재 합니까?',graphics=T),!0,!1)
    }
  }

  lb <- menu(colnames(dt),title = '라벨이 될 컬럼 선택 : ',graphics=T)
  tp <- as.numeric(readline(prompt='train data의 비율 입력 ex.0.9(90%일 경우) : '))

  set.seed(11)
  dt_shuffle <- dt[sample(nrow(dt)), ] # 데이터 셔플

  train_num<-round(tp*nrow(dt_shuffle),0)

  dt_train <- dt_shuffle[1:train_num,]
  dt_test <-dt_shuffle[(train_num+1):nrow(dt_shuffle),]

```



```

dt_model <- C5.0(dt_train[, -lb], dt_train[, lb] , trials = 100)
print(dt_model)
dt_result <- predict(dt_model, dt_test[, -lb])

CrossTable(dt_test[, lb], dt_result ,prop.chisq = F)
tree1 <- rpart( dt_train[, lb] ~., data = dt_train[, -lb], method='class', control = rpart.control(minsplit = 3))

graphics.off()
fancyRpartPlot(tree1, type=2, palette=c('Spectral', 'RdYlGn'), caption = '의사결정나무 그래프')

}

rule_based <- function(){
  library(RWeka)
  library(OneR)
  library(gmodels)
  files <- res0

  delete <- 0
  while(TRUE){
    delete <- menu(c('없음', colnames(files)), title='삭제할 컬럼번호를 입력하세요', graphics=T)
    if(delete == 1) break
    files <- files[-(delete+1)]
  }
  label_num <- menu(colnames(files), title='라벨이 될 컬럼을 선택하세요', graphics=T)
  files <- data.frame(label=files[, label_num], files[-label_num])
  a <- readline(prompt='train data의 비율을 어떻게 하겠습니까? ex.0.9(90%일 경우)')
  a <- as.numeric(a)

  set.seed(123450)
  train_cnt <- round(a * dim(files)[1])
  train_idx <- sample(1:dim(files)[1], train_cnt, replace=F)
  data_train <- files[train_idx,]
  data_test <- files[-train_idx,]

  if (x1 == 12)
    model <- OneR(label ~., data=data_train)
  else if (x1 == 13)
    model <- JRip(label ~., data=data_train)

  result <- predict(model, data_test[-1])
  CrossTable(data_test[, 1], result, prop.chisq = F)

}

regression_func <- function(){
  library(stats)

  dt <- res0

```

```

x <- menu(colnames(dt),title = '독립변수가 될 컬럼 선택(x축): ')
y <- menu(colnames(dt[, -x]),title = '종속변수가 될 컬럼 선택(y축): ')

model2<-lm(dt[,y] ~ dt[,x]) # y축 x축

plot(dt[,y] ~ dt[,x], data=dt, pch=21,col=blues9,bg=blues9,
      main=paste('y=',round(model2$coefficients[2],3),'x + ',round(model2$coefficients[1],3)),
      xlab=colnames(dt[x]),ylab=colnames(dt[y]))

abline(model2,col='red',lwd=3)
}

switch(as.numeric(x1),barplot_func(),pie_func(),plot_func(),wordcloud_func(),
      boxplot_func(),variance_func(),distribution_his_func(),sanpodo_func(),
      knn_func(),naive_bayes_func(),decision_func(),rule_based(),rule_based(),regression_func())
}

baek_func()

```

실습 3. 코스피 지수 수익율과 삼성, 현대 자동차 주식 수익율 실습

1. 데이터를 R로 로드한다.

```

k_index.csv  코스피 지수
S_stock.csv   삼성 수익율
H_stock.csv   현대 자동차 수익율

```

데이터를 가지고 plot 그래프를 그린다.

데이터를 가지고 회귀모델을 만든다.

회귀모델로 회귀 직선을 그린다. (제목까지 포함)

2. 코스피 지수 데이터와 삼성 수익율 데이터를 merge 한다.

3. X축을 코스피 지수로하고, y축을 삼성수익율로 해서 plot 그래프를 그린다.

4. 회귀 함수를 이용해서 회귀 직선을 그린다.

```

k_index <- read.csv("c:\\data\\K_index.csv",header = T,stringsAsFactors = F)
s_stock <- read.csv("c:\\data\\S_stock.csv",header = T,stringsAsFactors = F)
h_stock <- read.csv("c:\\data\\H_stock.csv",header = T, stringsAsFactors = F)

```

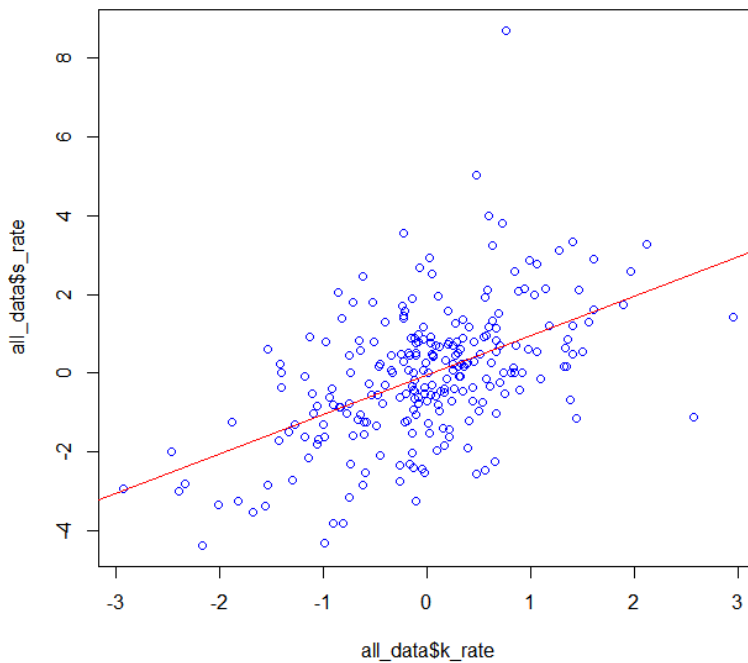
```
all_data <- merge(merge(k_index,s_stock),h_stock)
```

```
head(all_data)
```

```
plot(all_data$k_rate, all_data$s_rate, col="blue") # y: 코스피 등락비율, x : 삼성 수익율 등락비율
```

```
library(stats)
```

```
model_s <- lm(all_data$s_rate ~ all_data$k_rate, data=all_data) # 종속변수 ~ 독립변수 간의 기울기
abline(model_s, col="red")
```



설명 : 회귀 모수중 기울기 1보다 크면 공격적 주식이고 1보다 작으면 방어적 주식이다.

기울기가 낮을수록 시장과 덜 받는 것이고 기울기가 높을수록 탄력적으로 움직이는 것
수익률이 낮더라도 안전하게 벌고 싶다면 기울기가 낮은 것을 선택.
기울기가 높은 것 = (high return high risk)

문제 282. 현대 자동차의 회귀 직선 그래프와 회귀식을 구하시오.

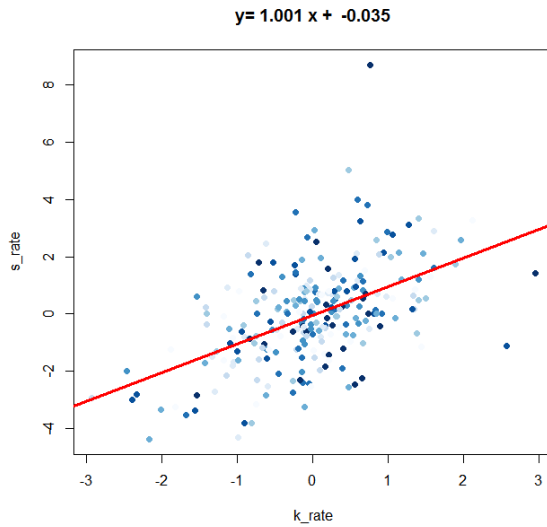
종합 함수 이용해서 출력함.

상관관계수의 기호는 "r"을 사용한다.

상관계수의 수치가 0에 가까울 수록 상관관계가 약하다는 뜻 이고 +1에 가까우면 양의 상관관계가 강하다. -1 에 가까우면 음의 상관관계가 강하다

EX. 온도와 O형링 파손수 두 변수간의 상관관계가 강한지 약한지를 알아내시오.

문제 283. 코스피 지수 수익율과 삼성전자 수익율의 상관관계를 구하시오.



문제 284. 코스피 지수 수익율과 현대 자동차 수익율의 상관관계를 구하시오.

설명 : 상관계수는 시장과 얼마나 비슷하게 움직이느냐를 찾는것이다. 현업에서 0.65~0.70 보다 큰 것 부터 가치가 있다고 판단하고 투자 분석을 한다.

3. 다중 선형 회귀분석 이론

"독립 변수가 여러 개인 경우의 회귀분석 "

다중선형 회귀분석의 목적이 하나의 독립변수만을 가지고 종속변수를 예측하기 위함이라면 다중회귀 분석의 목적은 여러 개의 독립변수(x)들을 가지고 종속변수(y)를 예측하기 위한 회귀모형을 만드는 것이다.

예: 집값을 종속변수라고 한다면, 집값에 영향을 주는 요소가 단순히 평수(독립변수)만 있는 것이 아니다.

집값 <----- 평수, 교통, 학군, 범죄율,
(종속변수)

단순 선형 회귀식	다중 선형 회귀식
$y = ax + b$	$Y = b + a1*x1 + a2*x2 +$ 평수 .. 학군 ..

문제 285. 스마트폰 만족감(종속변수)에 영향을 미치는 요소들 외관, 편의성, 유용성 중에 가장 영향도가 높은 것이 무엇인지 알아 내시오.

1. 데이터를 로드한다.
2. 회귀분석을 한다.

```
muti_hg <- read.csv("c:\\wwdata\\ww\\multi_hg.csv",header = T)
attach(mutu_hg)
```

```
lm(만족감 ~ 외관 + 편의성 + 유용성, data=mutu_hg)
```

Coefficients:			
(Intercept)	외관	편의성	유흥성
3.5136	0.2694	0.2105	0.1623

문제 286. 미국대학교 입학에 가장 영향을 미치는 요소가 무엇인가?

```
spt<-read.csv("c:\data\WWsports.csv",header = T)

head(spt)
spt<-spt[,-1]

normalize<-function(x) {
  return((x-min(x))/ (max(x)-min(x)))
}

spt<-data.frame(lapply(spt,normalize))

lm(spt$acceptance ~ spt$academic+spt$sports+spt$music)

> lm(spt$acceptance ~ spt$academic+spt$sports+spt$music)

call:
lm(formula = spt$acceptance ~ spt$academic + spt$sports + spt$music)

Coefficients:
(Intercept)  spt$academic  spt$sports  spt$music
    0.06122      0.48964      0.30195      0.11432
```

문제 287. 감기에 가장 영향을 미치는 대기오염이 무엇인지?

```
airpol <- read.csv("c:\data\WWpollution.csv")
loc_code <- read.csv("c:\data\WWarea_code.csv")
cold <- read.csv("c:\data\WWcold.csv")
cold_code <- merge(cold, loc_code )

cold_code <- cold_code[,-1]

cold_code$date <- substr(cold_code$date,1,6)

m <- aggregate(cnt ~ date + district, cold_code, sum)

airpol2 <- merge(airpol, m)

#-- 정규화

normalize <- function(x) {
  return ( (x-min(x)) / (max(x) - min(x)) ) }
```

```
airpol2_n <- as.data.frame(lapply(airpol2[3:9], normalize))
```

```
#-- 회귀분석
```

```
attach(airpol2_n)
```

```
lm(cnt ~ (NO2 + ozone + CO + SO2 + dust+ fine_dust), data=airpol2_n )
```

```
> lm(cnt ~ (NO2 + ozone + CO + SO2 + dust+ fine_dust), data=airpol2_n )
```

```
Call:
```

```
lm(formula = cnt ~ (NO2 + ozone + CO + SO2 + dust + fine_dust),  
    data = airpol2_n)
```

```
Coefficients:
```

```
(Intercept)      NO2      ozone      CO      SO2      dust  fine_dust  
    0.10436    0.19125   -0.01393    0.15594    0.09782    0.11498   -0.12586
```

문제 288. 감기에 가장 영향을 미치는 대기오염이 무엇인지?

1. 보험 데이터를 R로 로드한다.

```
head(insurance)
```

```
종속변수 : expense(보험비용)
```

```
독립변수 : age
```

```
          bmi (비만비용)
```

```
          children (아이명수)
```

2. 독립변수와 종속변수간의 상관관계 분석

```
cor(insurance[,c("age", "bmi", "children", "expense")])
```

설명 : 나이가 많을수록 높은 의료비가 예상된다는 것을 확인해야함

```
insurance<-read.csv('c:\ww\data\ww\insurance.csv',header = T)
```

```
head(insurance)
```

```
normalize <- function(x){
```

```
  return ((x-min(x))/(max(x)-min(x)))
```

```
}
```

```
ins<-data.frame(lapply(insurance[,c(1,3,4,7)],normalize))
```

```
cor(insurance[,c(1,3,4,7)])
```

```
lm(ins$charges~., data = ins)
```

```
> lm(ins$charges~., data = ins)
```

```
Call:
```

```
lm(formula = ins$charges ~ ., data = ins)
```

```
Coefficients:
```

```
(Intercept)      age      bmi  children  
   -6679.8    240.0    332.1    542.9
```

```
> lm(ins$charges~., data = ins)

Call:
lm(formula = ins$charges ~ ., data = ins)

Coefficients:
(Intercept)      age      bmi    children
    -6679.8      240.0     332.1      542.9
```

예를 들어 나이가 1살 증가했을 때 보험료가 얼마나 오르는지를 알고 싶을 때는 정규화를 하지 않는다.

정규화를 하는 경우	정규화를 하지 않는 경우
보험 비용에 가장 영향을 크게 미치는 변수가 무엇인지 확인할 때	나이가 한 살 더 늘어날 때 라든가 부양가족이 한 명 더 늘어날 때의 보험료가 얼마나 인상 되어야 하는지 예측할 때

문제 290. 보험 데이터의 결정계수를 확인하는데 독립변수로 bmi, age, children, sex, smoker, region 을 다 넣고 확인 하시오.

```
insurance<-read.csv('c:\\data\\insurance.csv',header = T)
normalize <- function(x) {
  return ( (x-min(x)) / (max(x) - min(x)) ) }
r<-lm(insurance$charges~., data = insurance)
summary(r)

> summary(r)

Call:
lm(formula = insurance$charges ~ ., data = insurance)

Residuals:
    Min       1Q   Median       3Q      Max
-11304.9  -2848.1   -982.1   1393.9  29992.8

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -11938.5      987.8  -12.086 < 2e-16 ***
age             256.9       11.9   21.587 < 2e-16 ***
sexmale       -131.3      332.9   -0.394 0.693348
bmi             339.2       28.6   11.860 < 2e-16 ***
children       475.5      137.8    3.451 0.000577 ***
smokeryes     23848.5     413.1   57.723 < 2e-16 ***
regionnorthwest -353.0      476.3   -0.741 0.458769
regionsoutheast -1035.0     478.7   -2.162 0.030782 *
regionsouthwest  -960.0     477.9   -2.009 0.044765 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6062 on 1329 degrees of freedom
Multiple R-squared:  0.7509,    Adjusted R-squared:  0.7494
F-statistic: 500.8 on 8 and 1329 DF, p-value: < 2.2e-16
```

문제 291. 흡연자는 비흡연자보다 의료비가 23,837 달러 더 증가하는지를 회귀분석으로 확인 하시오.

```
insurance<-read.csv('c:\\data\\insurance.csv',header = T)

lm(insurance$charges ~., data=insurance)

> lm(insurance$charges ~., data=insurance)

Call:
lm(formula = insurance$charges ~ ., data = insurance)
```

```
> lm(insurance$charges ~., data=insurance)
```

```
Call:
lm(formula = insurance$charges ~ ., data = insurance)
```

```
Coefficients:
(Intercept)          age          sexmale          bmi          children          smokeryes
-11938.5         256.9        -131.3        339.2         475.5        23848.5
regionnorthwest regionsoutheast regionsouthwest
-353.0        -1035.0        -960.1
```

문제 292. 자식 한 명당 의료비가 얼마나 더 증가하는지?

```
> lm(insurance$charges ~., data=insurance)
```

```
Call:
lm(formula = insurance$charges ~ ., data = insurance)
```

```
Coefficients:
(Intercept)          age          sexmale          bmi          children          smokeryes
-11938.5         256.9        -131.3        339.2         475.5        23848.5
regionnorthwest regionsoutheast regionsouthwest
-353.0        -1035.0        -960.1
```

문제 293. 비만지수와 나이에 증가에 따라 의료비가 얼마나 증가하는지 확인 하시오.

나이 1살 증가 당 256 달러, bmi 1 증가 당 256 달러 증가

```
> lm(insurance$charges ~., data=insurance)
```

```
Call:
lm(formula = insurance$charges ~ ., data = insurance)
```

```
Coefficients:
(Intercept)          age          sexmale          bmi          children          smokeryes
-11938.5         256.9        -131.3        339.2         475.5        23848.5
regionnorthwest regionsoutheast regionsouthwest
-353.0        -1035.0        -960.1
```

문제 294. 흡연자는 비흡연자보다 의료비가 23,847 달러 증가한다고 했는데 흡연과 비만이 같이 있는 경우 의료비가 얼마나 증가하는지 확인 하시오.

---> 흡연은 비만과 함께 질병을 더 악화시킨다는 점을 보여준다.

```
insurance<-read.csv('c:\\data\\insurance.csv',header = T)
```

```
head(insurance)
```

```
lm(insurance$charges ~., data=insurance)
```

```
insurance$bmi30 <- ifelse(insurance$bmi >= 30, 1, 0)
```

```
s<-lm(insurance$charges~ insurance$bmi30 + insurance$smoker + insurance$bmi30:smoker , data=insurance)
s
```

```
> s
```

```
Call:
lm(formula = insurance$charges ~ insurance$bmi30 + insurance$smoker +
    insurance$bmi30:smoker, data = insurance)
```

```
Coefficients:
(Intercept)          insurance$bmi30          insurance$smokeryes          insurance$bmi30:smokeryes
7977.0             865.7             13386.2             19329.1
```


문제 295.	단순회귀 함수를 수정하는데 회귀 직선 그래프의 제목에 회귀직선식과 결정계수가 같이 나오게 코드를 수정 하시오.
----------------	---

```
baek_func <- function() {

  # 현재 컴퓨터에 필요한 패키지 설치
  # 만약 패키지가 존재하지 않는 경우에만 설치

  packages <- c("XML","stringr","rJava","KoNLP","wordcloud","wordcloud2","lubridate")

  if (length(setdiff(packages, rownames(installed.packages())) > 0) {
    install.packages(setdiff(packages, rownames(installed.packages())))
  }

  graphics.off()

  slc<-c('막대그래프','원형그래프','산포도 그래프','워드클라우드','사분함수 그래프','분산과 표준편차',
        '정규분포와 히스토그램','산포도와 회귀직선','knn 머신러닝','naive bayes 머신러닝',
        'decision tree','규칙기반 oneR','규칙기반 JRip','단순회귀')

  x1<-menu(slc,title = '그래프 선택',graphics=T)

  h <- switch(menu(c('TRUE','FALSE'),title='header 옵션 : ',graphics=T),!0,!1)

  res0 <- read.csv(file.choose(),header = h, stringsAsFactors = T)

  #res0<- get(readline(prompt = '테이블명 입력 : '))

  barplot_func <-function(){#막대그래프
    res1<- menu(colnames(res0), title='토탈 값을 구할 컬럼 선택 : ',graphics=T)
    res2<- menu(colnames(res0), title='그룹핑할 컬럼 선택 : ',graphics=T)
    q<-tapply(res0[,res1], res0[,res2],sum)
    q[is.na(q)] <- 0
    barplot(q, col = rainbow(nrow(q)), main = paste( colnames(res0)[res2], '별', colnames(res0)[res1],'총합' ), beside =
T, ylim = c(0,max(q)*1.4))
    legend("topright", rownames(q),title = paste(colnames(res0)[res2],' 구분' ),inset = 0,fill = rainbow(nrow(q)),cex=0.8)
  }

  pie_func <-function(){#원형 그래프
    res1<- menu(colnames(res0), title='토탈 값을 구할 컬럼 선택 : ',graphics=T)
    res2<- menu(colnames(res0), title='그룹핑할 컬럼 선택 : ',graphics=T)
    q<-tapply(res0[,res1], res0[,res2],sum)
    label<-paste(unique(res0[,res2]), round(q/sum(q) * 100,1),'%')
    pie(q,col=rainbow(nrow(q)),label=label,main = paste( colnames(res0)[res2], '별',colnames(res0)[res1],'총합' ))
  }

  plot_func<-function(){#산포도 그래프
    x <- menu(colnames(res0), title='x축 컬럼 선택 : ')
    y <- menu(colnames(res0), title='y축 컬럼 선택 : ')
```

```

plot(res0[,x],res0[,y],pch=16, col=blues9,xlab = colnames(res0)[x] ,ylab = colnames(res0)[y],main =
paste(colnames(res0)[x],'와 ',colnames(res0)[y],'의 상관 관계 '))
}

wordcloud_func<-function(){#워드클라우드
library(wordcloud)
library(KoNLP)
library(plyr)
useSejongDic()          # 370957개의 한글 단어가 추가 (전희원 선생님이 만듦)

graphics.off()

res<-readline(prompt = 'c:WWWdata 경로에 위치한 txt 파일명 입력 : ')
word<-readLines(gsub(' ','',paste('c:WWWdataWWW',res,'.txt'))))

nouns <- extractNoun(word)  # 연결문에서 명사만 출력
nouns <- nouns[nchar(nouns)>=2]  #두글자 이상인 명사만 추출
cnouns <- count(unlist(nouns))  #단어와 건수 출력

pal <- brewer.pal(6,"Dark2")  # Dark2라는 색깔을 추가하는 작업
pal <- pal[-(1)]
windowsFonts(malgun=windowsFont("맑은 고딕"))  #맑은 고딕 폰트 추가
wordcloud(words=cnouns$x, freq=cnouns$freq, colors=pal, min.freq=3,
          random.order=F, family="malgun")
}

boxplot_func<-function(){#사분위수 그래프
res1<- menu(colnames(res0), title='컬럼 선택 : ')
boxplot(res0[,res1], horizontal = T, col = blues9)
}

variance_func<-function(){#분산시각화
res1<- menu(colnames(res0),title='컬럼 선택 : ')

plot(res0[,res1],main=paste(' 분산= ',round(var(res0[,res1]),4),'표준편차= ',round(sd(res0[,res1])),4),col='blue')
abline(h=mean(res0[,res1]),lty=2,col='red')
}

distribution_his_func<-function(){#정규분포&히스토그램
library(fBasics)
res1<-menu(colnames(res0),title='컬럼 선택 : ')

x<-sort(res0[,res1])

hist(x,col=blues9, axes = F, ann=F)
par(new=T)
plot(x,dnorm(x,mean=mean(x),sd=sd(x)),type='l', lwd=3, col='red',main=paste('왜도값 :',round(skewness(x),4)))
}

sanpodo_func<-function(){#산포도그래프&회귀직선

```

```

res2<-menu(colnames(res0),title = 'x축 데이터 입력 : ',graphics=T)
res3<-menu(colnames(res0),title = 'y축 데이터 입력 : ',graphics=T)

graphics.off()
model<-lm(res0[,res3]~res0[,res2])
plot(res0[,res2],res0[,res3])
abline(model,col="red")
}

knn_func<-function(){#knn_머신러닝
  library(caret)
  library(e1071)
  library(gmodels)
  library(class)
  #x <-readline("분석할 csv 파일명을 입력하세요~ ")
  y <-menu(colnames(res0),title = '라벨로 지정할 컬럼 선택 : ',graphics=T)
  y<-colnames(res0[y])

  #입력값받는 함수
  #k_n<-readline("k값을 입력하시오~ ")

  wbcd <- na.omit(res0)

  # set.seed(26)
  # wbcd <- wbcd[sample(nrow(wbcd)), ]

  normalize<-function(x) {
    return( (x-min(x))/ ( max(x)-min(x)))
  }

  wbcd <- wbcd[-1]
  ncol1 <- which(colnames(wbcd)==y)

  wbcd_n <- as.data.frame(lapply(wbcd[, -ncol1], normalize) )

  mm<-round(nrow(wbcd_n)*9/10)

  wbcd_train <- wbcd_n[1:mm, ]
  wbcd_test <- wbcd_n[(mm+1):nrow(wbcd_n), ]

  wbcd_train_label <- wbcd[1:mm,y]
  wbcd_test_label <- wbcd[(mm+1):nrow(wbcd_n),y]

  repeats = 3
  numbers = 10
  tunel = 10

  set.seed(1234)

  x = trainControl(method = "repeatedcv",

```

```

        number = numbers,
        repeats = repeats,
        classProbs = TRUE,
        summaryFunction = twoClassSummary)

model1 <- train( wbcd_train_label~. , data = data.frame(wbcd_train,wbcd_train_label), method = "knn",
  preProcess = c("center","scale"),
  trControl = x,
  metric = "ROC",
  tuneLength = tune1)

k_n<-model1$bestTune

result1 <- knn(train=wbcd_train, test=wbcd_test,
  cl= wbcd_train_label, k = k_n )
# prop.table( table(ifelse(wbcd[(mm+1):nrow(wbcd_n),y]==result1,"o","x" )))
CrossTable( x= wbcd_test_label, y= result1,prop.chisq=FALSE)

}

naive_bayes_fun<-function(){
  library(gmodels)
  library(e1071)
  dt<-res0
  res1 <- menu(colnames(dt), title='라벨이 될 컬럼번호 선택 : ',graphics=T)
  lplc <- as.numeric(readline(prompt = '라플라스 값 입력 (사용하지 않으려면 0 입력, ex.0.001): '))

  for (i in 1:length(dt)){
    dt[which(dt[,i]=='|dt[,i]==?'), i] <- NA
  }
  dt<-na.omit(dt) # na값 생략 ( ? or '')

  for(i in 1 : length(dt)){
    dt[, i] <- factor(dt[,i])
  }

  set.seed(123450)
  train_cnt<- round (0.75*nrow(dt))
  train_indx<-sample(1:nrow(dt),train_cnt,replace = F)

  dt_train <- dt[train_indx,]
  dt_test <- dt[-train_indx,]

  model2<-naiveBayes(dt_train[,res1]~. , data = dt_train, laplace = lplc) #라플라스 사용 o 0.1~~~0.000001
  result2<-predict(model2,dt_test[,1])

  CrossTable(dt_test[,1],result2)
}

decision_func <- function(){

```

```

library(C50)
library(gmodels)
library(rattle)
library(rpart)
library(RColorBrewer)
library(FSelector)

dt <- res0

slc <- switch(menu(c('yes','no'),title='삭제할 컬럼이 존재합니까? : ',graphics=T),!0,!1)

if(slc){
  stop<-T
  while(stop){
    del_col <- menu(colnames(dt),title='삭제할 컬럼 선택 : ',graphics=T)
    dt<-dt[,-del_col]
    stop <- switch(menu(c('yes','no'),title='삭제할 컬럼이 더 존재 합니까?',graphics=T),!0,!1)
  }
}

lb <- menu(colnames(dt),title = '라벨이 될 컬럼 선택 : ',graphics=T)
tp <- as.numeric(readline(prompt='train data의 비율 입력 ex.0.9(90%일 경우) : '))

set.seed(11)
dt_shuffle <- dt[sample(nrow(dt)), ] # 데이터 셔플

train_num<-round(tp*nrow(dt_shuffle),0)

dt_train <- dt_shuffle[1:train_num,]
dt_test <-dt_shuffle[(train_num+1):nrow(dt_shuffle),]
dt_model <- C5.0(dt_train[, -lb], dt_train[,lb] , trials = 100)
print(dt_model)
dt_result <- predict(dt_model, dt_test[, -lb])

CrossTable(dt_test[, lb], dt_result ,prop.chisq = F)
tree1<-rpart( dt_train[,lb]~., data = dt_train[, -lb], method='class',control = rpart.control(minsplit = 3))

graphics.off()
fancyRpartPlot(tree1,type=2,palette=c('Spectral','RdYlGn'),caption = '의사결정나무 그래프')
}

rule_based<-function(){
  library(RWeka)
  library(OneR)
  library(gmodels)
  files<-res0

  delete<-0
  while(TRUE){
    delete<-menu(c('없음',colnames(files)),title='삭제할 컬럼번호를 입력하세요',graphics=T)

```

```

    if(delete==1) break
    files<-files[-(delete+1)]
  }
  label_num<-menu(colnames(files),title='라벨이 될 컬럼을 선택하세요',graphics=T)
  files<-data.frame(label=files[,label_num],files[-label_num])
  a<-readline(prompt='train data의 비율을 어떻게 하겠습니까? ex.0.9(90%일 경우)')
  a<-as.numeric(a)

  set.seed(123450)
  train_cnt<-round(a * dim(files)[1])
  train_indx<-sample(1:dim(files)[1],train_cnt, replace=F)
  data_train<-files[train_indx,]
  data_test<-files[-train_indx,]

  if (x1==12)
    model<- OneR(label~.,data=data_train)
  else if(x1==13)
    model<- JRip(label~.,data=data_train)

  result<-predict(model,data_test[-1])
  CrossTable(data_test[,1],result,prop.chisq = F)
}

regression_func <- function(){
  library(stats)

  dt <-res0

  x <- menu(colnames(dt),title = '독립변수가 될 컬럼 선택(x축): ',graphics=T)
  y <- menu(colnames(dt),title = '종속변수가 될 컬럼 선택(y축): ',graphics=T)

  model2<-lm(dt[,y] ~ dt[,x]) # y축 x축

  plot(dt[,y] ~ dt[,x], data=dt, pch=21,col=blues9,bg=blues9,
    main=paste('y=',round(model2$coefficients[2],3),'x + ',round(model2$coefficients[1],3)),
    xlab=colnames(dt[x]),ylab=colnames(dt[y]),sub=paste("결정계수 : ",summary(model2)$r.squared ))
  abline(model2,col='red',lwd=3)
}

switch(as.numeric(x1),barplot_func(),pie_func(),plot_func(),wordcloud_func(),
  boxplot_func(),variance_func(),distribution_his_func(),sanpodo_func(),
  knn_func(),naive_bayes_func(),decision_func(),rule_based(),rule_based(),regression_func())
}

baek_func()

```

문제 296.	우주왕복선 챌린저호의 폭발원인인 o형링 파손에 영향을 주는 독립변수가 온도 말고도 다른 독립변수가 있는지 reg 함수를 이용해서 확인 하시오.
----------------	---

```

launch<-read.csv("c:\\data\\challenger.csv",header = T)
library(stats)

head(launch,4)
attach(launch)
model <- lm(distress_ct ~ temperature+ field_check_pressure +flight_num )

reg(distress_ct,launch[2:4])
reg<- function(y,x){

  x<-as.matrix(x)
  x<-cbind(Intercept = 1, x)
  b<-solve(t(x) %*% x)%*% t(x) %*% y
  colnames(b) <- 'estimate'
  print(b)
}

> reg(distress_ct, launch[2:4])
              estimate
Intercept      3.527093383
temperature    -0.051385940
field_check_pressure 0.001757009
flight_num      0.014292843

```

문제 297. 책 260 페이지 베타햇을 구하는 수학식을 도출하는 과정을 풀이 하시오.

(Theorem) $\frac{d(a^T x)}{dx} = \frac{d(a^T x)}{dx} = a^T$

$$\frac{\partial}{\partial \beta} \left[(x\beta)^T (x\beta) - 2(x\beta)^T Y + Y^T Y \right]$$

$$= \frac{\partial}{\partial \beta} \left[\beta^T x^T x \beta - 2\beta^T x^T Y + Y^T Y \right]$$

$$= \frac{\partial}{\partial \beta} \left[(x^T x \beta)^T + \beta^T x^T x - 2(x^T Y)^T \right] = 0$$
 Transpose all

$$= x^T x \beta + x^T x \beta - 2x^T Y = 0$$

$$= 2x^T x \beta - 2x^T Y = 0$$

$$\Rightarrow x^T x \beta = x^T Y$$

$$\beta = (x^T x)^{-1} x^T Y$$

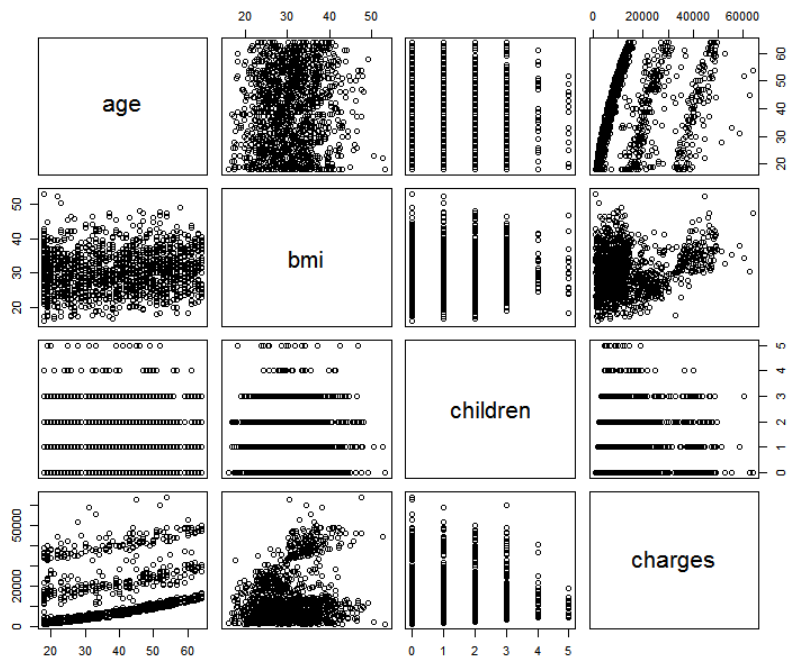
문제 298. 나이가 올라감에 따라서 의료비 지출이 증가한다는 것을 R의 pairs 함수로 확인 하시오.

책 269p

```

insurance <- read.csv("c:\\data\\insurance.csv",header = T)
pairs(insurance[c("age","bmi","children","charges")])

```

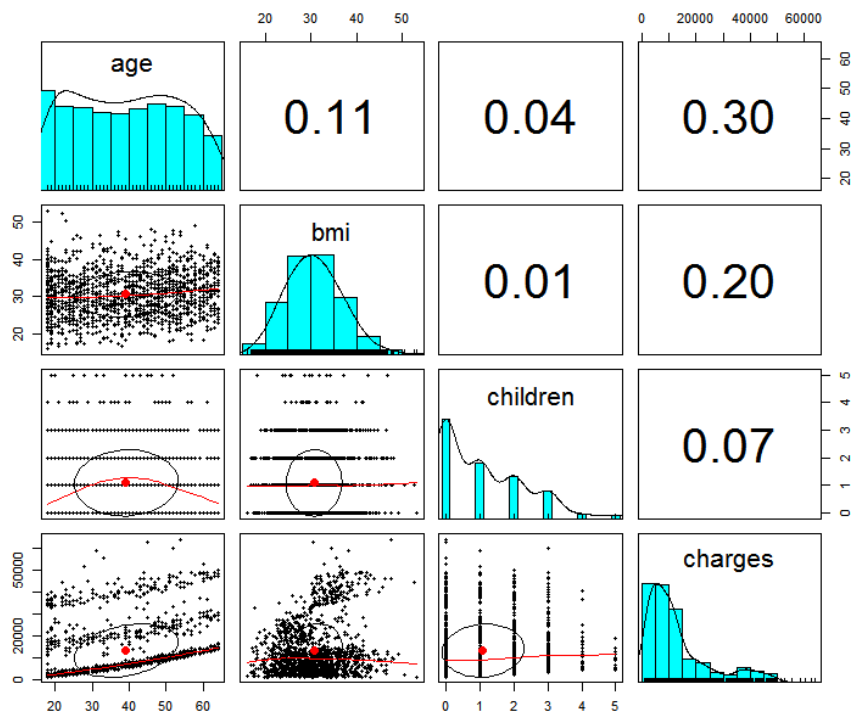


문제 299. 페이지 270 처럼 숫자로 출력되게 하시오.

```
install.packages("psych")
```

```
library(psych)
```

```
pairs.panels(insurance[c("age", "bmi", "children", "charges")])
```



문제 300. Lm 함수를 이용해서 회귀분석을 하는데 종속변수 charges로 두고 독립변수를 age, children, bmi, sex, smoker, region로 두고 회귀분석 하시오.

```
attach(insurance)
```



```
ins_model <- lm(charges ~ age + children + bmi + sex+smoker + region)
ins_model
```

```
> ins_model

Call:
lm(formula = charges ~ age + children + bmi + sex + smoker +
    region)

Coefficients:
    (Intercept)          age        children          bmi          sexmale
      -11938.5         256.9         475.5         339.2        -131.3
    smokeryes regionnorthwest regionsoutheast regionsouthwest
      23848.5        -353.0        -1035.0        -960.1
```

분석결과 :

1. 나이가 일년씩 더해질때마다 평균적으로 256달러의 의료비가 더 들고
2. 자녀 한명이 추가됨으로써 의료비가 475달러 더 들고
3. Bmi가 증가함에 따라 연간 339달러의 의료비가 더 든다.

문제 301. 책 274p 를 보고 남자가 여성에 비해 의료비가 131달러 적게 든다는 것을 성별 더미코딩을 통해 독립변수로 추가해서 알아내시오.

문제 302. 위의 회귀모델의 결정계수를 확인하시오.

```
> summary(ins_model)

Call:
lm(formula = charges ~ age + children + bmi + sex + smoker +
    region)

Residuals:
    Min       1Q   Median       3Q      Max
-11304.9  -2848.1   -982.1   1393.9  29992.8

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -11938.5     987.8  -12.086 < 2e-16 ***
age             256.9       11.9   21.587 < 2e-16 ***
children       475.5       137.8    3.451 0.000577 ***
bmi            339.2       28.6   11.860 < 2e-16 ***
sexmale       -131.3       332.9   -0.394 0.693348
smokeryes     23848.5     413.1   57.723 < 2e-16 ***
regionnorthwest -353.0     476.3   -0.741 0.458769
regionsoutheast -1035.0    478.7   -2.162 0.030782 *
regionsouthwest -960.0    477.9   -2.009 0.044765 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6062 on 1329 degrees of freedom
Multiple R-squared: 0.7509, Adjusted R-squared: 0.7494
F-statistic: 500.8 on 8 and 1329 DF, p-value: < 2.2e-16
```

문제 303. 비선형 연령을 모델에 추가하기위해서 아래의 변수를 생성하고 age2가 회귀모델에 더 유사한 변수인지를 확인 하시오.

```
insurance$age2 <- insurance$age^2
ins_model <- lm(charges ~ age +age2+ children + bmi + sex+smoker + region)
```

```
> summary(ins_model)

Call:
lm(formula = charges ~ age + children + bmi + sex + smoker +
    region)

Residuals:
    Min       1Q   Median       3Q      Max
-11304.9  -2848.1   -982.1   1393.9  29992.8

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -11938.5      987.8  -12.086 < 2e-16 ***
age             256.9       11.9   21.587 < 2e-16 ***
children       475.5       137.8    3.451 0.000577 ***
bmi            339.2       28.6   11.860 < 2e-16 ***
sexmale       -131.3       332.9   -0.394 0.693348
smokeryes     23848.5      413.1   57.723 < 2e-16 ***
regionnorthwest -353.0      476.3   -0.741 0.458769
regionsoutheast -1035.0     478.7   -2.162 0.030782 *
regionsouthwest -960.0      477.9   -2.009 0.044765 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6062 on 1329 degrees of freedom
Multiple R-squared:  0.7509,    Adjusted R-squared:  0.7494
F-statistic: 500.8 on 8 and 1329 DF,  p-value: < 2.2e-16
```

문제 304. Bmi30 이라는 더미변수를 추가해서 bmi가 30이상인 사람들은 더 의료비가 들거라는 예측을 하는 회귀 모델에 결정계수를 확인해서 bmi30이 유사한 독립변수인지 확인 하시오.

```
attach(insurance)
insurance$bmi30 <- ifelse(insurance$bmi >=30 ,1 , 0)
ins_model <- lm(charges ~ age +age2+ children + bmi + sex+smoker + region+bmi30)
```

```
summary(ins_model)
```

```
> summary(ins_model)

Call:
lm(formula = charges ~ age + age2 + children + bmi + sex + smoker +
    region + bmi30)

Residuals:
    Min       1Q   Median       3Q      Max
-12494.2  -3362.2   137.6   1312.0  29321.2

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -2943.176    1828.115   -1.610 0.107646
age           -28.533      80.445   -0.355 0.722875
age2             3.603       1.004    3.590 0.000342 ***
children       630.402    142.366    4.428 1.03e-05 ***
bmi           153.905      46.056    3.342 0.000856 ***
sexmale       -166.295     328.315   -0.507 0.612584
smokeryes     23857.543    407.353   58.567 < 2e-16 ***
regionnorthwest -400.518    469.638   -0.853 0.393911
regionsoutheast -888.533    472.833   -1.879 0.060440 .
regionsouthwest -947.681    471.216   -2.011 0.044513 *
bmi30         2727.552     547.618    4.981 7.17e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5977 on 1327 degrees of freedom
Multiple R-squared:  0.7582,    Adjusted R-squared:  0.7564
F-statistic: 416.2 on 10 and 1327 DF,  p-value: < 2.2e-16
```

설명 : 비만지수 30미만인 사람도다 연간 275달러 더 의료비가 들거라 예상할 수 있다.

문제 305. bmi30*smoker를 추가해서 비만계수가 30 이상이면서 흡연자가 더 의료비는 내는지 확인하고 이 독립변수가 유의한 결정계수로 확인한다.

```
ins_model4 <- lm(charges~age+age2+bmi30+bmi30*smoker+children, data = insurance)
summary(ins_model4)
```

```
> summary(ins_model4)

Call:
lm(formula = charges ~ age + age2 + bmi30 + bmi30 * smoker +
    children, data = insurance)

Residuals:
    Min       1Q   Median       3Q      Max
-18718.8  -1641.8  -1313.8   -846.4   23799.7

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2237.3818   1083.9103    2.064  0.0392 *
age         -24.5114     60.2871   -0.407  0.6844
age2          3.6643     0.7524    4.870 1.25e-06 ***
bmi30         42.3927    276.9065    0.153  0.8783
smokeryes    13389.6765   442.5871   30.253 < 2e-16 ***
children      669.3870    106.6878    6.274 4.74e-10 ***
bmi30:smokeryes 19759.1119   608.6309   32.465 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

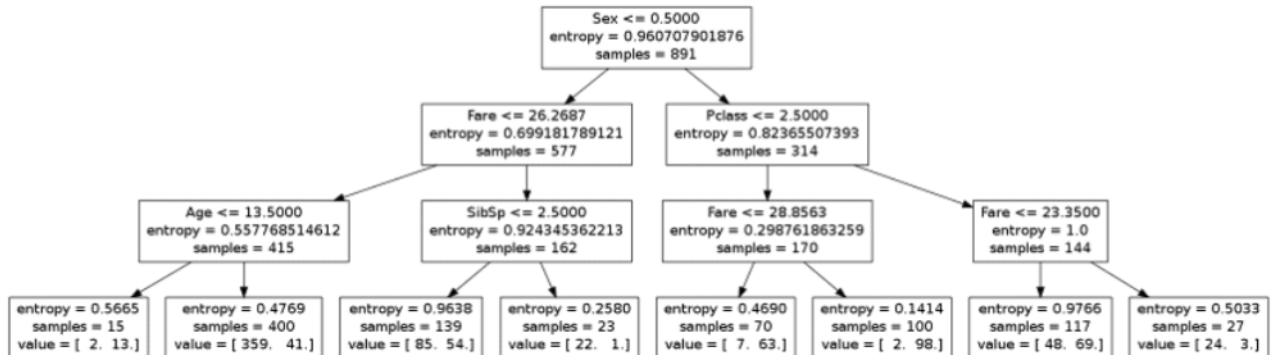
Residual standard error: 4483 on 1331 degrees of freedom
Multiple R-squared:  0.8636,    Adjusted R-squared:  0.8629
F-statistic: 1404 on 6 and 1331 DF,  p-value: < 2.2e-16
```

문제 306. 나이도 많은데, 비만이고 흡연하는 사람이 의료비가 더 드는지 확인 하시오.

```
insurance$age3 <- ifelse(insurance$age >=50 ,1 , 0)
ins_model4 <- lm(charges~age+age2+bmi30+bmi30*smoker*age3+children, data = insurance)
summary(ins_model4)
```

회귀트리

2018년 6월 5일 화요일 오후 3:23



위의 그림은 타이타닉 생존자를 찾는 의사결정트리 모델입니다.

첫번째 뿌리 노드를 보면 성별 ≤ 0.5 라고 되어있는데 이는 남자냐? 여자냐? 라고 질문하는 것과 같습니다.

결국, 모든 승객에 대한 분류(Classification)를 통해 생존확률을 예측할 수 있을 것 입니다.

이처럼, 숫자형 결과를 반환하는 것을 **회귀나무(Regression Tree)**라고 하며,

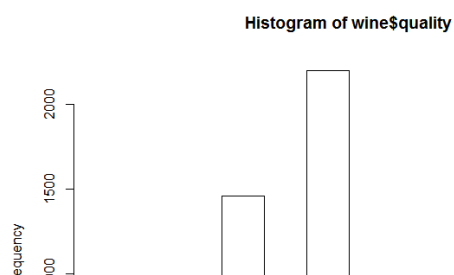
범주형 결과를 반환하는 것을 **분류나무(Classification Tree)**라고 합니다.

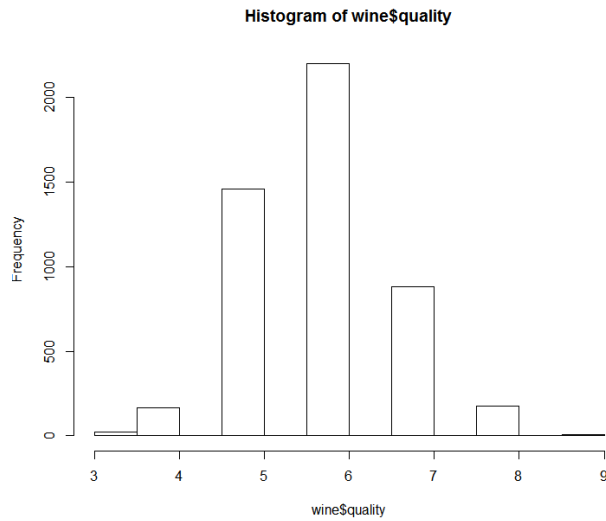
1. 와인 데이터에 대한 소개

#fixed.acidity : 고정 산도
#volatile.acidity : 휘발성 산도
#citric.acid : 시트르산
#residual.sugar : 잔류 설탕
#chlorides : 염화물
#free.sulfur.dioxide : 자유 이산화황
#total.sulfur.dioxide : 총 이산화황
#density : 밀도
#pH : pH
#sulphates : 황산염
#alcohol : 알코올
#quality : 품질

2. 와인 quality 데이터가 정규분포에 속하는 안정적인 데이터인지 확인

hist(wine\$quality)





3. wine 데이터를 train 데이터와 test 데이터로 나눈다.

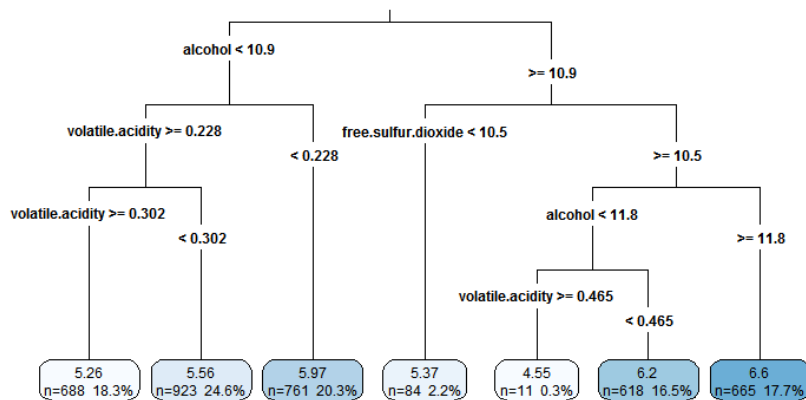
```
wine_train <- wine[1:3750,]
wine_test <- wine[3750:nrow(wine),]
```

4. train 데이터를 가지고 model을 생성한다.

```
library(rpart)
model<- rpart(quality~., data=wine_train)
model
```

5. 위에서 나온 모델로 트리를 시각화 하시오.

```
library(rpart.plot)
rpart.plot(model, digits = 3, fallen.leaves = T, type = 3, extra = 101)
```



6. 위에서 만든 모델로 테스트 데이터의 라벨을 예측 하시오.

```
result<-predict(model, wine_test)
```

7. 테스트 데이터의 실제 라벨(품질)과 예측결과(품질)을 비교한다

```
cbind(round(result),wine_test$quality)
```

8. 테스트 데이터의 라벨과 예측 결과와 상관관계가 어떻게 되는지 확인한다.

```
cor(result,wine_test$quality)
```

```
> cor(result,wine_test$quality)
[1] 0.5369593
```

설명 : 0.53은 두 데이터간의 연관 강도만 측정하는 것이다. 그래서 두 데이터간의 오차율이 어떻게 되는지 확인해서

9. 두 데이터간의 오차율을 확인

```
MAE <- function(actual,predicted){
  mean(abs(actual-predicted))
}
```

```
MAE(result,wine_test$quality)
```

```
> MAE(result,wine_test$quality)
[1] 0.5867792
```

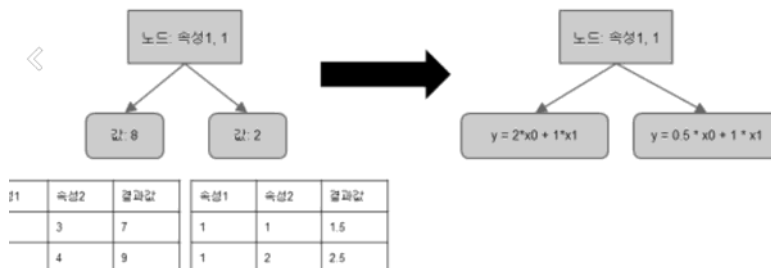
0.58 <--- 이 모델의 경우 다른 모델인 서포트 벡터 머신에서의 오차는 0.45 인데 0.58이면 상대적으로 좀 큰 오차이므로 개선의 여지가 필요하다.

개선방법이 회귀트리 ----> 모델트리로 변경해서 개선을 한다.

모델트리 ?

모델 만들기

- 값 대신 회귀식 사용 → 수치 예측에 사용



트리에서 값대신에 회귀식을 사용해서 수치를 예측한다.

10. 모델트리를 구현하기 위한 패키지를 설치

```
library(RWeka)
```

11. 와인의 품질을 예측하는 모델을 생성한다.

```
model_tree <- M5P(quality ~ ., data=wine_train)
model_tree
```

12. 위에서 만든 모델로 테스트 데이터의 라벨을 예측 하시오.

```
model_result <- predict(model_tree,wine_test)
```

13. 테스트 데이터의 실제 라벨(품질)과 예측결과(품질)을 비교한다.

```
cbind(round(model_result),wine_test$quality)
```

14. 테스트 데이터의 라벨과 예측 결과와 상관관계가 어떻게 되는지 확인한다.

```
cor(model_result,wine_test$quality)
```

```
> cor(model_result,wine_test$quality)
[1] 0.6273025
```

15. 두 데이터간의 오차율을 확인

```
MAE(model_result, wine_test$quality)
```

```
> MAE(model_result, wine_test$quality)
[1] 0.5458398
```

문제 307.

보스턴 하우스 데이터(보스턴 지역의 집값)를 이용해서 회귀트리를 그리시오.
범죄율, 방의 개수, 지역 학교의 교사 숫자, 강과 인접한 거리등의 데이터를 확인해서 회귀트리를 그리시오.

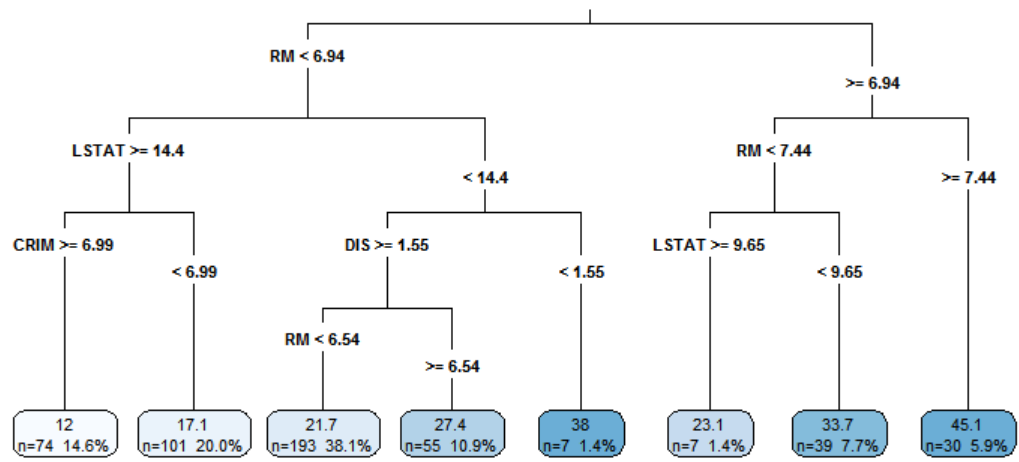
라벨 : MEDV(집값) , cat.MEDV(주택가격이 3만 달러가 넘는지 안넘는지에 대한 라벨)

[01]	CRIM	자치시(to wn) 별 1인당 범죄율
[02]	ZN	25,000 평방피트를 초과하는 거주지역의 비율
[03]	INDUS	비소매상업지역이 점유하고 있는 토지의 비율
[04]	CHAS	찰스강에 대한 더미변수(강의 경계에 위치한 경우는 1, 아니면 0)
[05]	NOX	10ppm 당 농축 일산화질소
[06]	RM	주택 1가구당 평균 방의 개수
[07]	AGE	1940년 이전에 건축된 소유주택의 비율
[08]	DIS	5개의 보스턴 직업센터까지의 접근성 지수
[09]	RAD	방사형 도로까지의 접근성 지수
[10]	TAX	10,000 달러 당 재산세율
[11]	PTRATIO	자치시(to wn)별 학생/교사 비율
[12]	B	$1000(Bk - 0.63)^2$, 여기서 Bk는 자치시별 흑인의 비율을 말함.
[13]	LSTAT	모집단의 하위계층의 비율(%)
[14]	MEDV	본인 소유의 주택가격(종양값) (단위: \$1,000)

```
boston <- read.csv("c:\\data\\boston.csv", header = T)
```

```
model_b <- rpart(MEDV~.,data=boston[,-15])
```

```
rpart.plot(model_b, digits = 3, fallen.leaves = T, type = 3, extra = 101)
```



상관분석

2018년 6월 5일 화요일 오후 7:46

둘 또는 그 이상의 변수들이 서로 관련성을 가지고 변화 할 때, 그 관계를 분석하는데 사용되는 대표적인 분석 방법

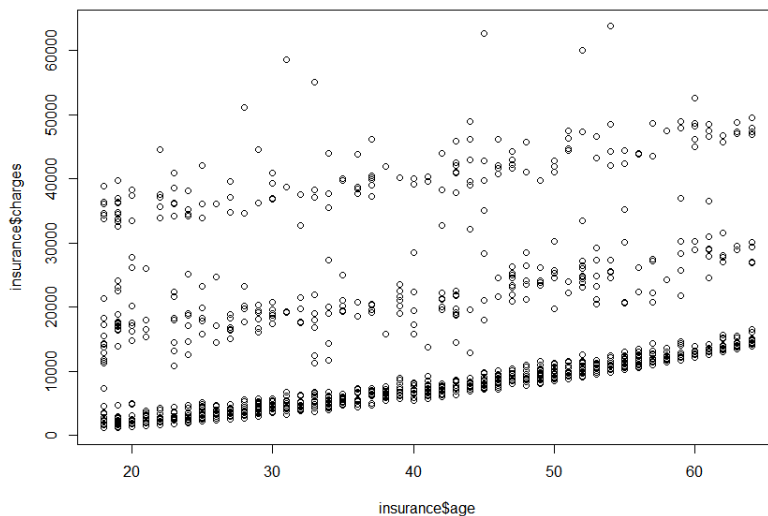
1. 상관분석 (correlation analysis)
2. 회귀분석 (regression analysis)

1. 산점도 (Scatter plot)

두 개 변수 간의 관계를 나타내는 방법이다.

상관계수를 파악하기 전에 우선, 산점도를 이용해 두 변수 간에 관련성을 시각적으로 파악할 수 있다.

```
insurance <- read.csv("c:\ww\data\ww\insurance.csv",header = T)
plot(insurance$charges~insurance$age)
```



2. 공분산과 상관계수

산점도를 이용하면 두 변수간의 직선적인 관계를 개략적으로 파악할 수 있지만, 두 변수 사이의 관계를 정확히 숫자로 나타낼 수 없기 때문에 공분산 및 상관계수를 이용한다.

2.1 공분산

2개의 확률변수의 상관정도를 나타내는 값인데, 만약 2개의 변수 중 하나의 값이 상승하는 경향을 보일 때 다른 값도 상승하면 공분산의 값은 양수, 반대로 다른 값이 하강하면 공분산 값은 음수이다.

```
cov(insurance$charges,insurance$age)
cov(insurance$age,insurance$charges)
```

```
> cov(insurance$charges, insurance$age)
[1] 50874.8
> cov(insurance$age, insurance$charges)
[1] 50874.8
```

```
> cov(insurance$charges, insurance$age)
[1] 50874.8
> cov(insurance$age, insurance$charges)
[1] 50874.8
```

위의 예제는 두 변수 간의 상관관계는 상승하는 경향이라고 이해할 수 있다.
하지만 2개의 변수의 측정 단위에 따라 값이 달라지므로 **절대적 정도**를 파악하기에는 한계가 있다.

-----> 이때 공분산을 표준화 시킨 **상관계수**를 통해 파악할 수 있다.

2.2 상관계수

상관계수는 -1 ~ 1 사이의 값을 가지며 0일 경우에는 두 변수 간의 선형관계가 전혀 없다는 것을 뜻 한다.

보통 상관계수 값이

0.3 ~ 0.7 : 뚜렷한 양적 선형관계

0.7~1.0 : 강한 양적 선형관계

라고 한다.

```
cor(insurance$charges, insurance$age, use='complete.obs', method = 'pearson')
```

use='complete.obs' : 결측값을 모두 제거된 상태에서 상관계수 계산

#method = 'pearson' : 피어슨 상관계수 지정 (가장 많이 사용)

```
> cor(insurance$charges, insurance$age, use='complete.obs', method = 'pearson')
[1] 0.2990082
```

* 상관계수는 특이 값에 민감하게 반응하며, 두 변수의 관련성만을 의미할 뿐, 원인과 결과의 방향을 알려주지는 못한다.

■ 상관계수의 검정

상관계수의 가설 검정은 cor.test() 함수를 사용하면 된다. 귀무가설 "상관관계가 없다" 에 대한 검정 결과 p-value < 2.2e-16 (=0.05) 값이 나왔으므로 귀무가설을 기각할 수 있다.

그외에 검정통계량의 값 (t) , 95% 신뢰구간, 표본상관계수 등을 확인할 수 있다.

```
cor.test(insurance$charges, insurance$age)
```

```
> cor.test(insurance$charges, insurance$age)

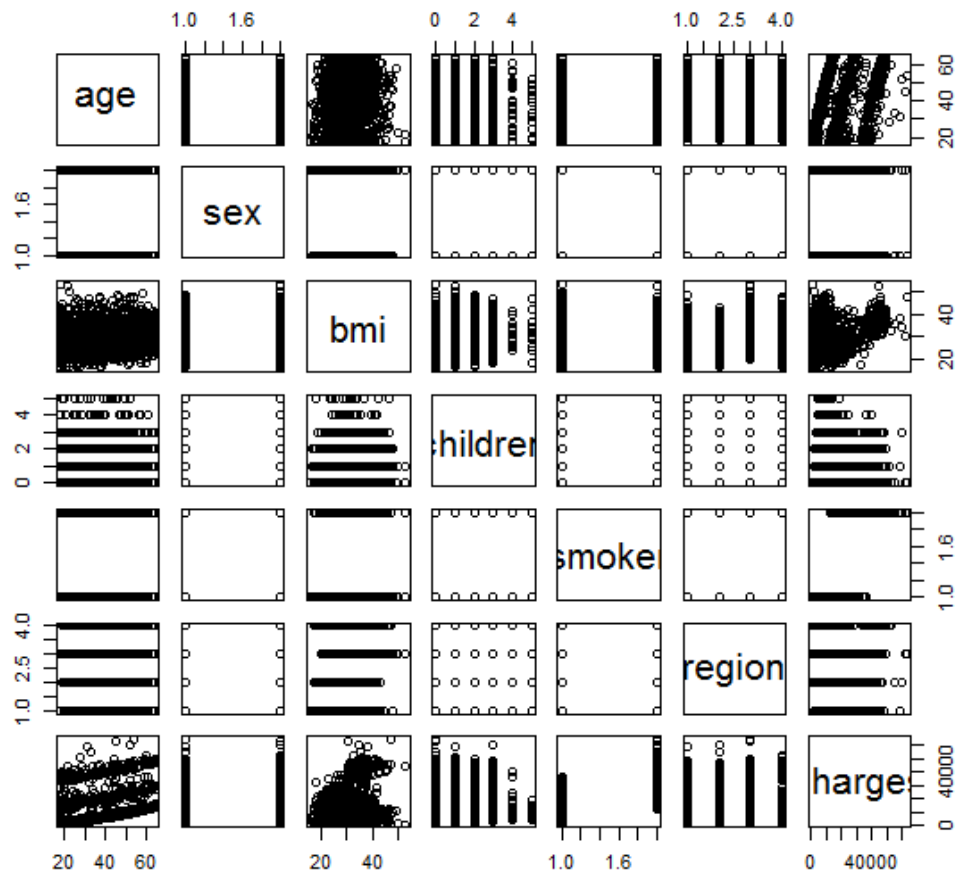
Pearson's product-moment correlation

data:  insurance$charges and insurance$age
t = 11.453, df = 1336, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.2494139 0.3470381
sample estimates:
cor
0.2990082
```

3. 상관계수 시각화

■ plot 함수

```
plot(insurance)
```



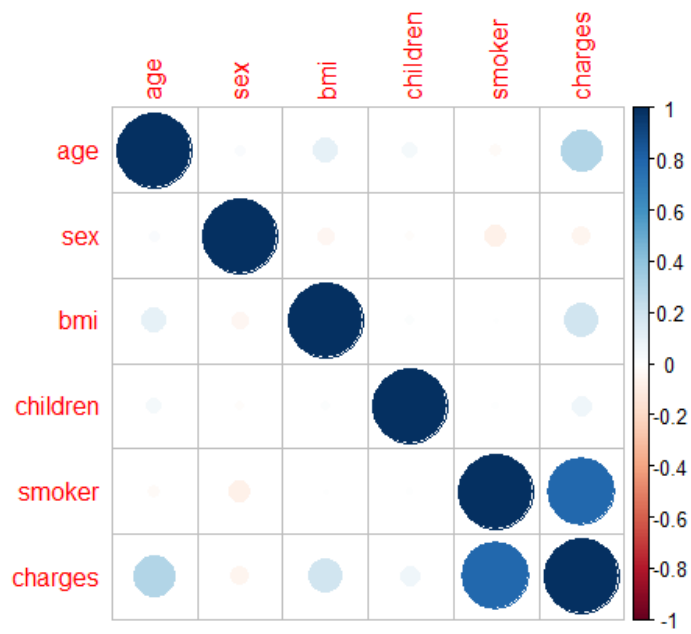
```
insurance[, "sex"] <- ifelse(insurance$sex == 'female', 1, 0)
insurance[, "smoker"] <- ifelse(insurance$smoker == 'yes', 1, 0)
cor(insurance[, -6])
```

```
> cor(insurance[, -6])
```

	age	sex	bmi	children	smoker	charges
age	1.00000000	0.02085587	0.109271882	0.04246900	-0.025018752	0.29900819
sex	0.02085587	1.00000000	-0.046371151	-0.01716298	-0.076184817	-0.05729206
bmi	0.10927188	-0.04637115	1.000000000	0.01275890	0.003750426	0.19834097
children	0.04246900	-0.01716298	0.012758901	1.000000000	0.007673120	0.06799823
smoker	-0.02501875	-0.07618482	0.003750426	0.00767312	1.000000000	0.78725143
charges	0.29900819	-0.05729206	0.198340969	0.06799823	0.787251430	1.000000000

■ corrplot 패키지 사용

```
install.packages("corrplot")
library(corrplot)
x <- cor(insurance[, -6])
corrplot(x)
```



신경망

2018년 6월 7일 목요일 오후 2:02

목차

1. 단층 퍼셉트론 and, or 게이트
2. 활성화 함수 소개
3. 단층 퍼셉트론을 R로 구현
4. 신경망 실습 1(콘크리트 데이터)
5. 신경망 실습 2(전력 생산량 데이터)
6. 신경망 실습 3(필기체 데이터 분류)

■ 활성화 함수

활성화 함수 ? 입력신호의 총합이 활성화를 일으킬지를 정하는 역할을 하는 함수

$$K = X0*W0 + X1*W1 + X2*W2$$

$$Y = f(k) \quad (1 : \text{신호가 흐른다}, 0 : \text{신호가 안흐른다})$$

* 활성화 함수의 종류

1. 계산함수

입력신호의 합이 임계치를 넘느냐 안 넘느냐에 따라서 0과 1을 리턴 (Ex. $f(0.3) = 1$, $f(-0.2) = 0$)

2. 시그모이드 함수

계단함수는 0과 1중 하나의 값만 리턴해주는 반면 시그모이드 함수는 연속적인 실수를 리턴한다.

단층 : 입력층 ---> 출력층 , 다층 : 입력층 ---> 은닉층 ---> 출력층

단층이 아니라 다층 신경망을 사용하게 되면 활성화 함수를 시그모이드 함수로 사용해야 한다.

$$f(k) = 1 / 1 + \exp(-k) \quad (\text{비선형 함수}) \quad \text{---> 미분이 가능함} \quad \exp(-k) = \text{자연상수에 } -k \text{승}$$

3. 렐루 함수 : Relu 함수

Rectified Linear Unit

(정류된) ---> 전기회로 용어

시그모이드 함수의 단점때문에 나온 함수

시그모이드 함수의 단점이 기울기 0이 되는 부분 때문에 학습이 느린 단점이 있다.

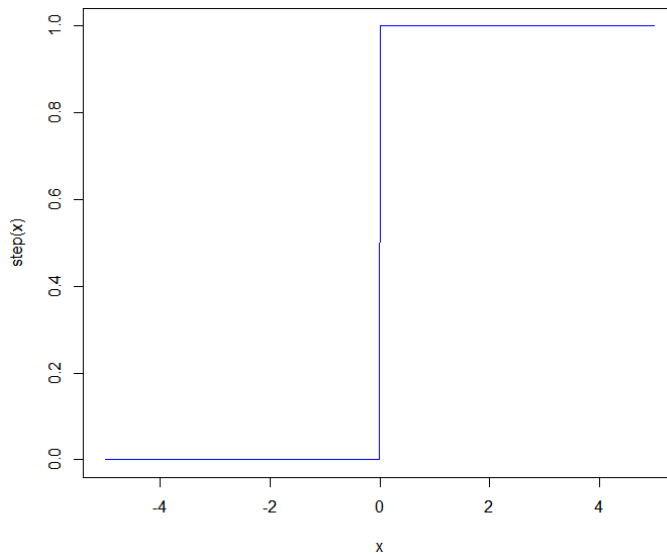
Relu는 입력이 0을 넘으면 그 입력을 그대로 출력하고 0 이하면 0을 출력하는 함수.

오래전부터 신경망은 시그모이드 함수를 이용했지만 최근에는 Relu 함수를 주로 이용한다.

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$$

문제 308. 활성화 함수인 계단함수를 R로 생성하고 계단 그래프를 그리시오.

```
step <- function(x){ ifelse (x>=0,1,0)}  
x<-seq(-5,5,0.01)  
plot(x,step(x),col="blue",type = 'l')
```



문제 309. R로 시그모이드 활성화 함수를 생성 하시오.

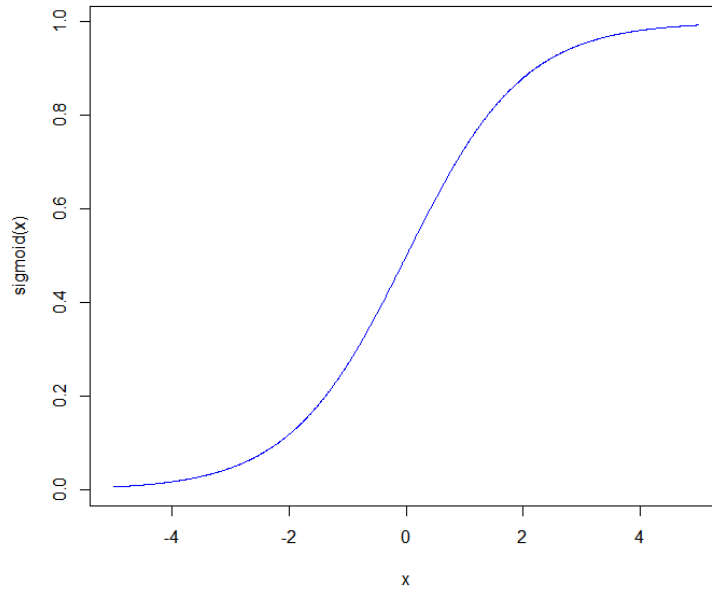
```
sigmoid <- function(x){  
  1/(1+exp(-x))  
}
```

```
sigmoid(1.0)
```

```
sigmoid(2.0)
```

```
plot(x,sigmoid(x),col="blue", type='l')
```

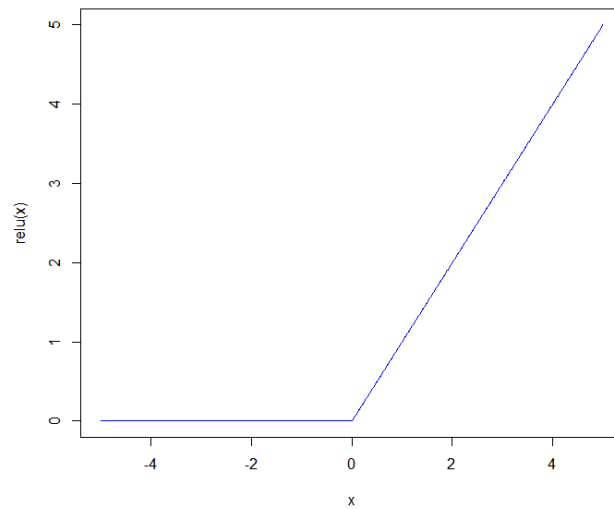
```
> sigmoid(1.0)  
[1] 0.7310586  
> sigmoid(2.0)  
[1] 0.8807971
```



문제 310. R로 Relu 함수를 만들고 Relu 함수 그래프를 그리시오.

```
relu <- function(x){
  ifelse(x>=0,x,0)
}

plot(x,relu(x),col="blue",type='l')
```



신경망 실습1 (콘크리트 데이터)

" 콘크리트 강도를 예측하는 신경망을 만드는 실습 "

자갈, 모래, 시멘트 등을 몇 대 몇 비율로 섞었을 때 어느정도 강도가 나오는지 예측하는 신경망

- 1.콘크리트 데이터 소개
- 2.콘크리트 데이터를 로드하고 데이터 정규화 작업 (0~1사이의 값을 가지게 됨)
- 3.훈련 데이터, 테스트 데이터를 나눈다 (8:2) 비율
- 4.neuralnet 패키지에 콘크리트 훈련 데이터를 넣어서 모델을 생성 (neuralnet 패키지 설치)
- 5.모델(신경망)을 시각화하고 테스트 데이터를 이용해 모델을 테스트한다.

6. 예측 값과 실제 값 간의 상관관계를 확인

1. 콘크리트 데이터 소개

1. mount of cement: 콘크리트의 총량
2. slag : 시멘트
3. ash : 분 (시멘트)
4. water : 물
5. superplasticizer : 고성능 감수제(콘크리트 강도를 높이는 첨가제)
6. coarse aggregate : 굵은 자갈
7. fine aggregate : 잔 자갈
8. aging time : 숙성시간

2. 콘크리트 데이터 로드 및 데이터 정규화 작업

```
concrete<-read.csv("c:\wwdata\wwconcrete.csv",header = T)
```

```
normalize <- function(x) {  
  return ( (x-min(x)) / (max(x) - min(x)) ) }
```

```
concrete_norm <- as.data.frame(lapply(concrete,normalize))
```

3. 훈련데이터와 테스트 데이터를 나눈다 (8:2)

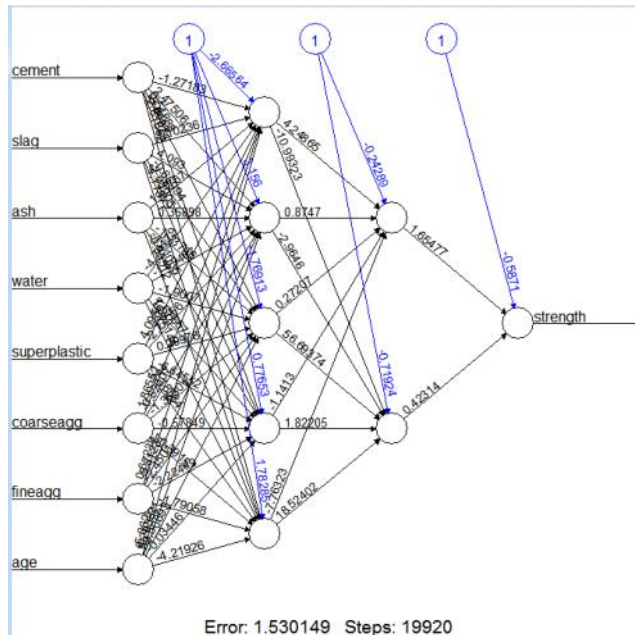
```
concrete_train <- concrete_norm[1:773, ]  
concrete_test <- concrete_norm[774:1030,]
```

4. neuralnet 패키지에 콘크리트 훈련 데이터를 넣어서 모델을 생성

```
install.packages("neuralnet")  
library(neuralnet)  
concrete_model <- neuralnet(formula = strength ~cement + slag + ash +  
                             water +superplastic + coarseagg + fineagg + age , data = concrete_train)  
model_result <- compute(concrete_model, concrete_test[1:8])  
  
predicted_strength <- model_result$net.result
```

5. 모델을 시각화 하고 테스트 데이터를 이용해 데이터 테스트

```
plot(concrete_model)
```

6. 예측 값과 실제 값 간의 상관관계 확인

```
cor(predicted_strength,concrete_test$strength)
```

```
> cor(predicted_strength,concrete_test$strength)
      [,1]
[1,] 0.8063612662
```

7. 모델 성능 개선

```
concrete_model2 <- neuralnet(formula = strength ~cement + slag + ash +
  water +superplastic + coarseagg + fineagg + age , data = concrete_train , hidden=c(5,2) )
```

```
model_result2 <- compute(concrete_model2, concrete_test[1:8])
predicted_strength2 <- model_result2$net.result
```

```
cor(predicted_strength2, concrete_test$strength)
```

```
> cor(predicted_strength2, concrete_test$strength)
      [,1]
[1,] 0.9378852843
```

문제 311. 회귀트리 그릴 때 사용했던 와인 데이터를 신경망으로 상관관계가 어떻게 되는지 확인해 보시오.

```
library(neuralnet)
```

```
wine <- read.csv("c:\data\whitewines.csv", header = T)
```

```
normalize <- function(x) {
  return ( (x-min(x)) / (max(x) - min(x)) )
}
```

```
wine_norm <- as.data.frame(lapply(wine,normalize))

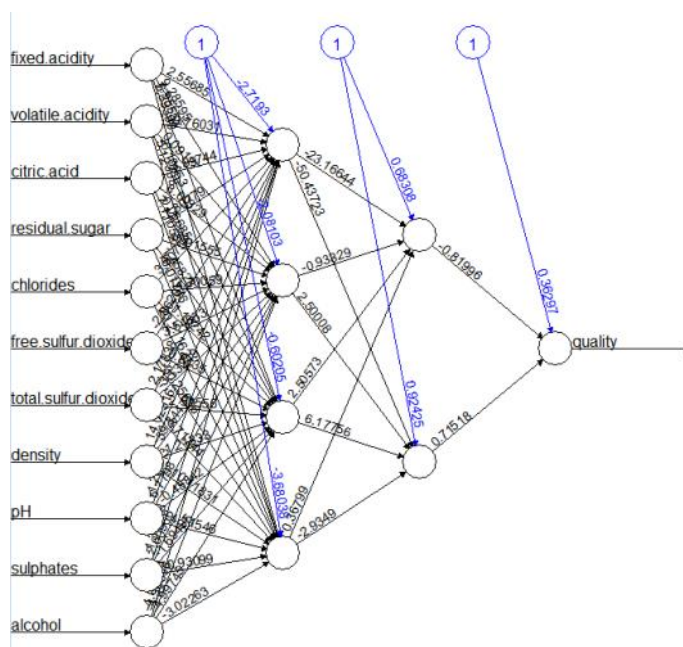
wine_train <- wine_norm[1:floor(nrow(wine_norm)*0.8), ]
wine_test <- wine_norm[ ceiling(nrow(wine_norm)*0.8):nrow(wine_norm),]

wine_model <- neuralnet(formula = quality ~ fixed.acidity + volatile.acidity
+ citric.acid + residual.sugar + chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
density + pH + sulphates + alcohol, data = wine_train,hidden=c(4,2))

model_result <- compute(wine_model, wine_test[1:ncol(wine)-1])

predicted_quality <- model_result$net.result
plot(wine_model)

cor(predicted_quality,wine_test$quality)
```



```
> cor(predicted_quality,wine_test$quality)
[1,]
[1,] 0.6245435662
```

문제 312. (자동화 스크립트) 15번에 신경망을 추가 하시오.

```
baek_func <- function() {

# 현재 컴퓨터에 필요한 패키지 설치
# 만약 패키지가 존재하지 않는 경우에만 설치

packages <- c("XML","stringr","rJava","KoNLP","wordcloud","wordcloud2","lubridate")

if (length(setdiff(packages, rownames(installed.packages())) > 0) {
install.packages(setdiff(packages, rownames(installed.packages())))
}

graphics.off()
```

```

slc<-c('막대그래프','원형그래프','산포도 그래프','워드클라우드','사분함수 그래프','분산과 표준편차',
      '정규분포와 히스토그램','산포도와 회귀직선','knn 머신러닝','naive bayes 머신러닝',
      'decision tree','규칙기반 oneR','규칙기반 JRip','단순회귀','신경망')

x1<-menu(slc,title = '그래프 선택',graphics=T)

h <- switch(menu(c('TRUE','FALSE'),title='header 옵션 : ',graphics=T),!0,!1)

res0 <- read.csv(file.choose(),header = h, stringsAsFactors = T)

#res0<- get(readline(prompt = '테이블명 입력 : '))

barplot_func <-function(){#막대그래프
  res1<- menu(colnames(res0), title='토탈 값을 구할 컬럼 선택 : ',graphics=T)
  res2<- menu(colnames(res0), title='그룹핑할 컬럼 선택 : ',graphics=T)
  q<-tapply(res0[,res1], res0[,res2],sum)
  q[is.na(q)] <- 0
  barplot(q, col = rainbow(nrow(q)), main = paste( colnames(res0)[res2], '별', colnames(res0)[res1],'총합' ), beside = T,
ylim = c(0,max(q)*1.4))
  legend("topright", rownames(q),title = paste(colnames(res0)[res2],' 구분' ),inset = 0,fill = rainbow(nrow(q)),cex=0.8)
}

pie_func <-function(){#원형 그래프
  res1<- menu(colnames(res0), title='토탈 값을 구할 컬럼 선택 : ',graphics=T)
  res2<- menu(colnames(res0), title='그룹핑할 컬럼 선택 : ',graphics=T)
  q<-tapply(res0[,res1], res0[,res2],sum)
  label<-paste(unique(res0[,res2]), round(q/sum(q) * 100,1),'%')
  pie(q,col=rainbow(nrow(q)),label=label,main = paste( colnames(res0)[res2], '별',colnames(res0)[res1],'총합' ))
}

plot_func<-function(){#산포도 그래프
  x <- menu(colnames(res0), title='x축 컬럼 선택 : ')
  y <- menu(colnames(res0), title='y축 컬럼 선택 : ')
  plot(res0[,x],res0[,y],pch=16, col=blues9,xlab = colnames(res0)[x],ylab = colnames(res0)[y],main =
paste(colnames(res0)[x],'와 ',colnames(res0)[y],'의 상관 관계 '))
}

wordcloud_func<-function(){#워드클라우드
  library(wordcloud)
  library(KoNLP)
  library(plyr)
  useSejongDic()          # 370957개의 한글 단어가 추가 (전희원 선생님이 만들)

  graphics.off()

  res<-readline(prompt = 'c:wwdata 경로에 위치한 txt 파일명 입력 : ')
  word<-readLines(gsub(' ','',paste('c:wwwdatawww',res,'.txt')))

  nouns <- extractNoun(word)  # 연설문에서 명사만 출력

```

```

nouns <- nouns[nchar(nouns)>=2]    #두글자 이상인 명사만 추출
cnouns <- count(unlist(nouns))    #단어와 건수 출력

pal <- brewer.pal(6,"Dark2")    # Dark2라는 색깔을 추가하는 작업
pal <- pal[-(1)]
windowsFonts(malgun=windowsFont("맑은 고딕"))    #맑은 고딕 폰트 추가
wordcloud(words=cnouns$x, freq=cnouns$freq, colors=pal, min.freq=3,
           random.order=F, family="malgun")
}

boxplot_func<-function(){#사분위수 그래프
  res1<- menu(colnames(res0), title='컬럼 선택 : ')
  boxplot(res0[,res1], horizontal = T, col = blues9)
}

variance_func<-function(){#분산시각화
  res1<- menu(colnames(res0),title='컬럼 선택 : ')

  plot(res0[,res1],main=paste('분산 = ',round(var(res0[,res1]),4),'표준편차 = ',round(sd(res0[,res1]),4),col='blue')
  abline(h=mean(res0[,res1]),lty=2,col='red')
}

distribution_his_func<-function(){#정규분포&히스토그램
  library(fBasics)
  res1<-menu(colnames(res0),title='컬럼 선택 : ')

  x<-sort(res0[,res1])

  hist(x,col=blues9, axes = F, ann=F)
  par(new=T)
  plot(x,dnorm(x,mean=mean(x),sd=sd(x)),type='l', lwd=3, col='red',main=paste('왜도값 : ',round(skewness(x),4)))
}

sanpodo_func<-function(){#산포도그래프&회귀직선
  res2<-menu(colnames(res0),title = 'x축 데이터 입력 : ',graphics=T)
  res3<-menu(colnames(res0),title = 'y축 데이터 입력 : ',graphics=T)

  graphics.off()
  model<-lm(res0[,res3]~res0[,res2])
  plot(res0[,res2],res0[,res3])
  abline(model,col="red")
}

knn_func<-function(){#knn_머신러닝
  library(caret)
  library(e1071)
  library(gmodels)
  library(class)
  #x <-readline("분석할 csv 파일명을 입력하세요~ ")
  y <-menu(colnames(res0),title = '라벨로 지정할 컬럼 선택 : ',graphics=T)
  y<-colnames(res0[y])

```

```

#입력값받는 함수
#k_n<-readline("k값을 입력하시오~ ")

wbcd <- na.omit(res0)

# set.seed(26)
# wbcd <- wbcd[sample(nrow(wbcd)), ]

normalize<-function(x) {
  return( (x-min(x))/( max(x)-min(x)))
}

wbcd <- wbcd[-1]
ncol1 <- which(colnames(wbcd)==y)

wbcd_n <- as.data.frame(lapply(wbcd[, -ncol1], normalize) )

mm<-round(nrow(wbcd_n)*9/10)

wbcd_train <- wbcd_n[1:mm, ]
wbcd_test  <- wbcd_n[(mm+1):nrow(wbcd_n), ]

wbcd_train_label <- wbcd[1:mm,y]
wbcd_test_label  <- wbcd[(mm+1):nrow(wbcd_n),y]

repeats = 3
numbers = 10
tunel = 10

set.seed(1234)

x = trainControl(method = "repeatedcv",
                  number = numbers,
                  repeats = repeats,
                  classProbs = TRUE,
                  summaryFunction = twoClassSummary)

model1 <- train( wbcd_train_label~. , data = data.frame(wbcd_train,wbcd_train_label), method = "knn",
                 preProcess = c("center","scale"),
                 trControl = x,
                 metric = "ROC",
                 tuneLength = tunel)

k_n<-model1$bestTune

result1 <- knn(train=wbcd_train, test=wbcd_test,
               cl= wbcd_train_label, k = k_n )
# prop.table( table(ifelse(wbcd[(mm+1):nrow(wbcd_n),y]==result1,"o","x" )))
CrossTable( x= wbcd_test_label, y= result1,prop.chisq=FALSE)

```

```

}

naive_bayes_fun<-function(){
  library(gmodels)
  library(e1071)
  dt<-res0
  res1 <- menu(colnames(dt), title='라벨이 될 컬럼번호 선택 : ',graphics=T)
  lplc <- as.numeric(readline(prompt = '라플라스 값 입력 (사용하지 않으려면 0 입력, ex.0.001): '))

  for (i in 1:length(dt)){
    dt[which(dt[,i]==""|dt[,i]=="?"), i] <- NA
  }
  dt<-na.omit(dt) # na값 생략 ( ? or "")

  for(i in 1 : length(dt)){
    dt[, i] <- factor(dt[,i])
  }

  set.seed(123450)
  train_cnt<- round (0.75*nrow(dt))
  train_indx<-sample(1:nrow(dt),train_cnt,replace = F)

  dt_train <- dt[train_indx,]
  dt_test <- dt[-train_indx,]

  model2<-naiveBayes(dt_train[,res1]~. , data = dt_train, laplace = lplc) #라플라스 사용 o 0.1~~~0.000001
  result2<-predict(model2,dt_test[, -1])

  CrossTable(dt_test[,1],result2)
}

decision_func <- function(){
  library(C50)
  library(gmodels)
  library(rattle)
  library(rpart)
  library(RColorBrewer)
  library(FSelector)

  dt <- res0

  slc <- switch(menu(c('yes','no'),title='삭제할 컬럼이 존재합니까? : ',graphics=T),!0,!1)

  if(slc){
    stop<-T
    while(stop){
      del_col <- menu(colnames(dt),title='삭제할 컬럼 선택 : ',graphics=T)
      dt<-dt[,-del_col]
      stop <- switch(menu(c('yes','no'),title='삭제할 컬럼이 더 존재 합니까?',graphics=T),!0,!1)
    }
  }
}

```

```

}

lb <- menu(colnames(dt),title = '라벨이 될 컬럼 선택 : ',graphics=T)
tp <- as.numeric(readline(prompt='train data의 비율 입력 ex.0.9(90%일 경우) : '))

set.seed(11)
dt_shuffle <- dt[sample(nrow(dt)), ] # 데이터 셔플

train_num<-round(tp*nrow(dt_shuffle),0)

dt_train <- dt_shuffle[1:train_num,]
dt_test <-dt_shuffle[(train_num+1):nrow(dt_shuffle),]
dt_model <- C5.0(dt_train[, -lb], dt_train[,lb] , trials = 100)
print(dt_model)
dt_result <- predict(dt_model, dt_test[, -lb])

CrossTable(dt_test[, lb], dt_result ,prop.chisq = F)
tree1 <-rpart( dt_train[,lb]~., data = dt_train[, -lb], method='class',control = rpart.control(minsplit = 3))

graphics.off()
fancyRpartPlot(tree1,type=2,palette=c('Spectral','RdYlGn'),caption = '의사결정나무 그래프')
}

rule_based<-function(){
  library(RWeka)
  library(OneR)
  library(gmodels)
  files<-res0

  delete<-0
  while(TRUE){
    delete<-menu(c('없음',colnames(files)),title='삭제할 컬럼번호를 입력하세요',graphics=T)
    if(delete==1) break
    files<-files[-(delete+1)]
  }
  label_num<-menu(colnames(files),title='라벨이 될 컬럼을 선택하세요',graphics=T)
  files<-data.frame(label=files[,label_num],files[, -label_num])
  a<-readline(prompt='train data의 비율을 어떻게 하겠습니까? ex.0.9(90%일 경우)')
  a<-as.numeric(a)

  set.seed(123450)
  train_cnt<-round(a * dim(files)[1])
  train_indx<-sample(1:dim(files)[1],train_cnt, replace=F)
  data_train<-files[train_indx,]
  data_test<-files[-train_indx,]

  if (x1==12)
    model<- OneR(label~.,data=data_train)
  else if(x1==13)
    model<- JRip(label~.,data=data_train)

```

```

result<-predict(model,data_test[-1])
CrossTable(data_test[,1],result,prop.chisq = F)
}

regression_func <- function(){
  library(stats)

  dt <-res0

  x <- menu(colnames(dt),title = '독립변수가 될 컬럼 선택(x축): ',graphics=T)
  y <- menu(colnames(dt),title = '종속변수가 될 컬럼 선택(y축): ',graphics=T)

  model2<-lm(dt[,y] ~ dt[,x]) # y축 x축

  plot(dt[,y] ~ dt[,x], data=dt, pch=21,col=blues9,bg=blues9,
       main=paste('y=',round(model2$coefficients[2],3),'x + ',round(model2$coefficients[1],3)),
       xlab=colnames(dt[x]),ylab=colnames(dt[y]),sub=paste('결정계수 : ',summary(model2)$r.squared ))
  abline(model2,col='red',lwd=3)
}

ann_func <- function(){

  dt<-res0

  library(neuralnet)

  slc <- switch(menu(c('yes','no'),title='삭제할 컬럼이 존재합니까? : ',graphics=T),!0,!1)

  if(slc){
    stop<-T
    while(stop){
      del_col <- menu(colnames(dt),title='삭제할 컬럼 선택 : ',graphics=T)
      dt<-dt[,-del_col]
      stop <- switch(menu(c('yes','no'),title='삭제할 컬럼이 더 존재 합니까?',graphics=T),!0,!1)
    }
  }

  lb <- menu(colnames(dt),title = '라벨이 될 컬럼 선택 : ',graphics=T)
  print(colnames(dt[lb]))
  normalize <- function(x) {
    return ( (x-min(x)) / (max(x) - min(x)) )
  }

  dt_norm <- as.data.frame(lapply(dt,normalize))

  dt_train <- dt_norm[1:floor(nrow(dt_norm)*0.8), ]
  dt_test <- dt_norm[ceiling(nrow(dt_norm)*0.8):nrow(dt_norm),]

  name<-names(dt_train)

```



```
f = as.formula(paste(colnames(dt[lb]), '~', paste(name[!name %in% colnames(dt[lb])], collapse = '+'))

dt_model <- neuralnet(formula = f, data = dt_train) # hidden = c(5,3) ... 처럼 줄수 있다.
model_result <- compute(dt_model, dt_test[-lb])

predicted_dt <- model_result$net.result
plot(dt_model)

cor(predicted_dt, dt_test$quality)
}

switch(as.numeric(x1), barplot_func(), pie_func(), plot_func(), wordcloud_func(),
       boxplot_func(), variance_func(), distribution_his_func(), sanpodo_func(),
       knn_func(), naive_bayes_func(), decision_func(), rule_based(), rule_based(), regression_func(), ann_func())
}

baek_func()
```

3. 단층 퍼셉트론을 R로 구현

문제 313. Input 라는 변수에 아래의 행렬을 입력 하시오.

0	0
1	0
0	1
1	1

```
x1<-c(0,1,0,1)
x2<-c(0,0,1,1)
x1<-cbind(x1,x2)
input<-matrix(x1,nrow=4,ncol=2)
```

```
> input
      [,1] [,2]
[1,]    0    0
[2,]    1    0
[3,]    0    1
[4,]    1    1
```

문제 314. Targets1 이라는 아래 행렬을 생성 하시오.

0
0
0
1

```
target1<-matrix(c(0,0,0,1),nrow=4,ncol = 1)
```

```
> target1<-matrix(c(0,0,0,1),nrow=4,ncol = 1)
> target1
     [,1]
[1,]    0
[2,]    0
[3,]    0
[4,]    1
```

■ 가중치를 만들기 위한 랜덤변수 runif(1) : 0~1 사이의 랜덤 값 생성

문제 315. 아래의 행렬과 input 행렬을 cbind로 묶어서 new_input이라는 변수에 넣으시오.

1
1
1
1

```
new_inputs<-cbind(c(1,1,1,1),input)

> new_inputs<-cbind(c(1,1,1,1),input)
> new_inputs
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    1    1    0
[3,]    1    0    1
[4,]    1    1    1
```

문제 316. New_inputs 행렬과 w행렬곱(내적)을 구하시오.

```
new_inputs %*% w

> new_inputs %*% w
     [,1]
[1,] 0.4916863
[2,] 0.7919858
[3,] 1.2327428
[4,] 1.5330423
```

문제 317. 어제 만들었던 계단 함수에 위의 값들을 넣고 값을 출력 하시오.

```
k<-new_inputs %*% w
step <- function(x){ ifelse (x>=0,1,0)}
x<-step(k)
x

> x
     [,1]
[1,]    1
[2,]    1
[3,]    1
[4,]    1
```

문제 318. step(k) 값과 targets1 과의 차이를 구하시오.

```
tfk <- targets1 - step(k)
```

```
tfk <- target1 - step(k)
```

tfk

```
> tfk
      [,1]
[1,]   -1
[2,]   -1
[3,]   -1
[4,]    0
```

신경망 R패키지 : 1. neuralnet 패키지 (콘크리트 데이터) 2. nnet 패키지 (와인데이터)

#1.nnet 패키지를 설치한다

```
install.packages("nnet")
library(nnet)
```

#2.wine 데이터를 로드한다

```
wine<-read.csv("c:\\data\\wine.csv",header = T)
head(wine)
```

#3. 정규화 작업을 진행한다.

```
wine_norm <- cbind(wine[1],scale(wine[-1]))
size <- nrow(wine_norm)
size
```

#4. 7:3으로 훈련데이터와 테스트 데이터를 분리한다.

```
set.seed(100)
index<-c(sample(1:size,size*0.7))
```

```
train<-wine_norm[index,]
test <- wine_norm[-index,]
```

#5. 훈련데이터로 신경망 모델을 생성한다.

```
wine_model<-nnet(Type~., data=train,size=2,decay=5e-04,maxit=200)
#size = 2 : 은닉층의 뉴런수 , decay = 5e-04 가중치 감소, maxit = 반복수
```

#6. 테스트 데이터를 모델에 넣어서 결과를 예측한다.

```
predicted_result <- predict(wine_model, test, type = "class")
predicted_result
```

#7. 이원 교차표를 그리고 정확도를 확인한다.

```
actual <- test$Type
model.confusion.matrix <- table (actual, predicted_result)
```

```
library(gmodels)
CrossTable(model.confusion.matrix)
```

actual	predicted_result			Row Total
	t1	t2	t3	
t1	20	0	0	20
	21.407	7.407	5.185	0.370
	1.000	0.000	0.000	
	1.000	0.000	0.000	
t2	0.370	0.000	0.000	21
	0	20	1	
	7.778	19.206	3.628	
	0.000	0.952	0.048	0.389
t3	0.000	1.000	0.071	
	0.000	0.370	0.019	
	0	0	13	13
	4.815	4.815	27.513	0.241
Column Total	0.000	0.000	1.000	
	0.000	0.000	0.929	
	0.000	0.000	0.241	
	20	20	14	54
	0.370	0.370	0.259	

문제 319. Wine 데이터의 맨 마지막 로우 하나만 분리해서 valid_wine.csv 라는 이름으로 저장하고 wine.csv는 wine2.csv 로 저장 하시오.

```
wine<-read.csv("c:\\data\\wine.csv",header = T)

valid_wine<- wine[nrow(wine), -1] #라벨값 (type) 지움
wine2<-wine[1:nrow(wine)-1,]

write.csv(valid_wine, file="c:/data/valid_wine.csv")
write.csv(wine2, file="c:/data/wine2.csv")
```

문제 320.

#1.nnet 패키지를 설치한다

```
install.packages("nnet")
library(nnet)
```

#2.wine 데이터를 로드한다

```
wine2<-read.csv("c:\\data\\wine.csv",header = T)
```

```
# valid_wine<- wine[nrow(wine),] #라벨값 (type) 지움
wine2<-wine2[1:nrow(wine2),]
```

```
head(wine2)
```

#3. 정규화 작업을 진행한다.

```
wine_norm <- cbind(wine2[1],scale(wine2[-1]))
#size <- nrow(wine_norm)
#size
valid_wine <- wine_norm[nrow(wine_norm),]
wine_norm<-wine_norm[1:nrow(wine_norm)-1,]
```

#4. 7:3으로 훈련데이터와 테스트 데이터를 분리한다.

```
set.seed(100)
index<-c(sample(1:size,size*0.7))
```

```
train<-wine_norm
```

```
#test <- wine_norm[-index,]
test <- valid_wine
```

#5. 훈련데이터로 신경망 모델을 생성한다.

```
wine_model<-nnet(Type~., data=train,size=2,decay=5e-04,maxit=200)
#size = 2 : 은닉층의 뉴런수 , decay = 5e-04 가중치 감소, maxit = 반복수
```

#6. 테스트 데이터를 모델에 넣어서 결과를 예측한다.

```
predicted_result2 <- predict(wine_model, test, type = "class")
predicted_result2
```

#7. 이원 교차표를 그리고 정확도를 확인한다.

```
actual <- test$Type
model.confusion.matrix <- table (actual, predicted_result2)
```

```
library(gmodels)
CrossTable(model.confusion.matrix)
```

Total Observations in Table: 1

actual	predicted_result2	
	t3	Row Total
t1	0 0.000	0
t2	0 0.000	0
t3	1 1.000	1
Column Total	1	1

R데이터 변환 : 표준화(Standardization)

2018년 5월 28일 월요일 오후 8:19

0. 데이터 변환

■ 데이터 변환 종류

(1) 표준화 (Standardization)

(2) 정규분포화

(3) 범주화

- 이산형화

- 이항변수화

(4) 개수 축소

(5) 차원 축소

- 주성분분석

- 요인분석

(6) 시그널 데이터 압축

다양한 소스로 부터 데이터를 R로 불러와서, 결합하고, **결측값**과 **특이값**을 확인 후 처리하고, 필요한 부분의 데이터만 선별적으로 선택 혹은 제거한 후에 분석의 목적과 필요에 따라서, 그리고 데이터의 형태에 따라서 다양한 데이터 변환 (data transformation) 작업을 수행한다.

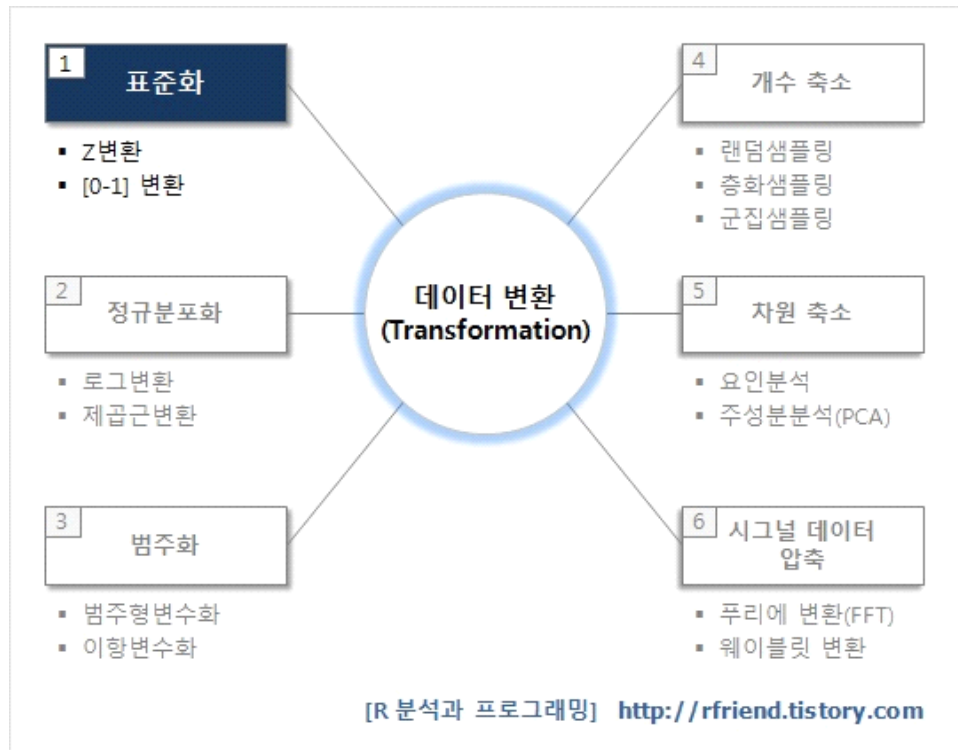
고급 분석가와 그렇지 않는 분석가가 나뉘는 부분, 데이터 엔지니어와 데이터 분석가가 나뉘어 지는 부분이 여기서 부터 이지 않을까 싶습니다. 업에 대한 지식과 더불어 분석의 목적과 분석의 기법에 대해서 정확히 알아야 하고, 데이터의 형태가 그에 맞는지, 맞지 않다면 어떻게 변환을 해야 하는지 알아야 하기 때문입니다. 그리고 데이터 변환을 하는데 있어 통계적인 기본 지식이 필요하다보니 여기부터는 프로그래밍을 잘하지만 통계를 잘 모르는 데이터 엔지니어의 경우 어려움을 겪기 시작합니다.

이 변환 작업에는 많은 시간과 노력이 필요합니다. 그래서 데이터 분석을 업으로 삼으려고 생각했던 사람이라도 소위 데이터 전처리, 데이터 변환의 지난한 과정에 대해서 재미를 느끼지 못하면 오래 견디지 못하고 다른 커리어로 전향을 하기도 합니다. 그만큼 본격적인 통계/데이터마이닝 과정에 진입하기 위한 전초 단계로 중요하지만 쉽지 않은 과정이라는 얘기입니다.

모델링을 하는데 있어 분석 목적에 유의미하고 적합한 파생변수를 개발하고 input으로 넣는 것이 정말 중요합니다. 개념적 정의, 조작적 정의를 통해 파생변수를 개발하는 과정에 필수로 필요한 이론적 지식이 이번부터 해서 총 6번에 나누어서 진행할 데이터 변환이 되겠습니다.

구분	데이터 변환 종류
데이터 분포나 속성을 변화시키는 기법	(1) 표준화, (2) 정규분포화, (3) 범주화
데이터 크기를 축소하는 기법	(4) 개수 축소(샘플링), (5) 차원 축소, (6) 시그널 데이터 압축

1. 표준화(Standardization)



1.1 표준정규분포 z 변환

우선 정규분포에 대해서 간략히 짚고 z 변환으로 넘어가겠습니다. 일상 생활 속에서 우리는 다양한 정규분포를 접하고 삽니다. 만약 100명의 수강생을 대상으로 통계와 R 분석 교육을 받고 시험을 치면 아마도 평균을 중심으로 종모양으로 좌우 분포가 비슷한 성적 분포를 띠 것입니다. 수강생 100명의 키와 몸무게를 조사를 해서 히스토그램을 그려보면 이 또한 평균을 중심으로 종모양으로 좌우 대칭인 정규분포를 띠 것입니다. 수강생 얼굴을 아직 본적도 없는데 이렇게 예언을 할 수 있다는거, 이게 참 신기한겁니다. ^^ 만약 키의 평균과 표준편차를 저한테 알려주고, 수강생 100명 중에서 한 명의 수강생을 뽑아서 키를 재서 저에게 알려주면 그 수강생이 전체 100명 중에서 상위 몇 % 키에 속할지도 추측할 수 가 있습니다. 놀랍지요?

통계학에서는 '중심극한정리(central limit theorem)'이 정말 중요한 역할을 하는데요, 중심극한정리란 분포의 모양을 모르는 모집단으로부터 표본을 추출할 때, 표본평균

$$\bar{X}$$

의 분포는 표본의 크기 n이 커짐(일반적으로)에 따라 점점 정규분포로 근사해 간다는 성질을 말합니다.

$$n \geq 30$$

참고) 중심극한정리 (Central Limit Theorem)

$$X_1, \dots, X_n$$

을 평균

$$\mu$$

, 분산

$$\sigma^2$$

인 모집단으로부터의 크기 n 인 확률표본이라고 했을 때,

표본평균

\bar{X}

의 분포는 n 이 커짐에 따라 정규분포

$N(\mu, \sigma^2/n)$

으로 근사해 간다.

중심극한정리에서 표본평균

\bar{X}

를 표준화하면

통계량

$$Z = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}}$$

는 근사적으로 표준정규분포

$N(0, 1)$

을 따른다.

이 중심극한정리에 근거해서 보통 샘플이 대략 30개 이상이면 표본평균이 정규분포로 근사한다고 가정하고 정규분포 가정에 근거한 다양한 통계분석 기법(추정과 검정 등...)을 적용할 수 있게 됩니다.

이때 두 개 이상의 모집단으로 부터 표본의 크기가 큰 표본을 추출했을 때, 각 집단의 평균과 표준편차가 다르거나, 혹은 측정 scale 이 다른 경우에는 다수의 집단 간, 변수 간 직접적인 비교가 불가능하게 됩니다. 미국 달러, 유럽의 유로화, 중국의 위안화, 일본의 엔화, 그리고 한국의 원화를 각 각 1000 단위를 가지고 있다고 했을 때, 이게 서로간에 대비해서 얼마나 많은 건지, 값어치가 있는건지 직접 비교하는게 불가능한 것과 같은 이치입니다. 이때 **특정 나라의 통화를 기준으로 삼고 다른 나라의 통화를 기준으로 변환을 하면 각 나라별 통화간의 돈의 가치를 비교할 수 있게 됩니다. 이게 표준화의 원리입니다.**

위에서 정규분포의 중요성에 대해서 설명했는데요, 정규분포 중에서도 평균이 0, 표준편차가 1인 정규분포를 **표준정규분포 (standardized normal distribution)** 이라고 합니다. 평균이 표준편차가 서로 다른 다수의 집합을 표준정규분포로 표준화를 하면 서로 비교를 할 수 있게 됩니다.

그러면, 이제 R로 표준정규화 하는 방법에 대해서 알아보겠습니다.

- 한국 성인 남성 1,000 명의 키가 평균 170cm, 표준편차 10cm의 정규분포
 - 남아프리카 부시맨 성인 남성 1,000명의 키가 평균 150cm, 표준편차 8cm의 정규분포
- 를 따른 다고 했을 때 두 집단의 키를 평균이 0, 표준편차가 1인 표준정규분포로 표준화를 해보도록 하겠습니다.

먼저, 데이터 생성은 아래와 같이 랜덤하게 생성하였습니다.

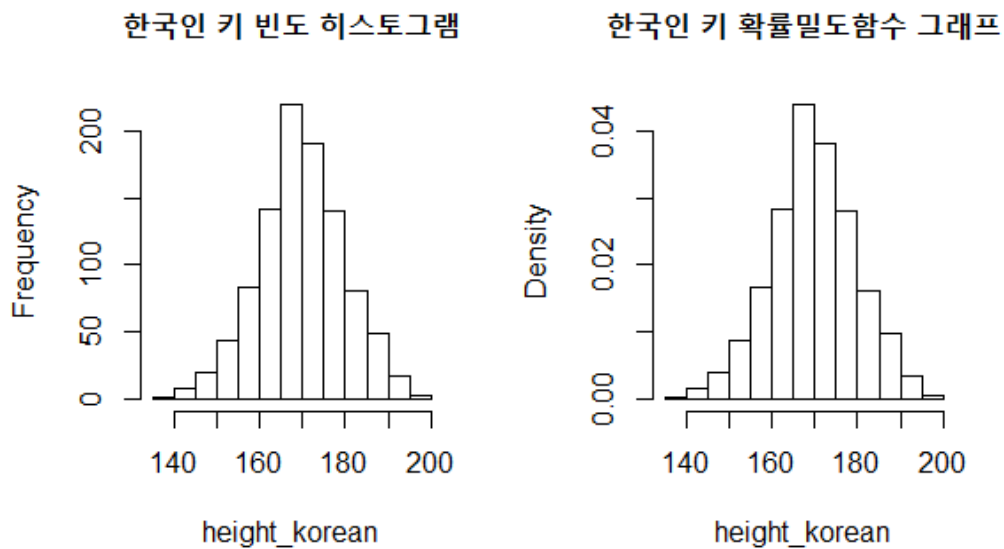
```
> ## 한국인, 부시맨 각 성인 1000명 키 데이터 생성
> height_korean <- rnorm(n=1000, mean = 170, sd = 10)
> height_bushman <- rnorm(n=1000, mean = 150, sd = 8)
>
> height <- data.frame(height_korean, height_bushman) # 데이터 프레임 생성
```



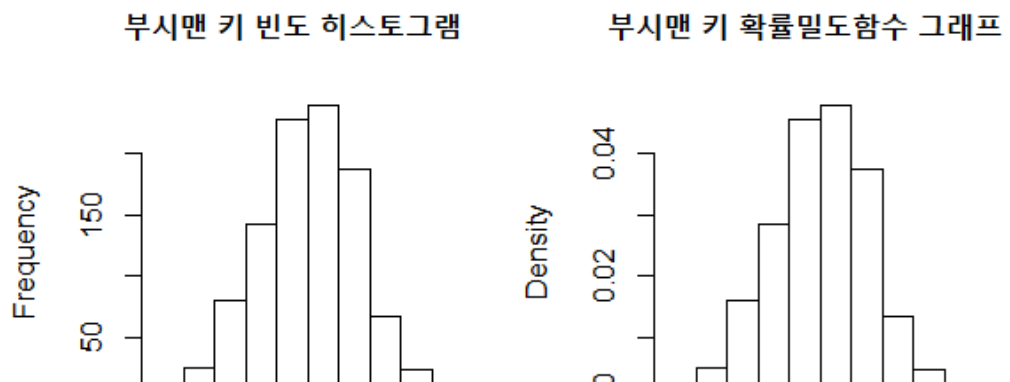
```
> rm(height_korean, height_bushman) # 벡터 삭제
>
> head(height) # 상위 6개 데이터 확인
  height_korean height_bushman
1    162.7654    132.5271
2    180.5701    135.5497
3    172.6752    142.5168
4    171.8035    156.7872
5    186.5258    154.3027
6    171.4634    156.1118
```

```
> ## 한국인, 부시맨 키 히스토그램
```

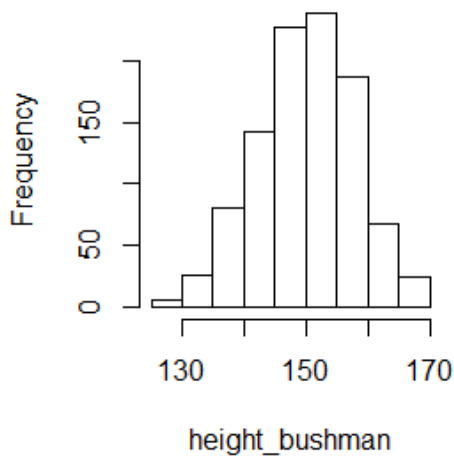
```
> attach(height)
> par( mfrow = c(1,2))
> hist(height_korean, freq = TRUE, main = "한국인 키 빈도 히스토그램")
> hist(height_korean, freq = FALSE, main = "한국인 키 확률밀도함수 그래프")
>
```



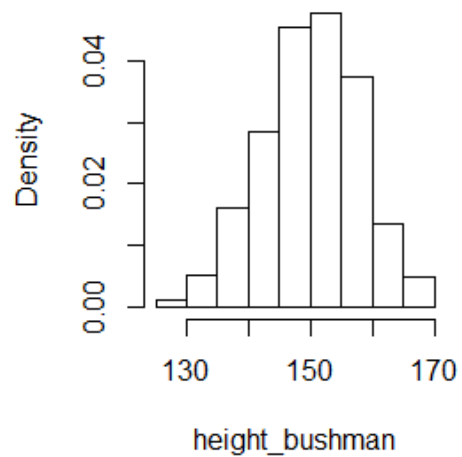
```
> hist(height_bushman, freq = TRUE, main = "부시맨 키 빈도 히스토그램")
> hist(height_bushman, freq = FALSE, main = "부시맨 키 확률밀도함수 그래프")
```



부시맨 키 빈도 히스토그램



부시맨 키 확률밀도함수 그래프



```
> detach(height)
```

그리고 표준정규화를 해보겠는데요, (a) **scale()** 함수를 쓰는 방법과 (b) $(x - \text{mean}(x)) / \text{sd}(x)$ 처럼 공식을 직접 입력하는 방법이 있습니다. 결과는 동일합니다.

```
> ## a. scale() 함수
```

```
>
> height <- transform(height,
+                       z.height_korean = scale(height_korean),
+                       z.height_bushman = scale(height_bushman)
+                       )
>
> head(height)
  height_korean height_bushman z.height_korean z.height_bushman
1      179.19      140.60      0.89308      -1.18393
2      164.54      152.70     -0.60892       0.35689
3      184.18      136.76      1.40477     -1.67426
4      196.37      144.26      2.65531     -0.71833
5      162.61      155.72     -0.80706       0.74198
6      158.02      147.19     -1.27775     -0.34510
```

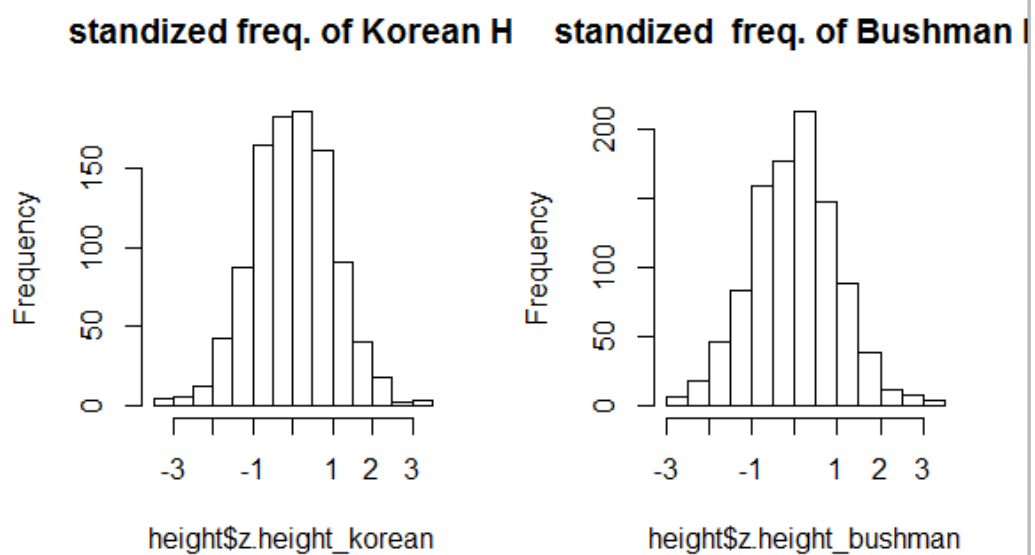
```
> ## b. z=(x-mean(x))/sd(x)
```

```
> height <- transform(height,
+                       z2.height_korean = (height_korean - mean(height_korean))/sd(height_korean),
+                       z2.height_bushman = (height_bushman - mean(height_bushman))/sd(height_bushman)
+                       )
>
> head(height)
  height_korean height_bushman z.height_korean z.height_bushman z2.height_korean z2.height_bushman
```

1	179.19	140.60	0.89308	-1.18393	0.89308	-1.18393
2	164.54	152.70	-0.60892	0.35689	-0.60892	0.35689
3	184.18	136.76	1.40477	-1.67426	1.40477	-1.67426
4	196.37	144.26	2.65531	-0.71833	2.65531	-0.71833
5	162.61	155.72	-0.80706	0.74198	-0.80706	0.74198
6	158.02	147.19	-1.27775	-0.34510	-1.27775	-0.34510

아래 히스토그램은 한국인과 부시맨의 성인 남자 키를 z 표준화 한 값에 대한 히스토그램이 되겠습니다. 둘다 평균이 0, 표준 편차가 1인 표준정규분포로 표준화 되었음을 확인할 수 있습니다.

```
> hist(height$z.height_korean, freq=TRUE, main="standized freq. of Korean H")
> hist(height$z.height_bushman, freq=TRUE, main="standized freq. of Bushman H ")
```



1.2 [0-1] 변환

연속형 변수의 값을 '0~1' 사이의 값으로 변환하는 [0-1]변환도 z변환과 함께 많이 쓰이는 표준화 기법입니다. 만약 변수들 간의 scale 이 다른 상태에서 인공지능망 분석을 하려면 [0-1]변환으로 단위를 표준화해준 후에 분석을 시행해야 합니다. Scale이 다른 두 변수를 [0-1] 변환하게 되면 상호간에 비교가 가능해집니다.

[0-1] 변환은 $(x - \min(x)) / (\max(x) - \min(x))$ 의 수식으로 계산하면 됩니다.

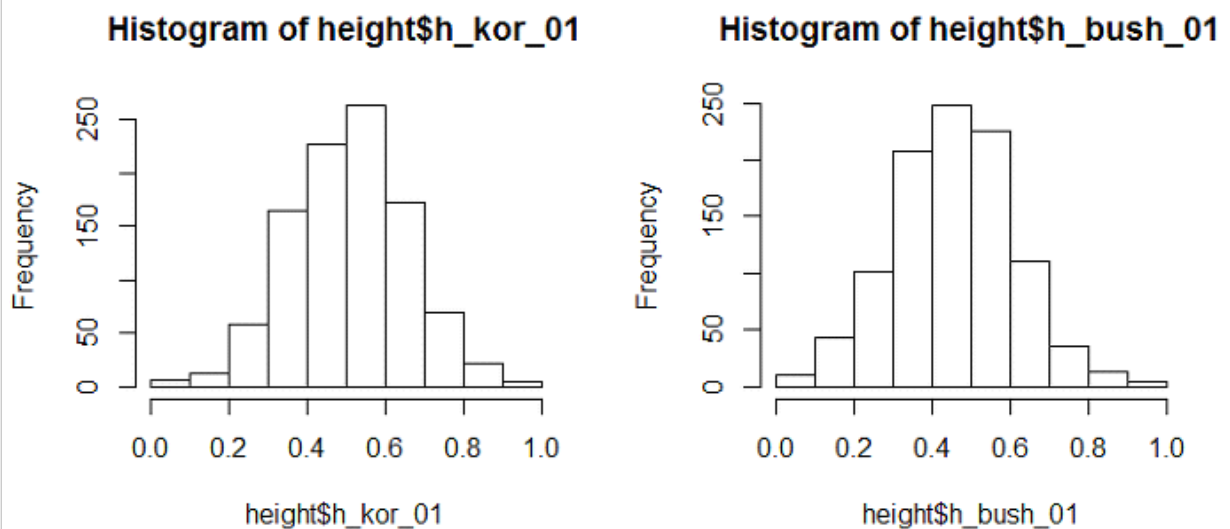
위의 한국 성인 남성과 부시맨 성인 남성 각 1,000명의 키 데이터를 가지고 이번에는 [0-1] 표준화 변환을 해보도록 하겠습니다. 일단 위 데이터셋 height에서 첫번째와 두번째 변수만 선택하고, 변수명이 너무 기므로 짧게 변수이름을 변경해보겠습니다.

```
> ## [0-1] transformation
> height <- height[,c(1:2)]
> library(reshape)
> height <- rename(height, c(height_korean = "h_kor", height_bushman = "h_bush"))
> head(height)
```

	h_kor	h_bush
1	179.19	140.60
2	164.54	152.70
3	184.18	136.76
4	196.37	144.26
5	162.61	155.72
6	158.02	147.19

그 다음 [0-1] 변환을 하고 히스토그램을 그려보겠습니다.

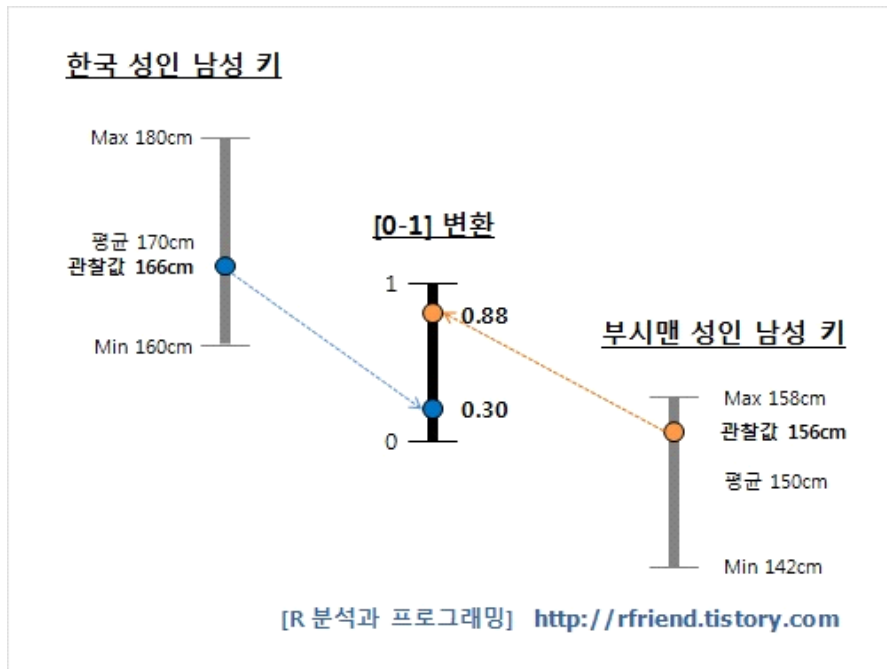
```
> height <- transform(height,
+       h_kor_01 = (h_kor - min(h_kor))/(max(h_kor) - min(h_kor)),
+       h_bush_01 = (h_bush - min(h_bush))/(max(h_bush) - min(h_bush))
+       )
>
> head(height)
  h_kor h_bush h_kor_01 h_bush_01
1 179.19 140.60  0.64341  0.27053
2 164.54 152.70  0.41760  0.51072
3 184.18 136.76  0.72034  0.19410
4 196.37 144.26  0.90835  0.34311
5 162.61 155.72  0.38781  0.57074
6 158.02 147.19  0.31705  0.40129
>
> hist(height$h_kor_01)
> hist(height$h_bush_01)
```



한국 성인 남성 키와 부시맨 성인 남성 키가 0~1 사이의 값으로 표준화되었음을 알 수 있습니다.

이해가 쉽도록 166cm의 한국 남성과 156cm의 부시맨 남성의 키를 가지고 [0-1] 변환 했을 때의 예시를 개념도로 아래에 작성 하였습니다. 참고하시기 바랍니다.

[0-1] 변환 예시 (한국 남성 166cm, 부시맨 남성 156cm)



요약

2018년 6월 10일 일요일 오후 9:31

Knn 알고리즘

새로 들어온 데이터가 기존 데이터의 그룹 중 어느 그룹에 속하는지 찾을 때 거리가 가장 가까운 데이터의 그룹을 자기 그룹으로 선택하는 아주 간단한 알고리즘

데이터 타입	수치형 데이터
개선 방법	최적의 k값
사용 패키지	
사용 함수	trainControl(), train(), knn(), normalize(사용자정의),
예시	

패키지정리

2018년 6월 10일 일요일 오후 10:10

library(data.table) # 데이터 테이블 패키지

library(doBy) # orderBy 함수를 쓸때 사용하는 패키지

library(lubridate) # 날짜함수를 쓸때 사용하는 패키지

library(dplyr) # 데이터 처리할수 있는 웬만한 함수가 있는 패키지

library(igraph) # 그래프 패키지

library(googleVis) # 구글에서 제공하는 그래프 패키지

library(maps) # 지도 그래프 패키지

library(mapproj) # 지도 그래프 패키지

library(ggplot2) # 구글에서 제공하는 지도 그래프 패키지

library(ggmap) # 구글에서 제공하는 지도 그래프 패키지

library(tuneR) # 소리를 시각화 할 수 있는 패키지

library(KoNLP) # 한국어를 R에서 인식할 수 있도록 하는 패키지

library(wordcloud) # 워드 클라우드를 그리는 패키지

library(plyr) # 워드 클라우드를 그릴때 필요한 패키지

library(outliers) # 이상치, 중앙값을 구할때 필요한 패키지(사분위수 그래프를 쓸때 쓰임)

library(FSelector) # 의사결정트리의 정보획득량을 구할때 필요한 패키지

library(gmodels) # 이원 교차표를 그리기 위해서 필요한 패키지

library(fBasics) # 왜도와 첨도를 구할때 필요한 패키지

library(shiny) # 샤이니 앱을 이용하기 위한 패키지

library(rsconnect) # 도메인 서버와 연결해주는 기능을 가진 패키지

library(xlsx) # xlsx 파일을 로드할때 필요한 패키지

```
library(class) # knn 알고리즘을 사용하기 위한 패키지
```

```
library(gmodels) # 이원 교차표를 나타내기 위한 패키지
```

```
library(e1071) # naiveBayes 함수를 사용하기 위한 패키지
```


연관분석(아포리오 알고리즘)

2018년 6월 11일 월요일 오전 10:08

8장 목차

1. Apriori 알고리즘 정의와 이론 설명
2. Apriori 알고리즘 실습1 (맥주와 기저귀)
3. Apriori 알고리즘 실습2 (국영수 성적 데이터)
4. Apriori 알고리즘 실습3 (상가 건물 데이터)
5. Apriori 알고리즘 자동화 스크립트 추가
6. NCS 평가문제 제출

Apriori 알고리즘이란?

1. 간단한 성능 측정치를 이용해 거대한 데이터베이스에서 데이터간의 연관성을 찾는 알고리즘
2. 맥주와 기저귀의 관계를 알아낸 대표적인 기계학습 방법

Apriori 알고리즘 사용 예시

1. 암 데이터에서 빈번히 발생하는 DNA 패턴과 단백질의 서열을 검색할 때
2. 사기성 신용카드 미 보험의 이용과 결합돼 발생하는 구매 또는 의료비 청구의 패턴 발견
3. 고객이 휴대폰 서비스를 중단하거나 케이블 TV 패키지를 업그레이드할 때 선행되는 행동의 조합 식별

연관규칙을 사람이 하기 어려운 이유 ?

1. 데이터가 너무 많아서 ...
= k개의 아이템이 있다면 잠재적인 규칙이 될 수 있는 2^k 개의 아이템 집합이 존재한다.

거래에서 나타나는 모든 항목들의 집합(item set)을

$$I = \{i_1, i_2, \dots, i_k\}$$

이라고 할 때, 모든 가능한 부분집합의 개수는 공집합을 제외하고

$$M = 2^k - 1$$

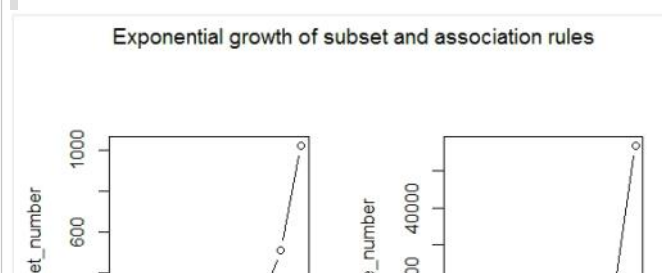
개 입니다.

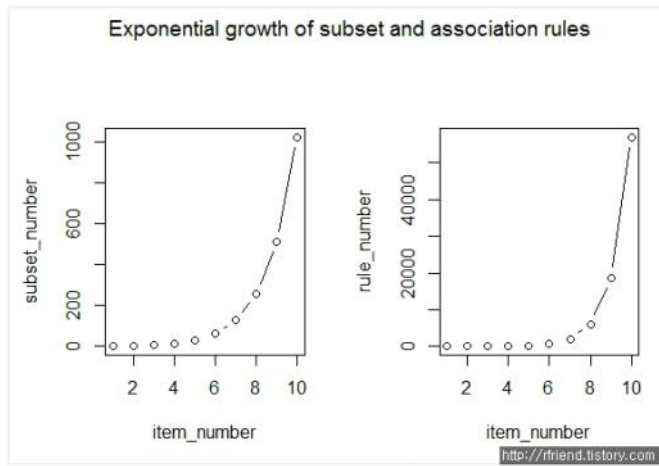
그리고 모든 가능한 연관규칙의 개수는

$$\text{number of rules} = 3^k - 2^{k+1} + 1$$

입니다.

이를 그래프로 나타내면 아래와 같은데요, 가능한 부분집합의 개수나 연관규칙의 개수가 item 이 증가할 때 마다 지수적으로 증가함을 알 수 있습니다.





5개의 원소 항목 집합 $I = \{A, B, C, D, E\}$ 일 때,

모든 가능한 항목집합 = $2^k - 1 = 2^5 - 1 = 32 - 1 = 31$

1-항목집합 : 5개

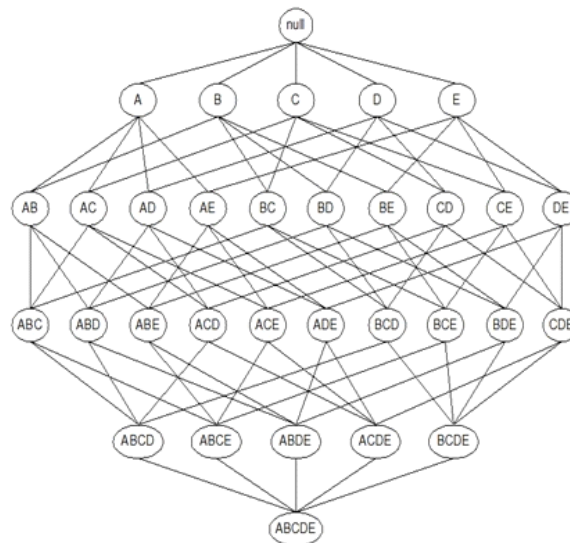
2-항목집합 : 10개

3-항목집합 : 10개

4-항목집합 : 5개

5-항목집합 : 1개

총 계 31개



[R 분석과 프로그래밍] <http://rfriend.tistory.com>

연관규칙에서 사용하는 두가지 통계 척도가 무엇인가?

1. 지지도 : 데이터에서 발생하는 빈도
2. 신뢰도 : 예측능력이나 정확도의 측정치

우유를 x라고 하고 시리얼을 y라고 하면 x와 y의 지지도와 신뢰도를 구하는데 모든 아이템들에 대해서 다 지지도와 신뢰도를 구한다. 그 중에 최소 지지도 이상인데이터만 필터링하고서 필터링 된 것 중에 신뢰도가 가장 좋은 것을 찾는다.

문제 328. 맥주와 기저귀 판매 목록 데이터를 가지고 기저귀를 사면 맥주를 산다는 연관규칙을 발견 하시오.

1. 데이터를 로드한다
2. arules 패키지를 로드한다.
3. apriori 함수를 이용해서 연관관계를 분석한다.

1. 데이터 로드

```
x <- data.frame(
  beer=c(0,1,1,1,0),
  bread=c(1,1,0,1,1),
  cola=c(0,0,1,0,1),
  diapers=c(0,1,1,1,1),
  eggs=c(0,1,0,0,0),
  milk=c(1,0,1,1,1) )
```

2. arules 패키지를 설치 및 로드한다.

```
install.packages("arules")
library(arules)

trans<-as.matrix(x,"Trasaction")
trans
```

3. apriori 함수를 이용해서 연관관계를 분석한다.

```
rules1<-apriori(trans, parameter = list(supp=0.2,conf=0.6, target='rules'))
rules1
```

inspect(sort(rules1))# support 지지도, confidence 신뢰도, lift 상관관계를 나타냄

#(연관규칙을 평가하는 지수는 지지도, 신뢰도 말고도 무수히 많은데 그중에 꽤 많이 쓰는게 신뢰도이다.),

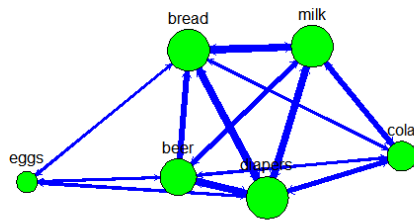
#신뢰도가 클수록 연관관계가 높다는 의미

```
> inspect(sort(rules1))
  lhs      rhs      support confidence lift      count
[1] {}      => {milk}    0.8      0.8000000 1.0000000 4
[2] {}      => {bread}   0.8      0.8000000 1.0000000 4
[3] {}      => {diapers} 0.8      0.8000000 1.0000000 4
[4] {}      => {beer}    0.6      0.6000000 1.0000000 3
[5] {beer}   => {diapers} 0.6      1.0000000 1.2500000 3
[6] {diapers} => {beer}    0.6      0.7500000 1.2500000 3
[7] {milk}   => {bread}   0.6      0.7500000 0.9375000 3
[8] {bread}  => {milk}    0.6      0.7500000 0.9375000 3
[9] {milk}   => {diapers} 0.6      0.7500000 0.9375000 3
[10] {diapers} => {milk}    0.6      0.7500000 0.9375000 3
[11] {bread}  => {diapers} 0.6      0.7500000 0.9375000 3
[12] {diapers} => {bread}   0.6      0.7500000 0.9375000 3
[13] {cola}   => {milk}    0.4      1.0000000 1.2500000 2
[14] {cola}   => {diapers} 0.4      1.0000000 1.2500000 2
[15] {beer}   => {milk}    0.4      0.6666667 0.8333333 2
[16] {beer}   => {bread}   0.4      0.6666667 0.8333333 2
[17] {cola,milk} => {diapers} 0.4      1.0000000 1.2500000 2
[18] {cola,diapers} => {milk}    0.4      1.0000000 1.2500000 2
[19] {diapers,milk} => {cola}    0.4      0.6666667 1.6666667 2
[20] {beer,milk} => {diapers} 0.4      1.0000000 1.2500000 2
[21] {beer,diapers} => {milk}    0.4      0.6666667 0.8333333 2
[22] {diapers,milk} => {beer}    0.4      0.6666667 1.1111111 2
[23] {beer,bread} => {diapers} 0.4      1.0000000 1.2500000 2
```

문제 329. 문제 328의 결과를 시각화 하시오.

```
install.packages("sna")
install.packages("rgl")

#visualization
b2 <- t(as.matrix(trans)) %*% as.matrix(trans)
library(sna)
library(rgl)
b2.w <- b2 - diag(diag(b2))
#rownames(b2.w)
#colnames(b2.w)
gplot(b2.w, displaylabel=T, vertex.cex=sqrt(diag(b2)), vertex.col = "green", edge.col="blue", boxed.labels=F,
arrowhead.cex = .3, label.pos = 3, edge.lwd = b2.w*2)
```



문제 330. 상가 건물 데이터의 연관성 분석을 하고 시각화를 하시오.
= 건물 상가에 서로 연관이 있는 업종이 무엇인가 ?
ex) 건물에 병원이 있으면 약국이 있는지 ?

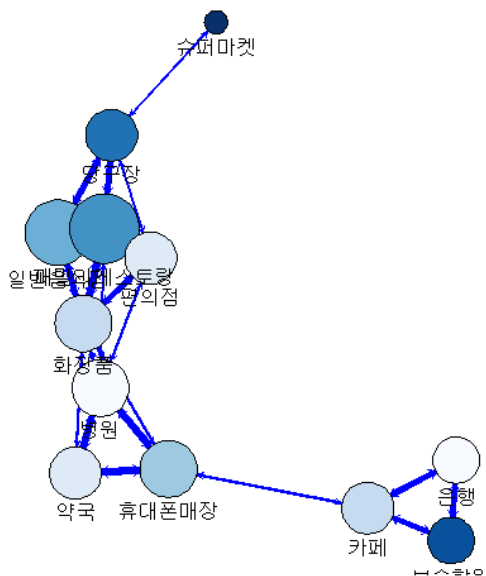
보습학원이 있는 건물에는 어떤 업종의 매장이 있는지를 알아내고 시각화 하시오.

```
building <- read.csv("c:\\data\\building.csv",header = T)
building[is.na(building)]<-0
```

```
trans2<-as.matrix(building[-1],"Trasaction")
rules2<-apriori(trans2, parameter = list(supp=0.2,conf=0.6, target='rules'))
inspect(sort(rules2))
```

```
b3 <- t(as.matrix(trans2)) %*% as.matrix(trans2)
library(sna)
library(rgl)
b3.w <- b3 - diag(diag(b3))
```

```
gplot(b3.w , displaylabel=T, vertex.cex=sqrt(diag(b3)), vertex.col = blues9, edge.col="blue",
      boxed.labels=F, arrowhead.cex = .3 , label.pos = 1 , edge.lwd = b3.w*1)
```



문제 331. Place 변수에 특수문자 \$를 추가하고 아래의 str_replace_all로 필터링 되는지 확인 하시오.

```
x<-c('가',12,'ab','@','$')
y <- str_replace_all(x,"[^:alpha:]", "") #알파벳,한글이 아닌것을 null로 바꾼다.
y
```

```
> x<-c('가',12,'ab','@','$')
> y <- str_replace_all(x, "[^[:alpha:]]","") #알파벳이 아닌것을 null로 바꾼다.
> y
[1] "가" "" "ab" "" ""
```

문제 332. 아래의 연관관계를 오전에 배운 방법으로 시각화 하시오.

```
#1. 상담일지 (advice.csv) 를 텍스트 마이닝한다.
install.packages("KoNLP") # 한글 데이터를 텍스트 마이닝
install.packages("tm") # 한글 데이터를 텍스트 마이닝
install.packages("stringr") # 텍스트 마이닝 지원

library(KoNLP)
library(tm)
library(stringr)

useSejongDic() # 세종사전 가져오는 작업
advice <- read.csv("c:\\data\\advice.csv",header=T,
                  stringsAsFactors=F)

# 명사만 추출하는 작업
place <- sapply(advice[,2],extractNoun, USE.NAMES=F)
c <- unlist(place)

# 철자가 2글자에서 5글자 사이인것만 추출한다.
place2 <- Filter(function(x) { nchar(x) >= 2 & nchar(x) <= 5 } , c)

# 숫자를 지우는 작업
res <- str_replace_all(place2, "[^[:alpha:]]","") #알파벳,한글이 아닌것을 null로 바꾼다.

# "" 를 삭제하는 작업
res <- res[res != ""]

# 위의 단어들로 워드 클라우드를 그린다.

wordcount <- table(res)
wordcount2 <- sort( table(res), decreasing=T)

wordcount2

library(wordcloud)
library(RColorBrewer)
palette <- brewer.pal( 8, "Set2")

wordcloud()

wordcloud(names(wordcount), freq=wordcount,
          scale=c(3,1), rot.per=0.25, min.freq=1,
          random.order=F, random.color=T,colors=palette)

# 자주 나오는 단어들만 추출해서 keyword 라는 변수에 입력

keyword <- dimnames(wordcount2[1:10])$res
keyword

# 단어를 연관관계 분석을 하겠금 표형태로 정재하는 작업
contents <- c()
for(i in 1:6) {
  inter <- intersect(place[[i]] , keyword)
```

```

contents <- rbind(contents,table(inter)[keyword])
}

rownames(contents) <- advice$DATE
colnames(contents) <- keyword
contents[which(is.na(contents))] <- 0
contents

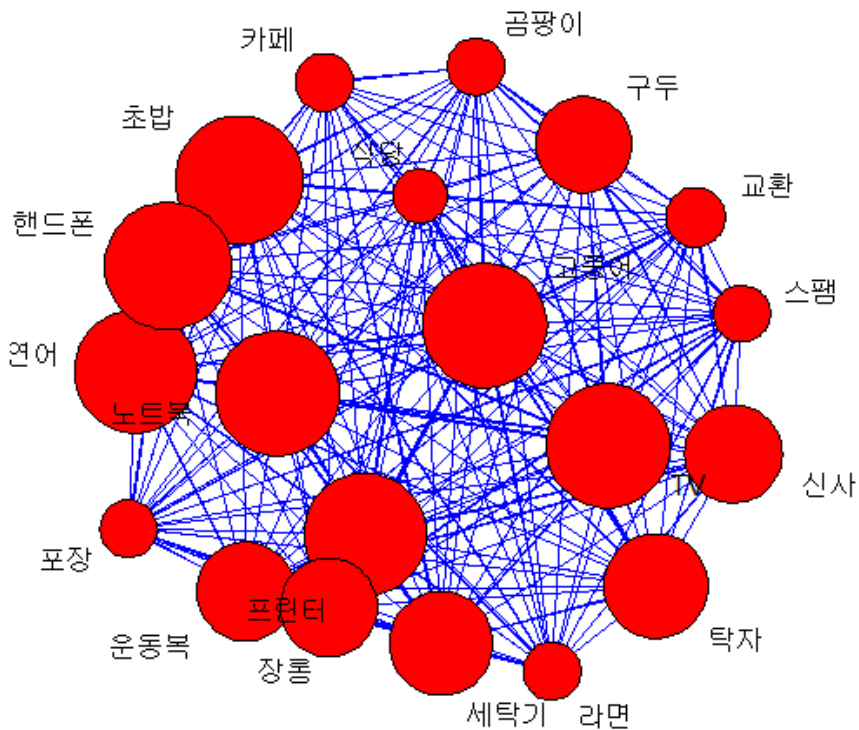
# 큰 상담일지를 정리한 advice2.csv 를 불러온다.
advice2 <- read.csv("c:\\data\\advice2.csv", header = T, stringsAsFactors = F)
rownames(advice2) <- advice2[,1]
advice2 <- advice2[-1]
advice3 <- ifelse(advice2 > mean(apply(advice2, 2, mean)), 1, 0)
advice3

cor(advice2)
install.packages("corrgram")
library(corrgram)
corrgram(cor(advice2))

library(corrplot)
corrplot(cor(advice2))

library(sna)
library(rgl)
advice_square <- t(as.matrix(advice2)) %*% as.matrix(advice2)
gplot(sqrt(sqrt(advice_square)), displaylabel=T, vertex.cex=sqrt(diag(advice_square))*0.01,
label=rownames(advice_square), edge.col="blue", boxed.labels=F, arrowhead.cex=0.01, edge.lwd=0.01, vertex.alpha=
0.01)

```



문제 333. 영화 라라랜드의 긍정적인 평가와 부정적인 평가에 대한 키워드를 워드 클라우드로 그리시오.

```

library(KoNLP)
library(wordcloud)

lala <- read.csv('c:\\data\\라라랜드.csv', header=T, stringsAsFactors = F)
lala_positive <- lala[lala$score>=9,c('content')]

```

```

lala_negative <- lala[lala$score<=2,c('content')]

head(lala_positive)
head(lala_negative)

#긍정 게시판 변수에서 명사만 추출하고 데이터 정제 작업을 한다.
po <- sapply(lala_positive, extractNoun, USE.NAMES=F)
po2 <- unlist(po)
po2 <- Filter(function(x){nchar(x)>=2},po2)

#너무 자주나오는 단어들을 제거버린다.
po3 <- gsub("\\d+", "", po2)
po3 <- gsub('관람객', "", po3)
po3 <- gsub('평점', "", po3)
po3 <- gsub('영화', "", po3)
po3 <- gsub('진짜', "", po3)
po3 <- gsub('완전', "", po3)
po3 <- gsub('시간', "", po3)
po3 <- gsub('올해', "", po3)
po3 <- gsub('장면', "", po3)
po3 <- gsub('남자', "", po3)
po3 <- gsub('여자', "", po3)
po3 <- gsub('만큼', "", po3)
po3 <- gsub('니가', "", po3)
po3 <- gsub('년대', "", po3)
po3 <- gsub('옆사람', "", po3)
po3 <- gsub('들이', "", po3)
po3 <- gsub('저녁', "", po3)

write(unlist(po3), 'c:\\data\\lala_positive.txt')

po4 <- read.table('c:\\data\\lala_positive.txt')
po_wordcount <- table(po4)

#라라랜드 영화에 부정적인 평가 게시글들을 명사로 변경하고 정제 작업을 수행한다.
ne <- sapply(lala_negative, extractNoun, USE.NAMES=F)
ne2 <- unlist(ne)
ne2 <- Filter(function(x){nchar(x)>=2},ne2)
ne3 <- gsub("\\d+", "", ne2)
ne3 <- gsub('관람객', "", ne3)
ne3 <- gsub('평점', "", ne3)
ne3 <- gsub('영화', "", ne3)
ne3 <- gsub('진짜', "", ne3)
ne3 <- gsub('완전', "", ne3)
ne3 <- gsub('시간', "", ne3)
ne3 <- gsub('올해', "", ne3)
ne3 <- gsub('장면', "", ne3)
ne3 <- gsub('남자', "", ne3)
ne3 <- gsub('여자', "", ne3)
ne3 <- gsub('만큼', "", ne3)
ne3 <- gsub('니가', "", ne3)

```

```

ne3 <- gsub('년대', '', ne3)
ne3 <- gsub('옆사람', '', ne3)
ne3 <- gsub('들이', '', ne3)
ne3 <- gsub('저녁', '', ne3)

write(unlist(ne3), 'c:\\data\\lala_negative.txt')
ne4 <- read.table('c:\\data\\lala_negative.txt')
ne_wordcount <- table(ne4)

#긍정단어와 부정단어를 각각 워드클라우드로 한 화면에 출력한다.
graphics.off()
palette <- brewer.pal(9,'Set1')
par(new=T, mfrow=c(1,2))

wordcloud(names(po_wordcount), freq=po_wordcount, scale=c(3,1), rot.per=0.1, random.order = F,
  random.color = T, col=rainbow(15))
title(main='라라랜드의 긍정적인 평가', col.main='blue')

wordcloud(names(ne_wordcount), freq=ne_wordcount, scale=c(3,1), rot.per=0.1, random.order = F,
  random.color = T, col=rainbow(15))
title(main='라라랜드의 부정적인 평가', col.main='red')

```

라라랜드의 긍정적인 평가



라라랜드의 부정적인 평가



문제 334. 라라랜드의 긍정적 평가 게시판의 글들을 명사만 추출한 다음 단어들간의 연관관계를 출력 하시오.

```

library(KoNLP)
library(tm)
library(stringr)
library(arules)

lala <- read.csv('c:\\data\\라라랜드.csv', header=T, stringsAsFactors = F)
lala_positive <- lala[lala$score>=9,c('content')]
lala_positive <- sapply(lala_positive, extractNoun, USE.NAMES=F)
head(lala_positive)

c <- unlist(lala_positive)
lala_positive2 <- Filter(function(x) { nchar(x) >= 2 &
  nchar(x) <= 5 }, c)
lala_positive2 <- gsub("\\d+", "", lala_positive2)
lala_positive2 <- gsub('관람객', "", lala_positive2)
lala_positive2 <- gsub('평점', "", lala_positive2)
lala_positive2 <- gsub('영화', "", lala_positive2)

```



```

lala_positive2 <- gsub('진짜', "", lala_positive2)
lala_positive2 <- gsub('완전', "", lala_positive2)
lala_positive2 <- gsub('시간', "", lala_positive2)
lala_positive2 <- gsub('올해', "", lala_positive2)
lala_positive2 <- gsub('장면', "", lala_positive2)
lala_positive2 <- gsub('남자', "", lala_positive2)
lala_positive2 <- gsub('여자', "", lala_positive2)
lala_positive2 <- gsub('만큼', "", lala_positive2)
lala_positive2 <- gsub('니가', "", lala_positive2)
lala_positive2 <- gsub('년대', "", lala_positive2)
lala_positive2 <- gsub('옆사람', "", lala_positive2)
lala_positive2 <- gsub('들이', "", lala_positive2)
lala_positive2 <- gsub('저녁', "", lala_positive2)

res <- str_replace_all(lala_positive2, "[^[:alpha:]]", "") #한글 또는 알파벳이 아니면 ""로 바꿈
res <- res[res != ""] # "" 값을 제외

wordcount <- table(res)
wordcount2 <- sort( table(res), decreasing=T) # 오름차순 정렬
keyword <- names( wordcount2[wordcount2>100] ) #값이 100개 이상인 키워드들만 저장

contents <- c() # contents 변수에 null값을 저장 (for문 안에 쓸 변수 선언 용도)
for(i in 1:length(lala_positive)) {
  inter <- intersect(lala_positive[[i]], keyword)
  contents <- rbind(contents,table(inter)[keyword])
}

colnames(contents) <- keyword
contents[which(is.na(contents))<-0]

#100/length(lala_positive)

rules_lala <- apriori(contents, parameter = list(supp = 0.007, conf = 0.3, target = "rules")) ## 지지도 신뢰도
rules_lala
inspect(sort(rules_lala))

> inspect(sort(rules_lala))

```

	lhs	rhs	support	confidence	lift	count
[1]	{영상}	=> {음악}	0.020449109	0.6354916	5.285838	265
[2]	{연기}	=> {음악}	0.016282121	0.5158924	4.291046	211
[3]	{스토리}	=> {음악}	0.016204954	0.4708520	3.916413	210
[4]	{영상미}	=> {음악}	0.014352959	0.4124169	3.430366	186
[5]	{배우}	=> {음악}	0.012655297	0.4795322	3.988612	164
[6]	{배우}	=> {연기}	0.011574967	0.4385965	13.896753	150
[7]	{연기}	=> {배우}	0.011574967	0.3667482	13.896753	150
[8]	{눈빛}	=> {마지막}	0.009414307	0.6777778	8.908035	122
[9]	{연출}	=> {음악}	0.007716645	0.3861004	3.211473	100
[10]	{색감}	=> {음악}	0.007407979	0.4247788	3.533189	96

K 평균 군집화

2018년 6월 12일 화요일 오전 9:49

9장 목차

1. K 평균 군집화 이론수업
2. K 평균 군집화 실습1 (국영수 점수를 가지고 학생분류)
3. K 평균 군집화 실습2 (소셜 미디어에 같은 성향을 갖는 사람들을 분류)

구분	설명	해당 알고리즘
지도학습	훈련 데이터와 정답을 가지고 데이터를 분류/예측하는 함수를 만들어내는 기계학습의 한 방법	분류 : Knn, 나이브베이지, 결정트리, rule-base 알고리즘, 서포트 벡터머신 회귀 (예측) : 선형회귀, 신경망
비지도학습	정답 없이 훈련 데이터만 가지고 데이터로부터 숨겨진 패턴/규칙을 탐색하는 기계학습의 한 방법	클러스터링 : k-menas , 연관규칙 (아프리오 알고리즘)
강화학습	어떤 환경에서 정의된 에이전트가 현재의 상태를 인식하여 선택 가능한 행동들 중 보상을 최대화 하는 행동 혹은 행동 순서를 선택하는 방법	

Ncs 문제 1. K 평균 군집화 알고리즘이란 무엇인가?

k-평균 알고리즘은 주어진 데이터를 k개의 클러스터로 묶는 알고리즘으로, 각 클러스터와 거리 차이의 분산을 최소화 하는 방식으로 동작한다. 이 알고리즘은 자율 학습의 일종으로, 레이블이 달려있지 않는 입력 데이터에 레이블을 달아주는 역할을 수행한다.

Ncs 문제 2. 컴퓨터가 어떻게 클러스터 구성에 대한 사전 지식이 없이 한 그룹이 끝나고 다른 그룹이 시작 하는 곳을 어떻게 알 수 있을까? 즉 컴퓨터가 어떻게 레이블 없는 데이터의 군집화를 가능하게 할까?

클러스터 안에 있는 아이템들은 서로 아주 비슷해야 하지만 클러스터 밖에 있는 아이템들과는 아주 달라야 한다는 원칙을 따르면 가능하다.

Ncs 문제 3. K평균 알고리즘의 목표는 무엇인가?

클러스터 내의 차이를 최소화하고, 클러스터 간의 차이를 최대화 하는 것이다.

Ncs 문제 4. Knn 과 k-평균의 공통점과 차이점은 ?

공통점 : 거리함수를 이용해서 중심에 가까운 거리에 있는 데이터를 클러스터링 한다.

차이점 : knn은 라벨이 있고 k-means는 라벨이 없다.

k-means 기본 실습

#1. 기본 데이터 셋을 만든다.

```
c <- c(3,4,1,5,7,9,5,4,6,8,4,5,9,8,7,8,6,7,2,1)
row <- c("A","B","C","D","E","F","G","H","I","J")
col <- c("X","Y")
data <- matrix( c, nrow= 10, ncol=2, byrow=TRUE, dimnames=list(row,col))
```

#2. 위에서 만든 데이터셋을 그린다.

```
plot(data)
```

#3. k-means 패키지를 설치한다.

```
install.packages("stats")
library(stats)
```

#4. k-means 함수로 데이터를 분류한다.

k 개 구하는 공식 : $k = \sqrt{n/2}$

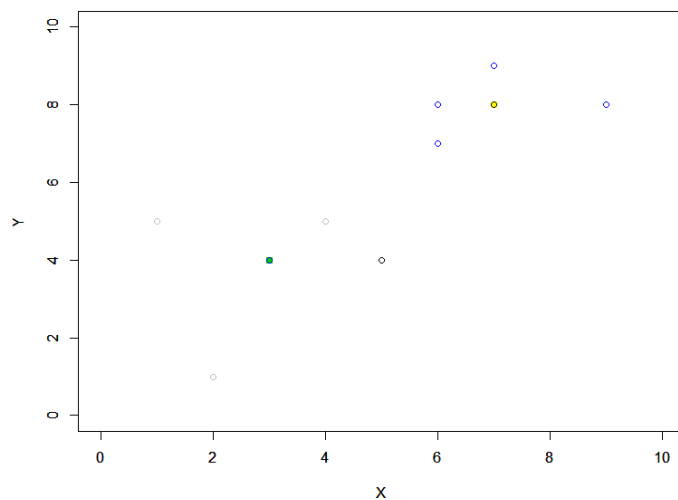
```
km <- kmeans(data,2)
km
```

#5. 분류한 파라미터값을 가지고 다시 1번 시각화한다.

```
km$centers # 클러스터의 중앙점
cbind(data,km$cluster)
plot(round(km$centers), col=km$centers, pch=22, bg=km$centers, xlim=range(0:10), ylim = range(0:10))
```

#6. 원래 데이터를 그린 plot 그래프와 위의 그래프를 합쳐서 출력한다.

```
plot(round(km$centers), col=km$centers, pch=22, bg=km$centers, xlim=range(0:10), ylim = range(0:10))
par(new=T)
plot(data,col=km$centers+1,xlim = range(1:10),ylim = range(1:10))
```

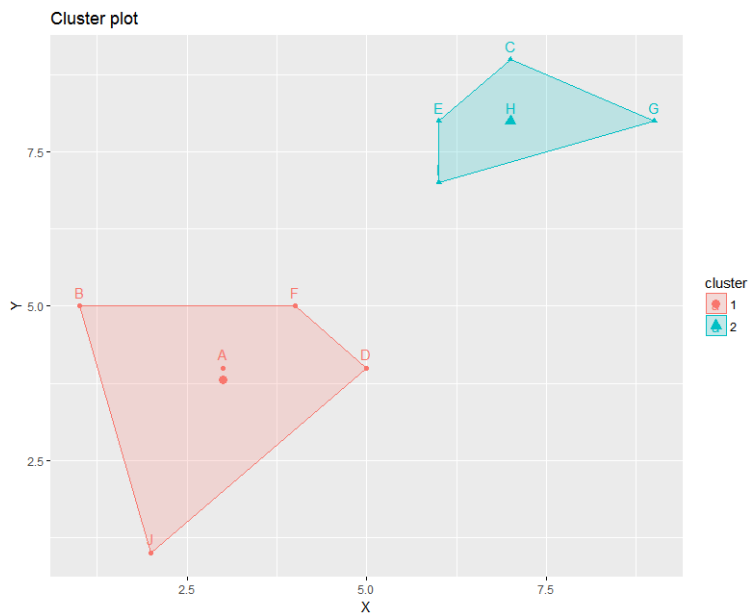


문제 335. 카페에 올라와져 있는 k-means 시각화 패키지(factoextra)를 이용해서 위의 data를 시각화 하시오.

```
install.packages("factoextra")
```

```
library(factoextra)
```

```
km <- kmeans(academy, 4)
fviz_cluster(km, data = data, stand = F)
```

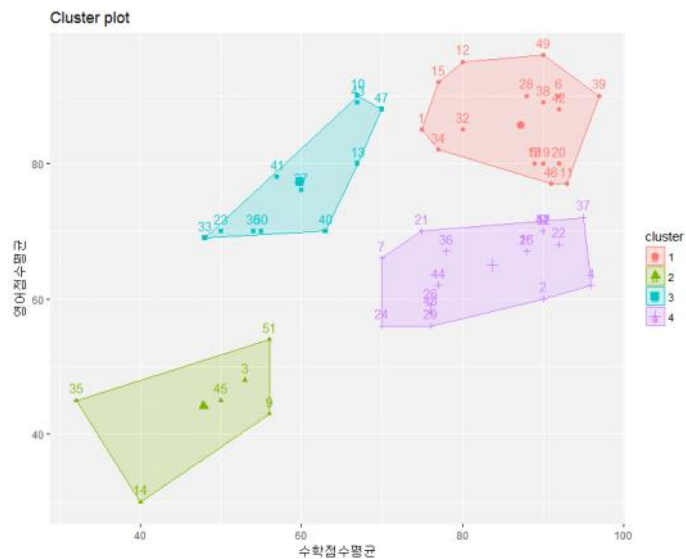


문제 336. 카페에 올라와져 있는 k-means 시각화 스크립트를 이용하여 위의 데이터를 시각화 하시오.

문제 337. Academy.csv 데이터를 k값을 4로 두고 시각화를 하시오.

```
academy<-read.csv("c:\\data\\academy.csv",header = T)
```

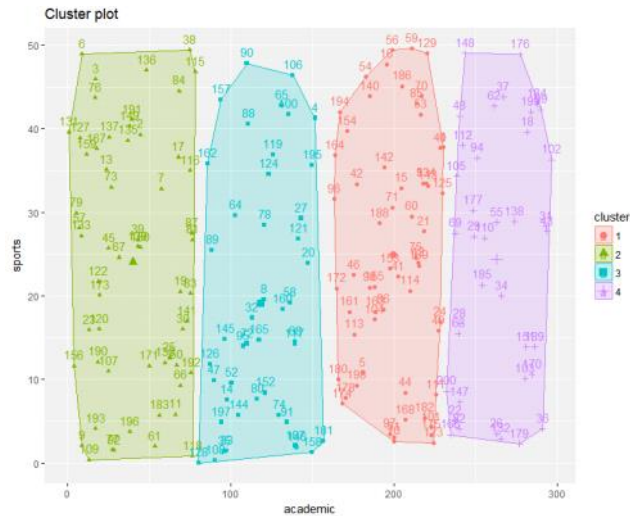
```
head(academy)
km2 <- kmeans(academy[,c(3,4)],4)
fviz_cluster(km2, data = academy[,c(3,4)], stand = F)
```



문제 338. 미국대학 입학 점수를 가지고 academic 점수와 sport 점수를 각각 x, y축으로 두고 분류 하시오. (k : 4)

```
sports<-read.csv("c:\\data\\sports.csv",header = T)
head(sports)
```

```
km2 <- kmeans(sports[,c(2,3)],4)
fviz_cluster(km2, data = sports[,c(2,3)], stand = F)
```



3. K-평균 실습데이터 (sns)

#1. 데이터를 로드한다.

```
teens<-read.csv("c:\\data\\snsdata.csv",header = T)
str(teens)
head(teens)
```

#2. 성별이 남자가 몇명이고 여자가 몇명인지 확인한다

```
table(teens$gender)
```

#3. 성별에 na가 몇개인지도 출력되게 하시오.

```
table(teens$gender, useNA = 'ifany')
```

#4. 연령을 이상하게 입력한 데이터를 na처리한다. (13~19살만 정상이라고 본다)

```
teens$age <- ifelse(teens$age >= 13 & teens$age <20, teens$age, NA)
summary(teens$age)
```

```
teens$female<-ifelse(teens$gender=="F" & !is.na(teens$gender),1,0)
teens$no_gender<-ifelse(is.na(teens$gender),1,0)
head(teens)
```

결측치인 나이

```
ave_age<-ave(teens$age,teens$gradyear,FUN=function(x) mean(x, na.rm=TRUE))
teens$age<-ifelse(is.na(teens$age),ave_age, teens$age)
summary(teens$age)
```

sns에 나타났던 관심사 횟수를 표현하는 36개의 수치형 데이터 컬럼을 정규화 시킨다.

```
interests <- teens[5:40]
interests_z <- as.data.frame(lapply(interests,scale))
head(interests_z)
```

```
# kmeans 함수로 5개의 클래스로 분류한다.

set.seed(2345)
teen_clusters<-kmeans(interests_z,5)

# 각 클래스의 갯수가 각각 어떻게 되는지 확인하시오...

teen_clusters$size

# 클러스터의 중심점의 좌표를 확인한다.

teen_clusters$centers
```

문제 339. 소개팅 데이터를 kmeans로 분석해서 label과 정확도가 일치하는지 확인 하시오.

```
like_n<-like[,1:7]
like_model <- kmeans(like_n,3)

cbind(like[,8],like_model$cluster)

> cbind(like[,8],like_model$cluster)
      [,1] [,2]
[1,]    1    1
[2,]    2    2
[3,]    2    2
[4,]    3    3
[5,]    1    1
[6,]    3    3
[7,]    1    1
[8,]    2    2
[9,]    3    3
[10,]   1    1
[11,]   3    3
```

문제 340. 동물 데이터를 가지고 k-mean 머신러닝 기법을 수행해서 동물 데이터의 라벨과 k-mean의 클러스터가 일치하는지 확인해보시오.

```
zoo <- read.csv("c:\\data\\zoo.csv",header = T)

zoo_model<-kmeans(zoo[,2:17],7)
cbind(zoo[, "type"],zoo_model$cluster)

> cbind(zoo[, "type"],zoo_model$cluster)
      [,1] [,2]
[1,]    1    1
[2,]    1    1
[3,]    4    3
[4,]    1    1
[5,]    1    1
[6,]    1    1
[7,]    1    1
[8,]    4    3
[9,]    4    3
```

문제 341. 부도여부 데이터를 kmeans 머신러닝 기법으로 분류해서 라벨과 일치하는지 확인해 보시오.

성능평가

2018년 6월 14일 목요일 오전 9:48

목차

1. 혼동 행렬을 사용한 성능 척도
2. 카파 통계량
3. 민감도와 특이도
4. 정밀도와 재현율
5. 성능 트레이드 오프 시각화(ROC 곡선)

Ncs 문제 1. 모델 성능 평가가 중요한 이유가 무엇인가?

머신러닝(학생)이 수행한 결과(분류, 예측)에 대한 공정한 평가를 통해 머신러닝(학생)이 앞으로도 미래의 데이터에 대해 더 잘 분류하고 예측할 수 있도록 해주고, 분류결과가 요행수로 맞힌게 아니다라는 것을 확실하게 해주며, 분류 결과를 좀 더 일반화 할 수 있기 때문이다.

Ncs 문제 2. 정확도란 무엇인가?

학습자가 맞거나 틀린 경우의 비율을 말한다.

Ncs 문제 3. 정확도가 성능을 측정하는데 충분치 않은 이유를 설명하고 대신 사용할 수 있는 성능척도가 무엇이 있는지 기술 하시오.

암을 판정하는 분류기가 99%의 정확도를 갖고 있다고 하면, 1%의 오류율이 있기 때문에 어떤 데이터에 대해서는 오류를 범할 수 도 있게 된다. 그래서 정확도만으로는 성능을 측정하는데 충분치 않다. 그래서 정확도 만으로는 성능을 측정하는데 충분치 않다. 정확도와 더불어서 분류기의 유용성에 대한 성능 척도를 정의 하는 것이 중요하다. (정확도 + 다른 성능 척도)

Ncs 문제 4. 카파 통계량이 무엇인지 설명하고 ham과 spam의 카파 통계량을 출력 하시오.

카파 통계량은 우연히 정확한 예측을 할 가능성을 설명함으로써 **정확도를 조정한다.**
카테고리 정보에 대한 2명의 평가자의 일치도를 측정하는 통계적 지표를 의미한다.

$$K = (pr(a) - pr(e)) / (1 - pr(e))$$

$$Pr(a) = (1203+152) / (1203+152+31+4) = 0.9748$$

$$Pr(e) =$$

평가자 a

$$Ham : ((1203 + 4) / (1203+152+31+4)) = 0.8683$$

$$Spam : ((31+152) / (1203+152+31+4)) = 0.1317$$

평가자 b

Ham : $((1203 + 31) / (1203+152+31+4)) = 0.8878$

Spam : $((4 + 152) / (1203+152+31+4)) = 0.1122$

$Pr(e) = (0.8683 * 0.8878) + (0.1317 * 0.1122) = 0.7857$

$K = (0.9748 - 0.7857) / (1 - 0.7857) = 0.8824$

예시.	카파 통계량이 무엇인지 설명하고 예시를 통해 카파 통계량을 구하시오.
-----	--

카파 통계량은 우연히 정확한 예측을 할 가능성을 설명함으로써 정확도를 조정한다.

평가자 b/a	합격	불합격
합격	40	10
불합격	20	30

설명 : 평가자 a와 b 모두 40명에게 합격을, 30명에게 불합격을 주었다.

$Pr(a)$ 는 2명의 평가자들이 일치할 확률이므로 0.7이 된다. (실제 확률)

$Pr(a) = 0.7$

$Pr(e)$ 를 구하기 위해서는 평가자 a와 b의 각각 합격과 불합격을 줄 확률을 구해야 한다.

- 평가자 a : 합격을 60번, 불합격을 40번 주었다.
합격을 $60/100 = 0.6$, 60% 확률
불합격을 $40/100 = 0.4$, 40% 확률
- 평가자 b : 합격을 50번, 불합격을 50번 주었다.
합격을 $50/100 = 0.5$, 50% 확률
불합격을 $50/100 = 0.5$, 50% 확률
- 평가자 a와 b 둘 모두 **확률적으로** '합격'을 줄 확률을 $0.6 * 0.5 = 0.3$,
불합격을 줄 확률은 $0.4 * 0.5 = 0.2$ 이므로

$Pr(e) = 0.3 + 0.2 = 0.5$

정확도 : $70/100 = 0.7$ ---> 70%의 정확도

카파 통계량 = $(pr(a) - pr(e)) / (1 - pr(e)) = 2 / 5 = 0.4$ ----> 카파지수 0.4 (보통 일치)

■ 민감도와 특이도

재현율과 민감도가 같은 대상

예측된 값

	NO	YES
NO	TN	FP
YES	FN	TP

특이도 : 실제 거짓인것 중에 거짓으로 예측한 것
Ex. 암이 아닌데 암이 아니다 라고 예측

$$\text{특이도} = \text{TN} / (\text{TN} + \text{FP})$$

민감도 : 실제 참인 것 중에 참으로 예측한 것
Ex. 암인데 암으로 예측한 것

$$\text{민감도} = \text{TP} / (\text{TP} + \text{FN})$$

병원이라면 특이도보다 민감도가 높은 모델을 선택하는 것이 좋다.

정밀도? 전체 예측한 것들 중에 암으로 예측한 것의 비율

$$\text{정밀도} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

재현율? 실제 암 환자중에서 암으로 예측한 것의 비율

$$\text{재현율} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

TP	FP
FN	TN

예측

소극적 예측 : 암이라고 판단하는 것 자체를 소극적으로 봐서 확실한 경우가 아니면 암으로 판단 안 하는것.

정밀도 ↑, 재현율 ↓ --> 안좋은 모델

ex) 암인 사람을 암이라고 판단을 잘 안한다.

공격적 예측 : 조금만 의심이 가도 다 암이라고 한다.

정밀도 ↓, 재현율 ↑

ex) 암인데 암이라고 판단하면되고 암이 아닌 사람들은 재검을 받으면 된다.

	ham	spam
ham	1203	31
spam	4	152

#스탬 정밀도

1203/1234

```
> 1203/1234  
[1] 0.9749
```

#스탐 정밀도

1203/1234

```
> 1203/1234  
[1] 0.9749
```

#스탐 재현율

1203/1207

```
> 1203/1207  
[1] 0.9967
```

성능개선

2018년 6월 15일 금요일 오전 9:54

Ncs 문제 1. 성능개선을 위한 방법에는 무엇이 있는가?

1. Caret 패키지의 자동 파라미터 튜닝
2. 앙상블 기법
 - Bagging
 - Random forest
 - Boosting

Ncs 문제 2. 정확도를 올리기 위한 방법에 질문을 3가지는 무엇인가?

1. 데이터에 대해 어떤 종류의 머신러닝 모델을 사용할 것인가?
2. 해당 모델에 대해서 파라미터 튜닝은 어떻게 할 것인가?
Ex. Knn의 k 값 파라미터
3. 데이터를 가지고 여러 모델을 만들었을 때 모델을 평가하는데 어떤 기준을 사용할 것인가?
= 정확도, 카파 상관계수 ..

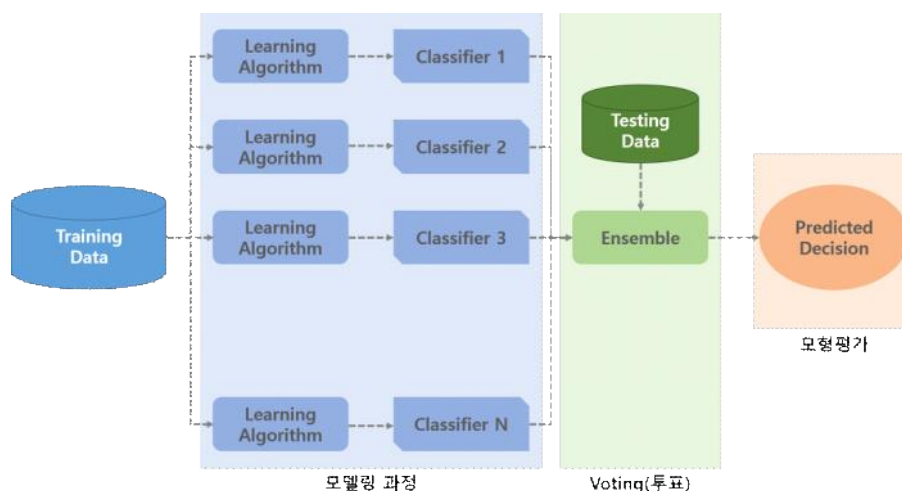
Ex. 정밀도 ↓, 재현율 ↑

병원의 경우는 공격적 예측 : 조금만 의심이 가도 암이라고 판단.

Ncs 문제 3. 머신러닝에서 앙상블이란 무엇인가?

약한 학습자 여러 개를 결합해서 보다 더 나은 강한 학습자를 만드는 기법

1. Bagging
2. Boosting



예제 : 정확도가 60% 밖에 되지 않는 분류기 모형들이 준비한데 이 모형들을 여

러 개 모아서 한꺼번에 평가하는 모형을 만드는데 90% 이상을 능가하게 만들려면 60% 분류기들을 최소 몇 개를 사용해야 하는가?

```
ret_err <- function(n,err) {
  sum <- 0
  for(i in floor(n/2):n) {
    sum <- sum + choose(n,i) * err^i * (1-err)^(n-i)
  }
  sum
}

for(j in 1:60) {
  err <- ret_err(j, 0.4)
  cat(j,'--->',1-err,'\n')
  if(1-err >= 0.9) break
}

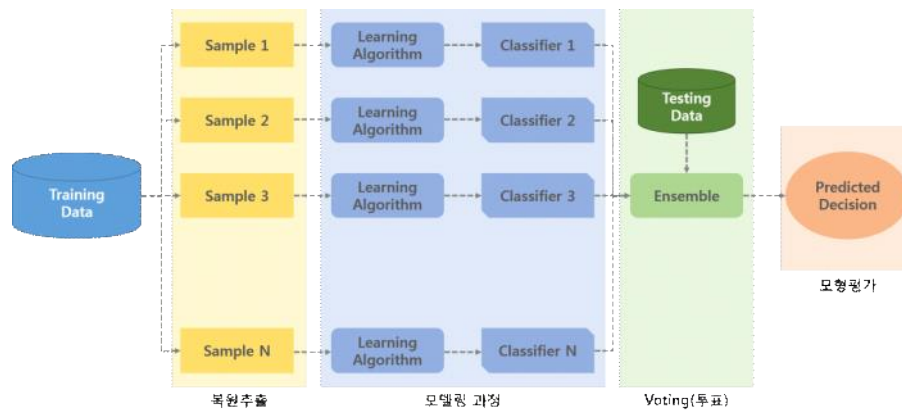
29 ---> 0.7658968
30 ---> 0.8246309
31 ---> 0.7805803
32 ---> 0.8352031
33 ---> 0.7940754
34 ---> 0.8449709
35 ---> 0.8065165
36 ---> 0.854019
37 ---> 0.8180171
38 ---> 0.8624195
39 ---> 0.8286737
40 ---> 0.8702343
41 ---> 0.8385691
42 ---> 0.8775173
43 ---> 0.847775
44 ---> 0.8843155
45 ---> 0.8563541
46 ---> 0.8906704
47 ---> 0.8643612
48 ---> 0.8966186
49 ---> 0.8718449
50 ---> 0.9021926
```

Ncs 문제 4.	배깅(bagging)이란 무엇인가?
-----------	---------------------

Bagging은 샘플을 여러 번 뽑아 각 모델을 학습시켜 결과를 집계하는 방법이다.

1. 훈련 데이터에서 샘플 데이터를 수집한다. (복원추출)
2. 샘플 개수만큼 동일한 알고리즘으로 모델을 학습시킨다.
3. 생성된 여러 개의 모델 중에서 투표를 통해서 최고의 모델을 선택한다.
4. 테스트 데이터로 성능 평가를 한다.

대표적인 bagging 알고리즘으로는 랜덤 포레스트 모델이 있다.



Randomforest ? 앙상블 기법 + Decision tree

■ random forest 실습

Kyphosis 데이터는 성형외과에서 아이들이 척추 수술후에 얼마만에 증상이 사라졌는지 아니면 그대로 존재하는지에 대한 데이터로서 독립변수가 처음 수술한 척추의 수와 관련된 척추의 수

Ncs 문제 4. 부스팅(boosting)이란 무엇인가?

앙상블 기법 중 하나인데, 배깅하고 원리는 같지만 다른 점은 잘 못 맞춘 문제에 대해 가중치를 부여해서 모델을 생성한다.

예 : 수학문제를 푸는데 9번 문제가 어려워서 계속 틀렸다고 가정하면, boosting 은 9번문제에 가중치를 부여해서 9번 문제를 잘 못 맞춘 모델을 최종 모델로 선정한다.

boosting 실습

```
install.packages("adabag")
library(adabag)

credit<-read.csv("c:\\data\\credit.csv")

set.seed(300)

model_ada <- boosting(default~, data = credit)
predict_ada <- predict(model_ada, credit)

#예측결과 확인
predict_ada #<-- train , test 안나누고 돌렸는데 100% 적중률

$confusion
      observed class
Predicted class no yes
no      700    0
yes      0   300
```

문제 343. heart.csv 데이터로 자동 파라미터 튜닝 했을 때와 안 했을 때의 정확도를 비교 하시오.
(자동 파라미터 튜닝 안했을 때)

```
heart <- read.csv("c:\\data\\Heart.csv",header = T)

# 정보획득량
library(FSelector)
information.gain(AHD ~ ., heart[-1])

# 데이터 준비
set.seed(1234)
heart_idx <- sample(2, nrow(heart), prob = c(0.7, 0.3), replace = T)

head(heart_idx)
head(heart_train)

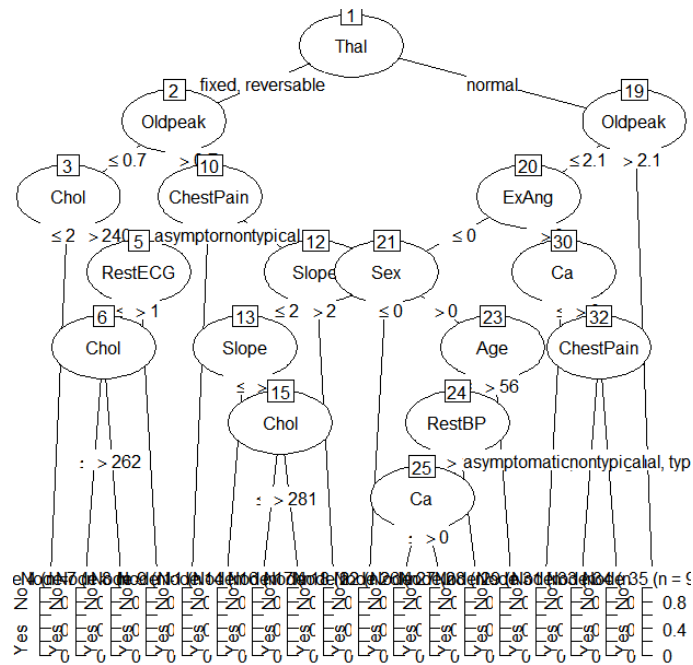
heart_train <- heart[heart_idx == 1, -1]
heart_test <- heart[heart_idx == 2, -1]

# 모델 훈련
library(C50)
heart_model <- C5.0(heart_train[, -14], heart_train[, 14], trial = 10)
heart_result <- predict(heart_model, heart_test[, -14])

# 정확도
library(gmodels)
CrossTable(heart_result, heart_test[, 14])

# 의사결정 트리
plot(heart_model)
```

heart_result	heart_test[, 14]		Row Total
	No	Yes	
No	37	9	46
	6.196	7.150	
	0.804	0.196	0.548
	0.822	0.231	
	0.440	0.107	
Yes	8	30	38
	7.501	8.655	
	0.211	0.789	0.452
	0.178	0.769	
	0.095	0.357	
Column Total	45	39	84
	0.536	0.464	



문제 344. heart.csv 데이터로 자동 파라미터 튜닝 했을 때와 안 했을 때의 정확도를 비교 하시오.
(자동 파라미터 튜닝 했을 때)

```
heart <- read.csv("c:\\data\\Heart.csv",header = T)

# 정보획득량
library(FSelector)
information.gain(AHD ~ ., heart[-1])

# 데이터 준비
set.seed(1234)
heart_idx <- sample(2, nrow(heart), prob = c(0.7, 0.3), replace = T)

head(heart_idx)
head(heart_train)

heart_train

heart_train <- heart[heart_idx == 1, -1]
heart_test <- heart[heart_idx == 2, -1]

library(C50)

# 정확도
library(gmodels)
heart_train<-na.omit(heart_train)
heart_test<-na.omit(heart_test)
m<-train(AHD~.,data=heart_train, method='C5.0')
p<-predict(m,heart_test[, -14])
table(p,heart_test[, 14])

> table(p,heart_test[, 14])

p      No Yes
No    34  8
Yes   11 30
```

문제 345. 부도예측 데이터를 이용해서 C5.0 알고리즘의 자동 파라미터 튜닝 했을 때와 안했을 때의 정확도 차이를 비교 하시오.

```
bankrupt <- read.csv("c:\\data\\부도예측데이터3.csv", header=T)
colnames(bankrupt) <- c("class","매출액","자기자본","총자본투자효율","부가가치
율","매출액증가율","재고자산증가율","총자산증가율","금융비용대매출액비
율","대출효율성계수","매출액순이익률","매출원가율","손익분기점율","순금융비
용대매출액비율","이자보상배율","자기자본순이익률","총자본경상이익률","총자
본순이익률","고정장기적합율의역","단기부채대총차입금","당좌비율","매출채권
대매입채무","순운전자본비율","유동비율","유동부채대총자본","유보액대총자산
비율","자기자본비율","차입금의존도","총차입금대매출액","금융비용부담율증가
분","매입채무회전율","순운전자본대매출액","운전자금대매출액","재고자산회전
율","총자본회전율","현금흐름지표(1)","현금흐름지표(2)","현금흐름지표(3)","현금
흐름지표(4)","현금흐름지표(5)","현금흐름지표(6)","현금흐름지표(7)","현금흐름지
표(8)","현금흐름지표(9)" )
```

```
bankrupt[,1]<-as.factor(bankrupt[,1]) #라벨값은 팩터값이어야한다.
```

```
# 훈련과 테스트 데이터에 대한 무작위 샘플 생성
```

```
# 예제와 같은 무작위 수열을 사용하기 위해 set.seed 사용
```

```
set.seed(12345)
```

```
bankrupt_rand <- bankrupt[order(runif(1200)), ]
```

```
train_cnt <- round(0.7*dim(bankrupt)[1])
```

```
train_idx <- sample(1:dim(bankrupt_rand)[1], train_cnt, replace=F)
```

```
bankrupt_train <- bankrupt_rand[train_idx, ]
```

```
bankrupt_test <- bankrupt_rand[-train_idx,]
```

```
###튜닝전
```

```
bankrupt_model <- C5.0(bankrupt_train[, -1], bankrupt_train[, 1], trial = 10)
```

```
bankrupt_result <- predict(bankrupt_model, bankrupt_test[, -1])
```

```
CrossTable(bankrupt_test[,1], bankrupt_result)
```

bankrupt_test[, 1]	bankrupt_result		Row Total
	0	1	
0	116	52	168
	22.228	17.984	
	0.690	0.310	0.467
	0.720	0.261	
	0.322	0.144	
1	45	147	192
	19.450	15.736	
	0.234	0.766	0.533
	0.280	0.739	
	0.125	0.408	
Column Total	161	199	360
	0.447	0.553	

```
###튜닝후
```

```
m<-train(class~.,data=bankrupt_train, method='C5.0')
```

```
p<-predict(m,bankrupt_test[, -1])
```

```
table(p,bankrupt_test[, 1])
```



```
> table(p,bankrupt_test[ , 1])
```

```
p      0    1
0 119  41
1  49 151
```

문제 346. 독일은행 채무 불이행자를 예측하는 모델의 성능을 boosting을 이용해서 향상 시키시오.

```
# boosting 실습
```

```
install.packages("adabag")
library(adabag)
```

```
credit<-read.csv("c:\\data\\credit.csv")
```

```
credit_shuffle <- credit[sample(nrow(credit)),]
```

```
#5. 데이터를 9대1로 나눈다.
```

```
train_num<-round(0.9*nrow(credit_shuffle),0)
```

```
credit_train <- credit_shuffle[1:train_num,]
```

```
credit_test <- credit_shuffle[(train_num+1):nrow(credit_shuffle),]
```

```
set.seed(300)
```

```
model_ada <- boosting(default~., data = credit_train)
```

```
predict_ada <- predict(model_ada, credit_test)
```

```
#예측결과 확인
```

```
predict_ada  #<-- train , test 만나누고 돌렸는데 100% 적중률
```

```
sum(predict_ada$class == credit_test$default)/NROW(credit_test)
```

```
sum(predict_ada1$class == credit_test$default)/NROW(credit_test)
```

```
sum()
```

```
> sum(predict_ada$class == credit_test$default)/NROW(credit_test)
[1] 0.75
```

특화된 머신러닝 주제

2018년 6월 20일 수요일 오전 10:32

목차

1. 오라클과 R 연동
2. R로 웹스크롤링 하는 방법 (파이썬)
3. 네트워크 데이터 분석과 시각화
4. R에서 병렬처리

책의 예제와 달리 실제 데이터는 그냥 가져와서 사용하는 CSV 형태로 패키징되는 경우는 거의 없다. 분석을 위한 데이터를 준비하는 과정에 상당한 노력이 필요하다. 학습 알고리즘의 요구 사항을 맞추려면 데이터를 수집, 병행, 정렬, 필터링, 포맷해야 한다. 이 과정을 비공식적으로 **데이터 먼징** 또는 **데이터 랭글링** 이라고 부른다.

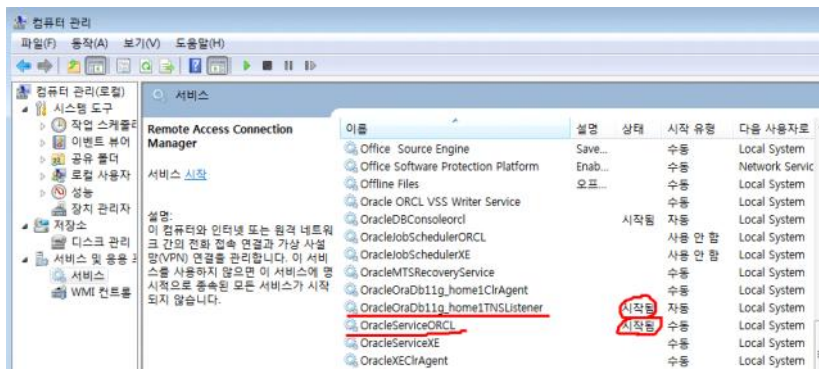
R에서는 rio 패키지를 이용해 다양한 형식에서 데이터를 끌어오고, 결합하고, 내보내기를 할 수 있다.

가져오기	<code>library(rio)</code> <code>credit <- import("credit.csv")</code> # credit.csv 파일을 데이터 프레임으로 가져옴
내보내기	<code>export(credit, "credit.xlsx")</code> # 엑셀형식으로 credit 데이터 프레임을 내보냄
변경하기	<code>convert("credit.csv", "credit.dta")</code> # credit.csv 파일을 credit.dta 형식으로 변경한다

■ 1. 오라클과 R 연동

문제 347.	회사 퇴사율에 가장 영향을 미치는 요소가 무엇인지에 대한 데이터를 오라클 db에 입력 하시오.
---------	--

1. 오라클 서비스를 올린다(orccl 서비스)



2. 오라클 리스너의 상태를 확인한다.

lsnrctl status

3. 오라클 리스너가 다운되었다면 올린다.

lsnrctl start

lsnrctl status

```

C:\Users\Administrator>lsnrctl start

LSNRCTL for 64-bit Windows: Version 11.2.0.1.0 - Production on 20-6월 -2018 10:45:31
Copyright (c) 1991, 2010, Oracle. All rights reserved.

TNS-01106: LISTENER 리스너명을 이용한 리스너는 이미 시작되었습니다

C:\Users\Administrator>lsnrctl status

LSNRCTL for 64-bit Windows: Version 11.2.0.1.0 - Production on 20-6월 -2018 10:45:46
Copyright (c) 1991, 2010, Oracle. All rights reserved.

<DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1522))>에 연결되었습니다
리스너의 상태
-----
명칭          LISTENER
버전          TNSLSNR for 64-bit Windows: Version 11.2.0.1.0 - Production
시작 날짜      20-6월 -2018 09:38:18
업타임        0 일 1 시간. 7 분. 33 초
트래픽 상태    off
로그인         ON: Local OS Authentication
OFF 리스너 매개변수 파일  C:\Users\Administrator\Myproduct\11.2.0\bin\home_1\
                        c:\Users\Administrator\Mydiag\tnslsnr\SPC1\1\listener\Alert\log.xml
리스너 로그 파일
로그 요약 정보 중...
<DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(PIPENAME=\\.\pipe\EXTPROC1522ipc))>
<DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=127.0.0.1)(PORT=1522))>
서비스 요약...
"CLRExtProc" 서비스는 1개의 인스턴스를 가집니다.
"CLRExtProc" 인스턴스(UNKNOWN 상태)는 이 서비스에 대해 1 처리기를 가집니다.
"orcl" 서비스는 1개의 인스턴스를 가집니다.
"orcl" 인스턴스(READY 상태)는 이 서비스에 대해 1 처리기를 가집니다.
"orclXDB" 서비스는 1개의 인스턴스를 가집니다.
"orcl" 인스턴스(READY 상태)는 이 서비스에 대해 1 처리기를 가집니다.
명령이 성공적으로 수행되었습니다

```

- 리스너를 통해서 오라클에 접속이 되는지 확인한다.

```

C:\Users\Administrator>sqlplus scott/tiger@orcl

SQL*Plus: Release 11.2.0.1.0 Production on 수 6월 20 10:47:17 2018

Copyright (c) 1982, 2010, Oracle. All rights reserved.

다음에 접속됨:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

```

- 퇴사율.csv 데이터를 오라클 db에 입력한다.

-----오라클 -----

```

create table hr
( sat_lvl float,
  last_eval float,
  number_project number(10),
  avg_hours number(10),
  time_spend_comp number(10),
  work_accident number(10),
  left number(10),
  promotion number(10),
  dept varchar2(30),
  sal number(10));

```

```

SELECT COUNT(*)
FROM hr;

```

COUNT(*)
1 14999

```

##### R studio #####
driver <- JDBC('oracle.jdbc.driver.OracleDriver', 'c:\\data\\ojdbc6.jar')

```

```

oracle_db <- dbConnect(driver, 'jdbc:oracle:thin:@//127.0.0.1:1522/orcl', 'scott', 'tiger')

```

```

query <- 'select * from hr'
hr <- dbGetQuery(oracle_db, query)
hr

```

문제 348. 구글 지도에서 서울의 위도, 경도 정보를 조회 하시오.

```
library(httr)

map_search <- GET("https://maps.googleapis.com/maps/api/geocode/json", query = list(address = "seoul"))
map_search

#html 문서를 아래의 순서대로 검색해서 내용을 조회
content(map_search)

content(map_search)$results[[1]]$formatted_address
content(map_search)$results[[1]]$geometry$location$lat
content(map_search)$results[[1]]$geometry$location$lng

> content(map_search)$results[[1]]$formatted_address
[1] "Seoul, South Korea"
> content(map_search)$results[[1]]$geometry$location$lat
[1] 37.56654
> content(map_search)$results[[1]]$geometry$location$lng
[1] 126.978
```

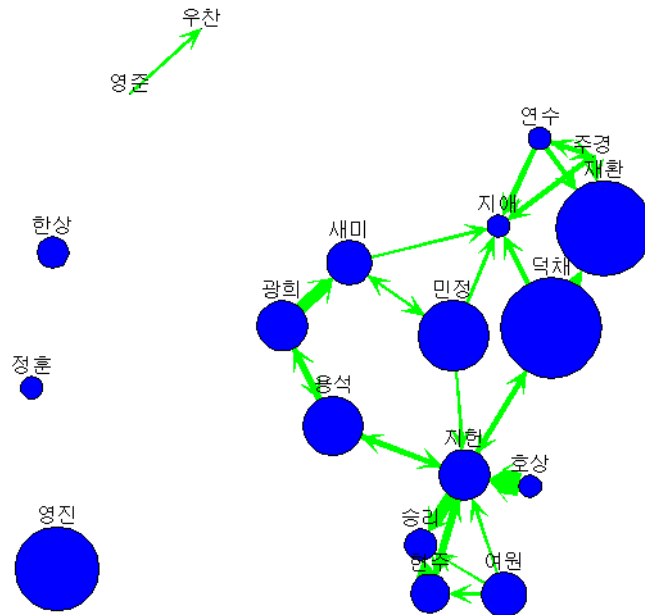
문제 349. 책을 많이 읽는 학생들의 친구 관계를 네트워크 관계도로 시각화 하시오.

질문 : 독서를 좋아하는 학생이 친구와의 교류가 많을까?

```
paper <- read.csv("c:\\data\\paper1.csv", header = T)
paper[is.na(paper)] <- 0
rownames(paper) <- paper[,1]
paper <- paper[-1]
paper2 <- as.matrix(paper)
book <- read.csv("c:\\data\\book_hour.csv", header = T)
paper2
book

library(sna)
x11()

gplot(paper2, displaylabels = T, boxed.labels = F, vertex.cex = sqrt(book[,2]), vertex.col = "blue", vertex.sides = 20,
      edge.lwd = paper2*2, edge.col = "green", label.pos = 3)
```

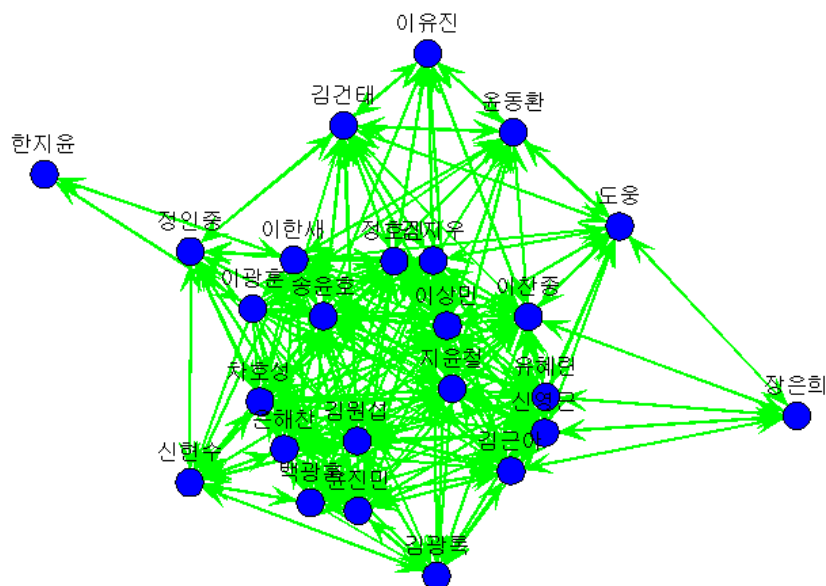


문제 350. 우리반 데이터를 가지고 시각화 하시오.

```
meal<-read.csv("c:\\data\\emp2_meal.csv",header = T)
rownames(meal) <- meal[,1]
meal <- meal[-1]
meal2 <- as.matrix(meal)
```

```
meal2
```

```
gplot(meal2 , displaylabels = T, boxed.labels = F , vertex.col = "blue" , vertex.sides = 20 ,
      edge.col = "green" , label.pos = 3)
```

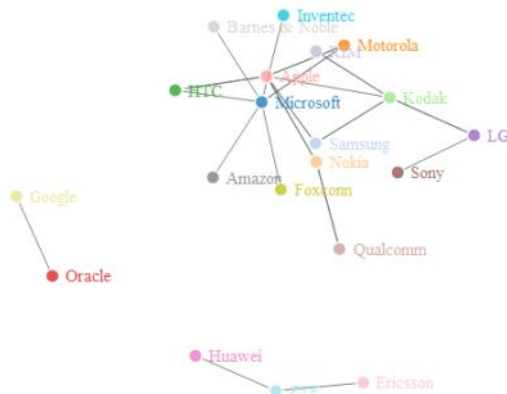


문제 351. 아이티 글로벌 기업들 간의 재판 소송 현황을 시각화 하시오.

```
library(networkD3)
library(dplyr)
```

```
node_df <- read.csv("c:\\data\\node_df.csv",header = T)
link_df <- read.csv("c:\\data\\link_df.csv",header = T)
D3_network_LM<-forceNetwork(Links = link_df,
                             Nodes = node_df,
                             Source = 'source_idx', Target = 'target_idx',
                             NodeID = 'node', Group = 'idx',
                             opacityNoHover = TRUE, zoom = TRUE,
                             bounded = TRUE,
                             fontSize = 15,
                             linkDistance = 75,
                             opacity = 0.9)
```

D3_network_LM



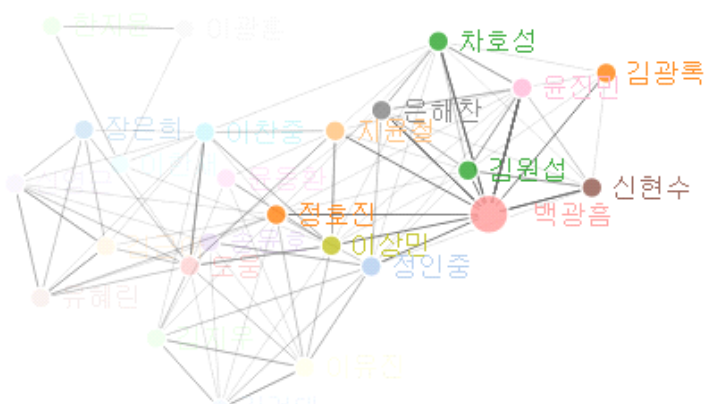
문제 352. 우리반 데이터로 시각화 하시오.

```
emp_df <- read.csv("c:\\data\\emp2_node_df.csv",header = T,stringsAsFactors = F)
head(emp_df,4)
emp_df<-rbind(c('김동윤',0),emp_df) # 0부터 시작해야됨
```

```
link_df <- read.csv("c:\\data\\link_emp2.csv",header = T)
```

```
D3_network_LM<-forceNetwork(Links = link_df,
                             Nodes = emp_df,
                             Source = 'source_idx', Target = 'target_idx',
                             NodeID = 'node', Group = 'idx',
                             opacityNoHover = TRUE, zoom = TRUE,
                             bounded = TRUE,
                             fontSize = 15,
                             linkDistance = 75,
                             opacity = 0.9)
```

D3_network_LM



■ R에서 병렬처리

```
system.time(l1<-rnorm(1000000))
```

사용자	시스템	elapsed
0.08	0.00	0.08

```
system.time(l2<-unlist(mclapply(1:2,function(x){rnorm(500000)}),mc.cores = 4))) # 윈도우에서는 1 코어만 지원
```

```
library(doParallel)
```

```
registerDoParallel(cores = 4)
```

```
system.time(l4p<-foreach(i=1:4,.combine = 'c')%dopar% rnorm(250000))
```

```
> system.time(l4<-foreach(i = 1:4, .combine = 'c')%do% rnorm(250000))
사용자 시스템 elapsed
0.08 0.00 0.07
> registerDoParallel(cores = 4)
> system.time(l4p<-foreach(i=1:4,.combine = 'c')%dopar% rnorm(250000))
사용자 시스템 elapsed
0.02 0.01 0.06
```

서포트 벡터 머신

2018년 6월 21일 목요일 오전 9:56

■ letterdata.csv 실습 _ 필기체 데이터 SVM 으로 분류 모델 생성하기

라벨값

letter	xbox	ybox	width	height	onpix	xbar	ybar	x2bar	y2bar	xybar	x2ybar	xy2bar	xedge	xedgey	yedge	yedgex
T	2	8	3	5	1	8	13	0	6	6	10	8	0	8	0	8
I	5	12	3	7	2	10	5	5	4	13	3	9	2	8	4	10
D	4	11	6	8	6	10	6	2	6	10	3	7	3	7	3	9
N	7	11	6	6	3	5	9	4	6	4	4	10	6	10	2	8
G	2	1	3	1	1	8	6	6	6	6	5	9	1	7	5	10
S	4	11	5	8	3	8	8	6	9	5	6	6	0	8	9	7
B	4	2	5	4	4	8	7	6	6	7	6	6	2	8	7	10
A	1	1	3	2	1	8	2	2	2	8	2	8	1	6	2	7
J	2	2	4	4	2	10	6	2	6	12	4	8	1	6	1	7
M	11	15	13	9	7	13	2	6	2	12	1	9	8	1	1	8

1. 데이터 로드

```
letters <- read.csv("letterdata.csv")
str(letters)
```

2. 훈련 데이터와 테스트 데이터 구분

```
letters_train <- letters[1:16000, ]
letters_test <- letters[16001:20000, ]
```

3. 데이터로 모델 훈련 ----

단순 선형 SVM을 훈련으로 시작

```
install.packages("kernlab")
library(kernlab)
letter_classifier <- ksvm(letter ~ ., data = letters_train, kernel = "vanilladot")
```

4. 모델에 대한 기본 정보 확인

```
letter_classifier
```

> letter_classifier

Support Vector Machine object of class "ksvm"

SV type: C-svc (classification)
parameter : cost C = 1

Linear (vanilla) kernel function.

Number of Support Vectors : 7037

서포트 벡터(margin 선에 걸리는 벡터)가 7037개

5. 모델 성능 평가 ----

테스트 데이터셋에 대한 예측

```
letter_predictions <- predict(letter_classifier, letters_test)
head(letter_predictions)
table(letter_predictions, letters_test$letter)
```

> table(letter_predictions, letters_test\$letter)

letter_predictions	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
A	144	0	0	0	0	0	0	0	0	1	0	0	1	2	2	C
B	0	121	0	5	2	0	1	2	0	0	1	0	1	0	0	2
C	0	0	120	0	4	0	10	2	2	0	1	3	0	0	2	C
D	2	2	0	156	0	1	3	10	4	3	4	3	0	5	5	3

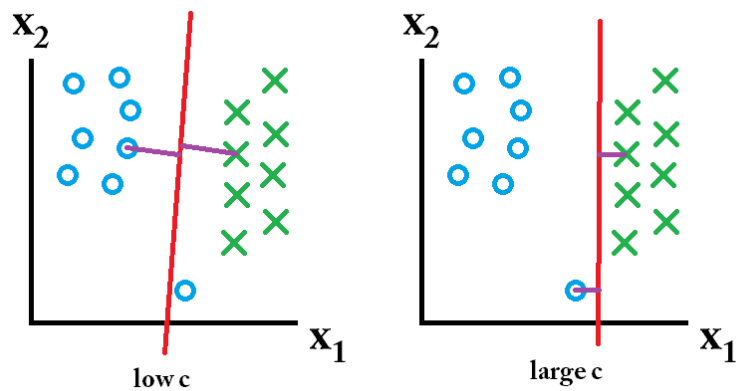

```
> table(letter_predictions, letters_test$letter)
```

letter_predictions	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
A	144	0	0	0	0	0	0	0	0	1	0	0	1	2	2	0
B	0	121	0	5	2	0	1	2	0	0	1	0	1	0	0	2
C	0	0	120	0	4	0	10	2	2	0	1	3	0	0	2	0
D	2	2	0	156	0	1	3	10	4	3	4	3	0	5	5	3
E	0	0	5	0	127	3	1	1	0	0	3	4	0	0	0	0
F	0	0	0	0	0	138	2	2	6	0	0	0	0	0	0	16
G	1	1	2	1	9	2	123	2	0	0	1	2	1	0	1	2
H	0	0	0	1	0	1	0	102	0	2	3	2	3	4	20	0
I	0	1	0	0	0	1	0	0	141	8	0	0	0	0	0	1
J	0	1	0	0	0	1	0	2	5	128	0	0	0	0	1	1
K	1	1	9	0	0	0	2	5	0	0	118	0	0	2	0	1

6. 일치/불일치 예측을 표시하는 TRUE/FALSE 벡터 생성

```
agreement <- letter_predictions == letters_test$letter
table(agreement)
prop.table(table(agreement))
```

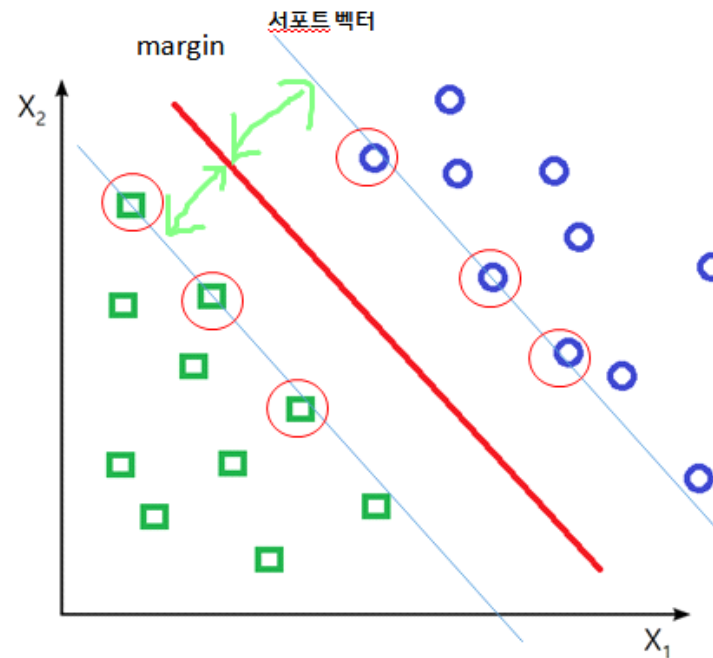
```
> table(agreement)
agreement
FALSE  TRUE
 643   3357
> prop.table(table(agreement))
agreement
FALSE  TRUE
0.16075 0.83925
```



왼쪽의 선이 테스트 데이터에 더 적합하다. (적절한 c 값을 정해 줘야함)

훈련데이터에선 잘 분류하지만 테스트 데이터에선 잘 분류 못하는 것 (너무 훈련데이터에 맞춰짐)

-----> 오버피팅 (c 값이 높을 경우)



마진의 값을 높이는게 목표인데 마진의 경계선에 있는 것들을 서포트 벡터라고 부른다.

■ 한국인 신체 데이터를 가지고 SVM 분류 모델을 생성하시오

성별	나이	키	가슴둘레	허리둘레	배둘레	엉덩이둘레	거드랑이둘레	얼굴수직길이	머리둘레	골격근량	체지방량	체수분	단백질	무기질	체지방률	복부지방률	기초대사량	복부지방률	대사량평가
남	23	1740	979	806	828	982	455	107	560	32.6	12.4	42	11.5	3.9	17.8	0.85	1609	표준	표준
남	22	1722	957	763	797	909	436	109	566	33.7	7.7	43.5	11.8	3.86	11.5	0.84	1649	표준	표준
남	24	1788	883	693	726	874	383	126	568	29.6	6.3	38.6	10.4	3.57	10.6	0.8	1507	표준	표준
남	23	1770	972	731	764	879	430	115	568	34.4	6	44.9	12.1	3.89	9	0.84	1686	표준	표준
남	23	1697	985	827	828	895	430	116	560	31.5	10.4	41.1	11.1	3.98	15.6	0.85	1584	표준	표준
남	24	1781	999	874	898	1010	423	130	561	40.8	11.8	52	14.2	4.92	14.2	0.84	1905	표준	표준
남	23	1673	919	805	826	912	400	126	587	32.3	10.4	41.3	11.4	3.74	15.5	0.85	1589	표준	표준
남	20	1830	901	778	802	920	407	119	577	31.1	12	40.6	10.9	3.85	17.8	0.84	1566	표준	표준
남	23	1710	975	922	961	992	438	120	572	30.1	22.5	39.3	10.6	3.95	29.4	0.88	1535	표준	표하
남	26	1726	966	903	934	1022	411	128	570	31.4	22.7	41.1	11	3.88	28.9	0.88	1582	표준	표준
남	25	1704	1039	854	884	951	429	114	568	30.1	18.2	39.3	10.7	3.64	25.3	0.88	1528	표준	표하
남	22	1778	976	763	813	931	442	134	589	33.9	12.9	44.3	11.9	4.19	17.6	0.86	1675	표준	표준
남	24	1784	1036	907	936	1002	479	123	601	38.8	18.6	49.6	13.6	4.61	21.5	0.91	1835	복비	표준

#1. 워킹디렉토리 소환

```
library(e1071)
setwd("c:/data")
getwd()
```

#2. 데이터 로드

```
## [1] "D:/data"
body <- read.csv("kbody2.csv", header = F)
```

#3. na값 제거

```
body1 <- na.omit(body)
```

#4. 컬럼명 줌

```
colnames(body1) <- c("gender", "age", "height", "chest", "heory", "bae", "ass", "kyeo",
  "face_vertical", "head", "bone", "body_fat", "body_water",
  "protein", "mineral", "body_fat_per", "bae_fat_per", "work", "bae_fat_test",
  "work_test")
```

#5. 랜덤시드 생성

```
set.seed(12345)
body_ran <- body1[order(runif(12894)),]
body_train <- body_ran[1:10314,]
```

```

body_test <- body_ran[10315:12894,]

body_train

#6. 선형SVM 훈련
body_svm <- svm(work_test~, data = body_train, kernel="linear")
body_svm

> body_svm

Call:
svm(formula = work_test ~ ., data = body_train, kernel = "linear")

Parameters:
  SVM-Type:  C-classification
 SVM-Kernel: linear
    cost:    1
   gamma:    0.0001348436

Number of Support Vectors: 6145

#7. 모델 테스트
p <- predict(body_svm, body_test, type="class")
p

#8. 정확도 파악
agreement<-p==body_test$work_test
table(agreement)

prop.table(table(agreement))

> table(agreement)
agreement
FALSE TRUE
  293 2287
> prop.table(table(agreement))
agreement
      FALSE      TRUE
0.1135659 0.8864341

```

문제 352.	Svm 머신러닝 함수로 알파벳 데이터 분류 모델을 생성하고 테스트 데이터의 정확도가 어떻게 나오는지 확인 하시오.
----------------	---

```

letters <- read.csv("c:\\data\\letterdata.csv")
str(letters)

#2. 훈련 데이터와 테스트 데이터 구분
letters_train <- letters[1:16000,]
letters_test <- letters[16001:20000,]

#3. 데이터로 모델 훈련 ----
# 단순 선형 SVM을 훈련으로 시작
install.packages("kernlab")
library(kernlab)

letter_classifier <- svm(letter ~ ., data = letters_train, kernel = "linear")

#4. 모델에 대한 기본 정보 확인
letter_classifier

#5..모델 성능 평가 ----

```

```
# 테스트 데이터셋에 대한 예측
letter_predictions <- predict(letter_classifier, letters_test)
head(letter_predictions)
table(letter_predictions, letters_test$letter)

#6. 일치/불일치 예측을 표시하는 TRUE/FALSE 벡터 생성
agreement <- letter_predictions == letters_test$letter

table(agreement)
prop.table(table(agreement))

> table(agreement)
agreement
FALSE    TRUE
   636   3364
> prop.table(table(agreement))
agreement
FALSE    TRUE
0.159 0.841
```

숫자 필기체 맞추기 실습

■ R에서 mnist로 필기체 인식 학습시키는 코드

```
rm(list=ls())
```

#1. 필요한 패키지를 로드한다.

```
install.packages("caret")
```

```
install.packages("doParallel")
```

```
install.packages("kernlab")
```

```
install.packages("ggplot2")
```

```
install.packages("lattice")
```

```
library(ggplot2)
```

```
library(lattice)
```

```
library(kernlab)
```

```
library(caret)
```

```
library(doParallel)
```

#2. 병렬 작업을 할 수 있도록 설정한다.

```
setwd('c:\\data')
```

```
cl <- makeCluster(detectCores())
```

```
registerDoParallel(cl)
```

#3. 28행의 28열의 행렬의 필기체 데이터 6만개를 사용한다.

```

# Load the MNIST digit recognition dataset into R

# http://yann.lecun.com/exdb/mnist/

# assume you have all 4 files and gunzip'd them

# creates train$n, train$x, train$y and test$n, test$x, test$y

# e.g. train$x is a 60000 x 784 matrix, each row is one digit (28x28)

# call: show_digit(train$x[5,]) to see a digit.

# brendan o'connor - gist.github.com/39760 - anyall.org

#4. 필기체 데이터 불러오는 함수이다.

load_mnist <- function() {

  load_image_file <- function(filename) {

    ret = list()

    f = file(filename,'rb')

    readBin(f,'integer',n=1,size=4,endian='big')

    ret$n = readBin(f,'integer',n=1,size=4,endian='big')

    nrow = readBin(f,'integer',n=1,size=4,endian='big')

    ncol = readBin(f,'integer',n=1,size=4,endian='big')

    x = readBin(f,'integer',n=ret$n*nrow*ncol,size=1,signed=F)

    ret$x = matrix(x, ncol=nrow*ncol, byrow=T)

    close(f)

    ret

  }

  load_label_file <- function(filename) {

    f = file(filename,'rb')

    readBin(f,'integer',n=1,size=4,endian='big')

    n = readBin(f,'integer',n=1,size=4,endian='big')

    y = readBin(f,'integer',n=n,size=1,signed=F)

    close(f)

    y

  }

  train <- load_image_file('train-images.idx3-ubyte')

  test <- load_image_file('t10k-images.idx3-ubyte')

  train$y <- load_label_file('train-labels.idx1-ubyte')

```

```

test$y <- load_label_file('t10k-labels.idx1-ubyte')
}

#5. 필기체 데이터중 하나를 시각화 하는 함수이다.

show_digit <- function(arr784, col=gray(12:1/12), ...) {
  image(matrix(arr784, nrow=28)[,28:1], col=col, ...)
}

#6. 데이터를 로드한다.

train <- data.frame()
test <- data.frame()

# Load data.

load_mnist()
# Normalize:  $X = (X - \min) / (\max - \min) \Rightarrow X = (X - 0) / (255 - 0) \Rightarrow X = X / 255$ .
train$x <- train$x / 255

#7. 훈련 데이터와 테스트 데이터를 6 대 4로 나눈다.

# Setup training data with digit and pixel values with 60/40 split for train/cv.
inTrain = data.frame(y=train$y, train$x)
inTrain$y <- as.factor(inTrain$y)
trainIndex = createDataPartition(inTrain$y, p = 0.60,list=FALSE)
training = inTrain[trainIndex,]
cv = inTrain[-trainIndex,]

#8. svm 모델로 훈련시킨다.

# SVM. 95/94.

fit <- train(y ~ ., data = head(training, 1000), method = 'svmRadial',
  tuneGrid =data.frame(sigma=0.0107249, C=1))
# c : svm 분류선을 정하는 파라미터, c가 너무 크면 오버피팅 (훈련데이터에 최적화)
# c가 너무 작으면 분류를 잘 못함 (언더피팅) ---> 적절한 c값을 알아야 한다.

# sigma : svm 커널을 적용해서 space 를 변경할 때 발생하는 오버피팅을 줄이기 위해 조정하는 파라미터
# sigma가 너무 낮으면 오버피팅 발생, space 변경 한다는 것은 비선형 --> 선형으로 변경한다는 것

results <- predict(fit, newdata = head(cv, 1000))
results
confusionMatrix(results, head(cv$y, 1000))

#9. 만든 svm 모델이 필기체 데이터를 잘 맞추는지 확인한다.

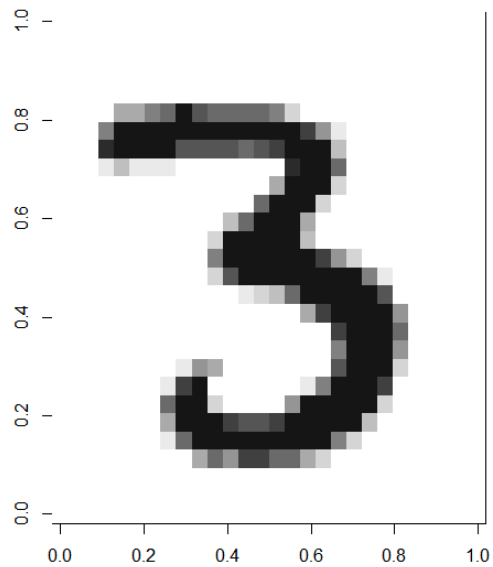
show_digit(as.matrix(training[5,2:785]))

# Predict the digit.
predict(fit, newdata = training[5,])

# Check the actual answer for the digit.

training[5,1]

```



```
> predict(fit, newdata = training[5,])
[1] 3
Levels: 0 1 2 3 4 5 6 7 8 9
> training[5,1]
[1] 3
Levels: 0 1 2 3 4 5 6 7 8 9
```

문제 353. 우리가 처음에 실습했던 알파벳 데이터 정확도가 84% 였는데 그때는 ksvm을 사용했다. 위에서 사용한 train 함수를 이용해서 알파벳 데이터의 정확도를 90% 이상 올리시오.

#1. 데이터 로드

```
rm(list=ls())
```

```
letters <- read.csv("letterdata.csv")
str(letters)
```

#2. 훈련 데이터와 테스트 데이터 구분

```
letters_train <- letters[1:16000,]
letters_test <- letters[16001:20000,]
```

#3. 데이터로 모델 훈련 ----

단순 선형 SVM을 훈련으로 시작

```
install.packages("kernlab")
library(kernlab)
letters_ksvm <- ksvm(letter~, data = letters_train, kernel="rbfdot", C = 15)
```

#4. 모델에 대한 기본 정보 확인

```
letter_classifier
# 서포트 벡터(margin 선에 걸리는 벡터)가 7037개
```

#5..모델 성능 평가 ----

테스트 데이터셋에 대한 예측

```
letter_predictions <- predict(letters_ksvm, letters_test)
head(letter_predictions)
```

```
table(letter_predictions, letters_test$letter)
```

#6. 일치/불일치 예측을 표시하는 TRUE/FALSE 벡터 생성

```
agreement <- letter_predictions == letters_test$letter  
table(agreement)  
prop.table(table(agreement))
```

```
> table(agreement)  
agreement  
FALSE    TRUE  
   128   3872  
> prop.table(table(agreement))  
agreement  
FALSE    TRUE  
0.032 0.968
```